Oxford Brookes University

Honor's Project undertaken in partial fulfillment for the

BSc (Hons) in Computing and Information Systems

"WADE: Web Analysis and Diagnosis Engine using Grid Computing Technology"

Project Code: U08096

June 19, 2009

Acknowledgement

First of all I would like to thank Oxford Brookes University and project supervisor Mr. Peter Lo, for all their supports and guidance given to me throughout this project, from the very beginning to the end, which certainly helped me to finish up the project with a positive motive. I also would like to thank Mr. Peter Lo for his guidance given to me at the start of the project in refining this project idea and helping me to enhance it. I also need to thank ICDSoft (Hong Kong) Limited provides several servers to me for evaluating my project in Grid Computing Architecture.

Abstract

Since 1990s, Grid computing becomes a popular research topic in Internet, Grid Computing allows several computers handle a single calculation at the same time, usually apply into scientific or technical problem. They are required a great number of computer processing to handle large amounts of data.

Thousands of Web Hosting Service Providers start up their business since 2000. This web hosting service provider not only providing Web Hosting Service, but also providing additional value-added services (such as SMTP, POP3, IMAP Servers, Free Sub-domains, Domain parking, Secure Socket Layer (SSL) and Graphical Hit Counter…etc.) From my research, all the Web Hosting Service Provider unable to provide a detail hosting reporting service to customer due to server utilization and time-consuming.

In this project, I will present how to apply the Grid Computing Technology in Deep Web Crawling and Analysis with limited server utilization and faster performance by using Grid Computing Framework, Alchemi. I will discuss how to split a thread for taking the highest performance, what are the unexpected error will occur when the thread splitting is wrong, how to crawl a website through multi-threads in Grid Computing and how Grid Computing can help researcher save more time to complete their calculation. It is an important topic in Grid Computing. I will explain which methodologies will be applied in the application. All selected methodologies can help the Application running parallel in the Grid Environment.

In the experiments session, we evaluate Web Analysis and Diagnosis Engine (WADE) by using over thousand of web pages from the education websites through Google Directory. The experiments will show the algorithms of the web crawling and grid computing with excellent accuracy and performance, and we will show the performance of different process separation. All researchers can find the solution to fix the bottleneck of performance when the application cannot meet the expectation in Grid Environment.

Table of Contents

Chapter 1. Introduction

1.1. Background

Grid Computing means the batch of computer works together to solve the single issue at the same time for saving time and resource. Grid Computing is a cheaper way to gain the same performance as Supercomputer. Since 1990s, this technology usually applies into research of medicine and science, not for the commercial. When the Google grows up, Grid Computing is started to involve the commercial activities, and many IT Company start to use Grid Computing to earn money through leasing the available computing resource to enterprise, called "Cloud Computing".

Web Analysis and Diagnosis Engine (WADE) design is base on Microsoft .NET Framework. It is suitable for all Microsoft Windows Platform. It also support execute on any Linux Platform through Novell Mono Project 2.0. WADE has to use a Grid Computing Technology, Alchemi, which is provided by The University of Melbourne for building the basic Grid Computing structure rapidly. Web Analysis and Diagnosis can provide a fast analysis and detail reporting via a new parser. That is using the formula of Regular Expression to find out all possible links for increasing the accuracy. The parser can detect all web format and file-type and telling you where the problem is, not only telling you has a problem.

WADE is design for Service-oriented Architecture (SOA). Industries can subscribe the Analysis and Diagnosis Report and feed into their existing Business System through SOAP or XML. SOA provides benefits in four basic categories: reducing integration expense, increasing asset reuse, increasing business agility, and reduction of business risk. SOA separates functions into distinct units, or services which developers make accessible over a network in order that users can combine and reuse them in the production of business applications.

Grid computing is a great platform to enlarge the performance of WADE and making the best use of computer resources. ICDSoft (Hong Kong) Limited, one of the leading Web Host Service provider in Hong Kong, responds many web servers haven't use more than 30% CPU. So I want to use the other 70% to do the right thing. For apply WADE, they need not to build up a new batch of server. They can gather all the existing Web Servers to be Grid Nodes for reducing cost.

1.2. Objectives

Web Analysis and Diagnosis Engine (WADE) will be the rapidest, most accurate analysis solution for Web Hosting Service Provider to provide a new service to their customer for increasing loyalty and enterprise image. Through supporting SOA, the industries can be easy to bundle the report to them reporting Application or for the administrator keep references. In this bad business environment, WADE should be a low cost and large benefit solution. Based on Grid Computing, they need not invest any additional budget to purchase a powerful server for serve WADE.

1.3. Chapter Summary

This chapter introduced what is the current problem that faced by web host industries, what is my objective to complete the project and what I will do in the project. Objective can guide my progress of development that is keeping correct without over the topic.

Chapter 2. Literature Review

The purpose of this part is to introduce a short overview on the literature used to create a technical foundation for the "WADE: Web Analysis and Diagnosis Engine using Grid Computing Technology" project. This part introduces the papers and documents used during research and giving an insight into, how those papers relate to this topic. For grant a better overview of the literature, it subdivides into five categories: Distributed Computing, Architectures, Algorithms and Web crawling.

2.1. Distributed Computing

This chapter introduces relevant paper regarding "Distributed Computing"

Distributed Computing Issues

This document is an "A Note on Distributed Computing" that is written by Jim Waldo, Geoff Wyant, Ann Wollrath, Sam Kendall; Sun Microsystems. It introduces almost common or known topics that related to the "Distributed Computing" and explains their issues. The topics covered by the different authors are not exclusively technical like Development in Distribution Environment, Resource Allocation, Synchronization and Failure recovery. It additionally addresses the three programming stages as well. At the end of the document the authors talk about the topics that they think the research is needed in the future. Two of them are "Guaranteed separation" and "Class Replacement without affecting the other parts of system".

The Vision of Unified Objects

The authors have used a point of view from programmer to explain what distributed system is and explain the operation of distributed object. This topic discusses some advantages of the remote class and identified some principles about design model in distributed system:

- There is a single natural object-oriented design for a given application, regardless of the context in which that application will be deployed;
- Failure and performance issues are tied to the implementation of the components of an application, and consideration of these issues should be left out of an initial design; and
- The interface of an object is independent of the context in which that object is used.

Local and Distributed Computing

This topic is all about the differences between local and distributed computing concern: Latency, Memory access, Partial failure and concurrency. Mainly that is focus on discussing the technical issues of resource allocation, synchronization and failure recovery.

Authors said a multi-threaded application needs to deal with Latency, Memory access, Partial failure and concurrency issues. There is a subtle difference. There is no real source for the indeterminacy invocation of operations in multi-threaded application development, so the programmer needs to fully control over invocation.

The Myth of "Quality of Service"

This topic is extending the previous discussion about how to base on resource allocation, synchronization and failure recovery to develop a Quality of Service.

It brings a summary that suppose that the interface describes the object, which supports a number of other objects. A definition of the sets is that there is no duplication. Thus, the implementation of this object makes a duplicate elimination. If the interface does not provide a way to check system information, a set of objects will be questioned to determine equality. Thus, duplicate elimination can only be done by interaction with the objects of the set. No matter how fast the objects of the set of the transaction. The overall efficiency of removing the duplicates will be governed by the latency to communicate over a slow connection is involved. There has not any change in the set of implementations that can overcome this. Interface design problem to determine the upper limit for this performance of operation.

Lessons from NFS

In this topic, Authors discussed some technical issues of NFS (Sun's distributed computing file system) and finding out what the functional limitations are. The limitations on the reliability and robustness of NFS cannot be fixed in the implementation of the parts of that system. There is no "quality of service" that can be improved to eliminate the need. Finally, Author provided some solution to solve the NFS problem. Require the centralized resource manager, which can detect the failure of resource recovery and begins to insure consistency of the system.

2.2. Architectures

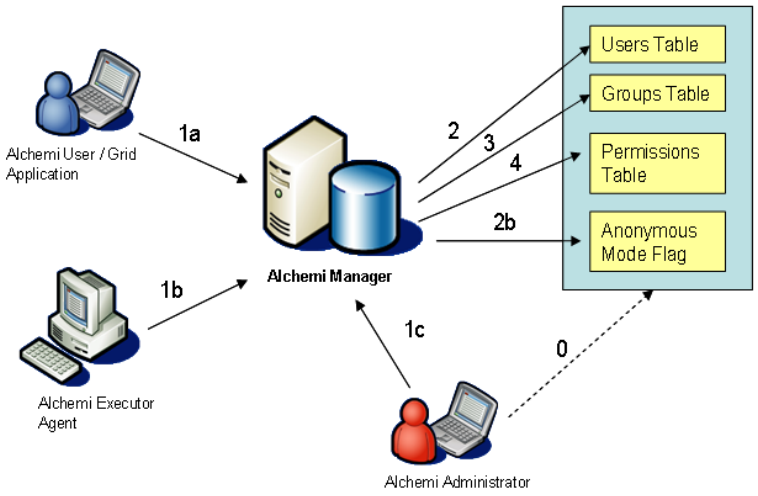
This chapter introduces relevant paper regarding "Enterprise Grid Framework"

Grid Computing Architecture Issues

This document is an "Alchemi: A .NET-based Enterprise Grid Framework" that is written by Krishna Nadiminti (Active developer), A. Luther (Project founder/Developer) and R. Buyya (CI/Mentor); University of Melbourne. Alchemi is the first Grid Computing Architecture using Microsoft .NET Framework Technology. The document mainly discusses what the benefits in Alchemi Architecture are. It also explains why a good Grid Architecture can improve the performance of Multi-thread Application. That is a topic extending "A Note on Distributed Computing".

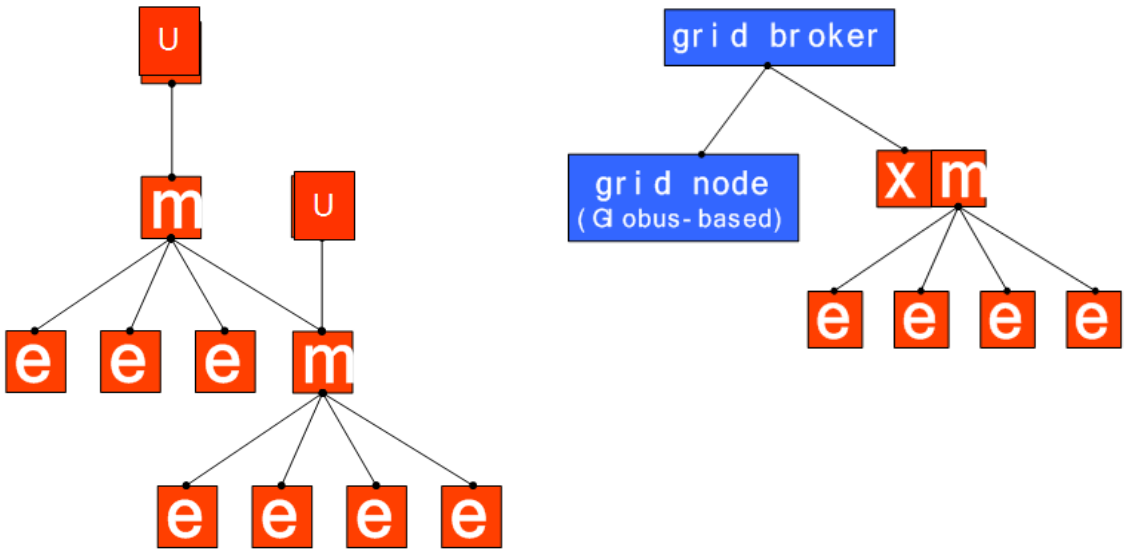
Alchemi Architecture Security

Alchemi is using Role-based Security to protect the Grid Environment to ensure no hacker can use the grid resources without authorization. The security contains three levels, Authentication, Authorization and Auditing. Authentication means checking the User Name and Password, if the login information is valid, the system will grant the permission to him base on his user account, it's Authorization. When the user submits a job to Grid Environment, All jobs/threads executed are recorded in a database and linked to user account used for Authentication.



Multi-level Grid Design

Alchemi supports a cross-domain level architecture. That means it can gather different century's computer network and work together hierarchically. It is an advanced method to apply into high-computing issues.



Just Use, without difficult technical concern

Alchemi provides a very simple programming model for programmer develops a multi-threaded application. Alchemi is an Object-Oriented Grid Thread Model. It contains those main components to provide service. Grid Application consists of independent grid threads. Manager, central controller is used to discovery, scheduling, dispatching and monitoring. Cross Platform Manager is a Web Service Interface for controlling the Grid Environment through Browser. Executor is a worker agent that can install any type of computer, such as Windows and Linux. User means the role that is running grid applications, monitoring and administration. It provides some functional design for the grid operation, transparent execution of threads, Event-driven and Reusable drag and drop components.

2.3. Enterprise Grid Computing

This chapter introduces relevant paper regarding "Enterprise Grid Computing"

Introduction

This document is an "Enterprise Grid Computing" that is written by Paul Strong, Sun Microsystems, ACM Digital Library, July 1, 2005. Paul has written about he has to admit a great measure of commiseration for the IT society at large, when it is confronted with a hail of hype about the network technologies, especially within the enterprise. He also talks about the Definition of Grid computing deeply and some topics about the implementation of Grid Computing in Enterprise Data Centre, and what should we care about the Grid Computing in the future.

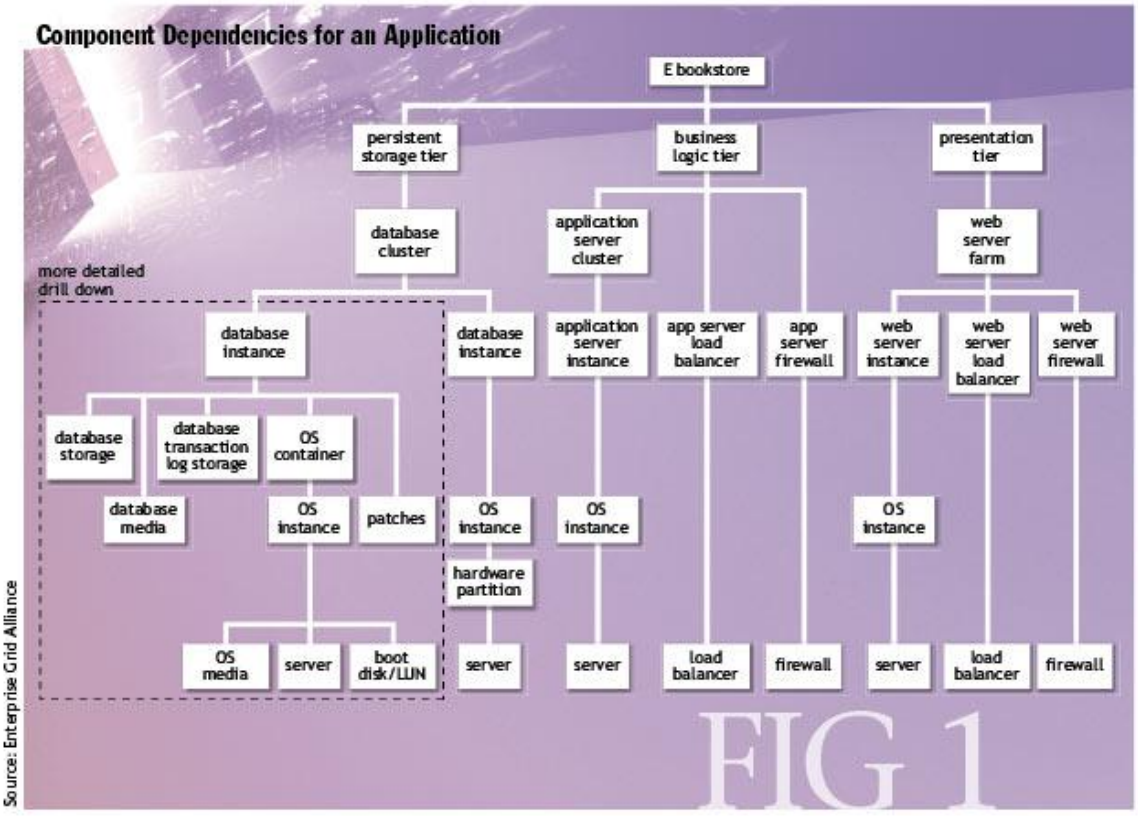
Grid Computing

At the heart of the network is the concept that applications and resources are connected in the form of a fabric or network ubiquitous network. In addition, the network concept implies both ubiquity and predictability, with networks being viewed as very similar to electrical power grids or 1, which are accessible everywhere and sharable by all.

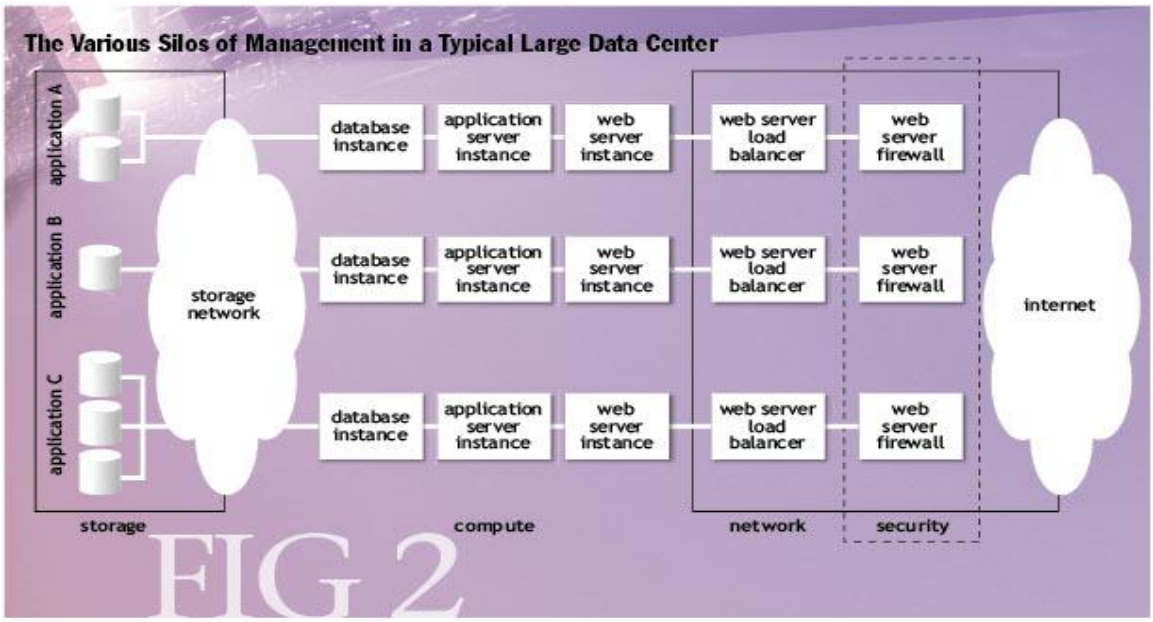
Grid computing is an inevitable consequence of a set of long-term technology trends. These trends have fueled each other, at least the last two or three decades, resulting in the application and infrastructure architectures we see today.

The Enterprise Data Center Today

Today the enterprise data center is a complex place. Each normally hosts a multitude of applications or services running on a large number of network resources. Each of the components of this tissue, either an application or resource, whether physical or logical, is relatively simple, but once you put all together, the complexity increases exponentially. When adding a component not only adds to the total number of components, you can also add a new type of component and a set of relationships with existing components in the tissue. View a typical enterprise application, such as an electronic library. The application can be divided into levels, such as storage or database, business logic, and presentation. Firewalls may exist between some of these levels. Each level may consist of a set of servers that run application components and perhaps a group or load balancing framework. Each server must run at least one application component, which may depend on some version of an operating system, along with a certain set of patches, all running on a particular type of processor. Figure 1 illustrates the complexity of the average data center in the form of a simplified graph for the unit only one application. Add another 10 or 100 such applications and the relationships between them and have an idea of the complexity that must be administered daily in a typical data center.



Today, companies mitigate the effects of complexity by creating silos of relatively stable infrastructure at a divide-and-conquer approach to management. In a typical data center, separate groups for managing their servers and operating systems, network components, storage components, security, and joint services applications. This is illustrated in Figure 2.



Complexity is addressed effectively by limiting the total number and types of components and their relationships. This allows the performance, scalability and availability of the inherent attributes of the distribution network architectures to be exploited, but is usually at the expense of efficiency and agility. Silos or replacement as a result of excess capacity in each silo, which is much less efficient than shared, dynamically allocate the excess capacity. These silos static as a result lack of agility, as new silos that have been created for new applications and services, rather than simply using perhaps an excess capacity.

Bringing the Threads Together

Virtualization, abstraction, and automation are the mechanisms that are keys to making the modern data center in a real network of an enterprise network and providing greater efficiency and agility. These mechanisms are usually performed in combination with a product, for example in the server and operating system provisioning tools, complex services and applications management lifecycle tools, service standards and management tools.

The key to extracting maximum value from these tools is that they share an architectural and operational.

A shared architecture should ensure that the right tools to solve problems the right way. This is the value of the various consortia in the network-for example, the EGA and GGF, which is leading to a series of requirements and an architectural model, respectively. Combining this with the use of standards for the various protocols and management mechanisms (many of which are incipient but nevertheless on the way) should allow data centers to choose the joint interoperability of tools appropriate to their needs, without fear of vendor lock-in.

2.4. Web Crawling Algorithms

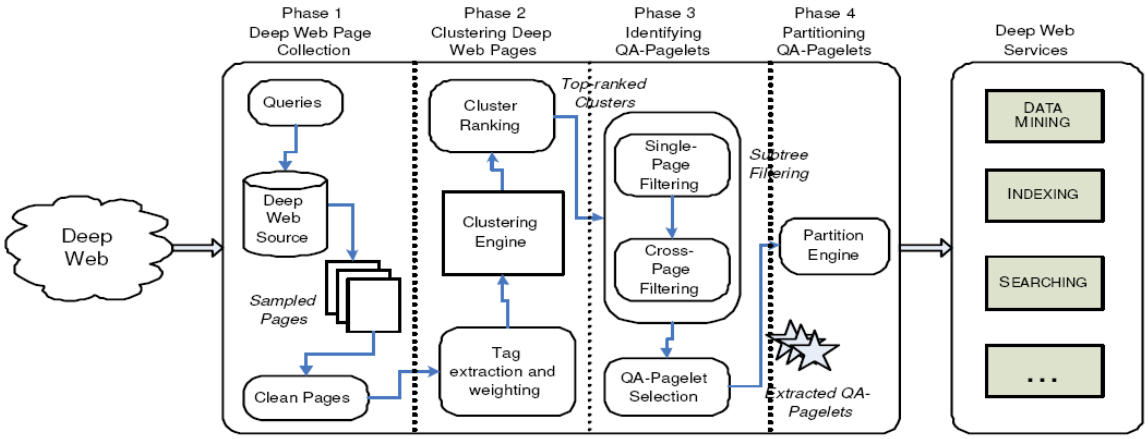
This chapter introduces relevant paper regarding "Alchemi QA-Pagelet: Data Preparation Techniques for Large Scale Data Analysis of the Deep Web".

Grid Computing Architecture Issues

This document is an "Alchemi QA-Pagelet: Data Preparation Techniques for Large Scale Data Analysis of the Deep Web" that is written by James Caverlee and Ling Liu College of Computing, Georgia Institute of Technology. It provides complete research information for reader to learn the foundation of Web Crawling in Grid Computing and also let the reader avoid some common technical issues through reading the documentation.

Authors discuss many related algorithms in the document. Each stage has different methodology to handle the current issue. In the following content, I will drifty explain the algorithms that I will apply in my project.

Web Crawling Stages



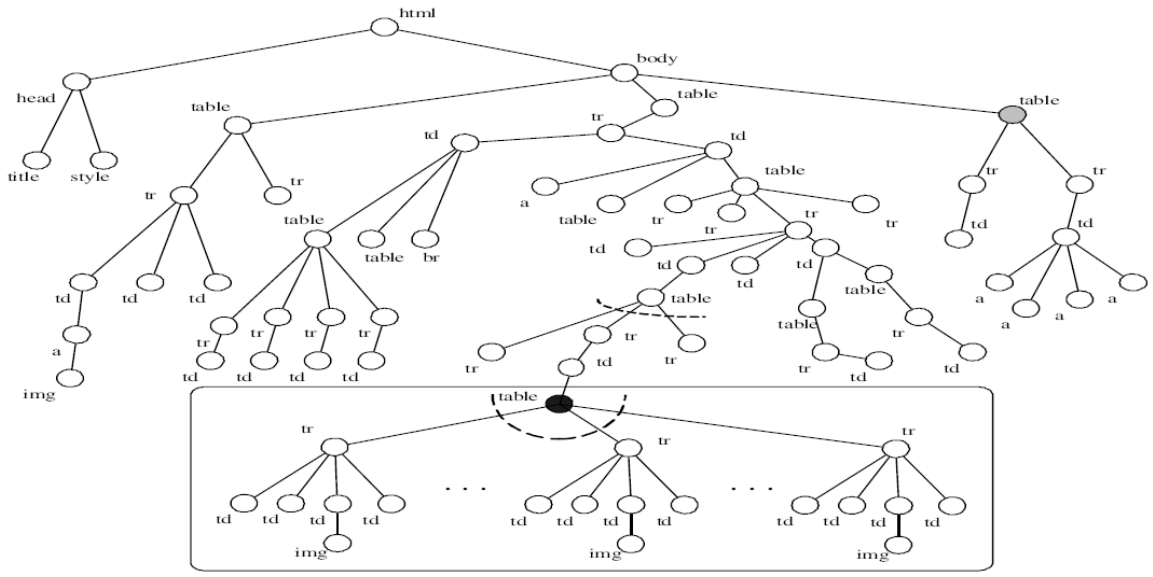
In Web Crawler, it contains 4 phases to process the website, including Web Page Collection, Clustering Web Page, Identifying QA-Pagelets and Patitioning QA-Pagelet. At the end, all result will export into database for Data Mining, Indexing and Searching…

These observations suggest that naturally, they have to take Thor's four stages, as shown. RST stage collects the sample pages of the answer, in response to queries over the Deep Web source. The second phase of the sample groups in response to individual groups of pages to pages that are related to their common control flow dependence, each corresponding to one type of site to answer: whether it is road games pages, page one game, does not match the pages except for pages. The third stage determines the QA-

Page-lets high position on the page of each group according to the separation of sub-tree in a set of clusters on a single page list of common sub-tree sets ranked by their actual diversity. Every single sub-tree corresponds to one set of content-type of the region set by the control-flow dependent answer pages. Then, using the internal cluster of indicators common to filter out content, and enhance the KU-Page-lets. At the end of the third stage, Thor recommends ranking list QA-Page-lets. The fourth phase of the bulkhead is the place to Page-lets KU-KU-detailed objects, which, in turn, add to the other Thor Deep Web information platform.

In our case, those 4 phases are suable for the project to analysis the website.

Tag Tree



Sample Tag Tree from IBM.com

Using the known variations of a Document Object Model, their system changed the website as a tag tree composed of tags and text. Under the tag, it is all the characters between the opening bracket "<" and a closing bracket ">", where each tag in the tag name (eg, BR, TD), and set attributes. The sequence of text characters between the sequence tags.

To translate the website into a tree tag requires that the page be formed. Requirements page is well formed, only the following: start tag including standalone guidelines, must have a corresponding end tag, all attribute values must be in quotes; tags are strictly slot. Pages which do not meet these criteria are automatically converted into well-formed using the Tidy [http://tidy.sourceforge.net/]. Properly developed site can be modeled as a

guideline the tree T consists of tag nodes and content nodes. Tag node consists of all characters from in particular to initiate the appropriate tag and end tag is marked with the name of the start tag. Content of the node consists of all characters between the start tag and the corresponding end tag or between the end tag and the start of the next tag. They mark the node to its content. All content nodes leave the tag tree.

They have made some definition of Tag Tree in a Web Objectization:

Definition 1 (Tag Tree): A tag tree of a page p is defined as a directed tree $\mathcal{T} = (V, E)$ *where* $V = V_T \cup V_C$, V_T *is a finite set of tag nodes and* V_C *is a finite set of content nodes;* $E \subset (V \times V)$, *representing the directed edges.* $\mathcal{T}$ *satisfies the following conditions:* $\forall (u, v) \in E, (v, u) \notin E$; $\forall u \in V, (u, u) \notin E$; *and* $\forall u \in V_C$, $\nexists v \in V$ *such that* $(u, v) \in E$.

Definition 2 (Subtree): Let $\mathcal{T} = (V, E)$ *be the tag tree for a page d, and* $\mathcal{T}' = (V', E')$ *is called a subtree of* $\mathcal{T}$ *anchored at node* u, *denoted as* $\text{subtree}(u)$ $(u \in V')$, *if and only if the following conditions hold: (1)* $V' \subseteq V$, *and* $\forall v \in V, v \neq u$, *if* $u \Longrightarrow^* v$ *then* $v \in V'$; *and (2)* $E' \subseteq E$, *and* $\forall v \in V', v \neq u, v \notin V_C, \exists w \in V', w \neq v$, *and* $(v, w) \in E'$

Definition 3 (Minimal Subtree with Property P): Let $\mathcal{T} = (V, E)$ *be the tag tree for a page p, and* $subtree(u) = (V', E')$ *be a subtree of* $\mathcal{T}$ *anchored at node* u. *We call* $subtree(u)$ *a minimal subtree with property* P, *denoted as* $subtree(u, P)$, *if and only if* $\forall v \in V, v \neq u$, *if* $\text{subtree}(v)$ *has the property* P, *then* $v \Longrightarrow^* u$ *holds.*

Definition 4 (QA-Pagelet): A QA-Pagelet is a minimal subtree that satisfies the following two conditions: (1) A QA-Pagelet is dynamically-generated in response to a query; and (2) it is a page fragment that serves as the primary query-answer content on the page.

Condition 1 the definition does not cover all the static parts of a page that is common to many Deep Web sites, such as navigation bars, the standard explanation, Standard, etc. However, not all regions of dynamically generated content, these definitions are designed to be direct answers to the query. Condition 2 is necessary to exclude from the definition of those regions, such as advertising, which are dynamically generated but is of secondary importance. The subtree corresponding to the QA-Pagelet is in dashed box. KU-Page-let roots are shaded in black and an assembly table.

Website Clustering

It notified about the page clustering problem, the explanation of Concrete similarity metrics is telling us how to select a suitable clustering algorithm to do a job. It analyzed

URL-based, Link-based, Content-based and the Size-based. It also introduces the two well-known clustering algorithms, Simple K-Means and Bisecting K-Means :

SimpleKMeans(Number of Clusters k, Input Vectors $\mathcal{D}$)
Let $\mathcal{D} = \{d_1, ..., d_n\}$ denote the set of n page vectors
Let N denote the total number of distinct tags in $\mathcal{D}$
Let $d_j = < (tag_1, w_j 1), \ldots, (tag_N, w_{jN}) >$ denote a page vector of N elements, w_{jl} is the TFIDF weight of the tag_l in page j $(l = 1, \ldots, N)$
Let $\mathcal{C} = \{C_1, ..., C_k\}$ denote a clustering of $\mathcal{D}$ into k clusters
Let μ_i denote the center of cluster C_i
foreach cluster C_i
  Randomly pick a page vector, say d_j from $\mathcal{D}$
  Initialize a cluster center $\mu_i = d_j$, where $d_j \in \mathcal{D}$
repeat
  foreach input page vector $d_j \in \mathcal{D}$
    foreach cluster $C_i \in \mathcal{C}$ $i = 1, \ldots, k$
      compute $\delta_i = sim(d_j, mu_i)$
    if δ_h is the smallest among $\delta_1, \delta_2, \ldots, \delta_k$
      mu_h is the nearest cluster center to d_j
    Assign d_j to the cluster C_h
  // refine cluster centers using centroid of each cluster
  foreach cluster $C_i \in \mathcal{C}$
    foreach tag l in d_j $(l = 1, \ldots, N))$
      $cw_{ij} \leftarrow \frac{1}{|C_i|} \sum_{l=1}^{N} w_{jl}$
    $\mu_i \leftarrow < (tag_1, cw_{i1}), \ldots, (tag_N, cw_{iN}) >$
until cluster centers no longer change
return $\mathcal{C}$

Tag Tree Signature Simple K-Means Page Clustering Algorithm

BisectingKMeans(Number of Clusters k, Input Vectors $\mathcal{D}$, Iterations I)
Define a clustering $\mathcal{C} = \{C_1\}$
foreach input vector $d_j \in \mathcal{D}$
  Assign d_j to C_1
for $i = 1$ **to** $k - 1$
  Select a cluster $C_i \in \mathcal{C}$
  Let $\mathcal{D}_i$ denote the set of page vectors in C_i
  Define a set of candidate clusterings $Candidate = \{Candidate_1, ..., Candidate_I\}$
  for $j = 1$ **to** I
    $Candidate_j \leftarrow$ SimpleKMeans(2, $\mathcal{D}_{C_i}$)
  $\hat{C} \leftarrow$ BestClustering($Candidate$)
  $\mathcal{C} \leftarrow \{\mathcal{C} \cup \hat{C} \setminus C_i\}$
return $\mathcal{C}$

Tag Tree Signature Bisecting K-Means Page Clustering Algorithm

And it recommended a better way for selecting the page clustering algorithm:

Average Fanout: Clusters that have pages with higher average fanout may be more likely to contain QA-Pagelets. The average fan-out for a $Cluster_i$ can be computed by the average of the largest fanout of a node in each page of the cluster. Namely,

$$\frac{1}{|Cluster_i|} \sum_{p \in Cluster_i} \max_{u \in p.V} \{ fanout(u) \}$$

The $p.V$ denotes the set of nodes in page p.

Average Page Size: Larger pages may tend to be more likely to contain QA-Pagelets. We define the average page size for a $Cluster_i$ as

$$\frac{1}{|Cluster_i|}\sum_{p \in Cluster_c} Size(p)$$

The $Size(p)$ denotes the size of page p in bytes.

An excellent Algorithm can make the calculation rapidly. It tells that when splitting a thread to Cluster Server, we should use domain based, not page based. Because the thread too small, that will increase the Server loading to collect, filtering and sorting the distributed threading from different Node Servers.

The researchers have made a detail experimental to prove their theory. Test in 50 websites and within 100 pages for each site to create data sets of 55,000 pages (1,100 pages per site), 550,000 pages (11,000 pages per site), and 5,500,000 pages (110,000 pages per site).

2.5. Web Services in Grid Computing

This chapter introduces relevant paper regarding "Experiences with GRIA – Industrial applications on a Web Services Grid" that is written by Mike Surridge and Steve Taylor, IT Innovation Centre, IEEE.

GRIA Project

The GRIA project is designed to be used by the Net industry. The GRIA middleware is based on Web Services, and aims to meet the needs of industry for security and business-to-business (B2B) service procurement and operation. This offers a well-defined B2B models for accounting and QoS agreement, and proxy-free delegation's support for account management and service federation. The GRIA v3 software is currently used in industry. A business-oriented approach, irrespective of the Open Grid Services Architecture proposals for changing the Global Grid Forum, GRIA has demonstrated the need for a wider understanding of Virtual Organizations (Vos). The traditional academic Vos are continual, resourceful, and, logically centralized, membership-oriented management structures. In contrast, the GRIA experience has been that the business is likely to project focused Vos and distributed process-oriented management structures.

Starting with the seemingly more modest goal of a business support existing Grid system, the GRIA project, new software is fully Web Services, and focused on the beginning of commercial business applications and business models. This includes the off the-side Web Services technologies, security add-ons, and the model-based access control process, which is in the business processes. It is also a stimulus for the development and standardization in the field of B2B negotiation, mediation and resource methods. GRIA stresses the need for a wider range of different VO models, including the fast and agile B2B models, as well as the large, long-VO models feature a number of large-scale scientific research cooperation. GRIA also highlights the need for the Semantic Grid to support open markets and processes. These will be addressed in future work using the EC IST GRIA middleware project Next GRID and SIMD.

2.6. Advanced Grid Computing Control

This chapter introduces relevant paper regarding "Self Adaptivity in Grid Computing".

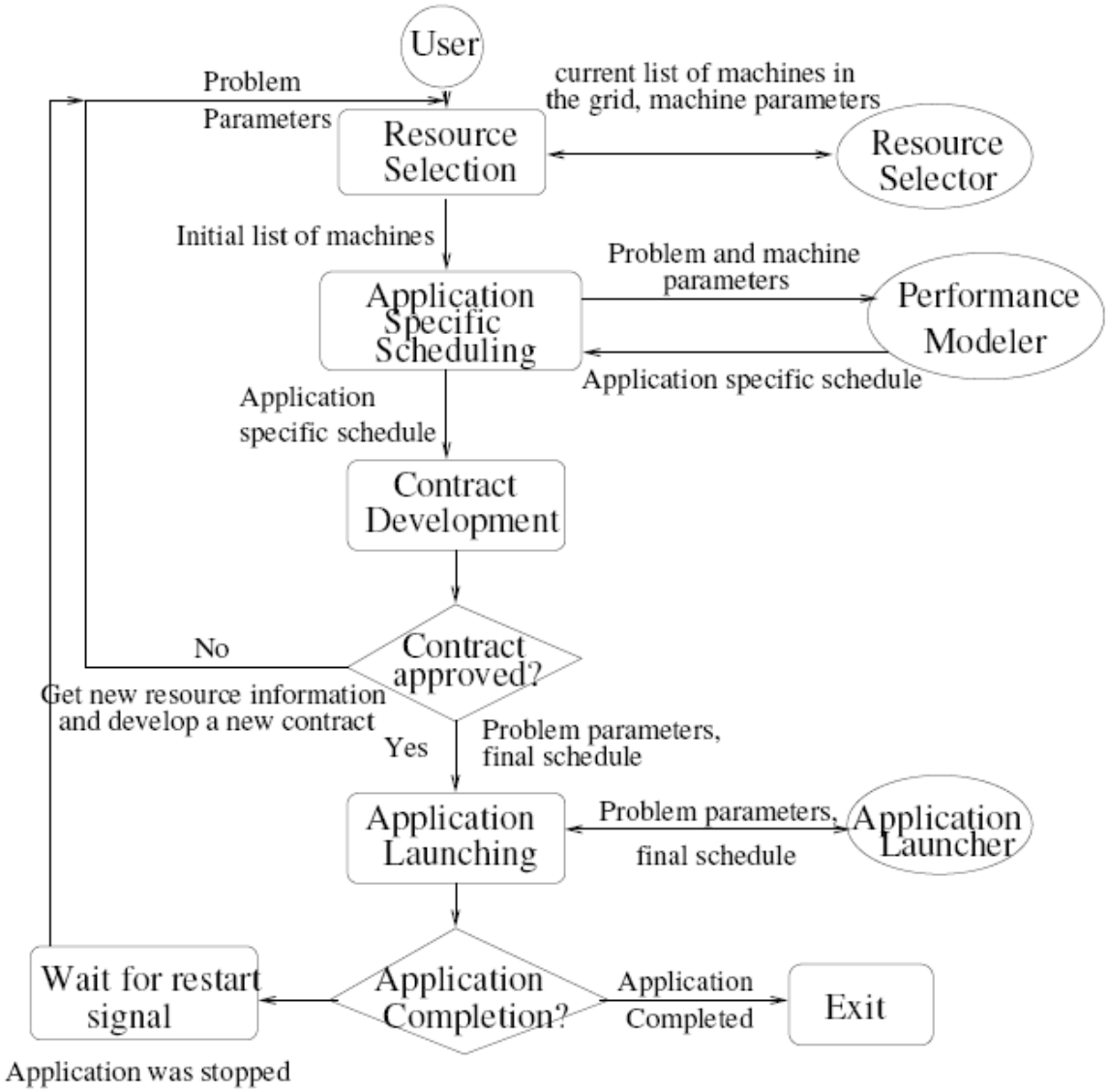
Introduction

This document is an "Self Adaptivity in Grid Computing" that is written by Sathish S. Vadhiyar and Jack J. Dongarra; Supercomputer Education and Research Centre Indian Institute of Science, Computer Science Department, University of Tennessee Knoxville, Computer Science and Mathematics Oak Ridge National Laboratory. It talks about how the Grid Computing has a methodology for dynamically balance work-load in Grid Environment. It's an advanced topic for a large Grid Computing Application. They found few self-adaptive software systems to do a comparison and deeply analyses them, all systems that dynamically adapt to changes in load characteristics, resources, computational Grids. Computational Grids include the dynamics of large stocks, so the opportunity to migrate executing programs in the different resources assumes great importance. Specifically, the main reasons of migration programs and grid systems to provide fault tolerance and to adapt to changes in system load. In this paper, we focus on executing the migration of applications and the Grid systems in order to adapt to the dynamics of the load of resources. Two disadvantages found in these systems, First, the individual policies of those working in the migration system of suspension and migration of applications to carry out programs for different systems, applications, may experience a long waiting time between when they are suspended, if they are new on the new system. Second, due to the use of the predefined conditions for suspension and migration and due to lack of knowledge about the remaining execution time of programs, applications may be suspended and moved, even if they intend to complete in a short period of time. This, of course, less desirable results of the network-oriented systems, where a large load dynamics can lead to the frequent satisfaction of predefined conditions and therefore may lead to the frequent invocations of suspension and migration decisions.

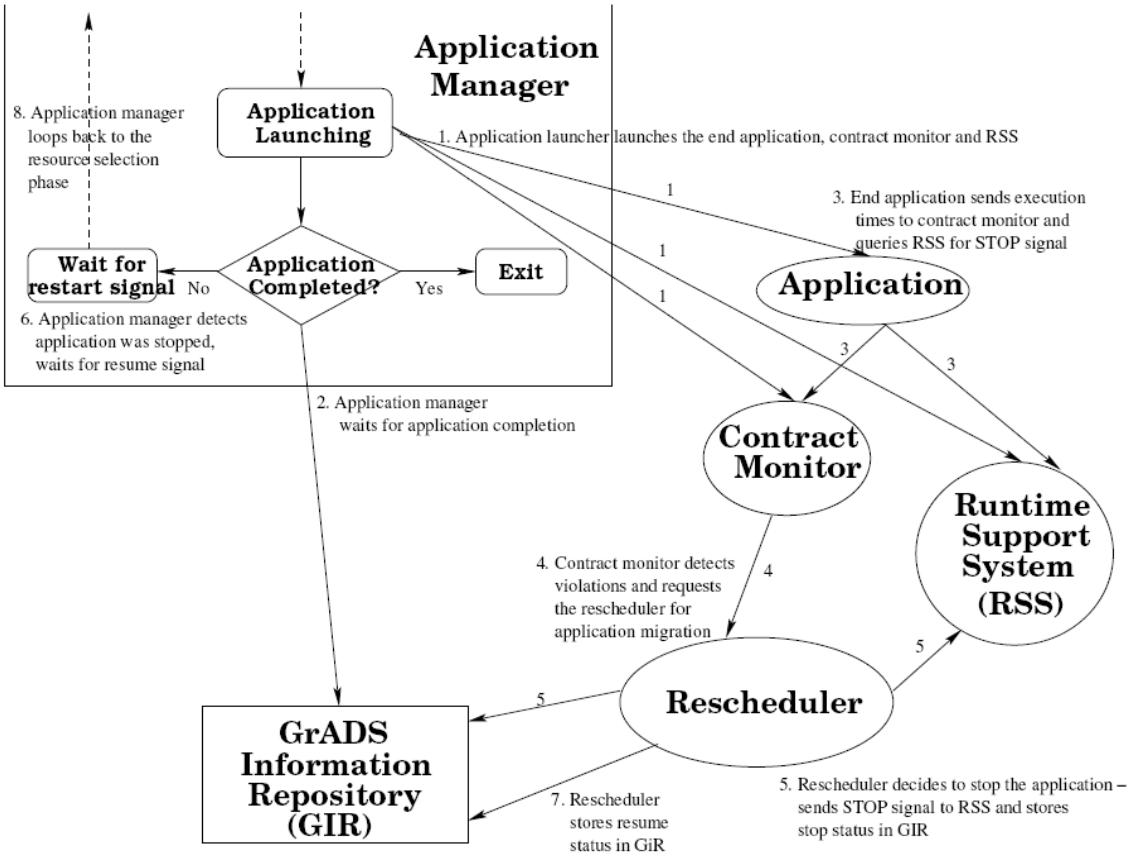
GrADS architecture

They introduce GrADS architecture. GrADS (Grid Application Development Software) is an ongoing research project involving many institutions and its aim is to simplify distributed heterogeneous computing in the same way that World Wide Web Simplified the exchange of information over the Internet. University of Tennessee examines issues related to the integration of libraries in the GrADS system. In his previous work, they have demonstrated ease with which the number of libraries as ScaLAPACK can be integrated into the Grid system and the ease with which the library can be used over the Grid. They also showed that some results demonstrate the benefits of Grid solution of a large number of problems. In the architecture of GrADS, a user wanting to solve through

the application of grid based on the GrADS manager. The life cycle and the manager were shown GrADS:



GrADS application manager



Interaction of Migration Framework

Many of the migration of existing systems, migrating applications are to the resources under the loading conditions of simple policies that cannot be applied to Grid systems. They implement the migration system, which takes into account both system load and application characteristics. Migrant decisions based on factors including the amount of resources, load, point, application of life, when the load is introduced, and from applications. They also implemented the system, that is opportunistically migrating executive applications to use the additional free resources. The experiments were performed and the results were presented to demonstrate the possibilities of migration system.

They aim to provide more reliable system and the SRS system, and provide cost effective Reschedule redistribution of data. In addition, instead of fix reschedule threshold is 30% of their future work will participate in the determination of the term limits dynamically based on the observation of dynamic load behavior of the system resources.

They offer their approach to examine the usefulness of complex applications involving multiple components and / or written in multi-programming languages similar to the

efforts of Mayes. Now, the average efficiency ratio is used when the track will be contacted reschedule migration. And in the future, their plan to investigate a more restrictive policy on contact with reschedule. Mechanisms to quantify the defects discovered in the implementation of the model to monitor and transmit information, the application developer must also be investigated.

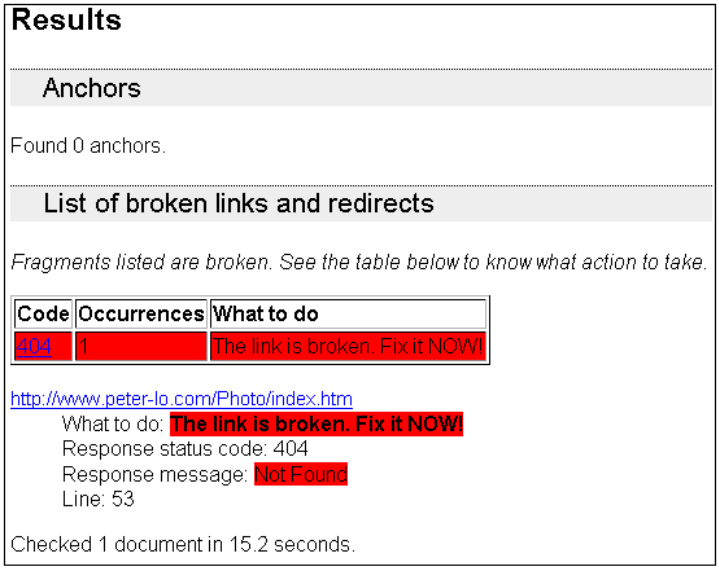
2.7. Chapter Summary

This chapter discussed what issues I need to face in development. The paper's authors gave me a lot of resources and suggestions to implement into my project. All information is very valuable for Grid Computing Development.

Chapter 3. System Analysis

3.1. Competitive Comparison

W3C Link Checker is developing by The World Wide Web Consortium. It is written in Perl. It provides a free link checking service to user who wants to find out the error from a selected URL. You also can install it in your own server to keep monitoring your links. W3C Link Checker can detect the status of webpage and image. It has a detail reporting to explain the problem:



W3C Link Checker is a One-Time Service. That means you cannot regular to check your selected URL automatically. If you want to batch check your domain list, you may need to develop a program to use the W3C Link Checker as an API. In the picture, you may see the whole checking time around 15.2 seconds. That is a too slow operation time for providing service by Service Providers. In my research, W3C Link Checker is hosting by a single server that is without any distributed computing technology. User can set the recursion depth for crawl the website deeply. Control the recursion depth can increase the analysis speed, the crawler need not to scan the whole site, just scan the few top level website. Administrator need not fully scan the website for saving the band wide and resource. W3C Link Checker can export a report as an html format. That is not enough for the enterprise to feed it into their database. Their developers need to convert the data before insert them in the analysis machine. W3C Link Checker can tell you the link is broken, but it won't tell you which website contained the error link. The error may be a wrong typing of the link, not really the file unavailable. So the developer may need to take a lot of time to trace the wrong link where it is. W3C Link Checker cannot detect any attachments, such as Office Documents (Word, Excel, Power Point, PDF & flash). It

just detects standard web page and java script document, not fulfill current IT administration requirement. There is the summary of comparison:

Items	W3C Link Checker	WADE
Computing Methodology	Single CPU	Grid Computing
Executive Environment	Perl: Linux, Windows	C#: Linux*, Windows
Batch Analysis	No	Yes
Customize the recursion depth	Yes	Yes
Report Broken Link	Yes	Yes
Report the location of Broken Link	No	Yes
Publish XML Report	No	Yes
Support Attachment (Office Document, PDF, flash)	No	Yes

*C# can be executed on Linux through Novell Mono Framework. Mono is an open source project to let .NET Application running without Microsoft Windows System.

Andrew Brian Cryer, MSc in CAD/CAM, has made a detail comparison for some well-known Link Checker Application. Andrew had compared in seven areas, Program Type, Scope, Images, In-Page Links, Check locally, JavaScript Links and the license type. The table shows the two different program types, exe and on-line, exe is an executive application in Microsoft Windows Series that is not for the Linux platform. On-line means the application is hosting by Service Provider, the program type is not a necessary. A completed analysis needs a large scope to crawl the webpage, if the analysis is depend on a single page, that means if the page does not contain the whole site map, some page will not be found by Link Checker, the result will not be trustable. So if the scanning depends on a site, the Link Checker will try to get the directory list from the server and one-by-one parses all available pages. Image is the main element in the website. Users cannot imagine the webpage is plaintext only. Detect the image is very necessary. In-Page Links is the web crawling function. Link Checker will collect all visible links that are including in <a> element. That is a common way to collection all links from a website without the directory list. Basically, Check locally is checking the language does it suitable from some countries. For example, Japanese may not understand English content, so the locally checking will suggest you to modify the content to be local language. Some link may not contain in <a> element, they may exist in a JavaScript language. Some professional crawler can find their location and capture them, but many common link checkers cannot do it well. Some Link Checkers need to pay the license, some is not. The consumers have a selective choice for their need. There is the Comparison Table:

Link Checkers	Program type	Scope	Images	In-Page Links	Check locally	JavaScript Links	Type
SortSite	Exe	Site	✔	✔	✔	✔	Commercial, 30 day trial
DeepTrawl	Exe	Site	✔	✔	✔	✖	Commercial, 30 day trial
Link Checker Pro	Exe	Site	✔	✖	✔	✖	Commercial, 30 day trial
Xenu's Link Sleuth	Exe	Site	✔	✖	✔	✖	Free
InfoLink Link Checker	Exe	Site	✔	✖	✔	✖	Free
Dead-Links	On-line	Site	✖	✖	✖	✖	Free
REL Link Checker Lite	Exe	Site	✖	✖	✖	✖	Free
404 error page	N/A	Site	✖	✖	✖	✔	Free
Google Sitemaps	N/A	Site	✖	✖	✖	✖	Free
1-hit.com Bad Link Checker	On-line	Page	✔	✔	✔	✔	Free
FWB Broken Link Checker	On-line	Page	✔	✖	✖	✖	Free
W3C Link Checker	On-line	Page	✖	✔	✖	✖	Free
Indiabook.com Free Link Checker	On-line	Page	✖	✖	✖	✖	Free
LinkChecker by 2bone	On-line	Page	✖	✖	✖	✖	Free
LinkTraX	On-line	Page	✖	✖	✖	✖	Free

The table can tell us if you want have a complete feature to monitor your website, you may need to pay for the commercial organization. But they are not suitable for all web-hosting company. Web Hosting Company is usually running their hosting service at Linux platform for the great performance and decreasing the license cost. Through the table, you can see the top three commercial application types that is exe file. That means it only for Microsoft Windows Series.

In the next page, there are the detail functional comparisons for those Link Checkers by Andrew Brian Cryer. I had made some comment and list a new problem for each Link Checker.

Finally, those Link Checkers are not running in Grid Computing Environment. They are running on a single Machine only. If the project needs not to analysis the batch domain list, the top three applications is the best choices. They can fulfill the basic requirement to

complete the web analysis job and research. For the highest requirement, we need to develop a Link Checker that is running in Grid Computing Methodology. That means WADE is the first application developed for Distribution Environment.

3.2. Problem & Justification

3.2.1. Hyper Link Identification

Identifying all hyper links in a webpage is a basic work to crawl a website deeply. Webpage are using HTML Format to present their content. HTML is the leading markup language for Web pages. It offers a method to describe the format of plain text information in a document — by indicating positive text as links, headings, paragraphs, lists, footer and other web components, and to reinforcement some text with interactive forms, embedded images, and other objects. HTML is written in the form of tags, enclosed by <> tags. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript) which can affect the behavior of Web browsers and other HTML processors. In a webpage, all links are not only presented by Link Element (<a href>), but also images are using Image Element (<img>) to present in the browser. For detecting all file is available, a Link Crawler should know how to find them out.

In some developer's habit, they won't enter a full URL for the Link in each page. Because they rely on the Web Server parsing function, the function can automatically plug the full URL path into the website to replace the short URL path. For example: (www.domain.com)

The standard format should:

Service

But they always use this format for the same level:

Service

And some case for taking back to top level:

Back to service

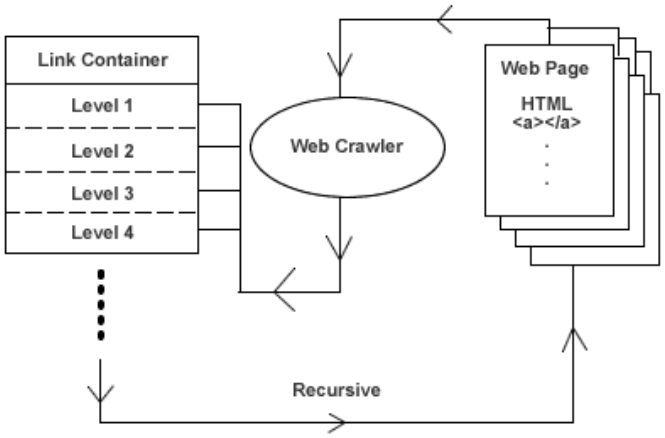
So the Link Crawler needs to handle this issue and re-engineer all the links to the standard format for the Link Parser use:

http://www.domain.com/service.html

If the Link Crawler does not handle this case, the parser cannot identify the URL, it will treat it as an Error Link, and then the result will be wrong and not trustable. Crawler will log down the current page location, (e.g. http://www.domain.com) and use the Regular Expression to capture all doubtful links. Crawler will try to use the logged path to complete the link if the link level is the same with logged path. If the logged path is the

second level and doubtful link is re-direct to the first level, the crawler will automatically bring the link to the top level. (Example: logged path: www.domain.com/sub/, the link is (./service.html), and then the completed link should be www.domain.com/service.html)

All captured link will store in a Level Container for the next level recursive crawling.



The Web Crawler will do the recursive crawling until all web page had been done or complete the scope of recursion depth only. Before insert into Link Container, the container will skip all duplication links and external domains. This action can reduce all dead looping. All captured links are includes web (e.g. html, php, asp, aspx, jsp), image (e.g. bmp, jpg, gif, png), flash (e.g. flv), document (e.g. doc, xls, ppt, pdf, xml) and other file type (zip, rar, exe, mov, mp3, avi, iso). System will try to understand their header to identify what they are. If it is a web, crawler will try to open and parse it to capture all links and put it into Link Container.

After the recursion completed, all links are spotless for the next checking (Live Detection). Live Detection will try to get the header or ping the file location for understanding the status of the link. If the target location is no response, that means the target may lost or the link is wrong. All error will keep into Error Container. Each Error Log will contain the Error Link, Error Type and Error Occurring Page. All Errors will feedback to the main server. Main server will remove all duplication error and generate the error report as XML for the client feed it.

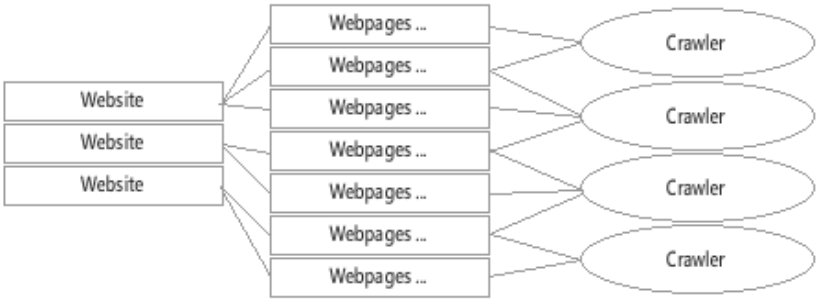
3.2.2. Web Crawl threading in different level

Common Web Crawlers are also using one thread to parse the website. That means all pages in the website will be one by one parse by the crawler. If the computer allows 100 threads running at the same time, this case will waste 99 threads for computing.

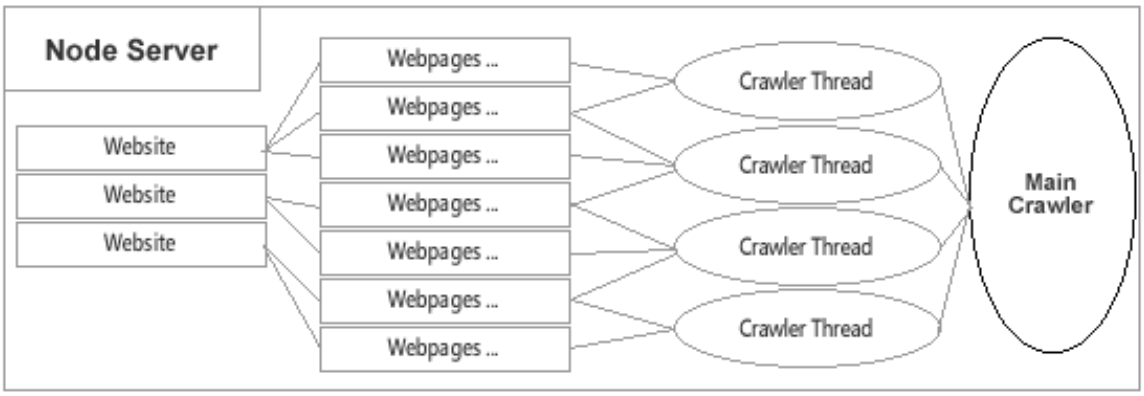
Through we are using Grid Computing. Crawler couldn't use a single thread to crawl the links in the batch of website. Single thread will waste the all surplus CPU resources and extend the crawling time. It's the same case with wasting 99 threads. In this case, if I have 100 node computers that support 10,000 threads, if I just assign 1 thread per node computer, that means I waste 9900 threads. So the crawler should support multi-threading technology.

For example, www.domain.com has 1000 pages, if it has few pages loading is more than 30 seconds, it will affect the following queue. That is the "One-by-One Situation".

For concerning the highest performance, Crawler should create a serial of threads to capture multi-pages, multi-website at same time to avoid page-loading issue. That is an advanced situation, but all crawlers are individual, all data need to combine in Master Server. That means each crawler may process duplication at the same time. It will waste many resources and increasing the Mast Server workload to filter useful information at the end. The speed will turning down.

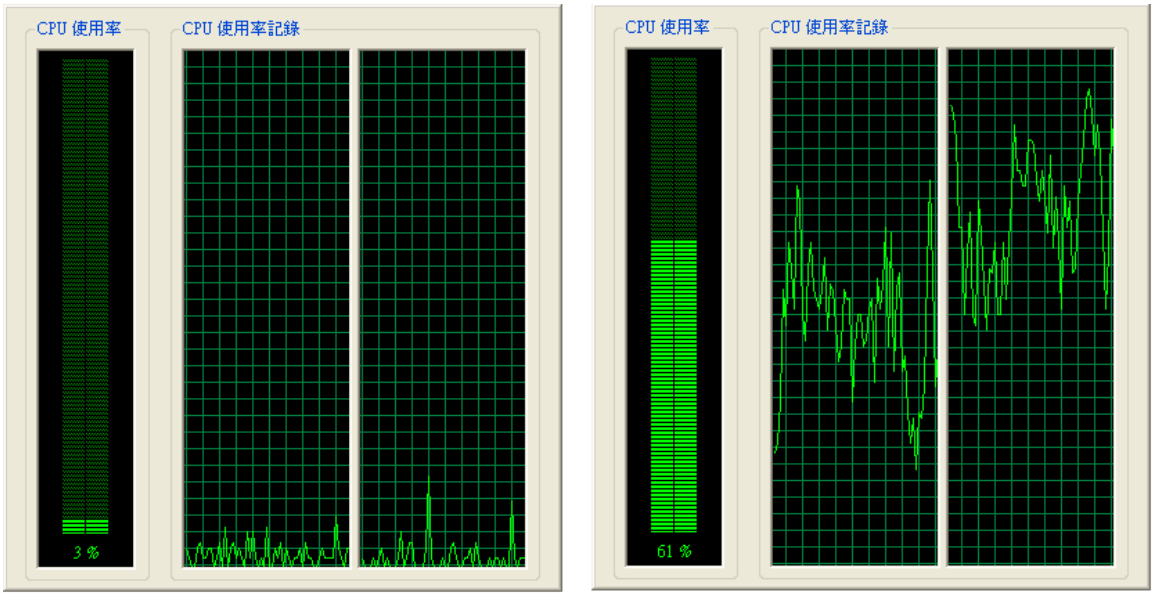


Through the Grid Computing, system will deploy many crawlers to different Node Servers. WADE should make the crawler that can use all surplus CPU resources to crawl the web per Node Server as the following picture:



In this case, each node server has a Main Crawler to control all sub threads. Each thread will use the same pool to store the web site map and result, so all data will not be duplicated. Master Server of Grid needs not to do the latter part processes from all nodes server, such as filtering, combining.

Each node server will handle different domain list for reducing the duplication, because there is not included any communication between them. They just can executive the command that is order by Master Server. If we want to use all available node resources, we need to support multi-threading. Each node server can spend more than 60% resource on the crawling. The speed will have a positive improvement.



3.2.3. Process Separation for Grid Computing

Process Separation is a difficult job in development. Grid Computing Performance is very depends on Process Separation. If the process split too small, Grid Master Server will need more resources to assign job to Node Servers and receive all response from Node Servers. For example, if the process has 10 parts and you split it as 20 parts and deploy to node servers. Master Sever need to stop and wait for their responses, after received all jobs, Master Server will combine and filter all needed data. If the Separation is too much, Node Server will return many duplicated data. Master Server need to spend more resources to process them. It will increase the workload of Master Server and decrease the whole job performance.

So I need to study and try the solution that is the preface to present a great performance of Process Separation.

There have 3 types to do the process separation:

1. Page-based
2. Domain-based
3. Sliding Window-based

Page-based

Page-based can let the series of thread processes the single page as the same time to increase the parsing. It is the fast method for a single domain checking. But it will increase the workload for the communication between both threads. The communication is for de-duplication and data status checking.

Domain-based

Domain-based can let the Master server assign the domains to each thread. Each thread will have a domain only, after they completed their domain crawling, the thread will return the result to Master Server and it closed. It is the better way for checking a domain list. It's no communication between threads. They just need to receive and send back the result to Master Server only. It's not a good idea for a large domain dataset. Because each thread just handles 1 domain, if the dataset has 1,000,000 domains, that means I need to deploy a lot of nodes to handle the list if I want to get to result with speedy?

Sliding Window-based

Each thread will handle 1-100 domains, and Each Node Server will handle 1-50 threads. After all the threads completed, Node Server will combine and filter all needed data, and report to Master Server. It is a final idea for a large domain dataset. Master Server needs to ensure all Node Server will not hold duplicated domain list.

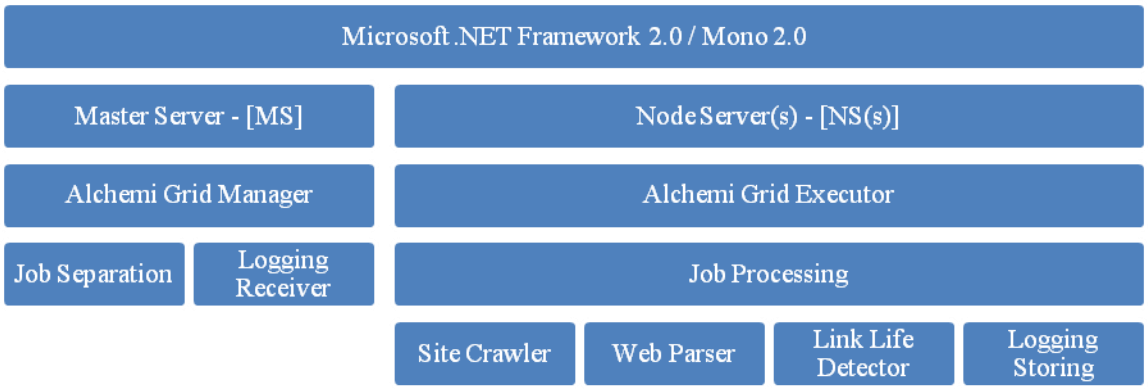
3.3. Chapter Summary

This chapter is about the problems that I found in research, I also provided the suggestion after the problem discussion. Those problems will appear into all related projects. So I listed them out for reference. In the initial part, I also make a Competitive Comparison for comparing the function difference between WADE and other common analysis tool. It is a great progress to show off the benefit of my application design.

Chapter 4. System Design

4.1. System Architecture

WADE System Distribution



The Workflow of WADE

WADE is running on those frameworks, Microsoft .NET Framework, Mono 2.0 and Alchemi Grid Computing Framework. Microsoft .NET Framework is a successful product that is developed by Microsoft Corporation. It's a secure, faster, and higher Compatibility. I'm using C# to do my development. C# is the one programming language under the .NET Framework. C# is standard by ECMA (ECMA-334) and ISO (ISO-23270) already, so the open source framework, Mono, is support C# too. Mono is an open source project that is hosted by Novell. It is a .Net Framework Linux Edition. It fully support Microsoft .NET Framework 2.0 through the standard of ECMA (ECMA-334). Mono allows Microsoft Developers running their .NET Application on Linux Operation System to decrease the difference level between Windows and Linux.

Alchemi Grid Computing Framework is developed by University of Melbourne, Australia. Alchemi fully practices OOP (Object-Oriented Programming) in Grid Computing. It allows you easy to change your application to Grid Computing structure. The developer should deeply understand the OOP and Multi-threading Technology before operate the Alchemi for getting full power from Grid Computing.

Alchemi contains 2 parts, Alchemi Grid Manager and Alchemi Grid Executor. Manager is installed at Master Server. It provides the command ordering, result gathering, Node

Server Management and Progress Management. It is the brain in the Grid Computing Architecture. Single Architecture supports single Manager Only. Manager has a User Management Module. Administrator can operate full function, User can view the status report only and the Executor can join into the Grid Architecture only, it hasn't any Management Access rights. Executor is a program in Node Server, It contains a Listener to receive the job, run it and output the result and feedback to Master Server. Node Server needs not to install your own application. Master Server will deploy your classes to all Node Servers. So, you can control the class in Grid Environment most easily.

After to describe the basic grid computing environment in the project, now I will explain the modules in WADE. WADE contains 2 parts, Server and Node. Server operates the Job Separation and Job Receiver. Separation is for make a single thread to multi-thread level. Receiver is for collecting and gathering all completed job from all managed Node Servers. That means Manager is an input & output, the process will pass to Node Server. Node Server contains 4 parts in Job Processing: Site Crawler, Web Parser, Link Life Detector and Logging Storing. Site Crawler will generate a sitemap for Web Parser to do a detail web page analysis. Site Crawler will go through the whole website, all page location will store in Link Container, Web Parser will parse and capture available links from all crawled links in Link Container. Web Parser will complete all contained links. Link Life Detector will try to access the target URL to get their header and more information. If it is a web link, it will store in Link Container for Parser uses. If it is an image, document or sound, Link Life Detector can try to access it. If they haven't any response, it will pass the link to Logging Storing. Logging Storing can de-duplicate all stored log and add the description for HTML Error Code. After all modules completed their job, Node Server will pass the result (Log) back to Master Server. WADE Manager will gather and export a report in XML format for enterprise feed it.

4.2. Major Functions

Rapid Sitemap Generation

Site Crawler will use Regular Expression to generate a Virtual Sitemap through Grid Computing for Web parser operation. End-user can control the number of sitemap level for system crawling deeply. Regular Expression can provide rapid link identification It's better than common character identification around 75-80%.

Detail Error Logging

End User can trace back the error easily. WADE is different than other Link Checkers. WADE will provide the detail occurred path of link error to end-user for finding out the problem source. All log will store in XML, Enterprise can feed it through Web Server and store the result into their database for detail analysis.

Unlimited Error Identification

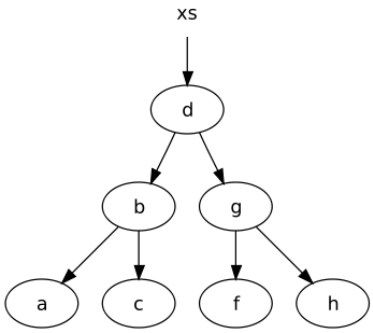
Web Parser does not only analyze webpage in the webpage, but also it can identify all file types and identify all the list of HTTP status codes (e.g. 200, 302, 404 and 502…etc) following the W3C Status Code Definitions for all the types of web page, image, document and large multimedia with Grid Computing Power. It also contains a Timer to control the expiry time of crawl waiting to avoid the website down. The referenced list of HTTP Status Codes is in **Appendix 8**.

4.3. Methodology

In the WADE design, it used some methodologies to implement the project. WADE cannot miss anyone of them. Every methodology has their function to complete some important processes of calculation. It includes Binary Tree Object Model, Multithreading, Regular Expression and Grid Computing. This chapter will explain what are they and what are the practical application.

4.3.1. Binary Tree Object Model (BTOM)

Through the BTOM, system can analyze all the level of web structure regularly. It also provide easy node finding method to the program for seeking target web element.



Definitions for rooted trees

- Edge-oriented refers to the mother-child (the arrows are the image of the fruit).
- Root node to node without parents is a fruit. Node to the root of the tree is rooted.
- A leaf node is not the children.
- Is the depth of node n is the length of the path from the root node. Set of all nodes of the depth is sometimes called the tree level. Root node is at a depth of zero.
- The height of a tree is the distance between the deepest node is a tree. A (rooted) tree with a node (the root) is in a height of zero.
- Siblings are nodes that share the same parent node.
- If a node P is a path node q, where p is the node nearest the root node q, then p is the ancestor is the EQF is a descendant P.
- Size of the node is the number of her descendants.
- Is the node degree is the number of edges to reach this node.
- Out-degree of a node is the number of edges leaving the node.
- Root tree node is the only curriculum = 0

4.3.2. Multithreading

System will apply **Amdahl's Law:** "…the performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used," (Hennessey and Patterson, 29) for increase each process performance.

$$\text{Speedup} = \frac{1}{\frac{\text{Fraction}_{\text{parallel}}}{\#\ \text{processors}} + (1 - \text{Fraction}_{\text{parallel}})}$$

The simplest multiple threading is where a thread runs until it is stopped by the event, which normally would result in long-latency stalls. These may be stand-cache miss, which is to reach out to the off-chip memory, which can take hundreds of CPU cycles for data to return. Rather than wait for a solution to the stand, which is threaded processor to switch execution to another thread, which was ready to go. Only when the data from the previous thread was received in the previous thread put back on the list of threads ready to run.

Conceptually, it is similar to many of the instructions used in real-time operating system that you voluntarily resign during the tasks when you need to wait until after a certain type of event.

Many of the threads to allow quick switching between blocked thread and another thread ready to run. To achieve this goal, equipment is the cost of reproduction of the program visible registers, and some of the CPU control registers (e.g. Program Counter). The transition from one topic to another thread is the use of switches from one register to another set.

This additional hardware such benefits:

- In the subject can be done in one processor cycle.
- It seems that every thread that they are self-execution and do not share any resources with any other hardware threads. This minimizes the amount of software necessary changes in the application and operating system to support multithreading.

Effectively switch between the active threads of each active thread must have its own set of registry. For example, to quickly switch between two threads, the register of equipment must be instantiated twice.

4.3.3. Regular Expression

RE can filter the target text (string) rapidly. It is a standard method to examine string and identifies parts that match the provided specification. I will use this formula to capture all links from the web.

```
(href|src)(=|
= )[\\\"\\\'](http:\\/\\/|\\.\\/|\\/)?\\w+(\\.\\w+)*(\\/\\w+(\\.\\w+)?)*(\\/|\\?\\w*=\\w
*(&\\w*=\\w*)*)?[\\\"\\\']
```

Regular Expressions (RES) is a mechanism to determine the selected chains, strings of characters. Regular expressions are independent of context, that the syntax of different fonts and character set ordering if its character sets the interpretation of the current location. Although the number of regular expressions can be interpreted differently depending on your current location, many features, such as an expression of nature, the contents of invariance across the region.

.	Match one character (except a lot of applications, newline, and figures that are exactly the taste and the platform specific newline character encoding, but it can be assumed that the lines are included). During the POSIX expression, location, literally meet the points. For example, the C "ABC", etc., but [C] is the only "A" and "." Or "c".
[ ]	A grouping expression. Corresponds to a character, which is closed in parentheses. For example, [ABC] matches "a", "b" or "c". [az] specifies the number corresponding to all lower-case letter "a" z ". These forms can be confused with: [abcx-z] matches' a ',' b ',' c ',' x ',' y 'or' Z 'and [a-CX-Z]. To - character is an accurate, if this is the first or last character in parentheses, or if it is escaped with backslash: [ABC-] [-ABC] or [a \-BC].
[^]	Matches a single character that is not included in parentheses. For example, [^ ABC] matches any character except "a", "b" and "C". [^ AZ] matches any single character is a letter from "A" with "." As before, the literal characters and ranges can be mixed.
^	Matches start position in the chain. Subject-based tool, it corresponds to the starting position of each row.
$	Coincides with the end position or the position of the string just before a string-ending newline. In accordance with the basic tools, putting an end to coincide with the position of any line.

\ (\)	Defines a marked sub expression. The string matched by the parentheses can be recalled later (see the next track, \ n). A marked sub expression is also called a block or capture group.
\n	Matches the nth marked sub expression harmonized, where n is a digit from 1 through 9. This construct is theoretically irregular and was not adopted in the POSIX ERE syntax. Some tools allow the claims of more than nine-capturing groups.
*	Matches the preceding element zero or more times. For example, ab * c matches "ac", "ABC", "abbbc" etc. [xyz] * matches "", "x" "y" "z", "zx", "zyx", "xyzzy" and so on. \ (Ab \) * matches "", "AB", "Ababa", "abababab ', and so forth.
\(m, n\)	Matches the preceding element at least m and at most n times. For example, \ (3.5 \) matches only "aaa", "aaaa" and "AAAAA". This is not found in some older cases, the regular expressions.

4.3.4. Grid Computing

Grid computing equipment and software, with more than one processing element or storage elements, processes, or more than one program, which freely or tightly controlled regime.

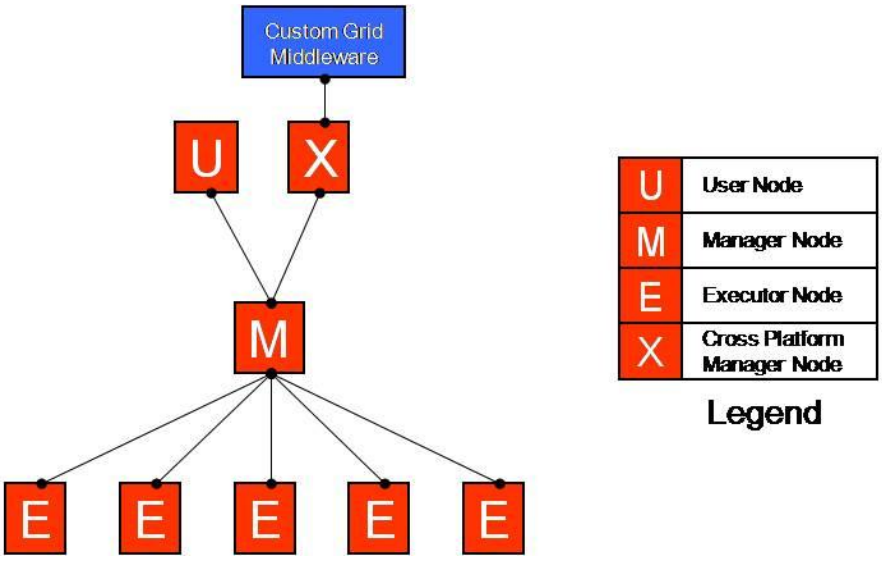
Today there are many definitions of Grid computing:

In Ian Foster's article "What is the Grid? A Three Point Checklist, he lists these primary attributes:

- Computing resources are not administered centrally.
- Open standards are used.
- Nontrivial quality of service is achieved.

The distributed program is divided into sections that operate at the same time a number of computers to communicate over the network. Broadcast is a type of parallel processing, but is also frequently used to describe parts of a program running simultaneously on multiple processors on the same computer. Both methods require the division of lots, which can operate simultaneously, but distributed by the program are often associated with a heterogeneous environment, various network delays and unexpected failures in the network or computers.

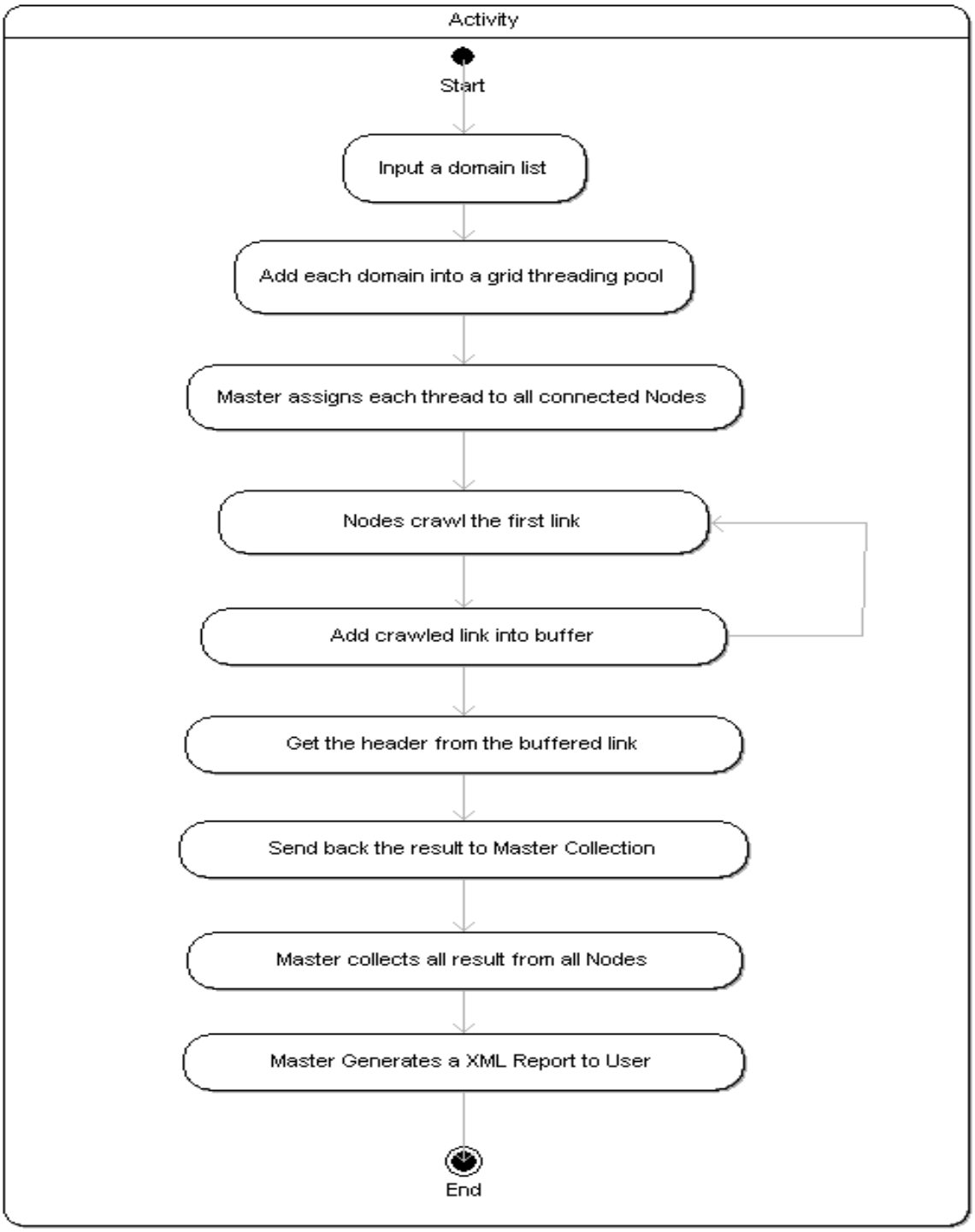
Organizing the interactions between computers that perform distributed calculations are extremely important. For the widest possible use of different computers, protocol or communication channel should not use any information that cannot understand some of the machines. Special care must be taken that the messages are indeed correct and that invalid messages, which would reduce the system and possibly the rest of the network are rejected. Another important factor is the ability to send software to another computer portable way so that they can meet and work with the existing network. It is not always practical in the use of different hardware and resources, in which case other methods, such as cross-compiling or manually porting this software to be used.



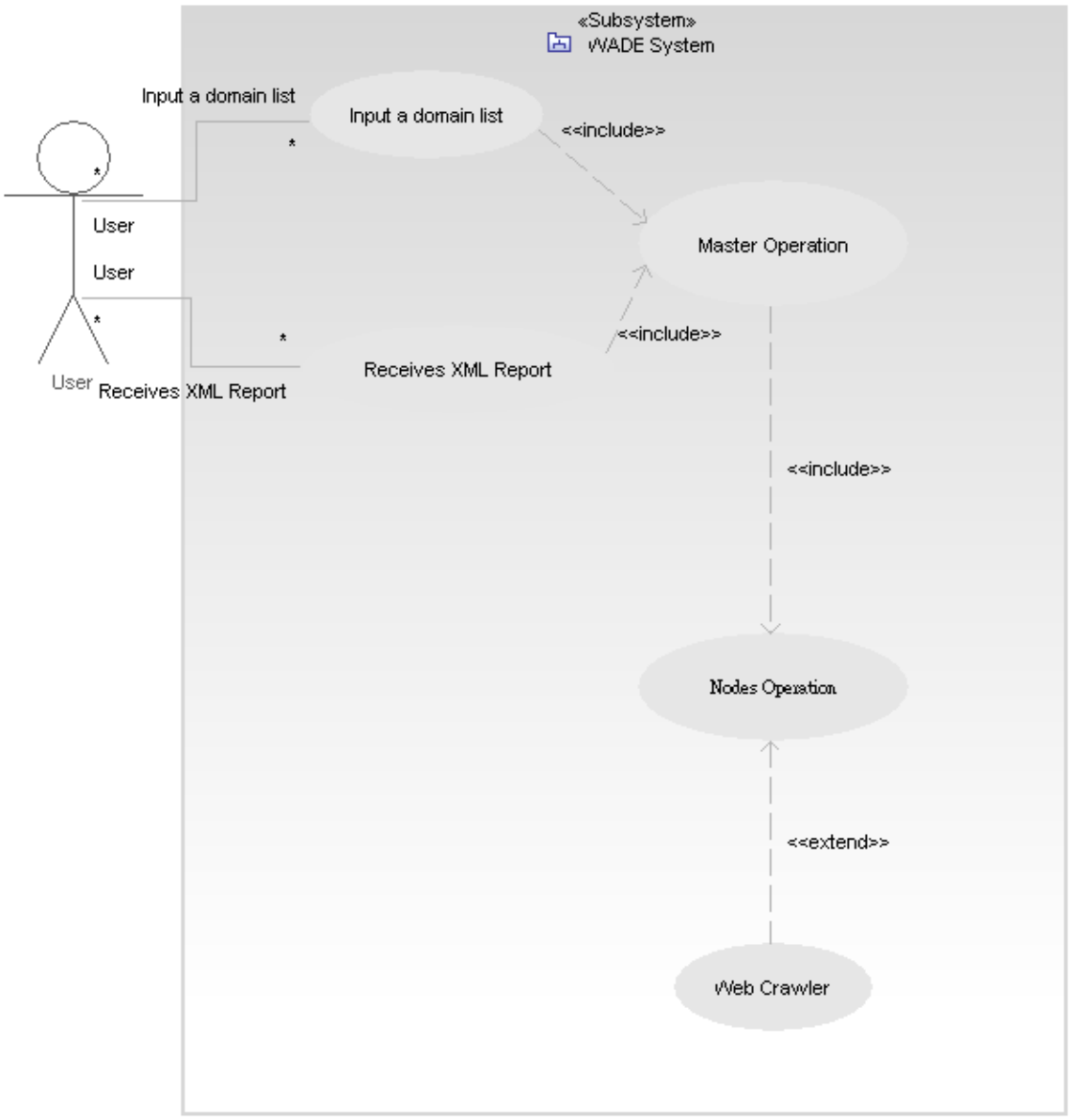
Grid Computing allows the system split the job and distribute to other node servers for increasing the speed of analysis calculation. Grid Computing has been applied to different scientific problems through loosely-coupled computers, and it is used in commercial enterprises for data analysis and processing in back-end.

4.4. UML

4.4.1. Activity Diagram

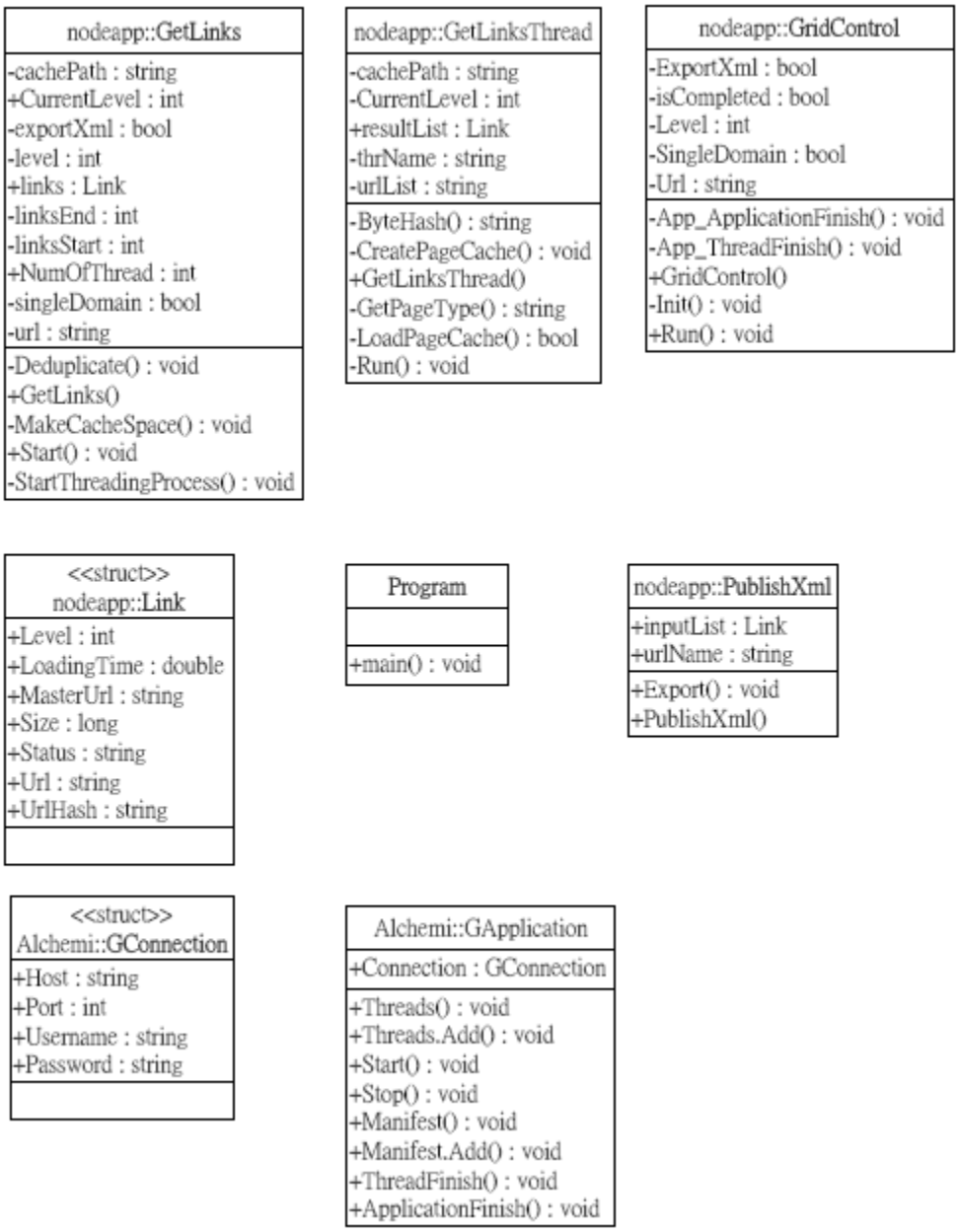


4.4.2. Use Case Diagram

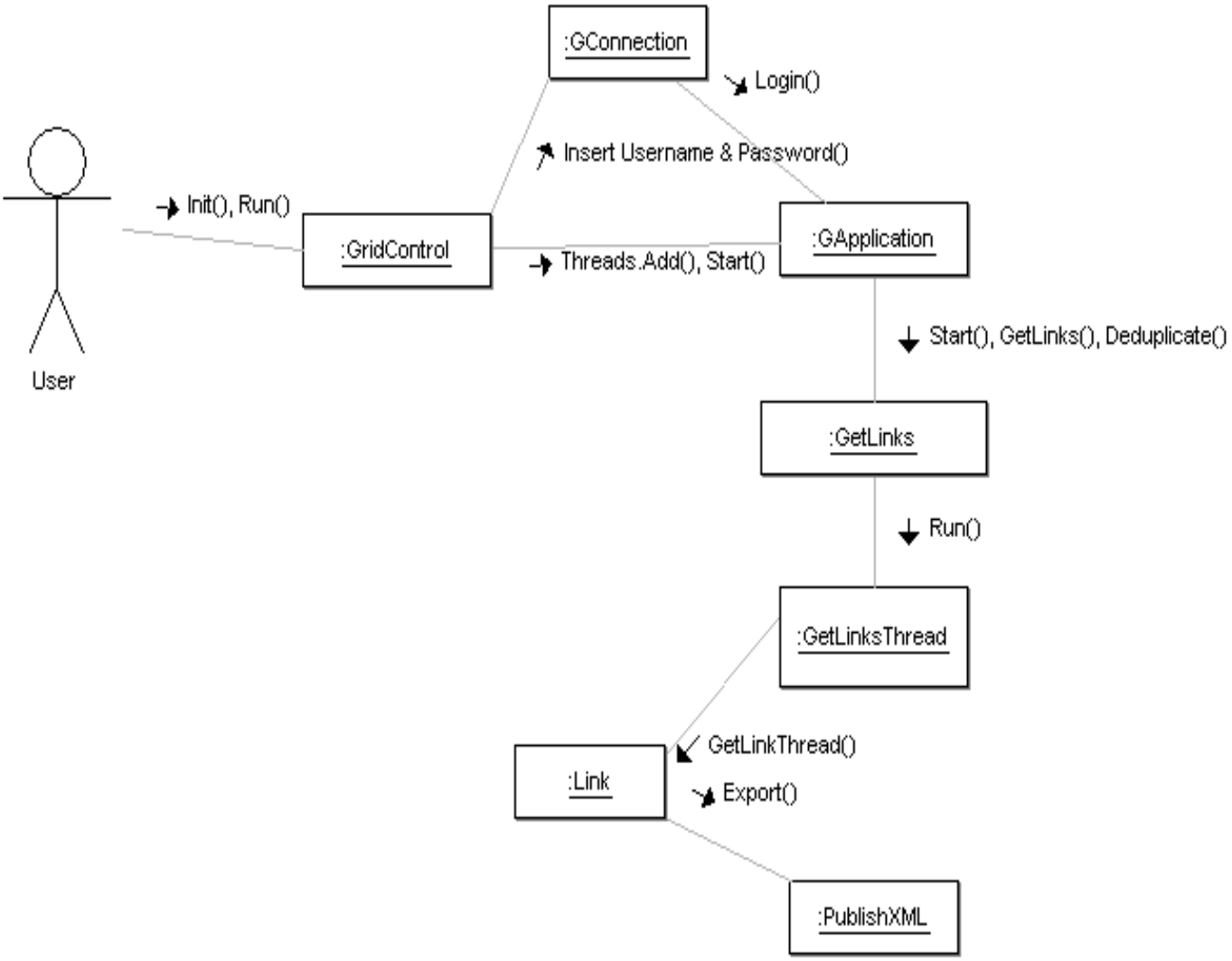


<<User check the link through WADE>>
1. Input a domain list
2. Add each domain into a grid threading pool
3. Master assigns each thread to all connected Nodes
4. Nodes crawl the first link
5. Add crawled link into buffer
6. Get the header from the buffered link
7. Send back the result to Master Collection
8. Master collects all result from all Nodes
9. Master Generates a XML Report to User
10. User receives the Report

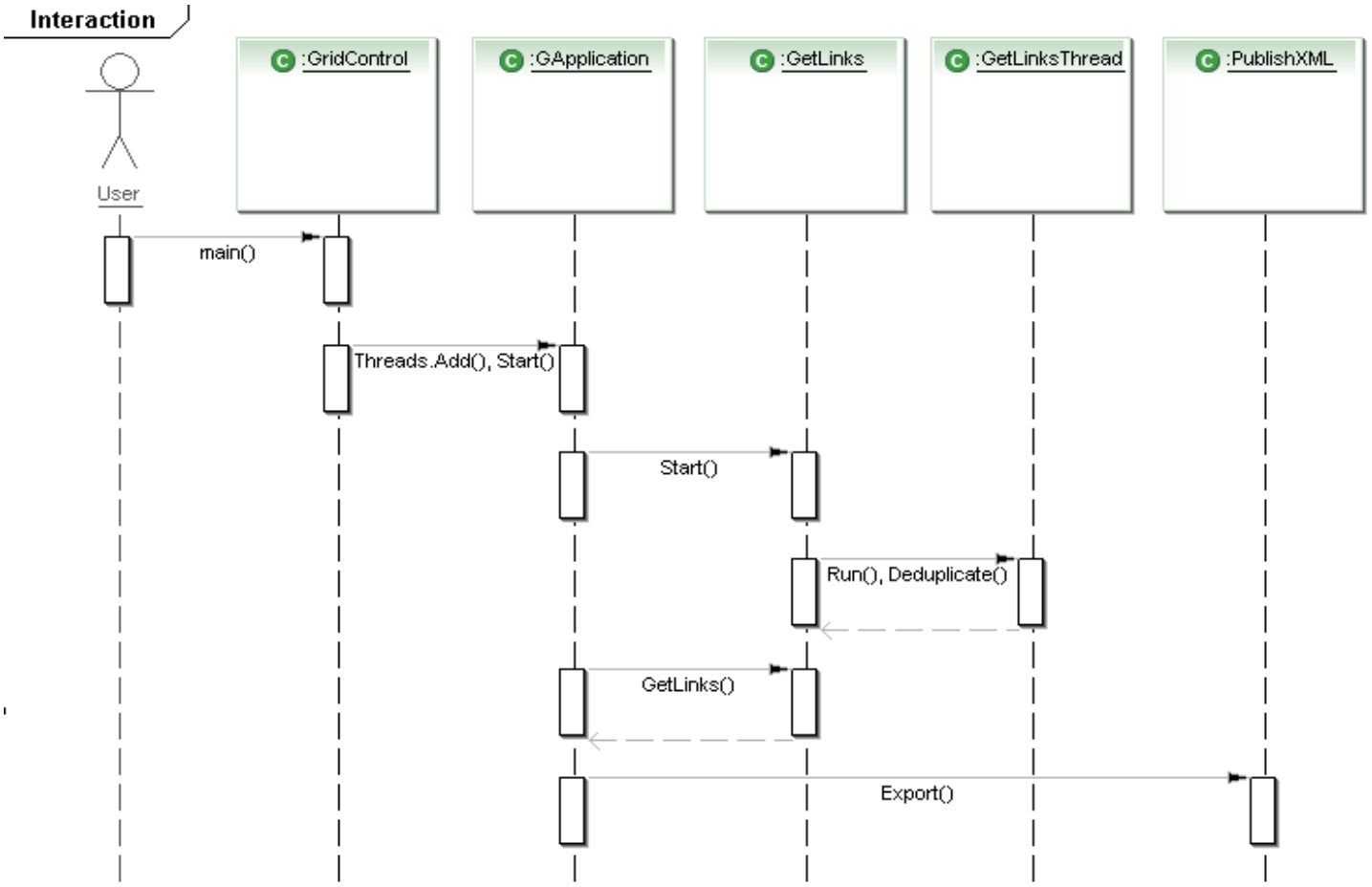
4.4.3. Class Diagram



4.4.4. Collaboration diagram



4.4.5. Sequence Diagram



4.5. Chapter Summary

This chapter covered much information about the system analysis. It included System Architecture, System Workflow, major functions of WADE, Methodologies and UML Diagram. I spend a lot of time to complete this part of documentation. I need to totally understand the methodologies that will be used. WADE is a Database-less design, because it is a Just-In-Time Web Analysis Tool, so it won't include a database. I used a plaintext document format to store the domain list, XML standard to keep the exported report. It's a new method to provide service;

Chapter 5. System Testing & Implement

5.1. Overview

Establishing software testing objective is an important part of planning the software testing cycle. Defining testing objective is also the difficult part of test planning activities. Cause of human always does not have a clear idea about what they want to do when they are beginning to do. So the best of laid test plans change during the process of test execution. That is a issue without a planned solution, there are some action testers can take for improving the test plan.

The establishment of clear testing objectives is a better way to reduce the future execution problems with less time. The tester should understand the objective of testing before execute the test plan, otherwise the test is not meaningful.

The goal of the testing is related to well-defined objective. It is a statement of the tester who planned to get the expected results from the specific testing activities. Each testing activity may have several objectives and two levels of objective specification to fulfill the tester expectation. A perfect test plan should include high-level general objectives in the initial section and specific low-level expected objectives for each particular type of the testing before implementation.

In this session, I will use Scenario Testing to draw a Testing Plan for the next system evaluation.

5.2. Test Scenarios Summary

Scenario Testing ensures the system operation flow that is successful and stable. Detail Scenario can increase the accuracy of testing to reduce the error that may occur at the real situation.

When the test scenario had executed, the Pass / Fail column should be changed. That is the indication of the status which is associated with test scenario. The Pass/ Fail should not indicate as Pass when the scenario has not completed. If the scenario is completed, the Pass / Fail should change to pass.

There are the scenarios which will be executed:

PASS / FAIL	Test Scenario ID	Test Description for this Scenario	Total Test Procedures for this Scenario
	S001	Setting up a Master Server	8
	S002	Setting up Node Servers	3
	S003	Nodes connect to Master Server	2
	S004	View status through Grid Management Console	5
	S005	Execute WADE	6

5.3. Testing Scenarios

Test Description S001- Setting up a Master Server

PASS / FAIL	Sequence of Execution #	Procedure	Setup / Initialization	Data
	1	PCD1001	Install Microsoft .NET Framework 3.5	None
	2	PCD1002	Install Windows Installer 4.5	None
	3	PCD1003	Restart the Server	None
	4	PCD1004	Install SQL Server 2008 Express	None
	5	PCD1003	Use PCD1003	None
	6	PCD1005	Install Alchemi Manager	None
	7	PCD1003	Use PCD1003	None
	8	PCD1006	Execute Alchemi Manager	Status to be Running

PASS / FAIL	Test Procedure [PCD1001] Step
	Ensure the Server is connected to Internet
	Download Microsoft .NET Framework 3.5 From Microsoft Website
	Execute the .NET Framework Installation File
	Pass next to continue the setup
	Waiting to complete the installation
	Ensure the Installation is successfully through viewing the Control Panel > Add and Remove Program
	Test Procedure [PCD1002] Step
	Ensure the Server is connected to Internet
	Download Windows Installer 4.5 From Microsoft Website
	Execute the Windows Installer Installation File
	Pass next to continue the setup
	Waiting to complete the installation
	Ensure the Installation is successfully through viewing the Control Panel > Add and Remove Program
	Test Procedure [PCD1003] Step
	Pass Start > Shutdown > Restart
	Waiting for the Server restart and show the Login Screen
	Login as Administrator again for the next progress
	Test Procedure [PCD1004] Step
	Ensure the Server is connected to Internet
	Download SQL Server 2008 From Microsoft Website
	Execute the SQL Server 2008 Installation File

	Pass next to continue the setup
	Enter the SQL Instance Name as "SQLEXPRESS"
	Grant the Administrator Right to SQL Account
	Ensure the SQL Server will start with Windows
	Waiting to complete the installation
	Confirm the SQL Server installation is successfully through view the status in Service > SQL Express
	Test Procedure [PCD1005] Step
	Ensure the Server is connected to Internet
	Execute the Alchemi Manager Installation File
	Pass next until complete the installation
	Select the SQL Server Name, Enter the Administrator Name and Password for completing the configuration
	Test Procedure [PCD1006] Step
	Click the Alchemi Manager icon that is stored in the desktop
	Waiting the status that will display after Alchemi Manager needs few seconds to load the system configuration

Expected Results

1. Microsoft .NET Framework 3.0, Windows Installer 4.5, SQL Server 2008 should be installed successfully before installing Alchemi Manager. All Server restart operation should not have any error occurrence.
2. After all installation completed, Alchemi Manager should be executed successfully. The status of Alchemi Manager should monitor all registration from all nodes.

Test Description S002- Setting up Node Server

PASS / FAIL	Sequence of Execution #	Procedure	Setup / Initialization	Data
	1	PCD1001	Use PCD1001	None
	2	PCD1003	Use PCD1003	None
	3	PCD2001	Install Alchemi Executor	None

PASS / FAIL	Test Procedure [PCD2001] Step
	Ensure the Server is connected to Internet
	Execute the Alchemi Executor Installation File
	Pass next until complete the installation

Expected Results

1. Microsoft .NET Framework should be installed successfully.
2. Restart the node should not have any error occurrence.
3. Alchemi Executor installation may not occur any error.

Test Description S003- Nodes connect to Master Server

PASS / FAIL	Sequence of Execution #	Procedure	Setup / Initialization	Data
	1	PCD3001	Execute Alchemi Executor	Running
	2	PCD3002	Connect to target Master	Connected

PASS / FAIL	Test Procedure [PCD3001] Step
	Click the Alchemi Executor icon that is stored in the desktop
	Waiting the status that will display after Alchemi Executor needs few seconds to load the system file
	Get the status is running
	Test Procedure [PCD3002] Step
	Enter the Master Server IP Address, Default Executor Right Name and Password in configuration page to complete the installation
	Pass "Connect" for connecting the Master
	Waiting the feedback from Master Server
	Get the status is connected

Expected Results

1. The execution of Alchemi Executor may not occur any error when starting the application.
2. Assume the configuration correct, the Alchemi Executor should connect to Master Server successfully.

Test Description S004- View status through Grid Management Console

PASS / FAIL	Sequence of Execution #	Procedure	Setup / Initialization	Data
	1	PCD4001	Install Alchemi Management Console	None
	2	PCD4002	Execute Alchemi Management Console	None
	3	PCD4003	Read the number of registered node	None
	4	PCD4004	Read the total power of Grid Environment	None
	5	PCD4005	Read the current job in each node	None

PASS / FAIL	Test Procedure [PCD4001] Step
	Ensure the Server is connected to Internet
	Execute the Alchemi Management Console Installation File
	Pass next until complete the installation
	Test Procedure [PCD4002] Step
	Enter the Master Server IP Address, Default Administrator Right Name and Password in login page to access the Grid Management Console
	Pass "Login" to access the Master Server
	Waiting the feedback from Master Server
	Test Procedure [PCD4003] Step
	Click the "Node" Tag, All registered node will record in there, if the node is offline, the icon will change to black, online is red
	Test Procedure [PCD4004] Step
	Click the "Statistic" Tag, All Power Information will display in this page. If the number of registered node is 6, the total power will sum all registered node.
	Test Procedure [PCD4005] Step
	Click the "Node" Tag, and select a Node with right click, it will display the current assigned job in the selected node/

Expected Results

1. If the Administrator Name and Password is valid, the login should be successfully.
2. If all nodes are registered successfully, PCD4003, PCD4004 and PC4005 should display correct information

Test Description S005- Execute WADE

PASS / FAIL	Sequence of Execution #	Procedure	Setup / Initialization	Data
	1	PCD5001	Install WADE	None
	2	PCD5002	Execute WADE	None
	3	PCD5003	Start the Crawl Operation	None
	4	PCD4005	Use PCD4005	None
	5	PCD5004	View the statistic at the end	None
	6	PCD5005	View the exported Report	None

PASS / FAIL	Test Procedure [PCD5001] Step
	Ensure the Server is connected to Internet
	Execute the WADE Installation File
	Pass next until complete the installation
	Test Procedure [PCD5002] Step
	Click the WADE icon that is stored in the desktop
	WADE will wait you to enter the location of Link Data File
	Test Procedure [PCD5003] Step
	Enter the location of Link Data File
	Press "Enter" to submit the location, it will capture the file and start the crawling
	Test Procedure [PCD5004] Step
	When the WADE completed the crawling, it will show the statistic about the mission information, such as how long does it work and how many page does it crawl. Ensure the information is near real timer
	Test Procedure [PCD5005] Step
	WADE will export the report for each domain. All report is stored in XML format. Open the report to basic check the domain information

Expected Results

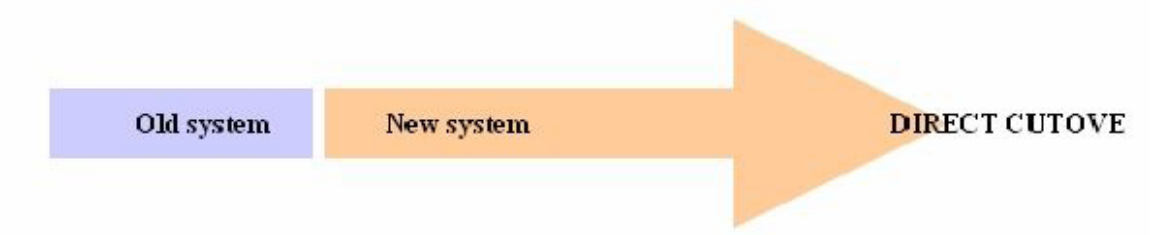
1. If all embedded configuration is correct, WADE will not have any error occurrence in initial running.
2. If the location of Link Data File is valid, WADE will not occur error.
3. If the Node and Internet is correct, WADE can assign Node to crawl the web smoothly.
4. If the crawling is successfully, the number of XML report should the same with the number of domain that are in Link Data File.

5.1. System Implementation plan

System implementation means going live the planned system. The stage of system development in approved the hardware and software requirement, confirmed the testing and documentation and completed the user training of system operation.

System implementation plan puts the new system live and replace the old system. There are four main styles of changing from old to new system. Each approach involves different cost and the factors of risk.

Direct Conversion is force the new system online and replacing the old system immediately. That is a quick and low-cost transition, but it may be unnerving if the changes have too big different. New System must be tested completely and ensure all users are totally understanding to whole system operation. If the new system goes live without all confirmation, it may occur many unexpected issues when the user operating the system. Error has a direct impaction on the users and the whole organization and the extent dependant on the centrality of the new system. Finally, Direct Conversion provides an opportunity for reducing old system problems. It may require the installation is completed at once and meticulous planning. Nevertheless it is not a better way to compare the result between old system and new system.



Phased Conversion is a changeover with phase. The system is implemented in managed stages or modules across the organization. Phased Conversion gives part of the system to entire organization, and it's an incremental and gradual change from the old system to new system. New system expands much functional components that are brought on-line and the old system and new system will use the same time as co-operation.

If the new system has new several components, it can be promoted at the initial time. Phased Conversion is impossible of the system is a single module, for example, when introducing a new hard disk that you cannot increase a little piece of a hard disk at a single period, it has to be full or do nothing.

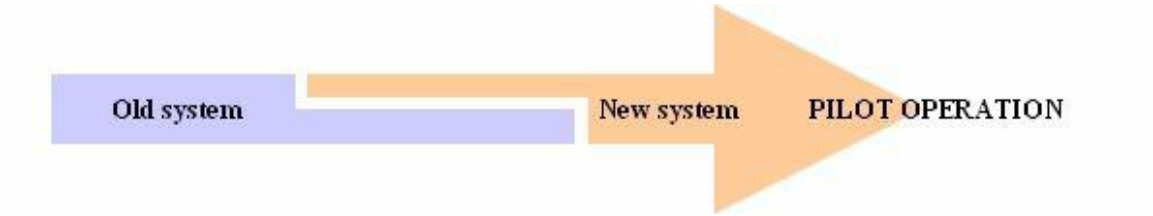
Old system and new system need to share the same data therefore a requirement for linking up the program in need. Both of them may be incompatible therefore phased may not be suitable. Phased Conversion needs a managed control. Cost and Risk is relatively moderate, because the system is implemented by stage-based with managed control.

Pilot Conversion means running the new system and old system at the same time, but only at a selected place, called a pilot site. It has been limited the potential risk when the other branches or departments are not converted all old problems are solved in new system. It's only worked in the same function system that is introduced many times across whole organization. It acts a trial before extending the deployment.

If the organization has several departments and located in different cities, Manager should decide where the site should be test the new system. Any errors and problems will be limited in a single location and it won't affect the whole organization.

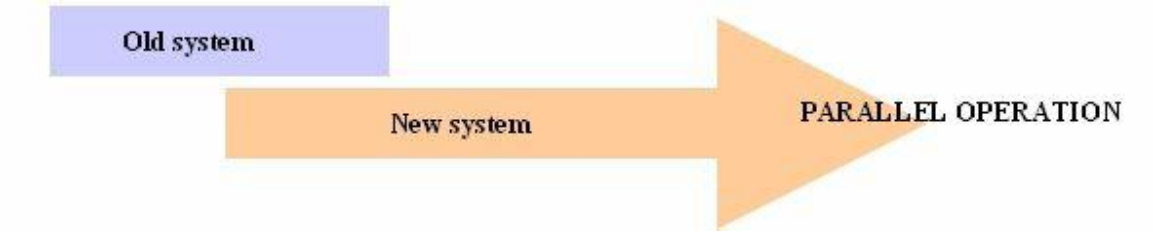
The testing is in a managed location, so that the cost and risk are in controlled. The success at the pilot location can help to convincing an organization to deploy the whole system. That is a good planning to get the experiences for deployment. All deployment problem can be solve in little location to avoid occur a big issue at a real whole site.

The application of Pilot Conversion needs extra cost and resources. It needs to take a long timescales to try the potential of causing motivational issues.



Parallel Conversion means both the new system and old system are operated totally in a specified period. Date input into both system and make a confirmation to both results. After testing the whole workflow of the system and getting the same result, that means the new system is suitable and matched with old data, then all users can be moved to new system and continue their work.

Parallel Conversion is only for if the old system and new systems are totally independent. If the new system and old system are not matched, this method is useless.



System Implementation Justification

Direct Conversion can be implemented in WADE project, because it is a new installation for the servers. It doesn't include any result comparison of implementation. Administrator needs not concern any fault handling. Direct Conversion is the fastest and low budget solution for implementation.

5.2. Tools Required For System Implementation

Selected Development Framework:	Microsoft .NET Framework 2.0
Selected Grid Computing Environment:	Alchemi
Selected Operation System:	Microsoft Windows Series
Selected Report Format:	XML

All additional specification is in **Appendix 9**.

The system can be installed into selected hardware specification for the testing purpose. The Steps of Software Installation shows you how to install the WADE and setup Grid Computing Environment. Before the evaluation, it is the first step that must be done.

The Detail Steps in **Appendix 10**.

5.3. User Training and Manual

User training and manual are going to avoid the error occurrence when they are using the system in progress. It is importance phase before implement WADE. A great training and detail manual can reduce user unexpected action after WADE implemented.

User Training

User training should be archived the following objective:

1. All administrators should understand the operation of WADE
2. All administrators should have a knowledge of Grid Computing Environment
3. All administrators should know the installation of Node
4. All sales should know how to explain the new service to end-users
5. All end-users should understand new report architecture
6. All end-users should know how to feed the report through web service

Training Location

Training course should be held at different sites for each type of user. Administrator should take the course in Server Room. Sales should take the course in Conference Room and the end-users (client) may take the course at their place. Out consultant will offer a onsite presentation to them.

Training Period

Company should provide 2-3 training in office hour to ensure all target can attend the courses. Each course should be around 30minutes to 60 minutes. WADE is a Server-based Application, so end-users need not to take much time to learn the back-end technologies.

5.4. Chapter Summary

This chapter included all progress and decision about the Testing Period. From this chapter, I learned how to create a Test Plan, why we need inactive with end-user. It is a great experience about communicating with the users and administration.

Chapter 6. Evaluation

There is the plan of WADE Evaluation:

1. Basic Demonstration
2. Dataset Sampling
3. Basic comparison between single computer and grid environment
4. Random comparison between single computer and grid environment
5. Detail comparison between Domain-Based and Page-Based

Basic Demonstration will show you how to control the WADE and reading the XML report.

Dataset Sampling will select 168 website from large dataset randomly for evaluation. All websites are captured from Google.com through the core of WADE.

Basic comparison will show you what is the different of hardware performance between single computer and grid environment.

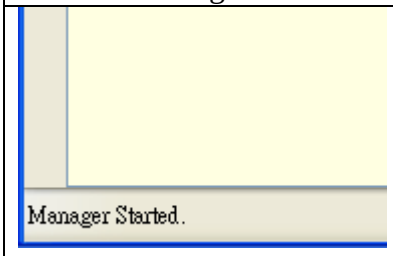
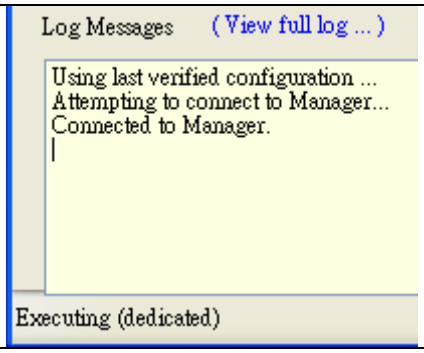
We will evaluate the WADE through Random comparison in 30 websites. It will show the simple result of crawling performance between single computer and grid environment.

Finally, we will try to understand what is the different between Domain-Based and Page-Based, so we can know what the bottleneck of performance in Process Separation is. The result can help the following researcher to choose a right solution.

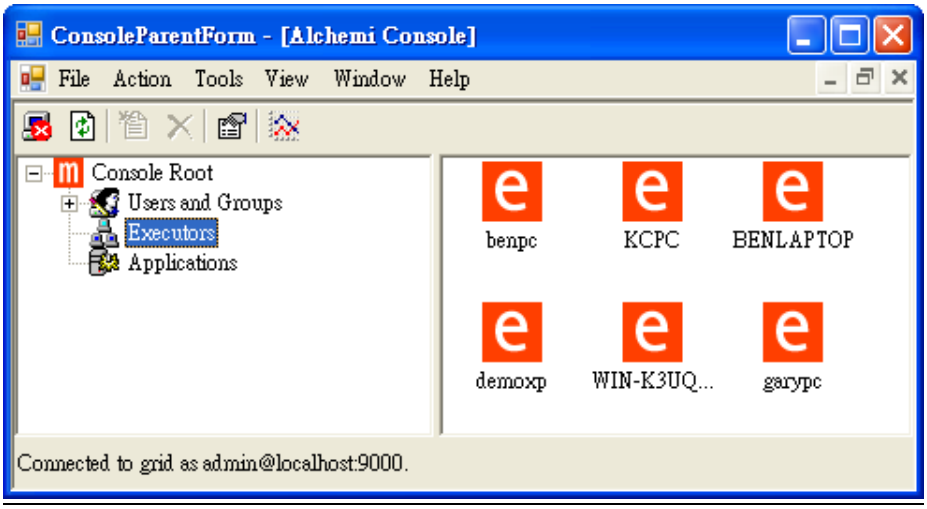
6.1. Basic Demonstration

First of all, I prepared the link list for the WADE scanning:

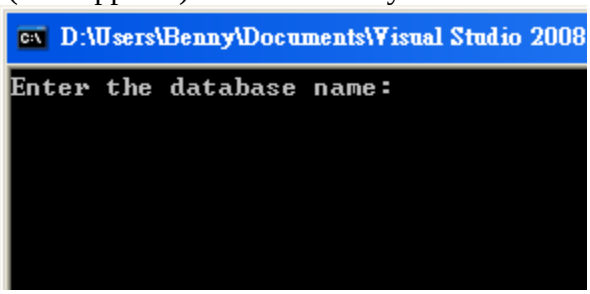
Ensure Alchemi Manager is running and all Alchemi Executors are connected to Alchemi Manager:

Alchemi Manager	Alchemi Executor
	

If all Executors are connected, you can see all nodes are available in Alchemi Console:

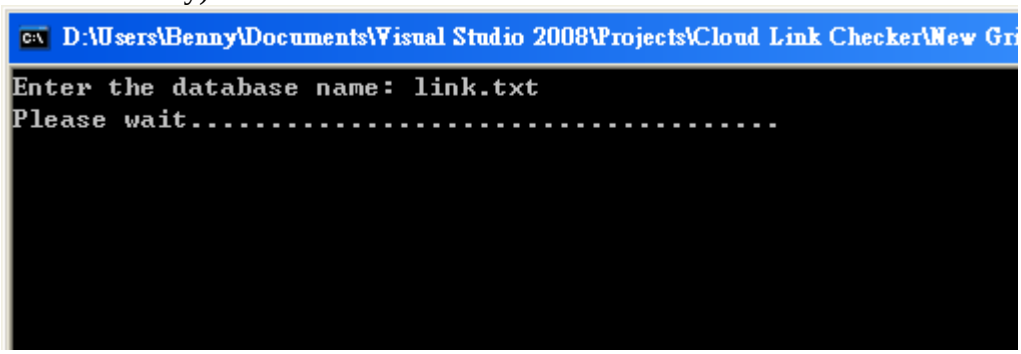


Now, Run the WADE (nodeapp.exe) and it will ask you the name of Link File:

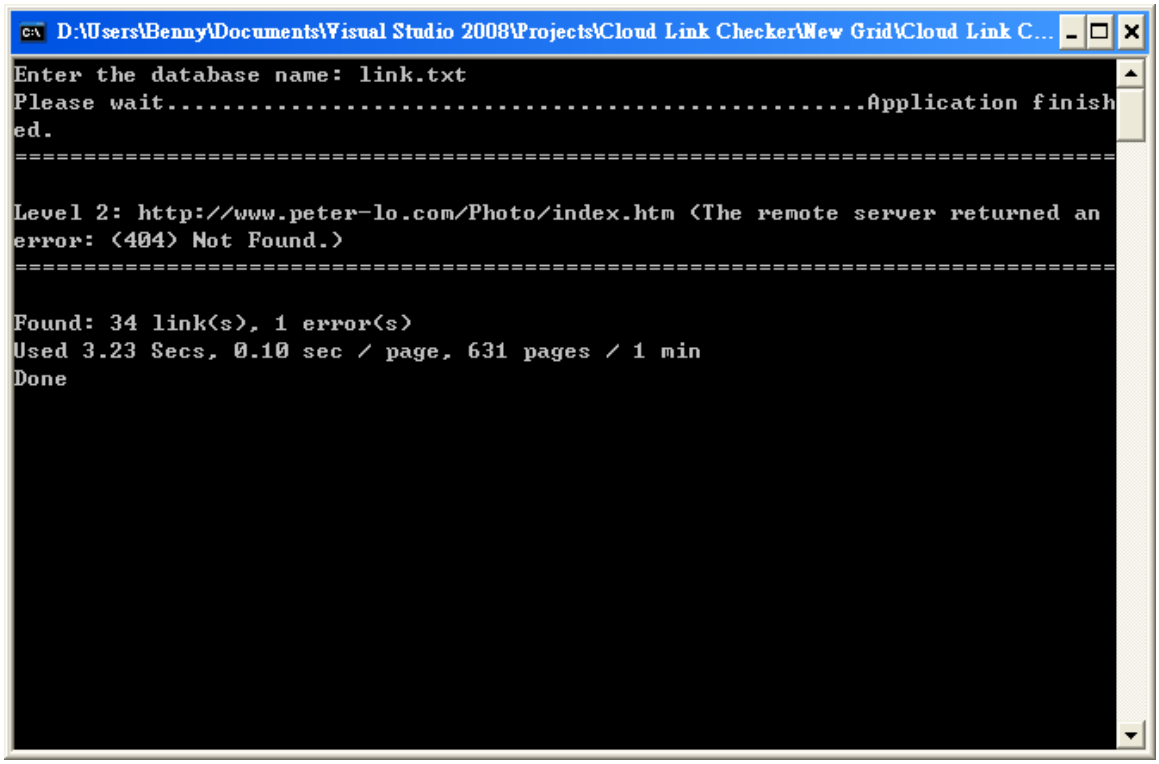


Please enter "link.txt" to select prepared Link File, then press Enter.

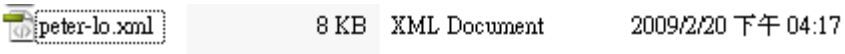
You can see WADE is running and please wait for it completed (Default the Crawler will capture 2 levels only):



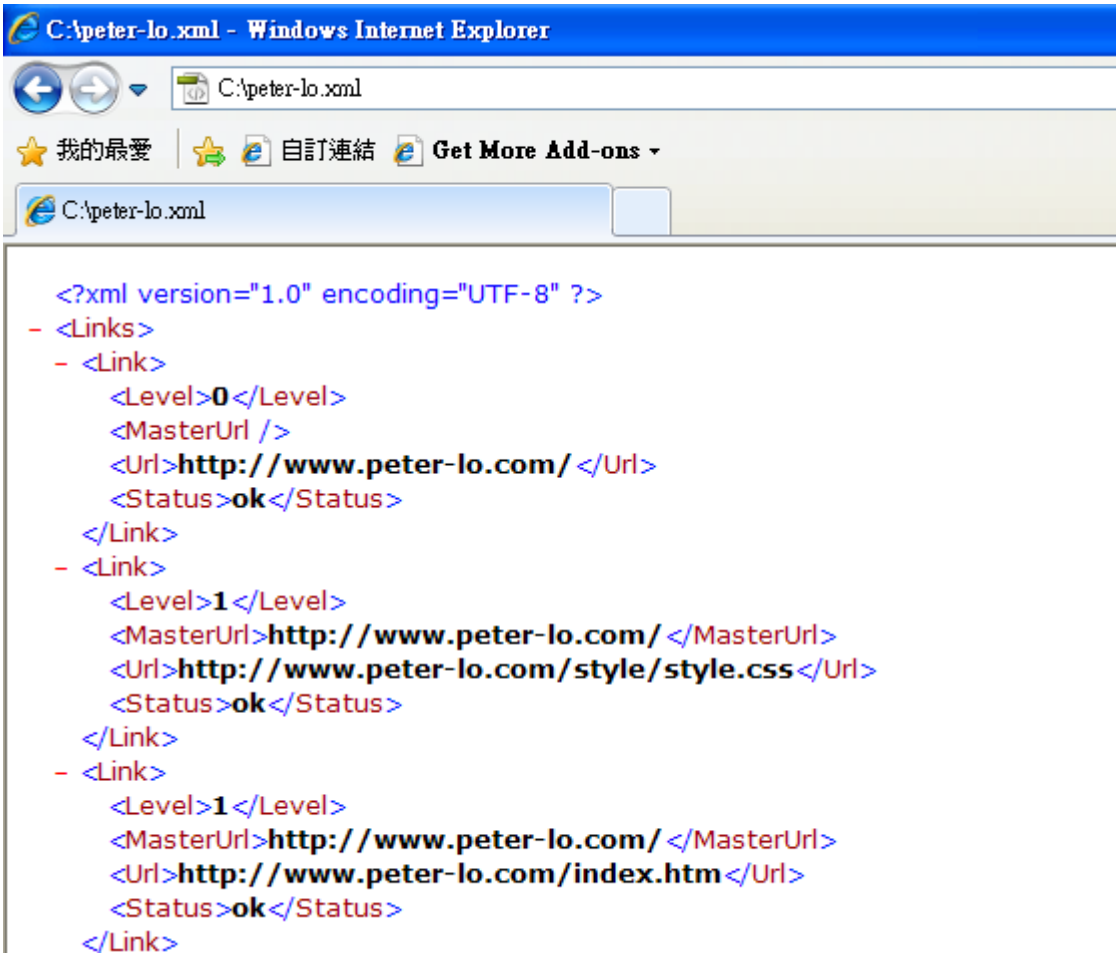
When the crawling is completed, you will get the detail information about concurrent mission:



The report will be exported to "C:\peter-lo.xml":



Developer can move it to Web Server or import it to database for enterprise feed it. There is the XML report content:


C:\peter-lo.xml - Windows Internet Explorer
C:\peter-lo.xml
我的最愛
自訂連結
Get More Add-ons
C:\peter-lo.xml
<?xml version="1.0" encoding="UTF-8" ?>
- <Links>
- <Link>
<Level>0</Level>
<MasterUrl />
<Url>http://www.peter-lo.com/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/style/style.css</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/index.htm</Url>
<Status>ok</Status>
</Link>

From the XML Content, You can trace the error location through (MasterUrl). MasterUrl means the Url is found in that page. That can easy to identify the problem is wrong typing, not the page or file lost. Level is telling you how deep the crawler phased. In this case, it just phased 2 levels (Default Setting).

For example:

```
- <Link>
  <Level>2</Level>
  <MasterUrl>http://www.peter-lo.com/contact/index.htm</MasterUrl>
  <Url>http://www.peter-lo.com/contact/images/</Url>
  <Status>ok</Status>
  </Link>
```

The *Url* is found in *MasterUrl*, and *MasterUrl* is found at *Level 2*, the *Url Status* is *OK*.

If the link is available and no error, the Status will show OK, if the link is unreachable, the Status will be the Html Error code with the Error Description. It can notify the user in detail. In this case, *http://www.peter-lo.com/Photo/index.htm* is 404 Not Found. WADE will add a description to it: **The remote server returned an error: (404) Not Found.** When the enterprise user feed it, they need to parse the Error Code and easy to know what is the problem in the link. Error Code Description is following the standard of www.w3.org/Protocols.

Through this XML Report, Website Administrator can trace the error easily and rapidly. In other applications, you also can get the error, but you never know where the error is. WADE provides MasterUrl Information. Administrator need not search all document to find the error location. It's clear and readable XML Report. Each website has a single XML Report. WADE is database-less architecture design, all data will not store in the server. All analysis is running just in time. Every Crawling will replace the existing XML Report, so the Enterprise should feed it per day or per hour.

6.2. Data Sampling

I prepared 996 non-duplicated links that are the website of education for evaluation. All links are captured from Google.com. There are some links in the database:

http://www.iipm.edu/	http://www.oberlin.edu/	http://www.ajula.edu/
http://www.mc3.edu/	http://www.vsc.edu/	http://www.ceram.edu/
http://www.chapman.edu/	http://www.sbts.edu/	http://www.nap.edu/
http://www.fccj.edu/	http://www.saintpaul.edu/	http://www.utsouthwestern.edu/
http://www.univdhaka.edu/	http://www.uncc.edu/	http://www.chestercollege.edu/
http://www.mcw.edu/	http://www.salem.edu/	http://www.csufresno.edu/
http://lakeland.edu/	http://www.umich.edu/	http://www.mbl.edu/
http://www.uh.edu/	http://www.ovu.edu/	http://www.neco.edu/
http://www.transy.edu/	http://www.mssu.edu/	http://www.wcjc.edu/
http://www.pari.edu/	http://www.uccs.edu/	http://www.swlaw.edu/
http://www.highpoint.edu/	http://www.digipen.edu/	http://www.antiochne.edu/
http://www.icsi.edu/	http://www.ccccco.edu/	http://www.montana.edu/
http://www.agsird.edu/	http://www.nols.edu/	http://www.jhsph.edu/
http://www.jessup.edu/	http://www.rockefeller.edu/	http://www.udel.edu/
http://www.ucollege.edu/	http://www.sloankettering.edu/	http://www.alquds.edu/
http://www.colbycc.edu/	http://www.lclark.edu/	http://www.bhu.edu/
http://www.ohsu.edu/	http://www.iilm.edu/	http://www.deanza.edu/
http://www.eurasia.edu/	http://www.com.edu/	http://www.cn.edu/
http://www.oaklandcc.edu/	http://www.wts.edu/	http://www.belhaven.edu/
http://www.iu.edu/	http://www.pebblehills.edu/	http://www.elmira.edu/
http://www.eustatiusmed.edu/	http://www.whoi.edu/	http://www.manipal.edu/
http://www.rasmussen.edu/	http://www.pba.edu/	http://www.gcu.edu/
http://www.dts.edu/	http://www.dmacc.edu/	http://www.newbury.edu/
http://www.lasierra.edu/	http://www.fhsu.edu/	http://www.sdcity.edu/
http://www.sjcl.edu/	http://www.tsbvi.edu/	http://www.sasin.edu/
http://mansfield.edu/	http://www.career.edu/	http://www.cdrewu.edu/
http://www.brynmawr.edu/	http://www.tamiu.edu/	http://www.loyno.edu/
http://www.ciachef.edu/	http://www.nsula.edu/	http://www.biola.edu/
http://www.devry.edu/	http://www.morrisville.edu/	http://www.isunet.edu/
http://www.bethlehem.edu/	http://www.mnwest.edu/	http://www.bucknell.edu/
http://www.stcl.edu/	http://www.multimedia.edu/	http://www.svots.edu/
http://www.uniminuto.edu/	http://www.nrao.edu/	http://www.vccs.edu/
http://www.utm.edu/	http://www.mcg.edu/	http://www.gaston.edu/
http://www.tc3.edu/	http://www.willamette.edu/	http://www.phoenix.edu/
http://www.tricountycc.edu/	http://www.ucmo.edu/	http://www.uncp.edu/
http://www.usi.edu/	http://www.bethelks.edu/	http://www.kishwaukeecollege.edu/

http://www.ie.edu/	http://www.rowancabarrus.edu/	http://www.msjc.edu/
http://www.unh.edu/	http://www.dordt.edu/	http://www.pasadena.edu/
http://www.marianopolis.edu/	http://www.fredonia.edu/	http://www.bbc.edu/
http://www.thunderbird.edu/	http://www.oak.edu/	http://www.caspercollege.edu/
http://www.einstein.edu/	http://www.bxscience.edu/	http://www.stephens.edu/
http://www.oc.edu/	http://www.prgs.edu/	http://www.bhsu.edu/
http://www.ceu.edu/	http://www.cva.edu/	http://www.umbc.edu/
http://www.wpunj.edu/	http://www.berklee.edu/	http://www.uhd.edu/
http://www.uthscsa.edu/	http://www.radcliffe.edu/	http://www.indycc.edu/
http://www.ndm.edu/	http://www.utmb.edu/	http://www.euruni.edu/
http://www.fdu.edu/	http://www.westvalley.edu/	http://www.kent.edu/
http://www.csudh.edu/	http://www.nesl.edu/	http://www.tamut.edu/
http://www.goldenwestcollege.edu/	http://www.cwu.edu/	http://www.usw.edu/
http://www.maryville.edu/	http://www.davisandelkins.edu/	http://www.omsi.edu/
http://www.baystate.edu/	http://www.northcentralcollege.edu/	http://www.southalabama.edu/
http://www.cogswell.edu/	http://www.saic.edu/	http://www.northwestu.edu/
http://www.kutztown.edu/	http://www.catawba.edu/	http://www.uchospitals.edu/
http://www.park.edu/	http://www.emmaus.edu/	http://www.wou.edu/
http://www.rockvalleycollege.edu/	http://www.wallawalla.edu/	http://www.hccc.edu/
http://www.uapb.edu/	http://www.converse.edu/	http://www.dillard.edu/

Some website may be down and we don't know. So the WADE needs to identify the status of all different type of website. If the website is down, it should return Error Code 404 for user to identify. By the way, each website may be stored at different location, the data transferring response time may be different, and so the timeout detection needs to check the site status more flexible.

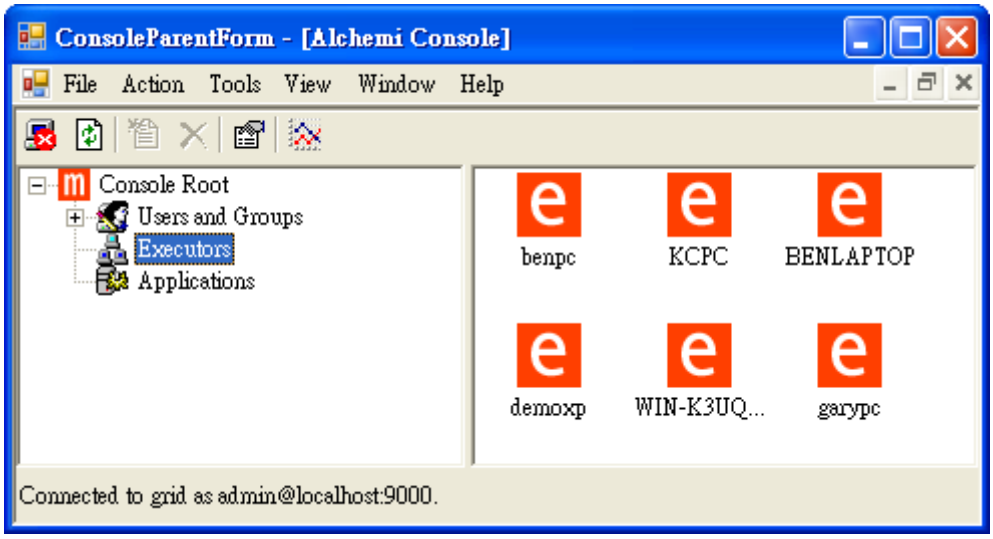
All sample links will store in "link.txt" for the next Demonstration use.

6.3. Basic comparison between single computer and grid environment

For a basic comparation between single PC and Grid Envirument, I added a single node into the Grid. The single PC contains *2.666GHz* computing power. It's not enough to handle a large computing mission. It's a basic desktop requirement.

No. of Executors	1	Max. Power Available	2.666 GHz	Current Power Available	99 %
No. of running Applications	0	Total Power Usage	3.7E-05 GHz*Hr	Current Power Usage	1 %
No. of running Threads/Jobs	0				

So in the next picture, 6 computers will be Nodes and logged into Master, Master will receive some technical information to understand the current status in the Grid Environment:



No. of Executors	6	Max. Power Available	10.64 GHz	Current Power Available	92 %
No. of running Applications	0	Total Power Usage	0.271629 GHz*Hr	Current Power Usage	0 %
No. of running Threads/Jobs	0				

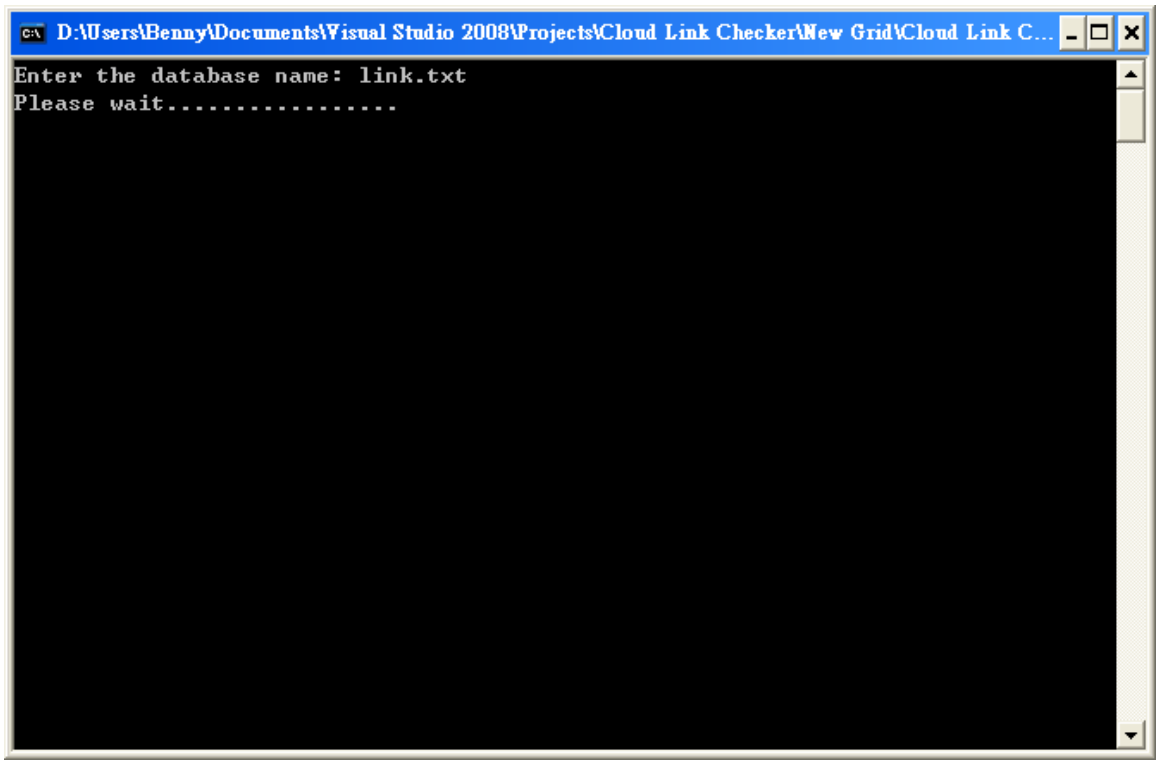
Through the picture, the Grid Environment contains 6 Executors and having *10.64GHz* Computing Power. That means it is basically over all existing powerful computer already. It can handle more calculation job now. The computing power is base on each Node CPU speed. For example, if I added 600 Nodes and each of them has 1.77GHz CPU, the total Computing Power will be *1,062GHz*. So you can understand why Google can use thousand and thousand desktops to be a Grid Computing Element and it is very fast. There is a simple comparation to see the different between single PC and Grid Envirument.

6.4. Random comparison between single computer and grid environment

In this part, I have prepared *30 random websites* for the next testing. There are the target website:

http://www.iipm.edu/	http://www.colbycc.edu/
http://www.mc3.edu/	http://www.ohsu.edu/
http://www.chapman.edu/	http://www.eurasia.edu/
http://www.fccj.edu/	http://www.oaklandcc.edu/
http://www.univdhaka.edu/	http://www.iu.edu/
http://www.mcw.edu/	http://www.eustatiusmed.edu/
http://lakeland.edu/	http://www.rasmussen.edu/
http://www.uh.edu/	http://www.dts.edu/
http://www.transy.edu/	http://www.lasierra.edu/
http://www.pari.edu/	http://www.sjcl.edu/
http://www.highpoint.edu/	http://mansfield.edu/
http://www.icsi.edu/	http://www.brynmawr.edu/
http://www.agsird.edu/	http://www.ciachef.edu/
http://www.jessup.edu/	http://www.devry.edu/
http://www.ucollege.edu/	http://www.bethlehem.edu/

All websites are hosted at the oversea. To reducing all unexpected issues (e.g. Transfer delay, website timeout and other technical issues), the test will *evaluate 3 times* using the same dataset for each environment (Single Computer and Grid Environment). It will record the number of link, number of error, total mission time and how long the webpage crawling.



Result of Random Comparison

Item	Single PC	Grid Environment
Round 1		
Total Link	42,974	43,169
Found Error	37	35
Total Time (sec.)	1,984.89	919.70
Seconds Per Page	0.05	0.02
Minute Per Page	1,299	2,816
Round 2		
Total Link	43,164	34,710
Found Error	36	33
Total Time (sec.)	1,785.42	537.59
Seconds Per Page	0.04	0.02
Minute Per Page	1,451	3,874
Round 3		
Total Link	43,169	32,105
Found Error	35	31
Total Time (sec.)	1,831.28	617.30
Seconds Per Page	0.04	0.02
Minute Per Page	1,414	3,121
Average		
Total Time (sec.)	**1,867.20**	**691.53**
Seconds Per Page	**0.043**	**0.02**
Minute Per Page	**1,388**	**3,270**

Total Link and the number of Found Error may not the same with last round. It's related the issues of dynamic page, too many users access, the web server is not available temporary and firewall blocked. You can see the different in the comparison table.

Through the Average Result, you can see it a big performance different between Single PC and Grid Environment. *Grid Computing is faster than Single PC more than 270%*. As you know, this Grid Environment is containing 6 Desktops Computer and using a single 6Mbps broadband line. That means *each Node can provide around 45% additional performance (averagely)* to the whole computing. In ICDSoft Web Hosting Company's situation, they have 330 Servers to provide web hosting service to 80,000 customers in the world. If they installed *Grid Environment in their servers, the total performance of WADE in their site will be up to 14,850%*. Additionally, each server has a single broadband line to provide service. Their crawling speed must over 10Mbps. So, total performance must be over the expected.

6.5. Detail comparison between Domain-Based and Page-Based

In the previous topic, *"Process Separation for Grid Computing"*, it introduced about the type of Process Separation, there is the three types:

A. Page-based
B. Domain-based
C. Sliding Window-based

Domain-based is our main evaluation in this project. It is a method that is easy to manage and deploy, but how about the Page-based and Sliding Window-based? In this testing, I will compare domain-based and page-based to prove the bottleneck of performance in Process Separation.

Page-based means the Master Server will split a Sitemap of each Domain and assign some part of webpage to each Node, after nodes completed the crawling, they will return the analysis result to Master Server. Master Server will combine and filter all received result, and then export a report to user.

Domain-based means the Master Server will split a Domain List and assign a series of domain to each Node, Node will handle the crawling, combine and filtering. After completed the operation, Nodes will return the processed report Master. Master will gather all report and export to user.

In this session, I will reuse the same method of random comparison to evaluation Page-based:

Item	Single-based	Page-based	Domain-based
	Round 1		
Total Link	43,164	43,164	43,169
Found Error	35	37	35
Total Time (sec.)	4,596.2	17,265.6	919.70
Seconds Per Page	0.1	0.4	0.02
Page Per Minute	564	150	2,816
	Round 2		
Total Link	34,710	43,169	34,710
Found Error	33	34	33
Total Time (sec.)	4,594.6	17,267.6	537.59
Seconds Per Page	0.13	0.4	0.02
Page Per Minute	453	150	3,874
	Round 3		
Total Link	32,105	36,713	32,105
Found Error	30	30	31
Total Time (sec.)	4,597.3	12,849.55	617.30
Seconds Per Page	0.14	0.35	0.02
Page Per Minute	419	171	3,121

	Average		
Total Time (sec.)	**4596**	**15,794.25**	**691.53**
Seconds Per Page	**0.12**	**0.38**	**0.02**
Page Per Minute	**478.7**	**157**	**3,270**

Why page-based performance is the worst in the evaluation? It is worse than Single PC. For explain this situation, there is some result of analysis:

Item	Single PC	Page-based	Domain-based
Domain List Separation	Master	Master	Master
Sitemap Generation	Master	Master	Node
Job Separation	-	Master	Node
Web Crawl	Master	Node	Node
Data Combine and filtering	-	Master	Node
Result Collection	Master	Master	Master
Reporting	Master	Master	Master

From the Table, Page-based does not fully deploy the job to node and many calculation and operation are handled by Master Server. In this case, if I increase more Nodes to this Page-based environment, the Master Server will need to handle more and more calculation and operation. If the Master Server is not a high performance machine, it will be over loading. Page-Based will make the job more complexity, because Master Server needs to collect a series of page from different Nodes, Node may be located at over sea. This will affect the collection performance. In a website, every page may contain a site menu, most links are duplicated. If the website is spitted to different Nodes, each Node may have a probability to crawl the same link. Duplicated job will increase total crawling time, and the Data combine and filtering are doing in Master Server. Each Node will send back a large unprocessed result data to Master. It will increase the band wide usage. Master needs to put a lot of resources to combine and filtering the large data, and then export a report.

In Domain-based, the role of Master is very simple, it just split the domain list, collect the result from Node and reporting. So, if I increase more Nodes to Domain-based environment, it will not decrease the performance of Master, and it will provide additional power to the whole environment for completing a mission cycle rapidly. Each node can handle a whole domain to avoid link duplication and complete the combine and data filtering at Node side. It will send back a small and de-duplicated data to Master for generating report.

Why the single PC is faster than Page-Based? Single PC needs not to handle Job Separation and Data Combination and filtering. Those items are the heavy job in Grid Computing, so Single PC is faster than Page-Based separation.

6.6. Chapter Summary

This chapter is using step by step method to present the whole System Evaluation. I tried use different case to test about why we need to use Grid Computing. At the end of this chapter I show the data comparison about Single Computing and Grid Computing. I also proved about Grid Computing contains performance bottleneck in job separation (Domain-Based and Page-Based). I hope these data may help the following related researcher.

Chapter 7. Conclusions

7.1. Project Achievement

The aim of the project intends to implement a new Web Crawling System, which will simplify the process of a Web Crawling Application through Grid Computing Environment. The main objective is to achieve success through the draft decision, to solve the current problem domain.

The WADE provides a rapid web analysis to Web Hosting Provider to decrease the Server workload and provide a new service to all potential consumers. Client can keep feed the Web Server status to ensure their Web Services always online. Server Administrator can reduce the manual monitoring and plan more IT strategies to main more of business.

Since thread splitting is very important in Grid Computing, incorrect separation (e.g. too big or too small) will affect the whole mission performance. In the last evaluation, it has a many data to prove this case. So in the following project in the real situation, I will know what the best method of thread splitting is.

The overall project stages, practices, developments and all other activities have helped the author to achieve self-meditative, independent learning and the project idea. This practice is the opportunity for me to acquire knowledge in the future, which gained from the study, and all the other modules of the degree.

7.2. Future enhancement

WADE is a database-less system, all data is stored in XML document for user feed it through web service. For Web Hosting Service Providers provide diversify services, I will increase a Database Server to store all data for more analysis or other application.

WADE should improve the algorithm of Web Crawler for reduce any downed website rapidly. WADE should a User-Friendly Interface for Server Administrators controlling the application. Command-line will decrease the working efficiency and interested. WADE should be easy to install and deploy. I will pack the installation file as MSI format. If the user is using Domain-Based Network environment, MSI installation file can be deployed automatically through Active Directory.

7.3. Aspects of resources

Each project stages need great amount knowledge and inspiration to complete the task. The project needs many researches specific and referencing to support. It spends lot of time to organize different information. To achieve this requirement, I had searched much reference article and documentation through Internet and University Libraries in Hong Kong. This experiences and proposes ideal help me to meet the project goal. If I had not paid more attention to study those documents, I think I won't have a best insight of this project.

7.4. Lessons learnt

It is a good practice about project management, planning and research, analysis and development and present personal ideas effectively. There is rare experience for me to handle the project and complete without teamwork.

7.5. Critical appraisal

The project idea is coming from the hot technology, Cloud Computing and my friend who is working in a Web Hosting Company. He told me about the business and technical issues that are facing by Web Hosting Provider. So they need a new web service with less investment to attract more consumers.

This idea development, testing and implementation stages require used many computers and development tools and purchasing development tool was a large investment in this project. I also spent lot of time to learn and study Grid Computing Technology. Finally, the planned results were under my expectation.

Chapter 8. References

[1] JIM WALDO, GEOFF WYANT, ANN WOLLRATH, SAM KENDALL. *"A NOTE ON DISTRIBUTED COMPUTING"*, SUN MICROSYSTEMS. (2004)

[2] AKSHAY LUTHER, RAJKUMAR BUYYA, RAJIV RANJAN, AND SRIKUMAR VENUGOPAL. *"ALCHEMI: A .NET-BASED ENTERPRISE GRID COMPUTING SYSTEM."* THE UNIVERSITY OF MELBOURNE, AUSTRALIA

[3] JAMES CAVERLEE AND LING LIU. *"QA-PAGELET: DATA PREPARATION TECHNIQUES FOR LARGE-SCALE DATA ANALYSIS OF THE DEEP WEB"*. GEORGIA INSTITUTE OF TECHNOLOGY. IEEE. 10.1109/TKDE.2005.151. VOLUME 17, ISSUE 9, PAGE(S): 1247 – 1262 (SEPT. 2005)

[4] PAUL STRONG. *"ENTERPRISE GRID COMPUTING"*. SUN MICROSYSTEMS. ACM. ISSN:1542-7730. VOLUME 3 , ISSUE 6 (JULY/AUGUST 2005)

[5] SURRIDGE, M., TAYLOR, S., DE ROURE, D., ZALUSKA, E. EXPERIENCES WITH *"GRIA – INDUSTRIAL APPLICATIONS ON A WEB SERVICES GRID"*. IEEE. 10.1109/E-SCIENCE.2005.38. VOLUME , ISSUE , PAGE(S):98 - 105 (JULY 2005)

[6] SATHISH S. VADHIYAR, JACK J. DONGARRA. *"SELF ADAPTIVITY IN GRID COMPUTING"*. ACM. ISSN:1532-0626. VOLUME 17 , ISSUE 2-4 (FEBRUARY 2005)

[7] LARRY SMARR, CHARLES E. CATLETT. *"METACOMPUTING"*. ACM. ISSN:0001-0782. VOLUME 35 , ISSUE 6 (JUNE 1992)

[8] BRAD ADELBERG. *"A TOOL FOR SEMI-AUTOMATICALLY EXTRACTING STRUCTURED AND SEMISTRUCTURED DATA FROM TEXT DOCUMENTS"*. ACM. ISSN:0163-5808. VOLUME 27 , ISSUE 2 (JUNE 1998)

[9] ARVIND ARASU, HECTOR GARCIA-MOLINA. "*EXTRACTING STRUCTURED DATA FROM WEB PAGES*". STANFORD UNIVERSITY. ACM. ISBN:1-58113-634-X . (2003)

[10] RICARDO A. BAEZA-YATES, BERTHIER RIBEIRO-NETO. *"MODERN INFORMATION RETRIEVAL"*. ACM. ISBN:020139829X. (1999)

[11] DOUG BEEFERMAN, ADAM BERGER. *"AGGLOMERATIVE CLUSTERING OF A SEARCH ENGINE QUERY LOG. IN KNOWLEDGE DISCOVERY AND DATA MINING"*. ACM. ISBN:1-58113-233-6. (2000)

[12] WILLIAM W. COHEN. *"RECOGNIZING STRUCTURE IN WEB PAGES USING SIMILARITY QUERIES"*. AAAI-99. (1999)

[13] JON M. KLEINBERG. *"AUTHORITATIVE SOURCES IN A HYPERLINKED ENVIRONMENT"*. ACM. VOLUME 46, ISSUE 5. ISSN:0004-5411 (SEPTEMBER 1999)

[14] RAGHAVAN, S. RAJAGOPALAN, R. KUMAR, P AND A. TOMKINS. *"TRAWLING THE WEB FOR EMERGING CYBER-COMMUNITIES"*. IN WWW '99. (1999)

[15] LIU, L.; PU, C.; HAN, W. *"XWRAP: AN XML-ENABLED WRAPPER CONSTRUCTION SYSTEM FOR WEBINFORMATION SOURCES".* IEEE. 10.1109/ICDE.2000.839475. VOLUME , ISSUE , 2000 PAGE(S):611 - 621 (2000)

[16] A. NIERMAN AND H. V. JAGADISH. *"EVALUATING STRUCTURAL SIMILARITY IN XML DOCUMENTS"*. IN PROC. OF THE 5TH INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES (WEBDB). PAGES 61--66. MADISON. WISCONSIN, (2002).

[17] YING ZHAO, GEORGE KARYPIS. *"CRITERION FUNCTIONS FOR DOCUMENT CLUSTERING: EXPERIMENTS AND ANALYSIS"*. TECHNICAL REPORT, UNIVERSITY OF MINNESOTA. UMN CS 01-040, 2001, (2002)

[18] *"COMMON OBJECT REQUEST BROKER: ARCHITECTURE AND SPECIFICATION."*,THE OBJECT MANAGEMENT GROUP. OMG DOCUMENT NUMBER 91.12.1 (1991).

[19] BLACK, A., N. HUTCHINSON, E. JUL, H. LEVY, AND L. CARTER. *"DISTRIBUTION AND ABSTRACT TYPES IN EMERALD."*,. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING SE-13, NO. 1, (JANUARY 1987).

[20] DASGUPTA, P., R. J. LEBLANC, AND E. SPAFFORD. *"THE CLOUDS PROJECT: DESIGNING AND IMPLEMENTING A FAULT TOLERANT DISTRIBUTED OPERATING SYSTEM."*, GEORGIA INSTITUTE OF TECHNOLOGY TECHNICAL REPORT GIT-ICS-85/29.(1985).

[21] MICROSOFT CORPORATION. *"OBJECT LINKING AND EMBEDDING PROGRAMMERS REFERENCE"*, VERSION 1. MICROSOFT PRESS, 1992.

[22] COOK, ROBERT. *"MOD- A LANGUAGE FOR DISTRIBUTED PROCESSING.",* PROCEEDINGS OF THE 1ST INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (OCTOBER 1979).

[23] ANDREW D. BIRRELL, BRUCE JAY NELSON. *"IMPLEMENTING REMOTE PROCEDURE CALLS"*. ACM. ISSN:0734-2071 . VOLUME 2 , ISSUE 1 (FEBRUARY 1984)

[24] TONY MASON. *"BOOK REVIEWS: NETWORK COMPUTING ARCHITECTURE AND NETWORK COMPUTING SYSTEM REFERENCE MANUAL".* ACM. ISSN:0146-4833 . VOLUME 20 , ISSUE 3 (JULY 1990)

[25] VASSOS HADZILACOS, SAM TOUEG. *"FAULT-TOLERANT BROADCASTS AND RELATED PROBLEMS".* ACM. ISBN:0-201-62427-3. (1993)

[26] *"FATHER OF THE GRID"*, THE UNIVERSITY OF CHICAGO MAGAZINE: APRIL 2004, [ONLINE], AVAILABLE: HTTP://MAGAZINE.UCHICAGO.EDU/0404/FEATURES/INDEX.SHTML ACCESSED 19 DEC,2008.

[27] BELL, MICHAEL. WILEY & SONS. *"INTRODUCTION TO SERVICE-ORIENTED MODELING"*. PP. 3. ISBN 978-0-470-14111-3. (2008).

[28] *"NOVELL MONO, OPEN SOURCE .NET DEVELOPMENT FRAMEWORK."* 03 FEB 2009, [ONLINE] AVAILABLE: HTTP://WWW.MONO-PROJECT.COM/MAIN_PAGE ACCESSED 03 FEB 2009

[29] MICHAEL DI STEFANO, JOHN WILEY & SONS, *"DISTRIBUTED DATA MANAGEMENT FOR GRID COMPUTING"*, Wiley-IEEE, (2005)

[30] IAN FOSTER. *"WHAT IS THE GRID? A THREE POINT CHECKLIST*", ARGONNE NATIONAL LABORATORY, MATHEMATICS & COMPUTER SCIENCE DIVISION, (MARCH 2007)

[31] DEPARTMENT OF COMPUTER SCIENCE, BALCI, O. "*SOFTWARE ENGINEERING LECTURE NOTES"*, VIRGINIA TECH, BLACKSBURG, VA, P. 24. (1998)

[32] *"LINK CHECKER COMPARISON"*, [ONLINE] AVAILABLE: HTTP://WWW.CRYER.CO.UK/RESOURCES/LINK_CHECKERS.HTM ACCESSED 03 FEB 2009

[33] STANLEY CHOW JEFF SMITH CHRISTOPHE GUSTAVE , GALASSO & ASSOCIATES, LP, *"VERIFYING AUTHENTICITY OF WEBPAGES"*, ORIGIN: AUSTIN, TX US, IPC8 CLASS: AH04L900FI, USPC CLASS: 713156

[34] *"HTML 4.01 SPECIFICATION"*, W3C RECOMMENDATION 24 DECEMBER 1999 [ONLINE], AVAILABLE: http://www.w3.org/TR/html401 ACCESSED 05 JAN 2009

[35] *"C# ISO/IEC 23270:2006"*, ISO/IEC 23270:2006 SPECIFIES THE FORM AND ESTABLISHES THE INTERPRETATION OF PROGRAMS WRITTEN IN THE C# PROGRAMMING LANGUAGE. [ONLINE], AVAILABLE:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42926 ACCESSED 10 FEB 2009

[36] DONALD KNUTH. "*THE ART OF COMPUTER PROGRAMMING"* VOL 1. FUNDAMENTAL ALGORITHMS, THIRD EDITION. ADDISON-WESLEY. ISBN 0-201-89683-4. SECTION 2.3, ESPECIALLY SUBSECTIONS 2.3.1-2.3.2 (PP.318-348). (1997)

[37] *"REGULAR EXPRESSIONS"*, THE SINGLE UNIX SPECIFICATION, VERSION 2, THE OPEN GROUP, (1997)

[38] RAY, ZHAO ZHANG. *"AN EFFICIENT ANONYMITY PROTOCOL FOR GRID COMPUTING"*, IEEE. 10.1109/GRID.2004.9. Volume , Issue , 8 Nov. 2004 Page(s): 200 - 207 (2004)

[39] ZHIGUO SHI, YEPING HE, XIAOYONG HUAI, HONG ZHANG, *"IDENTITY ANONYMITY FOR GRID COMPUTING COORDINATION BASED ON TRUSTED COMPUTING"*, IEEE, 10.1109/GCC.2007.77. Volume , Issue , 16-18 Aug. 2007 Page(s):403 – 410. (2007)

[40] DAVIES, ANTONY, *"COMPUTATIONAL INTERMEDIATION AND THE EVOLUTION OF COMPUTATION AS A COMMODITY"*, APPLIED ECONOMICS 36: 1131. DOI:10.1080/0003684042000247334, (JUNE 2004)

[41] CARL KESSELMAN. *"THE GRID: BLUEPRINT FOR A NEW COMPUTING INFRASTRUCTURE"*. MORGAN KAUFMANN PUBLISHERS, FOSTER, IAN; ISBN 1-55860-475-8. (NOVEMBER 1998)

[42] BERMAN, FRAN, ANTHONY J. G. HEY, GEOFFREY C. FOX, "*GRID COMPUTING: MAKING THE GLOBAL INFRASTRUCTURE A REALITY"*, ACM. ISBN 0-470-85319-0. (2003)

[43] LI, MAOZHEN, MARK A. BAKER. *"THE GRID: CORE TECHNOLOGIES"*, WILEY. ISBN 0-470-09417-6. (MAY 2005)

[44] *"GRID COMPUTING: A BRIEF TECHNOLOGY ANALYSIS"*. CTO NETWORK LIBRARY, SMITH, ROGER, 2005.

[45] STOCKINGER, HEINZ, *"DEFINING THE GRID: A SNAPSHOT ON THE CURRENT VIEW"*, SUPERCOMPUTING 42: 3. DOI:10.1007/S11227-006-0037-9, 2007

Appendix

Appendix 1. Resource Requirement

Hardware

- Processor type:
  - Itanium processor or faster
- Processor speed:
  - Recommended: 1.0 GHz or faster
- RAM:
  - Minimum: 512 MB
  - Recommended: 2.048 GB or more
  - Maximum: Operating system maximum
- Hard Disk
  - Disk space requirements will vary with the WADE components you install.
- Drive
  - A CD or DVD drive, as appropriate, is required for installation from disc.
- Display
  - WADE graphical tools require VGA or higher resolution: at least 1,024x768 pixel resolution.
- Other Devices
  - Pointing device: A mouse or compatible pointing device is required.

Software

- Operation System
  - Microsoft Windows
  - Linux
- Application Framework
  - Microsoft .NET Framework 2.0
  - Novell Mono 2.0
- Database System
  - MySQL
  - Microsoft SQL Server
- Grid Computing Framework
  - Alchemi - .NET based Enterprise Grid Computing Framework
- Development Tool
  - Microsoft Visual Studio 2008 Express Edition
- Others
  - Microsoft Windows Installer 4.5

Programming Language

- ISO/IEC 23270 C#

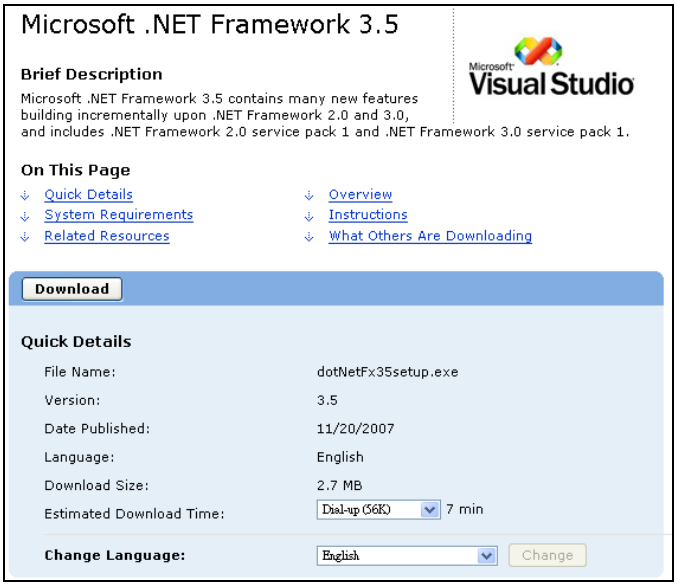
Appendix 2. The Steps of Software Installation

Before the evaluation, these applications should be installed into Server & Node PC:

1. Microsoft .NET Framework 3.5
2. Microsoft Windows Installer 4.5
3. Microsoft SQL Server 2008 Express
4. Alchemi Manager
5. Alchemi Executor

Alchemi Manager and Microsoft SQL Server 2008 Express are installed into Server. Alchemi Executor installs into each Node PCs. I will step by step to install a sample machine to evaluate the WADE:

Install Microsoft .NET Framework 3.5

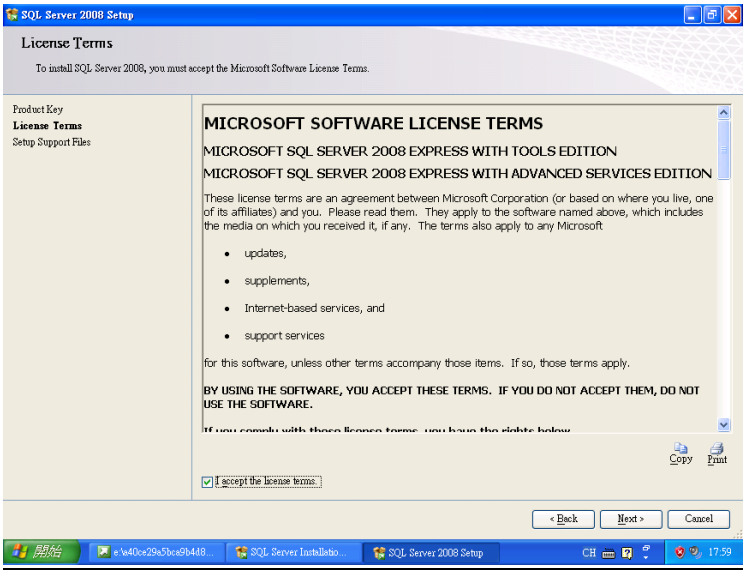


.NET Framework is the basic requirement of Alchemi & WADE, so it should be installed first. .NET Framework contains many useful components for developer to refer these classes to make the whole development cycle rapidly.

Install Microsoft Windows Installer 4.5

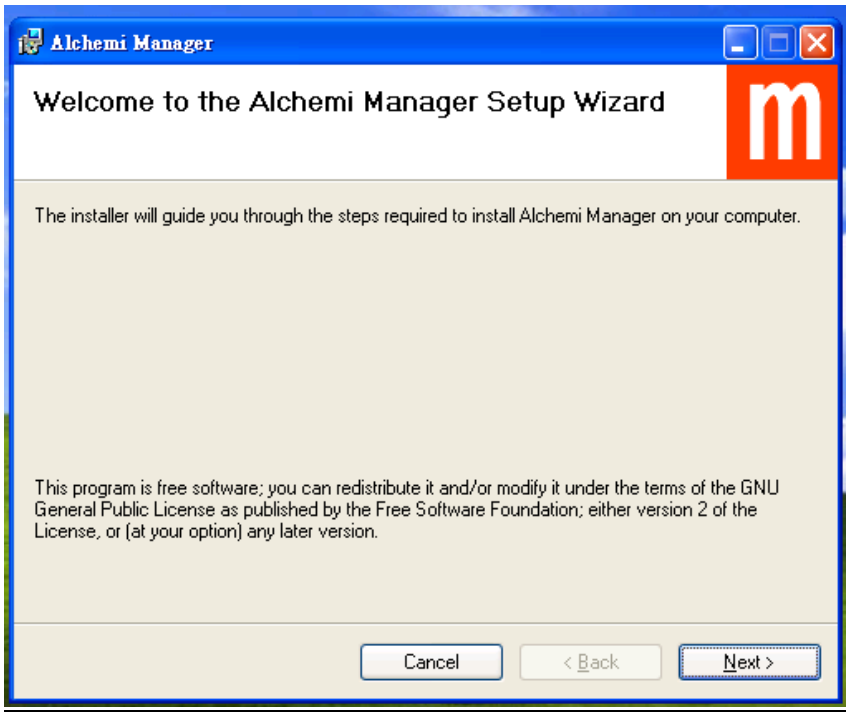
Microsoft SQL Server 2008 is required Windows Installer 4.5, so please install it before the SQL Server installation.

Install Microsoft SQL Server 2008 Express



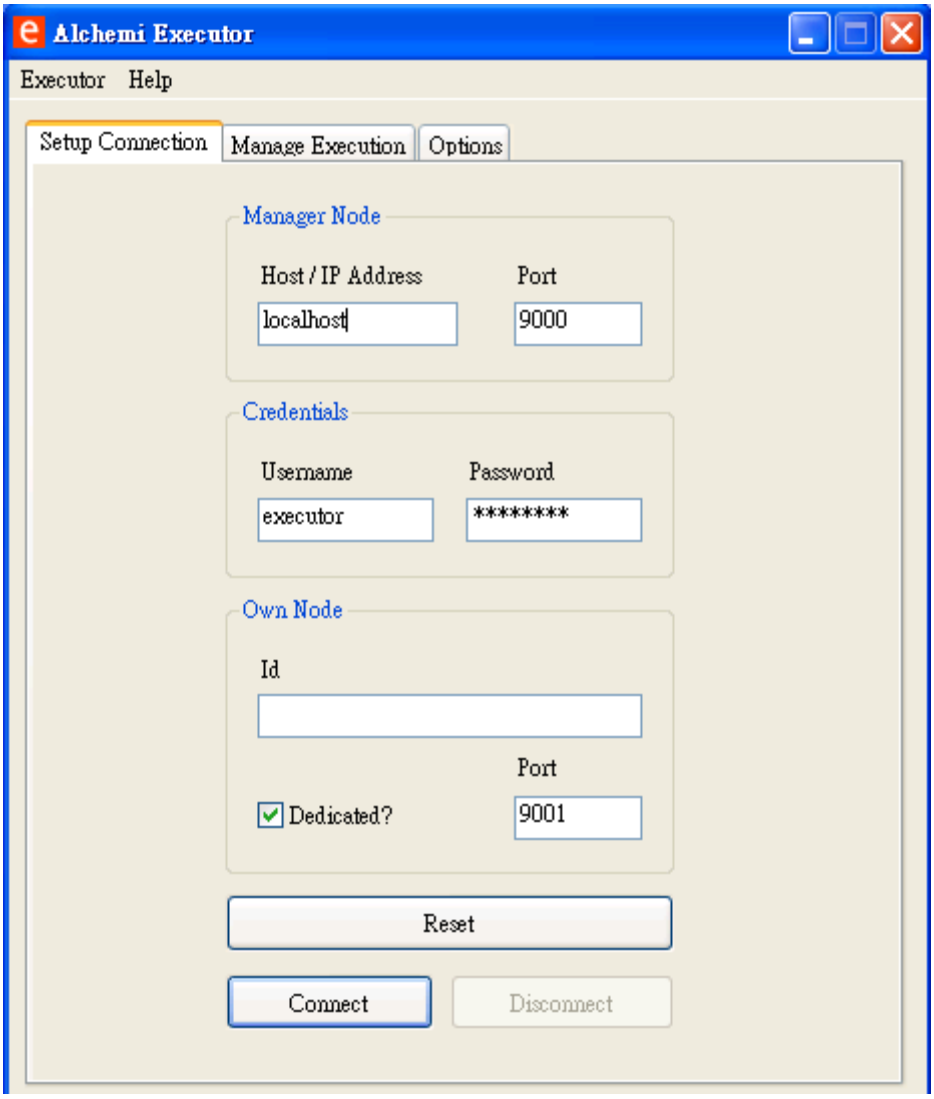
Microsoft SQL Server 2008 is for the Alchemi to store all nodes information and for the Master communicate with other Nodes. It is the communication platform in Grid Environment.

Install Alchemi Manager



Ahchemi Manager is the brain in Grid Environment. It handles Nodes Connection, Resource Management, Job Distribution, and Result Collection. Each Grid Environment must include a Manager. It's the same case as Domain/Client architecture. Ahchemi Manager is an Active Directory Server. Ahchemi Executor is Client Side Computer, called Node.

Install Alchemi Executor



Alchemi Executor should install into all nodes for receiving the operation command from Master PC (Alchemi Manager). Alchemi Grid Environment supports over 1,000 nodes in the same Grid Environment. All firewall should be opened port 9000 & 9001 and the host IP should be Master's IP address.

After those installations, you must execute Alchemi Manager and Alchemi Executors. Alchemi Executors should be all indicate to Master Server IP Address and connected with the Server. Alchemi Management Console will show the connected Executors and telling you all nodes information, such as concurrent process, hardware information and the total performance.

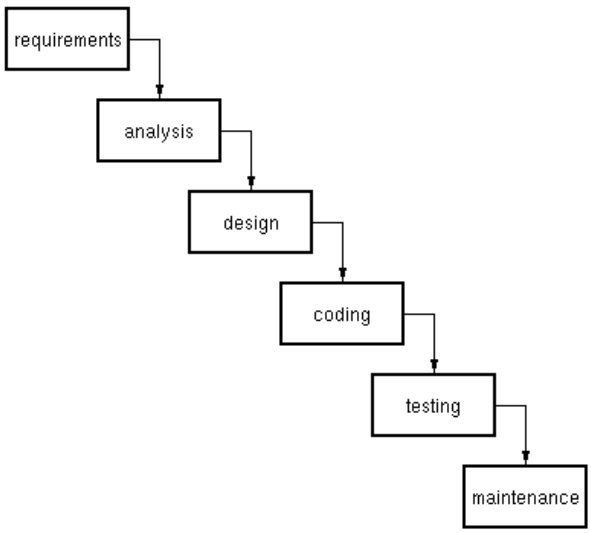
Appendix 3. System Development Schedule

Before starting the project development, though it is a one-man job, I have planned a milestone for guiding me to complete the project. I spitted my job in 11 missions, Concept Understanding, Methodology Study, Comparing Competitive Products, System Analysis, System Design, System Coding, System Testing, Review, Debug, Writing Report and Hand In. There is my project schedule:

Gantt chart

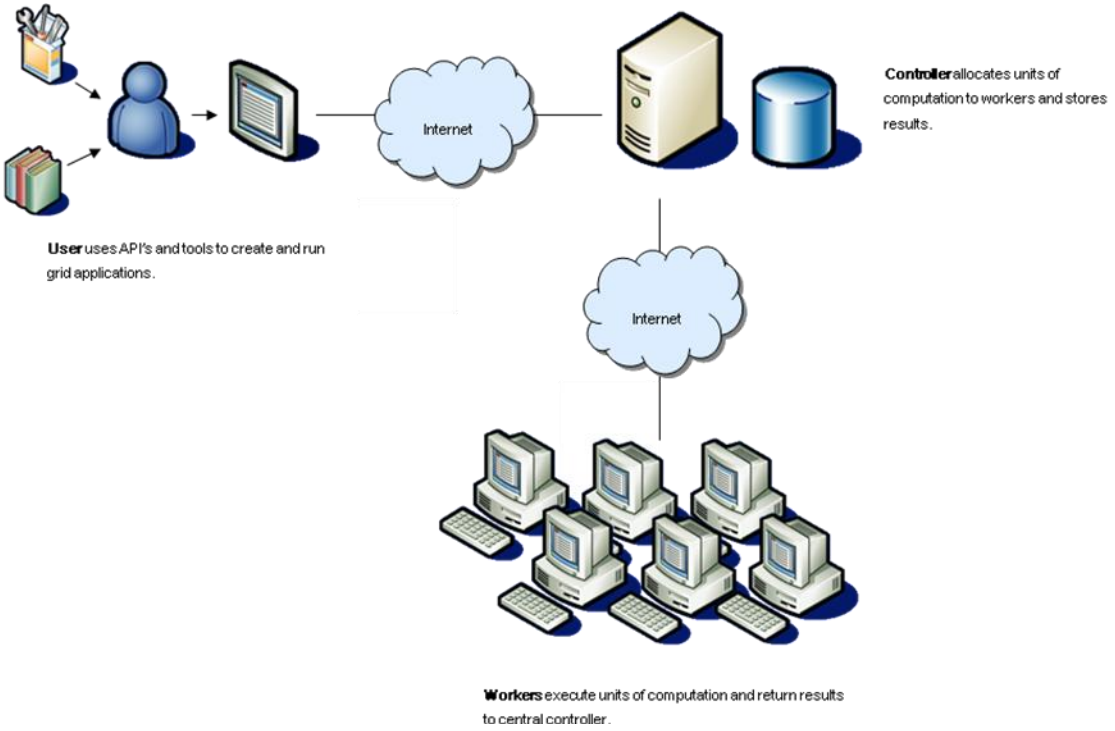
Mission Item	Nov 08	Dec 08	Jan 09	Feb 09	Mar 09
Concept Understanding					
Methodology Study					
Comparing Competitive Products					
System Analysis					
System Design					
System Coding					
System Testing					
Review					
Debug					
Writing Report					
Hand In					

I selected to use the Waterfall Model to run my development life cycle. The Waterfall Model is the consummate software life cycle model. Since 1980s, The model was the one throughout approved life cycle model. It describes the software life cycle of processes and products. Each process produces a product to a new product as output. Then the new product turns into the input of the next process **[7]**.

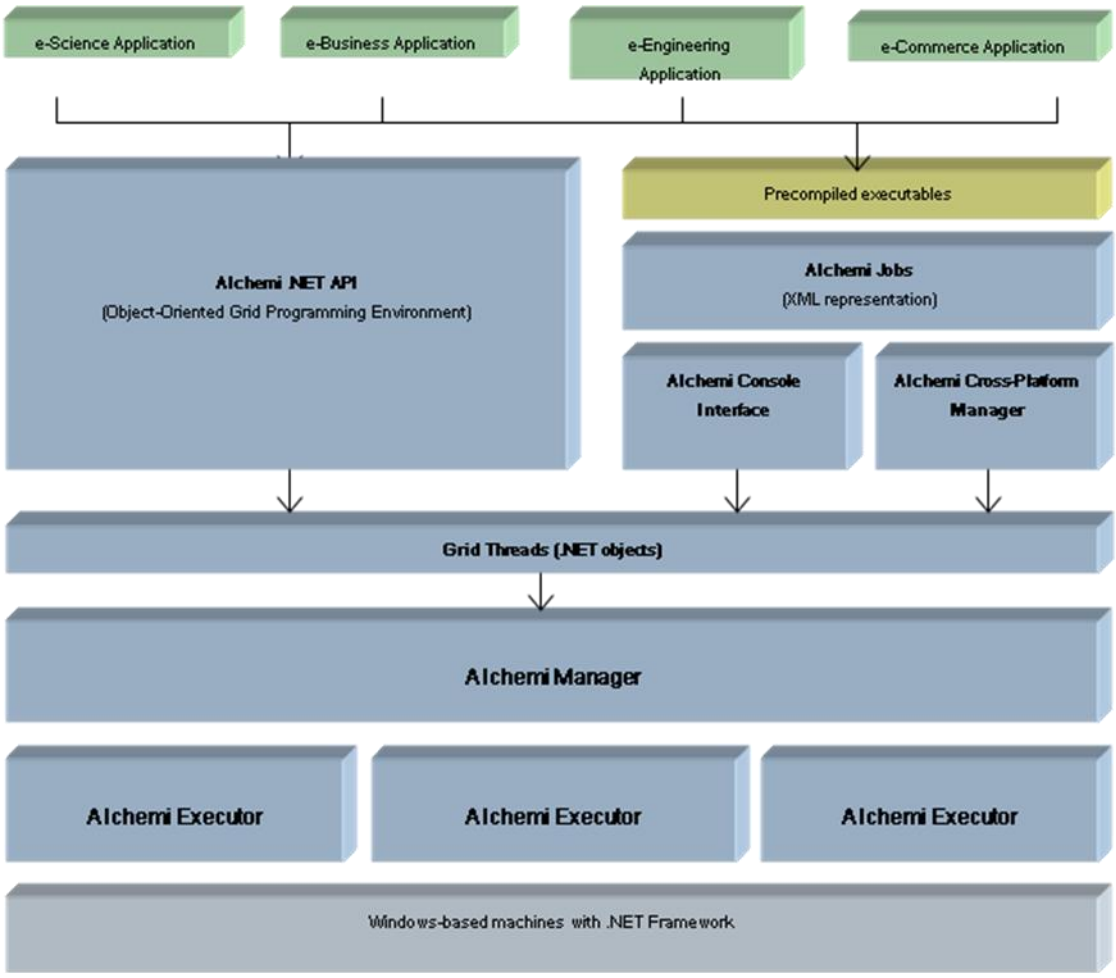


Gantt chart can tell the Concept Understanding, Methodology Study, Comparing Competitive Products and System Analysis is the major mission in the whole project. They will be explained in this paper. The schedule may be changed in the future for tracing the real process and it may not update in this Gantt chart.

Appendix 4. Basic Enterprise grid architecture

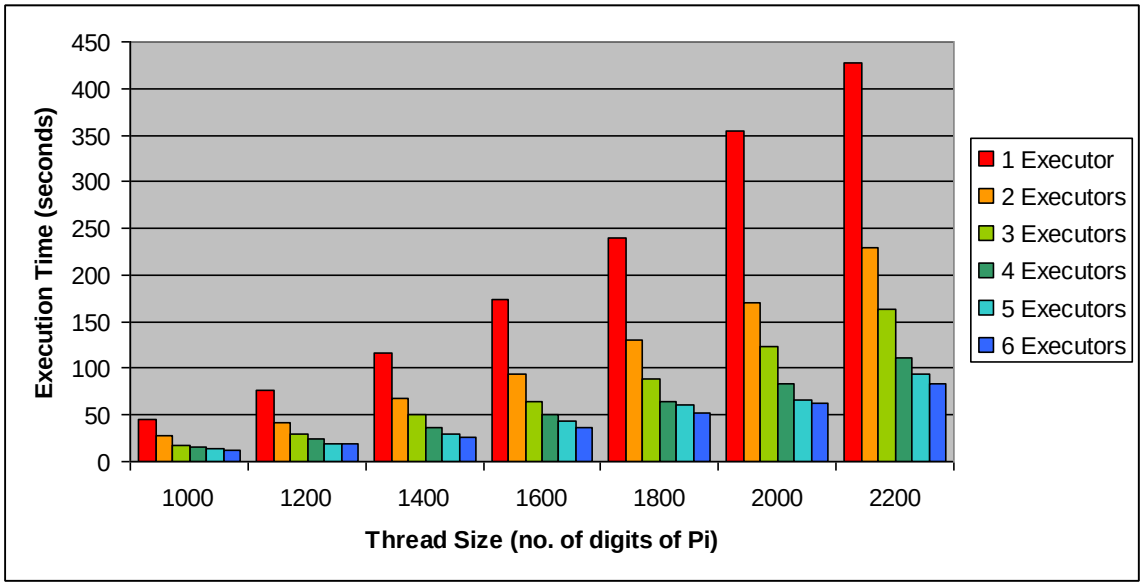


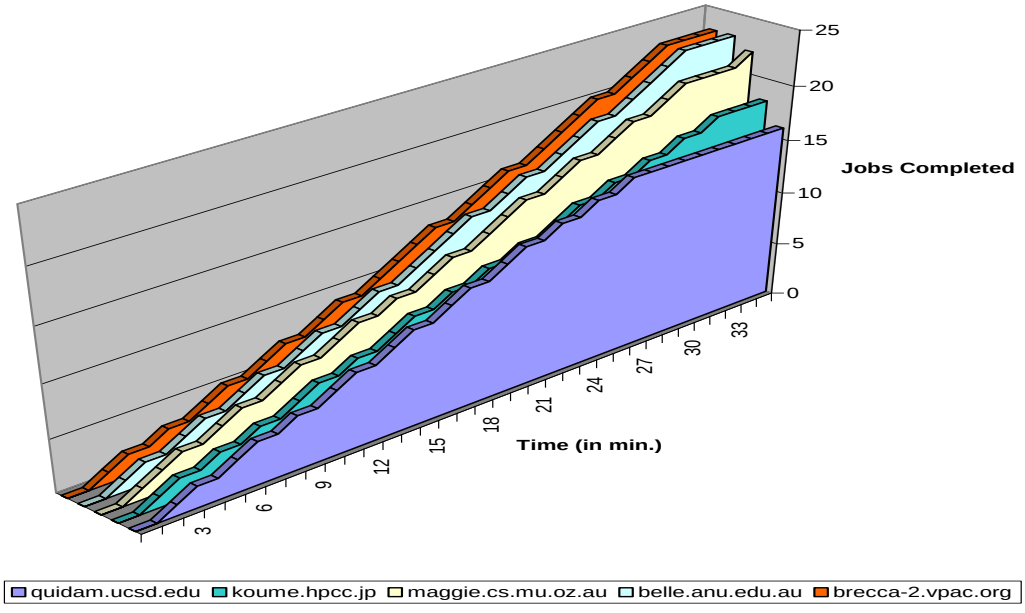
Appendix 5. Alchemi Architecture



Appendix 6. Alchemi Performance Evaluation Result

Standalone Node (High Precision Pi Calculation):





Appendix 7. Related Works and Comparison

System / Property	Alchemi	Condor	SETI@home	Entropia	XtermWeb	Grid MP
Architecture	Hierarchical	Hierarchical	Centralized	Centralized	Centralized	Centralized
Web Services Interface for Cross-Platform Integration	Yes	No	No	No	No	Yes
Implementation Technologies	C#, Web Services, & .NET Framework	C	C++, Win32	C++, Win32	Java, Linux	C++, Win32
Multi-Clustering	Yes	Yes	No	No	No	Yes
Global Grid Brokering Mechanism	Yes (via Gridbus Broker)	Yes (via Condor-G)	No	No	No	No
Thread Programming Model	Yes	No	No	No	No	No
Level of integration of application, programming and runtime environment	Low (general purpose)	Low (general purpose)	High (single purpose, single application environment)	Low (general purpose)	Low (general purpose)	Low (general purpose)

Appendix 8. Detail Competitive Comparison Table

SortSite

Version reviewed:	2.02
Program type:	Executable
Download from:	www.powermapper.com/download
Scope:	Site.
Check locally:	✔
In page links:	✔
JavaScript links:	✔
Images:	✔
Type:	Commercial, 30 day trial.
Can limit scope:	yes, depth limit and via wildcards e.g. http://www.site.com/keepout
Catches redirections:	✔ follows redirections but does not report them.
Notes:	also checks links in the CSS (background images, lack of @ import statements, etc.), and Flash. additional verification of reference checks as well - the server configuration issues, such as incorrectly configured MIME types - accessibility (WCAG 1.0 and Section 508) - all the sites HTML / XHTML validation - browser compatibility issues - Google / Yahoo / MSN search violations - and a bunch of other functions.

DeepTrawl

Version reviewed:	Not applicable, full review pending.
Program type:	Executable.
Download from:	www.deeptrawl.com/download.htm
Scope:	Site
Check locally:	✔
In page links:	✔
JavaScript links:	✘
Images:	✔
Type:	Commercial, 30 day trial.
Can limit scope:	✔
Catches redirections:	✘
Problems:	The full review is planned for this product. Please try again later.

Link Checker Pro

Version reviewed:	3.1.33
Program type:	Executable, must be run locally.
Download from:	www.link-checker-pro.com
Scope:	Site
Check locally:	✔ - but it is not apparent how to make it check a single page.
In page links:	✖
JavaScript links:	✖
Images:	✔
Type:	Commercial.
Trial period:	There is a free 30 day demo available for download from their website.
Can limit scope:	
Catches redirections:	
Problems:	When the tests, although it came with a message box saying: "This site has unlimited potential security flaw. Do you want to continue?" Does not catch all the pop-ups generated pages, it reads. When the tests, although it came with a message box saying "action can not be completed because a component (HTTPCheck) is not. You may be nervous about the URL, for example, that check "http://www.postgrad_resources.btinternet.co.uk" rather than "www.postgrad_resources.btinternet.co.uk."

Xenu's Link Sleuth

Version reviewed:	1.2e
Program type:	Executable, must be run locally.
Download from:	http://home.snafu.de/tilman/xenulink.html
Scope:	Site
Check locally:	✔
In page links:	✖
JavaScript links:	✖ It will find links which appear as simple text within JavaScript, but does not find links which are constructed by JavaScript.
Images:	✔
Type:	Free

Can limit scope:	You can specify not to check the URL, which is a start. This allows you to remove parts of your site. You can also specify how you want to go to the site.
Catches redirections:	✔ The default setting is not to flag these as errors.
Problems:	Xenus Link sleuth has problems with some types of links. Testing version 1.2e, it incorrectly interprets the link <a href="www.mysite.com"> when scanning www.mysite.com as "www.mysite.com / www.mysite.com. Presumably the link <a href =" http: / / www.mysite.com '> will be fine. The checking of the site must enter the complete URL of the page, for example, it cannot find "www.cryer.co.uk / Brian," but will succeed if given "www.cryer.co.uk / brian / index. Htm. In this report, several links, as does not work when they are not.

InfoLink Link Checker

Version reviewed:	1.9d
Program type:	Executable, must be run locally.
Download from:	www.biggbyte.com/infolink/download/download.html
Scope:	Site
Check locally:	✔
In page links:	✖
JavaScript links:	✖
Images:	✔
Type:	Free. Much of the website implies that it is commercial, but this shows its history. It was commercial before it was discontinued.
Can limit scope:	✔
Catches redirections:	✖
Problems:	It would not be created. This interface is a bit daunting, and it is a bit of time with him. The inspector and the report viewer is a separate, which makes good sense of integration. There may be a useful technical tool, but not very friendly webmaster tool.

Dead-Links.com

Version reviewed:	Not applicable. Last reviewed on 22nd March 2005.
Program type:	On-line link checker, must be connected to the internet.
Run from:	www.dead-links.com
Scope:	Site. It will start on the page you give it and will then check each of the (on-site) pages that that page links to. According to the dead-links FAQ it will check up to 25 pages or 150 pages if it finds a link back to dead-links.com, but in testing it checked 250 pages before stopping.
Check locally:	✖ Page must be publicly accessible on the internet to be checked.
In page links:	✖ The logic only tests links which point to other pages, not to bookmarks within the same page.
JavaScript links:	✖
Images:	✖
Type:	Free
Can limit scope:	Not applicable. Single page only.
Catches redirections:	✔ It will report on which pages (that you link to) redirect to somewhere else.
Problems:	On the final report, is simply a list of broken links, rather than the pages containing these links.

REL Link Checker Lite

Version reviewed:	1.0
Program type:	Executable, must be run locally.
Download from:	www.relsoftware.com
Scope:	Site
Check locally:	✖
In page links:	✖
JavaScript links:	✖
Images:	✖
Type:	Free
Can limit scope:	✔ - Can set to scan the entire site or just for those pages within a specified path.
Catches redirections:	
Problems:	If the site is general superintendent of missing pages (ie 404), then check this link fails to establish that the link does not work -

	this is my only show that the 404 error. Have problems with some types of links. It appears that the incorrect interpretation of the link <a href="www.mysite.com"> when scanning www.mysite.com as "www.mysite.com / www.mysite.com. Presumably the link <a href =" http:// www.mysite.com "> will be fine.

404 error page

Version reviewed:	Not applicable.
Program type:	Not applicable.
Download from:	Not applicable.
Scope:	Site
Check locally:	✖
In page links:	✖
JavaScript links:	✔ - but only after a surfer has clicked on the link.
Images:	✖ - missing images do not generate 404 errors.
Type:	Free
Can limit scope:	Not applicable
Catches redirections:	Not applicable
Problems:	This method finds a dead link when a user tries to link. It may only be working with the webmaster of a site. It is not always obvious from the page that link is broken one. It should be a 404. Demands that the 404 error page can be monitored and logged. Logs should be reviewed regularly to catch a lot of broken links.
Unique features:	Allows you (the webmaster) determine broken links to the websites of other persons, who refers to you.

Google Sitemaps

Version reviewed:	Not applicable. Last reviewed November 2005.
Program type:	On line. Reports on last ten errors on your site as encountered by google-bot. These may be days old.
Run from:	Go to www.google.com/webmasters/sitemaps to register or login.
Scope:	Site
Check locally:	✖ Site must be publicly accessible on the internet.
In page links:	✖
JavaScript links:	✖
Images:	✖
Type:	Free
Can limit scope:	Not applicable. No control over which pages errors are reported for.
Catches redirections:	✖
Problems:	It must be seen in the Google Sitemap is not a link checker. The fact that they provide assistance in this area is a coincidence. Only up to sixteen errors are reported. Is not updated regularly. So its worth a visit once or twice a week. No pages to control the error are reported. There is no evidence that the error caused by broken links on your site or broken link from another site.

1-hit.com Bad Link Checker

Version reviewed:	Not applicable. Last reviewed on 6th May 2004.
Program type:	On-line link checker, must be connected to the internet.
Available at:	www.1-hit.com/all-in-one/tool.broken-link-finder.htm
Scope:	Page. Will only check a single page.
Check locally:	✔ Page can be publicly accessible on the internet or a local file can be uploaded.
In page links:	✔
JavaScript links:	✔ It might find links which appear as simple text within JavaScript but this has not been confirmed.
Images:	✔
Type:	Free
Can limit scope:	Not applicable. Single page only.

Catches redirections:	✔
Problems:	If your site is user "on page 404, then the site will be marked as a redirect, and not missing, but this is cosmetic, because it still catches the link is not right.

FWR - Broken Link Checker

Version reviewed:	Not applicable. Last reviewed on 22nd March 2005.
Program type:	On-line link checker, must be connected to the internet.
Run from:	http://www.f-w-r.com/badlinkchecker.php
Scope:	Page
Check locally:	✖ Page must be publicly accessible on the internet to be checked.
In page links:	✖ The logic only tests links which point to pages external to the website.
JavaScript links:	✖ - however it does pick out "<a href=..." from any JavaScript and checks links on that basis. It identified the incorrect JavaScript link from the test cases used.
Images:	✔
Type:	Free
Can limit scope:	Not applicable. Single page only.
Catches redirections:	✖
Problems:	Very slowly. Did you see the results when it finishes, it would not be a problem if it was faster, but it is likely that some users will assume they do not work, because, as he slowly. It provides information on how to "search" link, which does not work, so you need to know that the "Total" refers to a dead link.

W3C Link Checker

Version reviewed:	3.6.2.3
Program type:	On-line link checker, must be connected to the internet.
Run from:	http://validator.w3.org/checklink
Scope:	Site. The default setting is just to check a single page, but ticking the option "check linked documents recursively" will allow it to check multiple pages.
Check locally:	✖ - pages must be published and available online before they can be checked.
In page links:	✔ - it refers to these as "broken fragments".
JavaScript links:	✖

Images:	✖
Type:	Free
Can limit scope:	You can limit the depth to which it scans.
Catches redirections:	✔
Problems:	Incorrectly displayed <a href="www.mysite.com"> when scanning www.mysite.com as "www.mysite.com / www.mysite.com ', but this is only cosmetic, because it will check that the correct reference. He is unable to treat a link to "# top" as the correct (at the top of the page), and flags this as a "broken fragment"

Indiabook.com Free Link Checker

Version reviewed:	Not applicable. Last reviewed on 25-Apr-04.
Program type:	On-line link checker, must be connected to the internet.
Run from:	www.indiabook.com/webmaster/link.html
Scope:	Page
Check locally:	✖ Page must be publicly accessible on the internet to be checked.
In page links:	✖ The report it generates lists in-page links, but this is misleading since it doesn't check them - the known broken in-page links on this page were listed as "OK"
JavaScript links:	✖
Images:	✖ The report it generates lists images, but this is misleading since it doesn't check them - the known missing/broken image on this page was listed as "OK"
Type:	Free
Can limit scope:	Not applicable. Single page only.
Catches redirections:	✖
Problems:	Once reviewed it listed all the links on the page, but could not pick which of those were damaged. He gave this site to a clean sanitation - despite the fact there are a few intentionally broken links. Conclusion: This is best avoided.

LinkChecker by 2bone

Version reviewed:	Not applicable. Last reviewed on 16th July 2004.
Program type:	On-line link checker, must be connected to the internet.
Run from:	www.2bone.com/links/linkchecker.shtml
Scope:	Page

Check locally:	✖ Page must be publicly accessible on the internet to be checked.
In page links:	✖
JavaScript links:	✖
Images:	✖
Type:	Free.
Can limit scope:	Not applicable. Single page only.
Catches redirections:	✖
Problems:	Once reviewed, the failure to take any of the broken links on this page. Probably it will be easy to find broken links, but nothing more complicated.

LinkTraX from ClientWorX

Version reviewed:	Not applicable. Last reviewed on 3rd November 2004.
Program type:	On-line link checker, must be connected to the internet.
Run from:	http://clientworx.com/LinkTraX/TestLink.shtml
Scope:	Page
Check locally:	✖ Page must be publicly accessible on the internet to be checked.
In page links:	✖ The logic only tests links which point to pages external to the website.
JavaScript links:	✖
Images:	✖
Type:	Free
Can limit scope:	Not applicable. Single page only.
Catches redirections:	✖
Problems:	Not reporting the internal links to your site, that means the will to spend some of the obvious broken links. (It may be that they intend to provide paid services, it would, but there is no evidence that this is their website).

Appendix 9. The HTTP Status Codes

Informational 1xx

This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and the blank line disappears. There is no need for headers for this class of status. Since HTTP/1.0 did not define any status 1XX, servers NOT send 1XX response to an HTTP/1.0 client except under experimental conditions.

100 Continue

The client should continue with the application. This is an interim response used to inform the client that the first part of the application has been received and has not been rejected by the server. The client continues to send the remaining request, or if the application has been completed, ignore this response. The server must send a final response after the request has been completed.

101 Switching Protocols

The server understands and is willing to comply, at the request of the customer through the Upgrade message header field, the change in the application protocol used for this connection. The server switch protocols to those of the response to the Upgrade header field immediately after the empty line which terminates the 101 response.

The Protocol should be switched, but only if it is beneficial. For example, to switch to a newer version of HTTP benefit from older versions, and switching to a real-time, synchronous protocol would be advantageous if the use of resources to such functions.

Successful 2xx

This type of status code indicates that, at the request of the client has been successfully received, understood and accepted.

200 OK

The request was successful. The information back to the answer depends on the method used in the request, for example:

GET unit corresponds to the requested resource is sent to the response;

HEAD the entity-header fields to the requested resource is sent without a reply message body;

POST entity or a description of the action;

TRACE an entity containing the request was received from the server.

201 Created

The performance of the application, and has led to a new resource is created. The newly created resource can be included in the URI (s) back to the entity of the response, and the most unique URI of the source given by a Location header field. The answer should include an entity containing a list of resource characteristics and location (s) to which the user or the user can choose the most appropriate representative. The entity format is specified by media type, the Content-Type header field. The origin server should be set up the resource before returning the 201 status code. If the action can not be implemented immediately, the server must comply with 202 (Accepted) response instead.

In response to ETAG 201 response may contain a header field indicates the current value of the entity tag of the requested variation

202 Accepted

The application was accepted for processing, but the process has not yet been completed. The application cannot be, or may be made, since this is actually refused to take place. There is no possibility for re-sending a status asynchronous operation such as this one.

The 202 non-committal is response intentionally. This is to the server to accept the request of another process (perhaps a batch process, which only runs once a day), it is not necessary to contact the user agent server persist until the process has been completed. The unit returns this response indicate that the application should include the current status, and / or monitor the status of the indicator, or an estimate of when the user is expected to request to be fulfilled.

203 Non-Authoritative Information

The returned meta information in the entity-header is not the definitive set available from the origin server, but to collect a local or a third-party copy. SUBSET it can be shown or revising the original version. For example, including local annotation information about the source of a superset of the meta information known to lead to the origin server. Use of this response code is not needed and can be justified only if the answer to any 200 (OK).

204 No Content

Server to fulfill the request, but shall not be required to return the unit-body, or that want to return updated meta information. The response contains the new or updated meta

information in the form of entity-headers, which if present should be associated with the requested version.

If the customer is a user agent, this does not alter the document, that the spectacle, making the request should be sent. This is the answer, first, that the input action without an agent causes a change in the user's active document, although any new or updated meta information of the document should apply to the User Agent is currently active view.

The answer does not contain a message body 204, and thus is always terminated in the first blank line after the header fields.

205 Reset Content

Server fulfills the request and the user agent should set up the document, which is due to the application must be sent. This response, first, that the measures to be taken through the user input, and then a clearing of the form in which the input is given so that the user can easily initiate another input action. The answer does not contain a unit.

206 Partial Content

The server has fulfilled the partial GET request to the source. The application must include the Range header field indicating the desired range, and possibly in a If-Range header field that the application requirement.

206 If the answer is the result of a Range, if required, to use a strong cache validator, the response should not include any other entity-headers. If the answer is the result of the If-Range request that used a weak validator, the response does not include other entity-headers. This will prevent the cached entity-bodies and updated headers. Anyway, the answer is to all of the entity-headers that would have been back to the 200 (OK) responses to that request. A cache cannot be combined with a 206 response with other previously cached content if the ETAG and Last-Modified headers do not match exactly. A cache that does not support the Range and Content-Range header is NOT cache 206 (Partial) responses.

Redirection 3xx

This class of status code indicates that further steps should be taken by the user agent to fulfill the request. The necessary action can be carried out, the user agent without interaction with the user, and if so, only if the method is applied to the second request is GET or HEAD. Customer's perception of the infinite redirection loops, since such loops generate network traffic for each redirection.

300 Multiple Choices

The requested resource corresponds to one of a series of missions, each with its own specific location, and agent-driven negotiation information is available to the user (user or agent) can select a preferred representation and redirect its request to that location.

Unless this is a HEAD request, the response should include an entity containing a list of resource characteristics and location (s) to which the user or the user can choose the most appropriate representative. The entity format is specified by media type, the Content-Type header field. Depending on the format and the capabilities of the user agent, selection of the most appropriate choice may be performed automatically. However, this standard does not define the standard of such automatic selection.

If the server is set to the choice of representation, it should be covered by this representation of the URI of the local area, then use the user's location field value for automatic redirection. This response is cacheable unless otherwise indicated.

301 Moved Permanently

The sources were required for a new permanent URL and any future references to this resource should use one of the returned URIs. Clients of the link editing capabilities should automatically be a link, the request-URI to one or more of the references back to the server, where possible. This response is cacheable unless otherwise indicated.

The new permanent URI field should be given to the place of the answer. Unless the request method is HEAD, the entity must be included in the response to a short hypertext note of a hyperlink to the new URI (s).

If the 301 status code does not get a response to the request GET or HEAD, the user agent must not automatically redirect the request unless they can demonstrate to the user, because it may change the conditions under which the request has been issued.

302 Found

The requested resource is temporarily staying in a URI. Since the redirection may be altered to the occasion, the client may continue to use the Request-URI future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

URI should be a transitional area, the location of the answer. Unless the request method was HEAD, the entity must be included in the response to a short hypertext note of a hyperlink to the new URI (s).

If the 302 status code does not get a response to the request GET or HEAD, the user agent must not automatically redirect the request unless they can demonstrate to the user, because it may change the conditions under which the request has been issued.

303 See Other

The answer can be found in the request URI using a GET method to retrieve the source. This method exists primarily to discharge the POST-activated script to the User Agent to a selected resource. The URI is not a substitute for the original reference to the requested resource. The 303 response is NOT to be cached, but the answer to the second (redirected) request from cacheable.

The URI in the location field should be the answer. Unless the request method was HEAD, the entity must be included in the response to a short hypertext note of a hyperlink to the new URI (s).

304 Not Modified

If the client made a conditional GET request and access is permitted, but the document is not modified, the server must comply with this status. The answer does not contain a message body 304, and thus is always terminated in the first blank line after the header fields.

If a clockless origin server obeys these rules, and proxies and clients add their own Date to any response received from one (as mentioned in [RFC 2068], section 14.19), caches work correctly.

If you get used to a strong cache validator, the conditional (see Section 13.3.3), the answer is not to be included in the other entity-headers. Otherwise (ie, the conditional quit weak validator), the reply does not include other entity-headers, this will prevent the cached entity-bodies and updated headers.

If a 304 response indicates an entity not currently cached, then the cache should be ignored in the response and repeat the request without any condition.

If a cache uses a received 304 response to update a cache entry in the cache entry should be updated to reflect any new field values given in the answer.

305 Use Proxy

The requested resources should be available in the proxy given by Location field. The location field specifies the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 Answers by origin servers only.

306 (Unused)

The 306 status code to a previous version of the specification is no longer used, and the code is reserved.

307 Temporary Redirect

The requested resource is temporarily staying in a URI. Since the redirection may be altered to the occasion, the client may continue to use the Request-URI future requests. This indicates the answer is only cacheable if the Cache-Control or Expires header field.

The URI must be given a provisional place in the area of the response. Unless the request method was HEAD, the entity must be included in the response to a short hypertext note of a hyperlink to the new URI (s), since many users pre-HTTP/1.1 agents do not understand the 307 status. Therefore, the note must contain the information necessary for the user to repeat the original request to the new URI.

If the 307 status code does not get a response to the request GET or HEAD, the user agent must not automatically redirect the request unless they can demonstrate to the user, because it may change the conditions under which the request has been issued.

Client Error 4xx

The 4xx class of status is when the client seems to have erred. Except when a HEAD request, the server must be included in the unit, which includes an explanation of the error situation, and whether it is temporary or permanent status. These status codes applicable to any request method. User agents need to be any person, to the user.

If the client sends data to the server implementation using TCP should be careful to accept that the customer received the package (s), which contains the answer, before the server closes the input connection. If the client continues after the closing of Sending data to the server, the server sends a TCP reset packet to the client's stack, which removes the client's input buffer is not known can be read and interpreted by the HTTP request.

400 Bad Request

The request could not be understood by the server due to incorrect syntax. The client SHOULD NOT repeat the request without modifications.

401 Unauthorized

The request requires user authentication. The answer must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. The customer may repeat the request to the appropriate Authorization header field. If the request already included Authorization credentials, then the 401 response indicates that authorization is denied the right. If the 401 response contains the same challenge as in the previous response, and the user agent has already attempted authentication at least once, then the user should be presented in the unit was the answer, as this is the unit for diagnostic information. HTTP access authentication explanation "HTTP Authentication: Basic and Digest Access Authentication".

402 Payment Required

This code is reserved for future use.

403 Forbidden

The server understood the request but refused to fulfill it. The license will not help, and the application will not be repeated. If the request method was not the head and the server wishes to disclose that the request was not met, you should describe the reason for the rejection of the unit. If the server does not want this information to the client, the status code 404 (Not Found) can be used.

404 Not Found

The server has not found anything matching the Request-URI. There is no indication whether the condition is temporary or permanent. The 410 (Gone) status code should be used if the server knows the individual's internal configurable mechanism, that an old resource is not available in final, and there is no forwarding address. This status code is used when the server does not wish to reveal, that the application is refused, or if there is no other response is applicable.

405 Method Not Allowed

The method of the Request Line is not allowed in the resource specified by Request-URI. The answer must include an Allow header containing the list of valid methods for the requested resource.

406 Not Acceptable

The source specified by the application will only be able to generate a response includes the characteristics of the content is not acceptable to accept headers sent in the request.

Unless you were at the request of the head, the response should include an entity containing a list of available entity characteristics and location (s) to which the user or the user can choose the most appropriate representative. The entity format is specified by media type, the Content-Type header field. Depending on the format and the capabilities of the user agent, selection of the most appropriate choice may be performed automatically. However, this standard does not define the standard of such automatic selection.

If the answer would not be acceptable, a user agent SHOULD temporarily stop receipt of more data on the user's query is a decision for further action.

407 Proxy Authentications Required

This code is similar to the 401 (not allowed), but indicates that the client must authenticate itself to the agent. The representative must return the Proxy-Authenticate header field containing a challenge applicable to the Agent on the resource. The customer may repeat the request to the appropriate Proxy-Authorization header field. HTTP access authentication explanation "HTTP Authentication: Basic and Digest Access Authentication".

408 Request Timeout

The client is not a request for the specified period of time. The server was willing to wait. At the request of the customer repeats without any amendments at a later date.

409 Conflict

The request could not be completed because of the conflict in the current state of the source. This code is only in cases where it is expected that the user may have to resolve the conflict and resubmit the request. The response body should include enough information to the user to recognize the source of the conflict. Ideally, the response of the organization includes sufficient information for the user or user agent to remedy the problem, however, is that it might not be possible, and it is not necessary.

Conflicts most likely to occur in response to a PUT request. For example, if versioning is used, and include changes in resource units, which were previously in conflict with (third party), at the request of the server to use the 409 response indicates that it is not able to complete the application. In this case, the answer is likely to include a list of units. The difference between the two versions is a form of the response Content-Type.

410 Gone

The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead. This response is cacheable unless indicated otherwise.

The 410 response is primarily intended to assist the task of web maintenance by notifying the recipient that the resource is intentionally unavailable and that the server owners desire that remote links to that resource be removed. Such an event is common for limited-time, promotional services and for resources belonging to individuals no longer working at the server's site. It is not necessary to mark all permanently unavailable resources as "gone" or to keep the mark for any length of time -- that is left to the discretion of the server owner.

411 Length Required

The server does not accept the request to the specified Content-Length. The customer may request a repeat if you add a valid Content-Length header field containing the length of the message body in the request message.

412 Precondition Failed

The precondition given in one or more of the request-header fields evaluated to false, when tested on the server. This response code allows the client to the condition of the current resource meta information (header field data), and thus prevent the requested method can be applied to the source is not the one intended.

413 Request Entity Too Large

The server refused to process the request because the request entity is larger than the server is willing or able to process. The server is a close relationship. The client will continue to prevent the application. If the condition is temporary, the server must include a Retry after header field indicates that the transitional period, after which the client tries again.

414 Request-URI Too Long

The server refused to service the request because the Request-URI is longer than the server is willing to interpret. This rare condition is only likely to occur when a customer is searching for transforming the long POST request to a GET request query information, when the client went to a URI "black hole" of (eg, a redirected URL prefix of a suffix in its own), or if the server is under attack on a client is trying to exploit the security hole is fixed in some servers using the length buffers for reading or manipulating Request URI.

415 Unsupported Media Type

The server refused to service the request because the entity of the request in a format not supported by the requested resource is in the method.

416 Requested Range Not Satisfiable

In response, the server should return the code, if this condition is included in the application of the Range request-header field, and not the domain-value in this field overlap the current extent of the selected source, and the application does not include the If - Range request-header field. (A byte-ranges, this means that the first-byte-pos is the byte-range-spec is greater than the current length of the selected source.)

If this is the status code returned for a byte-range request, the response should include a Content-Range entity-header field specifies the length of the selected source. This is not the answer to use the multipart / byte ranges content-type.

417 Expectation Failed

Expect a wait of request-header field could not be met by this server, or if the server is a proxy server, the clear evidence that the application does not meet the definition of a next-hop server.

Server Error 5xx

Reply from the status of the digit "5" indicates cases in which the server could not be that wrong, or is unable to carry out the request. Except when a HEAD request, the server must be included in the unit, which includes an explanation of the error situation, and whether it is temporary or permanent status. User agents SHOULD display any person of the user. These are the codes for the response to any request method.

500 Internal Server Error

The server encountered an unexpected condition which prevented to fulfill the request.

501 Not Implemented

The server does not support the functionality to fulfill the request. This is the appropriate response if the server does not recognize the request and the method is not able to support that each source.

502 Bad Gateway

The server, while a gateway or proxy server received a response to an invalid access to the upstream server is trying to fulfill the request.

503 Service Unavailable

The server is not able to handle the request due to a temporary overloading or maintenance on the server. The implication is that this is a transient state will be alleviated after some delay. If known, the length of the delay may be indicated in a Retry-After header. If no Retry after receiving the response to be managed by the client, since this is a 500 response.

504 Gateway Timeout

The server, while a gateway or proxy did not receive a timely response to the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (eg DNS), it is necessary to attempt to access the full application of the .

505 HTTP Version Not Supported

The server does not support or refuses to support the HTTP protocol version used in the application message. The server does not indicate that it is unable, or unwilling, to the application of the same major version as the client, other than this error message. The answer should include a person who explains why it is not supported version, and what other protocols are supported, to the server.

Appendix 10. WADE XML Report Sample

There is the detail XML Content:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Links>
- <Link>
<Level>0</Level>
<MasterUrl />
<Url>http://www.peter-lo.com/</Url>
<Status>ok</Status>
</Link>
+ <Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/index.htm</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/index.htm</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/Photo/index.htm</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/Resume/index.htm</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/contact/index.htm</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/images/sectitle_right.jpg</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/images/icon.gif</Url>
<Status>ok</Status>
```

```
</Link>
- <Link>
<Level>1</Level>
<MasterUrl>http://www.peter-lo.com/</MasterUrl>
<Url>http://www.peter-lo.com/images/cscwlo.jpg</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/MA104/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/A106/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/M014/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/M7011/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/M8034/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/M8748/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/U08096/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/U08182/</Url>
<Status>ok</Status>
</Link>
```

- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/U51020/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/B2001/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CS215/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CS211/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CS213/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CS218/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CS220/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CP582/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>
<Level>**2**</Level>
<MasterUrl>**http://www.peter-lo.com/Teaching/index.htm**</MasterUrl>
<Url>**http://www.peter-lo.com/Teaching/CP586/**</Url>
<Status>**ok**</Status>
</Link>
- <Link>

```
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/CE300/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/IS352/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/IT354/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/IT359/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Teaching/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Teaching/DB212/</Url>
<Status>ok</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/Photo/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/Photo/index.htm</Url>
<Status>The remote server returned an error: (404) Not Found.</Status>
</Link>
- <Link>
<Level>2</Level>
<MasterUrl>http://www.peter-lo.com/contact/index.htm</MasterUrl>
<Url>http://www.peter-lo.com/contact/images/</Url>
<Status>ok</Status>
</Link>
</Links>
```

Appendix 11. Progress Report (1 – 6)

INFORMATICS

OXFORD BROOKES UNIVERSITY

Progress Report No: **1** Date: **11 Nov 2008**

(to be completed by student)

Before I start to write the proposal of my selected topic, **Web Analysis and Diagnosis using Grid Computing**, I have to think how to improve the Business Environment for Web Hosting Service Provider and how to run a new service from the existing server without any new requirement involvement.

I found that, Web Hosting Service Provide hasn't the Web Analysis and Diagnosis Service to their customer. I think it is a charge to them for reaching consumer.

I have to search much documentation that is about Web Analysis and Grid Computing Technology, the competition will discuss in the project too. The documentation made me understood more about the existing technical problem in System Development.

I hope my project can guide the following researcher to learn more Web Analysis and Grid Computing Technology.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

OXFORD BROOKES UNIVERSITY

INFORMATICS

Progress Report No: **2** Date: **20 Dec 2008**
(to be completed by student)

I found some papers from university library, they are related with my project. These are the title of the paper:

1. **"A Note on Distributed Computing"**, *Jim Waldo, Geoff Wyant, Ann Wollrath, Sam Kendall; Sun Microsystems.*
2. **"Preparation Techniques for Large Scale Data Analysis of the Deep Web"**, *College of Computing, Georgia Institute of Technology.*
3. **"Self Adaptivity in Grid"**, *Sathish S. Vadhiyar[1], and Jack J. Dongarra[2,3]*

Those papers give me many concepts to design a workable Web Crawler in Grid Environment.

I also compared different Link Checkers and draw a table. All information can show the detail problem or feature of them. From the comparison, I can avoid all bad design in my project, and understanding the possibility of my project.

Finally, I prepared a basic System Development Schedule to guide me to complete the project.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

INFORMATICS

OXFORD BROOKES UNIVERSITY

Progress Report No: **3** Date: **18 Jan 2008**

(to be completed by student)

I identified what are the problems that needs to handle. In the system, I need to solve these problems:

1. Hyper Link Identification
2. Web Crawl threading in different level
3. Process Separation for Grid Computing

And I designed architecture of WADE, I had written a detail description to explain the whole workflow of the system.

I also describe three major functions in the document:

1. Rapid Sitemap Generation
2. Detail Error Logging
3. Unlimited Error Identification

Unlimited Error Identification will follow W3C standard to explain the error exception.

Finally, I did a research for all related methodologies. In the next month, I will start the system design and development.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

INFORMATICS

OXFORD BROOKES UNIVERSITY

Progress Report No: **4** Date: **28 Feb 2008**
(to be completed by student)

Before the design, I had written a Hardware and Software Specification.

In the design, I drawled some UML, such as, Activity Diagram, Use Case Diagram, Class Diagram, Collaboration diagram and Sequence Diagram. I also show some core code of the WADE for reference.

I made a detail Test Plan about the operation of WADE.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

INFORMATICS

OXFORD BROOKES UNIVERSITY

Progress Report No: **5** Date: **15 Mar 2008**
(to be completed by student)

After the development, I had completed following Demonstration & Evaluation:

6. The Steps of Software Installation
7. Basic Demonstration
8. Dataset Sampling
9. Basic comparison between single computer and grid environment
10. Random comparison between single computer and grid environment
11. Detail comparison between Domain-Based and Page-Based

Each step of Evaluation can find out many data to support Grid Computing is better than Single Computing.

The project will be completed soon. The conclusion and presentation will start in the next month before submit.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

INFORMATICS

OXFORD BROOKES UNIVERSITY

Progress Report No: **6** Date: **24 Mar 2008**
(to be completed by student)

I had completed the Conclusion and Presentation in this month. Conclusion included Project Achievement, Future enhancement Aspects of resources, Lessons learnt and Critical appraisal.

The presentation will go through the whole project and it will show the performance difference between Single Computing, Page-Based Computing and Domain-Based Computing.

Progress evaluation (to be completed by the supervisor)

General Progress:	(a) Very slow (b) behind (c) satisfactory (d) impressive
Research:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Tool Studying:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A
Implementation:	(a) Very slow (b) behind (c) satisfactory (d) impressive (e) done (f) N/A

Comments: _____

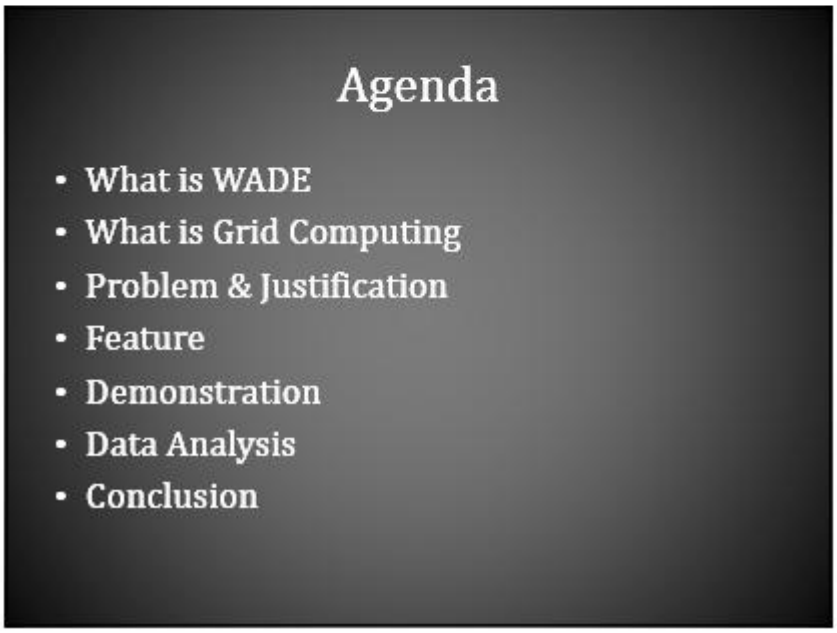
_____

_____

Supervisor Signature: _____ Duration _____ (hours)

Start Time: _____ End Time: _____

Appendix 12. Presentation Slides

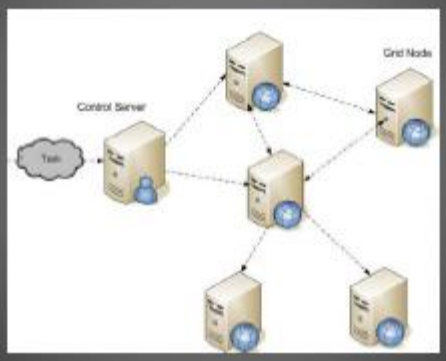


What is WADE

Web Analysis & Diagnosis Engine
is a Rapid Link Checking Service

My target: Web Hosting Service Provider

Grid Computing

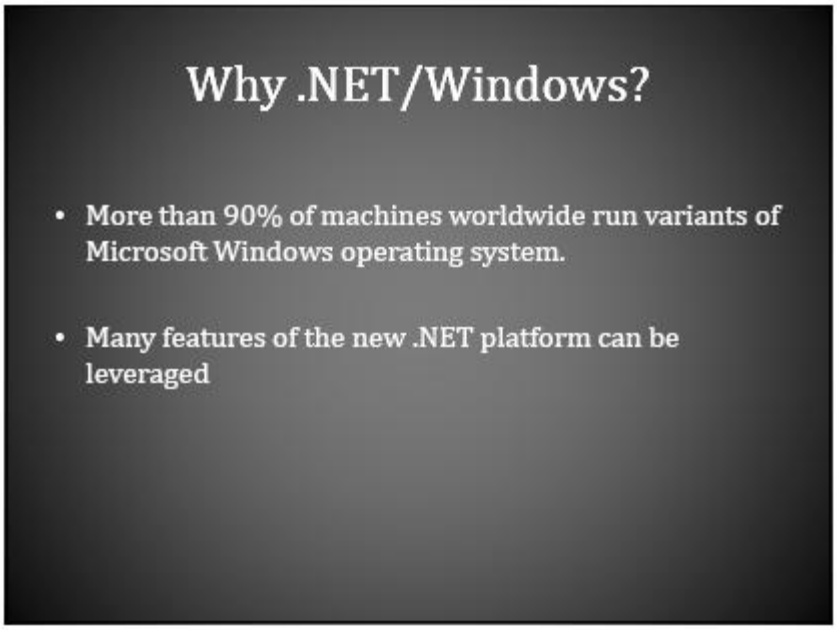


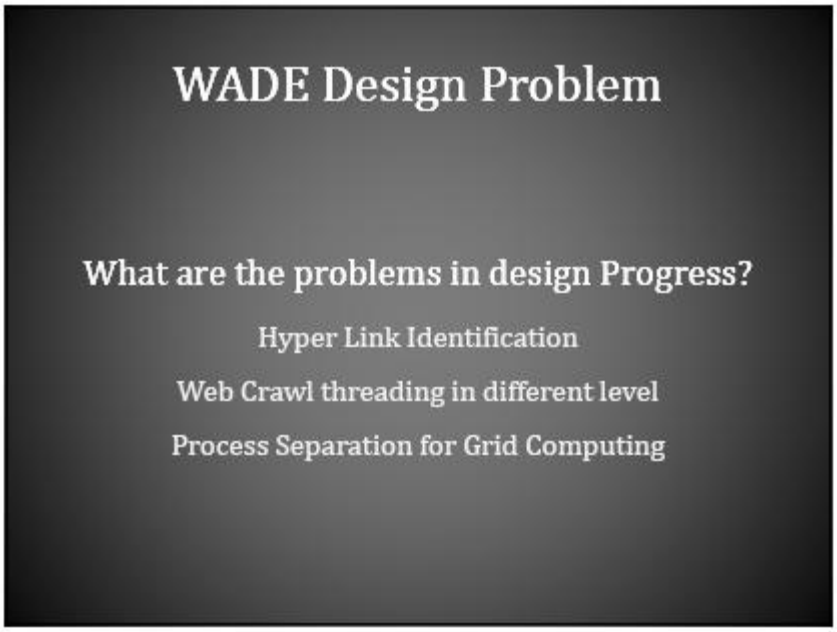
Why I choose Grid Computing

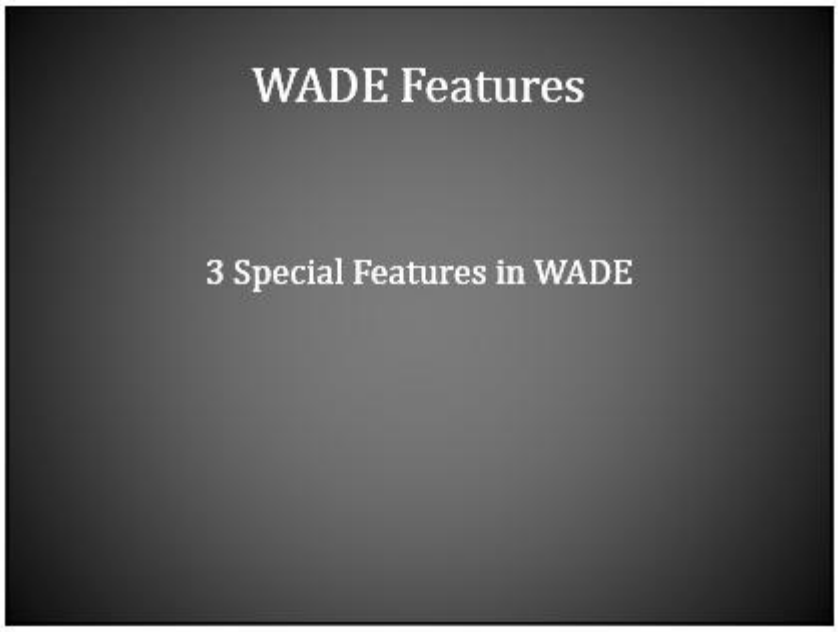
- Save calculation time
- Save resources

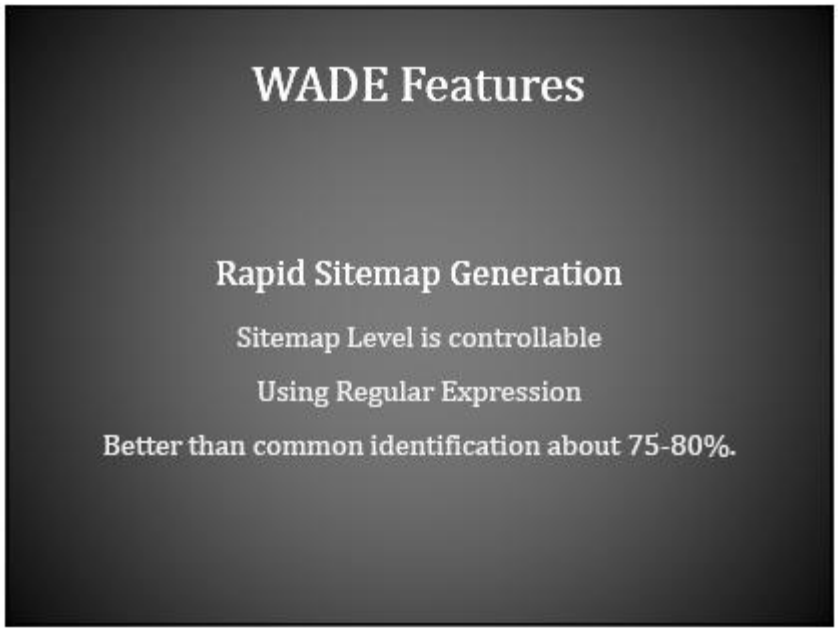
Alchemi: A .NET-based Enterprise Grid Framework

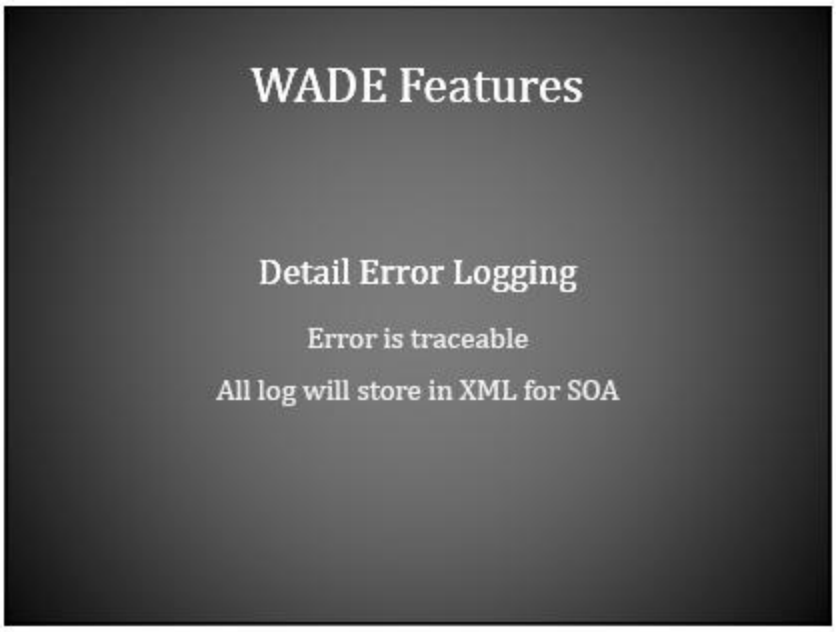
- Developed by The University of Melbourne, Australia
- .NET-based (Windows)

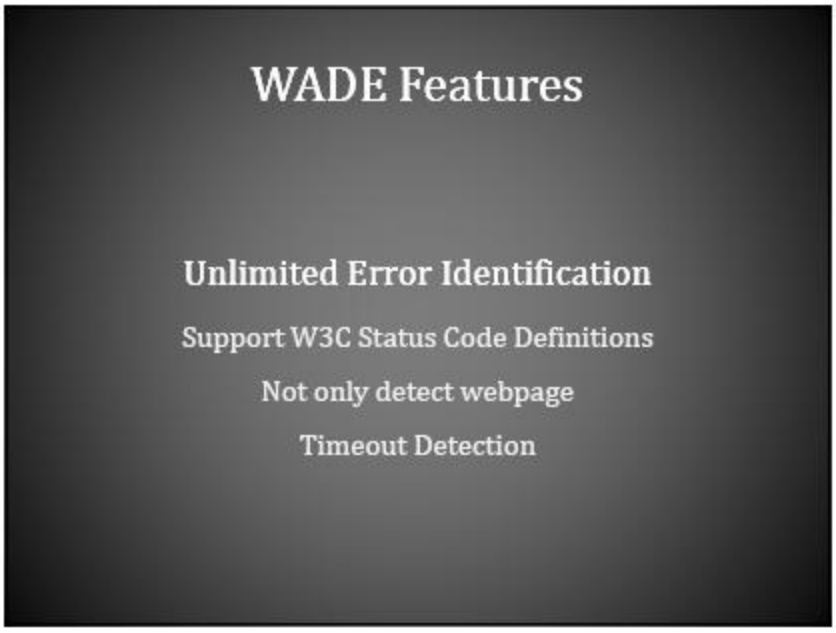

Why .NET/Windows?
• More than 90% of machines worldwide run variants of Microsoft Windows operating system.
• Many features of the new .NET platform can be leveraged


WADE Design Problem
What are the problems in design Progress?
Hyper Link Identification
Web Crawl threading in different level
Process Separation for Grid Computing


WADE Features
3 Special Features in WADE


WADE Features
Rapid Sitemap Generation
Sitemap Level is controllable
Using Regular Expression
Better than common identification about 75-80%.


WADE Features
Detail Error Logging
Error is traceable
All log will store in XML for SOA


WADE Features
Unlimited Error Identification
Support W3C Status Code Definitions
Not only detect webpage
Timeout Detection

Demonstration

What will be show?

Normal Web Crawler (Single PC)

Page-Based Web Crawler (Grid Computing)

Domain-Based Web Crawler (Grid Computing)

Data Analysis

Item	Single PC	Page-based	Domain-based
Domain List Separation	Master	Master	Master
Sitemap Generation	Master	Master	Node
Job Separation	-	Master	Node
Web Crawl	Master	Node	Node
Data Combine and filtering	-	Master	Node
Result Collection	Master	Master	Master
Reporting	Master	Master	Master

Conclusion

- Grid Computing can reduce the calculation time.
- Invalid Process Separation can decrease the performance of Grid Computing.
- WADE can provide the rapidest link checking service.

Q&A

Appendix 13. Coding

There will present about the Job Splitting, Thread Splitting and the core of Link Checking. It shows the calculation and the workflow of all functions.

Job Splitting

```
using System;
using System.Collections.Generic;
using System.Text;
using Alchemi.Core;
using Alchemi.Core.Owner;

namespace nodeapp
{
    class GridControl
    {
        public static bool isCompleted { get; private set; }
        public static GApplication App = new GApplication();
        public static List<Link> Results = new List<Link>();

        private string[] Url { get; set; }
        private bool SingleDomain { get; set; }
        private int Level { get; set; }
        private bool ExportXml { get; set; }

        public GridControl(string [] Url, bool SingleDomain, int Level, bool
ExportXml)
        {
            isCompleted = false;
            this.Url = Url;
            this.SingleDomain = SingleDomain;
            this.Level = Level;
            this.ExportXml = ExportXml;
        }

        [STAThread]
        public void Run()
        {
            App.ApplicationName = "Cloud Link Checker";
            Init();

            foreach (string u in Url)
                App.Threads.Add(new GetLinks(u, SingleDomain, Level, ExportXml));

            App.Start();

            try
```

```
            {
                App.Stop();
            }
            catch { }
        }

        private static void Init()
        {
            try
            {
                // get settings from user
                GConnection gc = new GConnection();
                gc.Host = "localhost";
                gc.Port = 9000;
                gc.Username = "user";
                gc.Password = "user";

                App.Connection = gc;

                // grid thread needs to
                App.Manifest.Add(new ModuleDependency(typeof(GetLinks).Module));

                // subscribe to ThreadFinish event
                App.ThreadFinish += new GThreadFinish(App_ThreadFinish);
                App.ApplicationFinish += new
GApplicationFinish(App_ApplicationFinish);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error: " + ex.Message);
            }
        }

        private static void App_ThreadFinish(GThread thread)
        {
            // cast the supplied GThread back to PrimeNumberChecker
            GetLinks pnc = (GetLinks)thread;
            Results.AddRange(pnc.links);
        }

        private static void App_ApplicationFinish()
        {
            isCompleted = true;
            Console.WriteLine("Application finished.");
        }
    }
}
```

Thread Splitting

```
using System;
```

```
using System.IO;
using System.Collections.Generic;
using System.Text;
using System.Threading;

using Alchemi.Core;
using Alchemi.Core.Owner;

namespace nodeapp
{
    [Serializable]
    class GetLinks : GThread
    {
        public int CurrentLevel { get; private set; }

        public int NumOfThread { get; private set; }
        private int linksStart { get; set; }
        private int linksEnd { get; set; }

        public List<Link> links { get; private set; }
        private string cachePath { get; set; }

        private string url { get; set; }
        private bool singleDomain { get; set; }
        private int level { get; set; }
        private bool exportXml { get; set; }

         public GetLinks(string Url, bool SingleDomain, int Level, bool ExportXml)
        {
            //Insert default value
            this.CurrentLevel = 1;
            this.NumOfThread = 2;
            this.linksStart = 0;
            this.linksEnd = 1;
            this.links = new List<Link>();
            //this.MakeCacheSpace();

            this.url = Url;
            this.singleDomain = SingleDomain;
            this.level = Level;
            this.exportXml = ExportXml;
        }

        private void MakeCacheSpace()
        {
            cachePath = @"c:\linkcache\";
            if (!Directory.Exists(cachePath)) Directory.CreateDirectory(cachePath);
        }
```

```
        public override void Start()
        {
            links.Add(new Link { Url = url, Status = "ok" });

            url = url.Split('/')[2];

            for (int lev = 1; lev < level + 1; lev++)
            {
                Console.WriteLine("Level {0}...", lev);

                List<string> selectedLinks = new List<string>();
                List<Link> tmplink = new List<Link>();

                for (int i = linksStart; i < linksEnd; i++)
                    selectedLinks.Add(links[i].Url);

                //Cal the each thread can handle how much url
                int perIndex = (selectedLinks.Count - (selectedLinks.Count %
NumOfThread)) / NumOfThread;

                List<string>[] tLists = new List<string>[NumOfThread];
                string[] tmpList = new string[perIndex];

                for (int i = 0; i < NumOfThread; i++)
                {
                    selectedLinks.CopyTo(perIndex * i, tmpList, 0, perIndex);

                    tLists[i] = new List<string>();
                    tLists[i].AddRange(tmpList);
                }

                //Add the last Url into last Thread List
                tmpList = new string[selectedLinks.Count % NumOfThread];
                selectedLinks.CopyTo(perIndex * NumOfThread, tmpList, 0,
tmpList.Length);
                tLists[NumOfThread - 1].AddRange(tmpList);

                ////Create specified num of Thread
                StartThreadingProcess(lev, ref tLists, ref tmplink);

                if (tmplink.Count == 0) break;

                //Remove all duplicated link
                Deduplicate(url, tmplink, singleDomain);

                //Setup the pointer
                linksStart = linksEnd;
                linksEnd = links.Count;
                Console.WriteLine("Found {0} links...", linksEnd - linksStart);
```

```
                CurrentLevel++;
            }

            if (exportXml)
            {
                PublishXml px = new PublishXml(url, links);
                px.Export();
            }
        }

        [STAThread]
        private void StartThreadingProcess(int currentLevel, ref List<string>[]
inputPool, ref List<Link> outputList)
        {
            //Create specified num of Thread
            GetLinksThread[] ts = new GetLinksThread[NumOfThread];

            //Add information into each thread
            for (int i = 0; i < NumOfThread; i++)
                ts[i] = new GetLinksThread("t" + i, inputPool[i], currentLevel,
cachePath);

            //Start all thread
            foreach (GetLinksThread t in ts)
            {
                t.thread.Priority = ThreadPriority.Highest;
                t.thread.Start();
            }

            //Wait all thread stopped
            foreach (GetLinksThread t in ts)
                t.thread.Join();

            //Collect all return information
            foreach (GetLinksThread t in ts)
                outputList.AddRange(t.resultList);
        }

        private void Deduplicate(string url, List<Link> inputLink, bool singleDomain)
        {
            for (int i = 0; i < inputLink.Count; i++)
            {
                //Check does it still under single domain
                if (singleDomain)
                    if (!inputLink[i].Url.Contains(url))
                        continue;

                bool dup = false;
                foreach (Link k in links)
```

```
                {
                    if (k.Url.Equals(inputLink[i].Url) &&
k.Status.Equals(inputLink[i].Status))
                    {
                        dup = true;
                        break;
                    }
                }

                if (!dup) links.Add(inputLink[i]);
            }
        }
    }
}
```

The Core of Link Checking

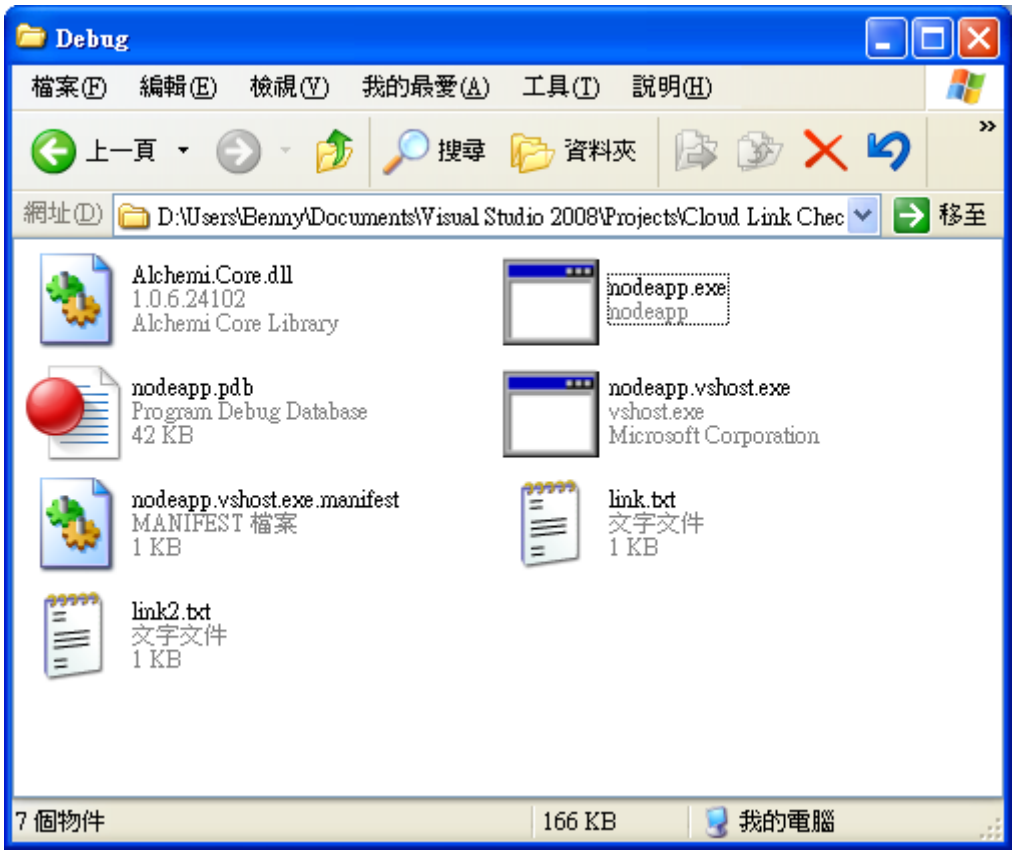
```
            foreach (string url in urlList)
            {
                //Console.WriteLine("{0}:{1}...", thrName, url);
                string CurrentUrl = url;
                string status = "ok";

                string[] linkE = CurrentUrl.Split('/');
                string lastE = linkE[linkE.Length - 1];

                System.Net.WebClient client = new WebClient();
                byte[] page = new byte[1];

                try
                {
                    //Check web format
                    string r = GetPageType(CurrentUrl).ToString();

                    switch (r)
                    {
                        case "text":
                            page = client.DownloadData(CurrentUrl);
                            break;

                        case "404":
                            throw new WebException("The remote server returned an
error: (404) Not Found.");

                        default:
                            throw new WebException("ok");
                    }

                }
                catch (WebException e)
                {
```

```
                    status = e.Message;
                    resultList.Add(new Link { Url = CurrentUrl, Status = status, Level
= CurrentLevel, MasterUrl = CurrentUrl });
                    continue;
                }

                //Load page cache
                //string cacheName = cachePath + ByteHash(page);
                //if (LoadPageCache(cacheName)) continue;

                string content = System.Text.Encoding.UTF8.GetString(page);
                string regex = "(href|src)(=|
= )[\\\"\\\']{http:\\/\\/|\\.\\/|\\/)?\\w+(\\.\\w+)*(\\/\\w+(\\.\\w+)?)*(\\/|\\?\\w*=\
\w*(&\\w*=\\w*)*)?[\\\"\\\']";
                Regex re = new Regex(regex);
                MatchCollection matches = re.Matches(content);

                System.Collections.IEnumerator enu = matches.GetEnumerator();
                while (enu.MoveNext() && enu.Current != null)
                {
                    Match match = (Match)(enu.Current);

                    //Remove the header [href="] and the footer ["]
                    string link = match.Value.Replace("\\", "").Replace(" ", "");

                    if (link.StartsWith("href"))
                        link = link.Substring(6, link.Length - 7);
                    else if (link.StartsWith("src"))
                        link = link.Substring(5, link.Length - 6);

                    //Console.WriteLine(link);

                    //Check does the path complete
                    if (link.StartsWith("./") || link.StartsWith("/")
|| !link.StartsWith("http"))
                    {
                        linkE = CurrentUrl.Split('/');
                        if (linkE[linkE.Length - 1].Contains("."))
                        {
                            string[] linkE2 = link.Split('/');

                            if (linkE2[linkE2.Length - 1].Contains(".") &&
linkE2.Length > 2) link = (link.Remove(link.LastIndexOf("/")));

                            link = (CurrentUrl.Remove(CurrentUrl.LastIndexOf("/")) +
"/" + link);
                        }
                        else
                        {
```

```
                            link = (CurrentUrl + "/" + link);
                        }

                        //If the link is not a web page or file, add [/] at the end
                        linkE = link.Split('/');
                        if (!linkE[linkE.Length - 1].Contains(".")) link += "/";

                        //Solve the duplicated [/]
                        link = link.Replace("//", "/").Replace(":/",
"://").Replace("./", "");
                    }

                    // Console.WriteLine(">> " + link);


                    //Add the url into list
                    resultList.Add(new Link { Url = link, Status = status, Level =
CurrentLevel, MasterUrl = CurrentUrl });
                }

                //Create Cache
                //CreatePageCache(cacheName);
            }
        }

        private string GetPageType(string link)
        {
            string result = "";

            try
            {
                bool isWeb = false;

                if (link.EndsWith("/"))
                {
                    isWeb = true;
                }
                else
                {
                    string[] formats = { "htm", "py", "php", "asp", "jsp", "js",
"msp", "?" };
                    foreach (string format in formats)
                        if (link.Contains(format)) { isWeb = true; break; }
                }

                if (isWeb)
                {
                    WebRequest myWebRequest = WebRequest.Create(link);
                    WebResponse myWebResponse = myWebRequest.GetResponse();
```

```
                result = myWebResponse.Headers.Get("Content-Type").Split('/')[0];

                myWebResponse.Close();
            }
            else return "Application";
        }
        catch { result = "404"; }

        return result;
```

Those codes can be compiled and it will be named “nodeapp.exe”. The same directory must include Alchemi.Core.dll for nodeapp.exe referred it. Nodeapp.exe is just running on the Server, need not to deploy to Node Server. Alchemi.Core.dll will help to deploy component to each registered Node Servers.



Appendix 14. Project Proposal

OXFORD BROOKES UNIVERSITY

informatics

Project Proposal Submission Form

Instructions : **1.** Please complete the form using Capital Letters. **2.** Completed proposal must be submitted through the Service counter			
Course :	**Oxford Brookes University**		
Name of Student :			
Email Address :		**Contact No :**	
Student ID :		**Class Code :**	**U08096**
Project Title :	**WADE: Web Analysis and Diagnosis Engine using Grid Computing Technology**		
Track	❑ ~~Computing and Information Systems~~ ❑ Computing and Software Engineering ❑ ~~Information Systems and Software Engineering~~ *(**Please tick the appropriate module)*		
Total Number of Pages including this cover page :	**30** *(** Please note the proposal should not exceed more than 8 pages. This excludes title, table of content, reference list and appendix pages)*		
Date of Submission			
Declaration : I declare that this assignment is my original work and that I have acknowledged any use of published or unpublished works of other people. I understand that I will be penalized for plagiarism and late submission.		**Signature :** ————————	
For Office Use Only			
Date of Reciept:		**Collected By:**	
For the Attention of:	**The Project Co-Ordinator, GSD Dept**		

Table of Contents

Abstract

Since 1990s, Grid computing becomes a popular research topic in Internet, Grid Computing allows several computers handle a single calculation at the same time, usually apply into scientific or technical problem. They are required a great number of computer processing to handle large amounts of data.

Thousands of Web Hosting Service Providers start up their business since 2000. This web hosting service provider not only providing Web Hosting Service, but also providing additional value-added services (such as SMTP, POP3, IMAP Servers, Free Sub-domains, Domain parking, Secure Socket Layer (SSL) and Graphical Hit Counter…etc.) From my research, all the Web Hosting Service Provider unable to provide a detail hosting reporting service to customer due to server utilization and time-consuming.

In this project, I will present how to apply the Grid Computing Technology in Deep Web Crawling and Analysis with limited server utilization and faster performance by using Grid Computing Framework, Alchemi. I will discuss how to split a thread for taking the highest performance, what are the unexpected error will occur when the thread splitting is wrong, how to crawl a website through multi-threads in Grid Computing and how Grid Computing can help researcher save more time to complete their calculation. It is an important topic in Grid Computing. I will explain which methodologies will be applied in the application. All selected methodologies can help the Application running parallel in the Grid Environment.

In the experiments session, we evaluate Web Analysis and Diagnosis Engine (WADE) by using over thousand of web pages from the education websites through Google Directory. The experiments will show the algorithms of the web crawling and grid computing with excellent accuracy and performance, and we will show the performance of different process separation. All researchers can find the solution to fix the bottleneck of performance when the application cannot meet the expectation in Grid Environment.

Proposal Narrative

Project Title

Web Analysis and Diagnosis using Grid Computing

Web Analysis and Diagnosis Engine (WADE) is a Utility tool for Web Hosting Service Provider. WADE can help Service Provider to find out the page error of their client domain and report to them for the best customer-service.

Grid computing is a great platform to enlarge the performance of WADE and making the best use of computer resources. ICDSoft (Hong Kong) Limited, the leader of Web Host Service provider in Hong Kong, responds many web servers haven't use more than 30% CPU. So I want to use the other 70% to do the right thing.

For apply WADE, they need not to build up a new server. They can gather all the Web Servers to be Grid Nodes for reducing cost.

Project Objectives

Web Analysis and Diagnosis Engine (WADE) will be the rapidest, most accurate analysis solution for Web Hosting Service Provider to provide a new service to their customer for increasing loyalty and enterprise image. Through supporting SOA, the industries can be easy to bundle the report to them reporting Application or for the administrator keep references. In this bad business environment, WADE should be a low cost and large benefit solution. Based on Grid Computing, they need not invest any additional budget to purchase a powerful server for serve WADE.

Problems & Justification

Hyper Link Identification

In a webpage, all links are not only presented by Link Element (<a href>), but also images are using Image Element (<img>) to present in the browser. For a Link Crawler, it should know how to find them out.

In some developer's habit, they won't enter a full URL for the Link in each page.

For example: (www.domain.com)

The standard format should:

Service

But they always use this format for the same level:

Service

And some case for taking back to top level

Back to service

So the Link Crawler needs to handle these issues and re-engineer all the links to the standard format for the Link Parser use.

Web Crawl threading in different level

Crawler couldn't use a single thread to capture the links in the page. Single thread will waste the all un-used CPU resources and extend the crawling time. For the highest performance, Crawler will create a serial of threads to capture multi-pages at the same time.

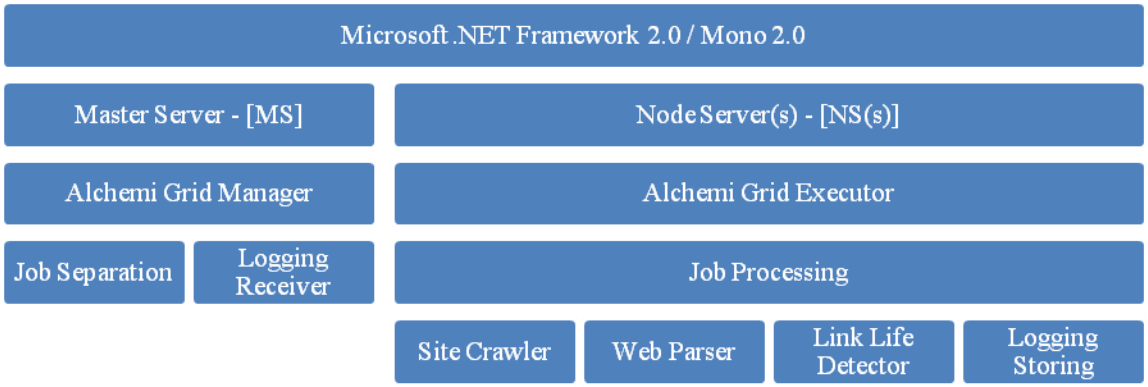
Process Separation for Grid Computing

Process Separation is a difficult job in development. Grid Computing Performance is very depends on Process Separation. If the process split too small, Grid Master Server will need more resources to assign job to Node Servers and receive all response from Node Servers. It will increase the workload of Master Server and decrease the whole job performance.

So I need to study and try the solution that is the preface to present a great performance of Process Separation.

System Architecture

WADE System Distribution



The Workflow of WADE

Major Functions

Rapid Sitemap Generation

Site Crawler will use Regular Expression to generate a Virtual Sitemap through Grid Computing for Web parser operation. End-user can control the number of sitemap level for system crawling deeply.

Unlimited Error Identification

Web Parser does not only analyze webpage in the webpage, but also it can identify all file types and identify all the list of HTTP status codes (e.g. 200, 302, 404 and 502…etc) following the W3C Status Code Definitions for all the types of web page, image, document and large multimedia with Grid Computing Power.

Detail Error Logging

End User can trace back the error easily. WADE is different than other Link Checkers. WADE will provide the detail occurred path of link error to end-user for finding out the problem source.

Literature Review

The purpose of this part is to introduce a short overview on the literature used to create a technical foundation for the "WADE: Web Analysis and Diagnosis Engine using Grid Computing Technology" project. This part introduces the papers and documents used during research and giving an insight into, how those papers relate to this topic. For grant a better overview of the literature, it subdivides into five categories: Distributed Computing, Architectures, Algorithms and Web crawling.

1.1. Distributed Computing

This chapter introduces relevant paper regarding "Distributed Computing"

Distributed Computing Issues

This document is an "A Note on Distributed Computing" that is written by Jim Waldo, Geoff Wyant, Ann Wollrath, Sam Kendall; Sun Microsystems. It introduces almost common or known topics that related to the "Distributed Computing" and explains their issues. The topics covered by the different authors are not exclusively technical like Development in Distribution Environment, Resource Allocation, Synchronization and Failure recovery. It additionally addresses the three programming stages as well. At the end of the document the authors talk about the topics that they think the research is needed in the future. Two of them are "Guaranteed separation" and "Class Replacement without affecting the other parts of system".

The Vision of Unified Objects

The authors have used a point of view from programmer to explain what distributed system is and explain the operation of distributed object. This topic discusses some advantages of the remote class and identified some principles about design model in distributed system:

- There is a single natural object-oriented design for a given application, regardless of the context in which that application will be deployed;
- Failure and performance issues are tied to the implementation of the components of an application, and consideration of these issues should be left out of an initial design; and
- The interface of an object is independent of the context in which that object is used.

Local and Distributed Computing

This topic is all about the differences between local and distributed computing concern: Latency, Memory access, Partial failure and concurrency. Mainly that is focus on discussing the technical issues of resource allocation, synchronization and failure recovery.

Authors said a multi-threaded application needs to deal with Latency, Memory access, Partial failure and concurrency issues. There is a subtle difference. There is no real source for the indeterminacy invocation of operations in multi-threaded application development, so the programmer needs to fully control over invocation.

The Myth of "Quality of Service"

This topic is extending the previous discussion about how to base on resource allocation, synchronization and failure recovery to develop a Quality of Service.

It brings a summary that suppose that the interface describes the object, which supports a number of other objects. A definition of the sets is that there is no duplication. Thus, the implementation of this object makes a duplicate elimination. If the interface does not provide a way to check system information, a set of objects will be questioned to determine equality. Thus, duplicate elimination can only be done by interaction with the objects of the set. No matter how fast the objects of the set of the transaction. The overall efficiency of removing the duplicates will be governed by the latency to communicate over a slow connection is involved. There has not any change in the set of implementations that can overcome this. Interface design problem to determine the upper limit for this performance of operation.

Lessons from NFS

In this topic, Authors discussed some technical issues of NFS (Sun's distributed computing file system) and finding out what the functional limitations are. The limitations on the reliability and robustness of NFS cannot be fixed in the implementation of the parts of that system. There is no "quality of service" that can be improved to eliminate the need. Finally, Author provided some solution to solve the NFS problem. Require the centralized resource manager, which can detect the failure of resource recovery and begins to insure consistency of the system.

1.2. Architectures

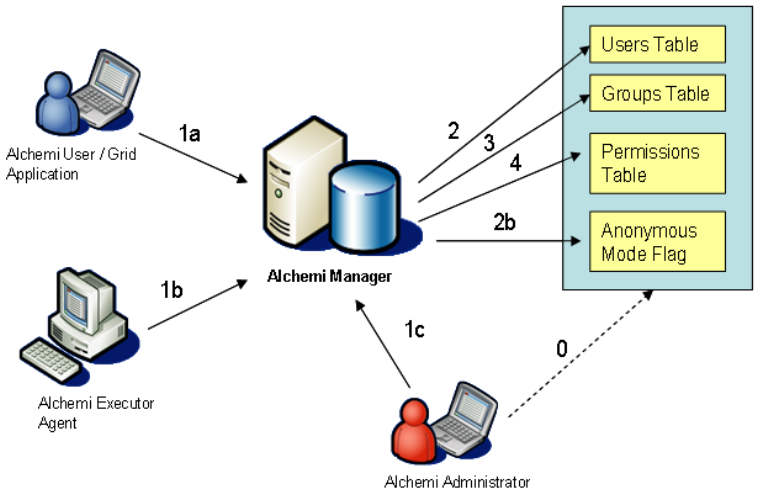
This chapter introduces relevant paper regarding “Enterprise Grid Framework”

Grid Computing Architecture Issues

This document is an “Alchemi: A .NET-based Enterprise Grid Framework” that is written by Krishna Nadiminti (Active developer), A. Luther (Project founder/Developer) and R. Buyya (CI/Mentor); University of Melbourne. Alchemi is the first Grid Computing Architecture using Microsoft .NET Framework Technology. The document mainly discusses what the benefits in Alchemi Architecture are. It also explains why a good Grid Architecture can improve the performance of Multi-thread Application. That is a topic extending “A Note on Distributed Computing”.

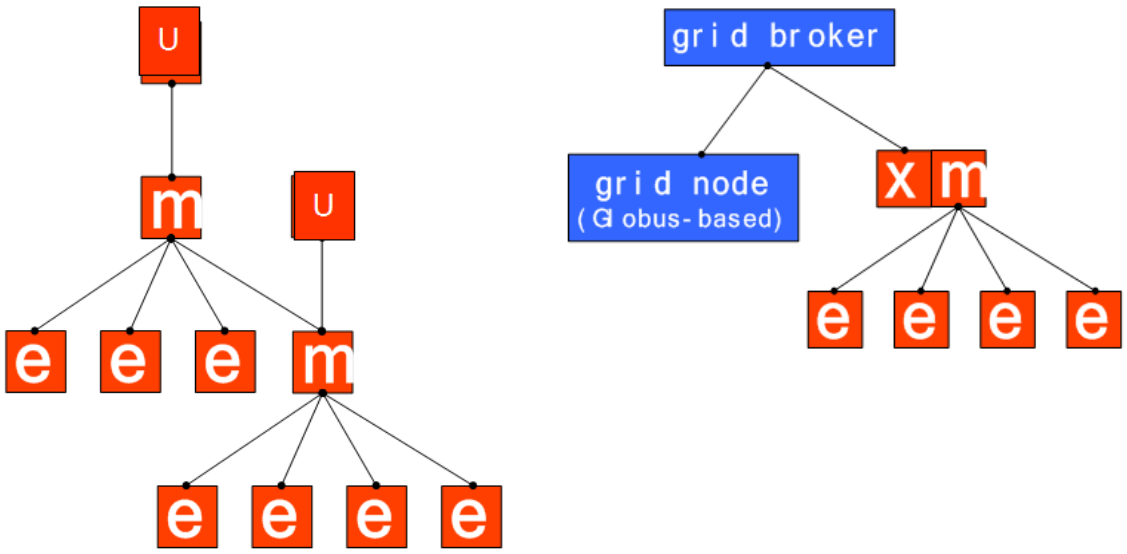
Alchemi Architecture Security

Alchemi is using Role-based Security to protect the Grid Environment to ensure no hacker can use the grid resources without authorization. The security contains three levels, Authentication, Authorization and Auditing. Authentication means checking the User Name and Password, if the login information is valid, the system will grant the permission to him base on his user account, it’s Authorization. When the user submits a job to Grid Environment, All jobs/threads executed are recorded in a database and linked to user account used for Authentication.



Multi-level Grid Design

Alchemi supports a cross-domain level architecture. That means it can gather different century's computer network and work together hierarchically. It is an advanced method to apply into high-computing issues.



Just Use, without difficult technical concern

Alchemi provides a very simple programming model for programmer develops a multi-threaded application. Alchemi is an Object-Oriented Grid Thread Model. It contains those main components to provide service. Grid Application consists of independent grid threads. Manager, central controller is used to discovery, scheduling, dispatching and monitoring. Cross Platform Manager is a Web Service Interface for controlling the Grid Environment through Browser. Executor is a worker agent that can install any type of computer, such as Windows and Linux. User means the role that is running grid applications, monitoring and administration. It provides some functional design for the grid operation, transparent execution of threads, Event-driven and Reusable drag and drop components.

1.3. Enterprise Grid Computing

This chapter introduces relevant paper regarding "Enterprise Grid Computing"

Introduction

This document is an "Enterprise Grid Computing" that is written by Paul Strong, Sun Microsystems, ACM Digital Library, July 1, 2005. Paul has written about he has to admit a great measure of commiseration for the IT society at large, when it is confronted with a hail of hype about the network technologies, especially within the enterprise. He also talks about the Definition of Grid computing deeply and some topics about the implementation of Grid Computing in Enterprise Data Centre, and what should we care about the Grid Computing in the future.

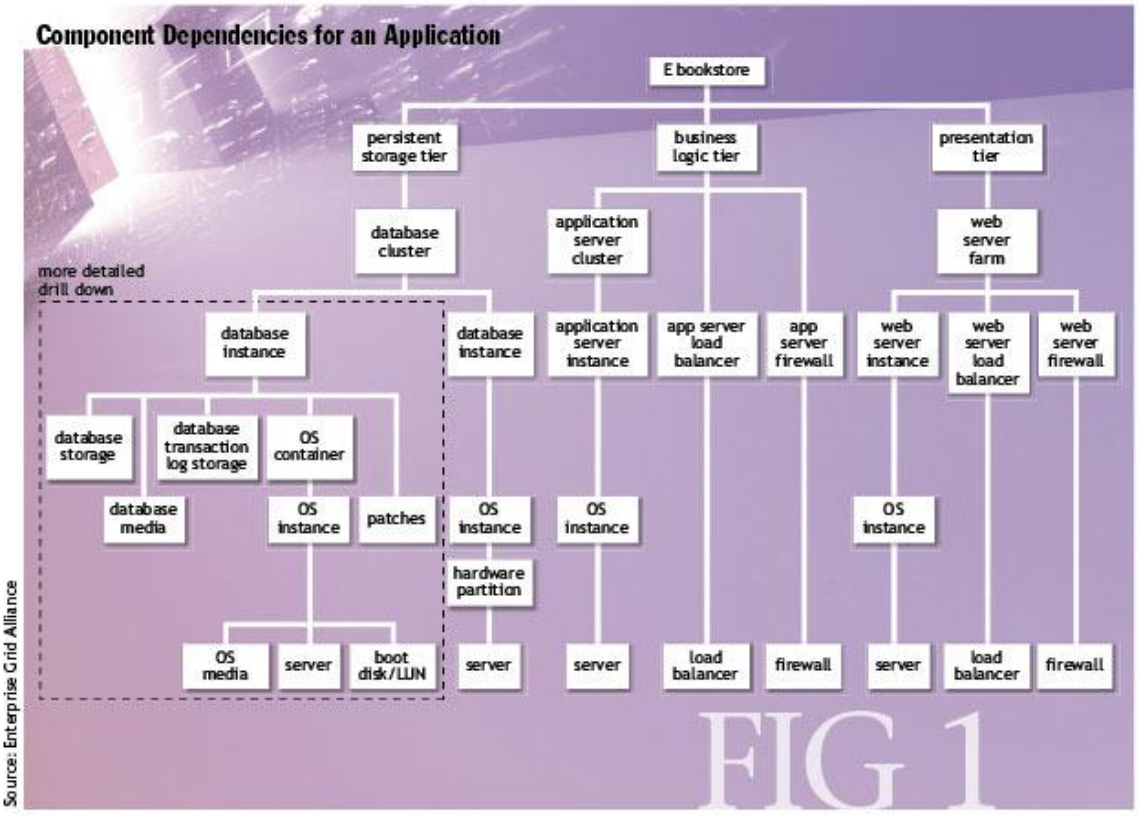
Grid Computing

At the heart of the network is the concept that applications and resources are connected in the form of a fabric or network ubiquitous network. In addition, the network concept implies both ubiquity and predictability, with networks being viewed as very similar to electrical power grids or 1, which are accessible everywhere and sharable by all.

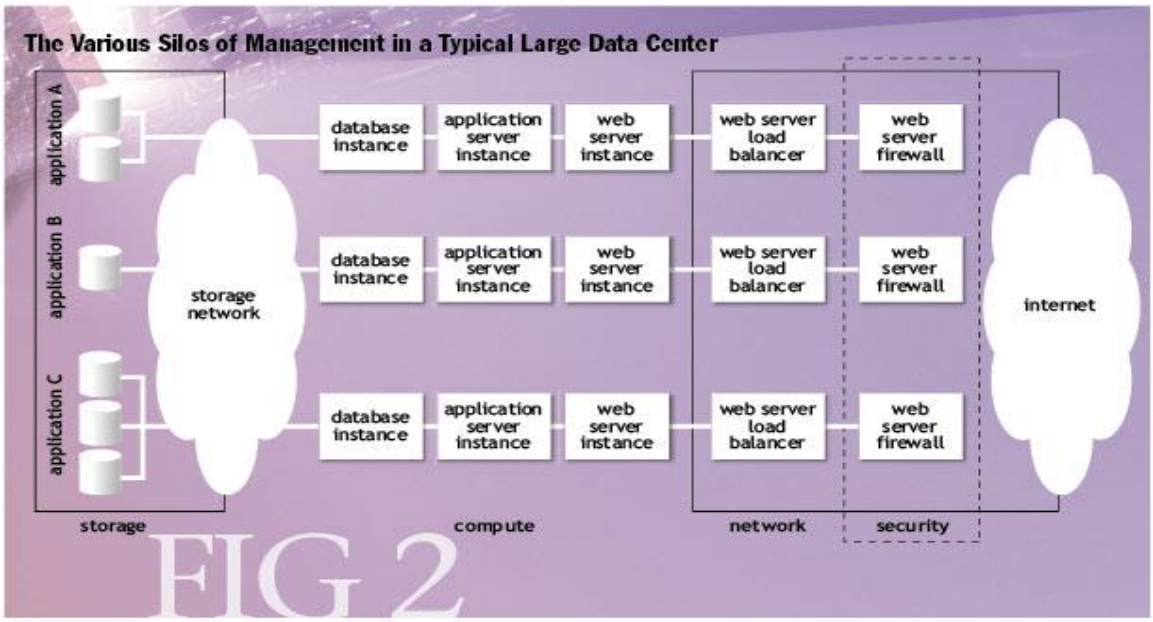
Grid computing is an inevitable consequence of a set of long-term technology trends. These trends have fueled each other, at least the last two or three decades, resulting in the application and infrastructure architectures we see today.

The Enterprise Data Center Today

Today the enterprise data center is a complex place. Each normally hosts a multitude of applications or services running on a large number of network resources. Each of the components of this tissue, either an application or resource, whether physical or logical, is relatively simple, but once you put all together, the complexity increases exponentially. When adding a component not only adds to the total number of components, you can also add a new type of component and a set of relationships with existing components in the tissue. View a typical enterprise application, such as an electronic library. The application can be divided into levels, such as storage or database, business logic, and presentation. Firewalls may exist between some of these levels. Each level may consist of a set of servers that run application components and perhaps a group or load balancing framework. Each server must run at least one application component, which may depend on some version of an operating system, along with a certain set of patches, all running on a particular type of processor. Figure 1 illustrates the complexity of the average data center in the form of a simplified graph for the unit only one application. Add another 10 or 100 such applications and the relationships between them and have an idea of the complexity that must be administered daily in a typical data center.



Today, companies mitigate the effects of complexity by creating silos of relatively stable infrastructure at a divide-and-conquer approach to management. In a typical data center, separate groups for managing their servers and operating systems, network components, storage components, security, and joint services applications. This is illustrated in Figure 2.



Complexity is addressed effectively by limiting the total number and types of components and their relationships. This allows the performance, scalability and availability of the inherent attributes of the distribution network architectures to be exploited, but is usually at the expense of efficiency and agility. Silos or replacement as a result of excess capacity in each silo, which is much less efficient than shared, dynamically allocate the excess capacity. These silos static as a result lack of agility, as new silos that have been created for new applications and services, rather than simply using perhaps an excess capacity.

Bringing the Threads Together

Virtualization, abstraction, and automation are the mechanisms that are keys to making the modern data center in a real network of an enterprise network and providing greater efficiency and agility. These mechanisms are usually performed in combination with a product, for example in the server and operating system provisioning tools, complex services and applications management lifecycle tools, service standards and management tools.

The key to extracting maximum value from these tools is that they share an architectural and operational.

A shared architecture should ensure that the right tools to solve problems the right way. This is the value of the various consortia in the network-for example, the EGA and GGF, which is leading to a series of requirements and an architectural model, respectively. Combining this with the use of standards for the various protocols and management mechanisms (many of which are incipient but nevertheless on the way) should allow data centers to choose the joint interoperability of tools appropriate to their needs, without fear of vendor lock-in.

1.4. Web Crawling Algorithms

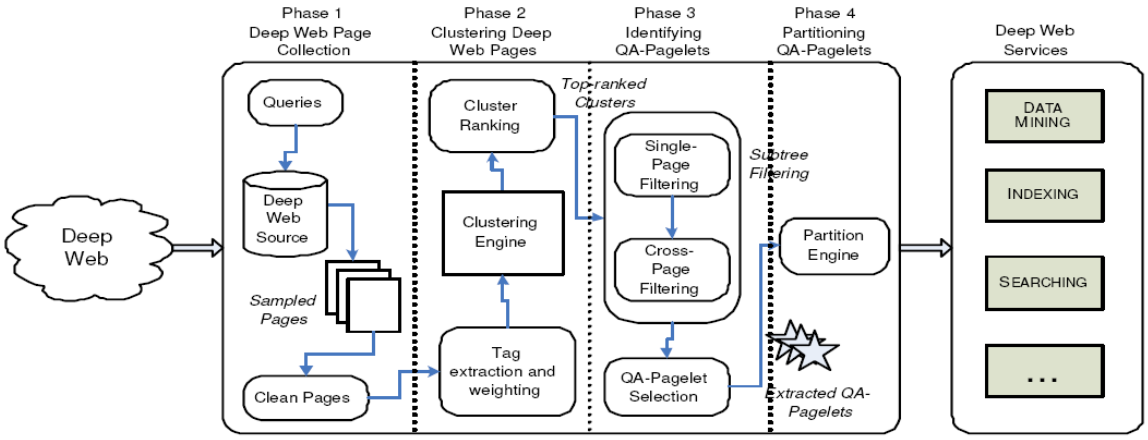
This chapter introduces relevant paper regarding "Alchemi QA-Pagelet: Data Preparation Techniques for Large Scale Data Analysis of the Deep Web".

Grid Computing Architecture Issues

This document is an "Alchemi QA-Pagelet: Data Preparation Techniques for Large Scale Data Analysis of the Deep Web" that is written by James Caverlee and Ling Liu College of Computing, Georgia Institute of Technology. It provides complete research information for reader to learn the foundation of Web Crawling in Grid Computing and also let the reader avoid some common technical issues through reading the documentation.

Authors discuss many related algorithms in the document. Each stage has different methodology to handle the current issue. In the following content, I will drifty explain the algorithms that I will apply in my project.

Web Crawling Stages



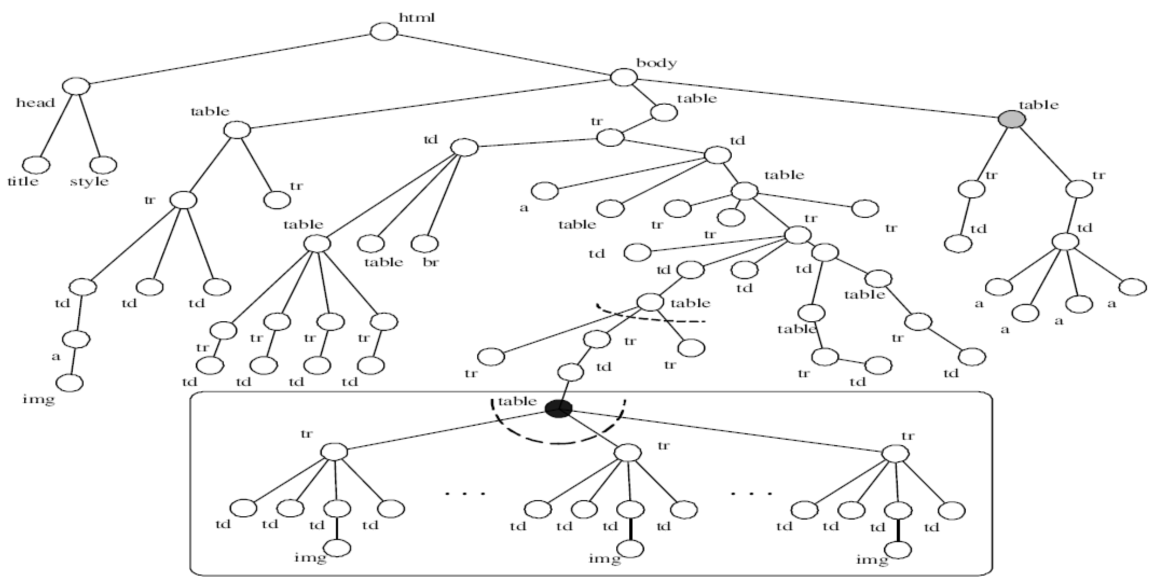
In Web Crawler, it contains 4 phases to process the website, including Web Page Collection, Clustering Web Page, Identifying QA-Pagelets and Patitioning QA-Pagelet. At the end, all result will export into database for Data Mining, Indexing and Searching…

These observations suggest that naturally, they have to take Thor's four stages, as shown. RST stage collects the sample pages of the answer, in response to queries over the Deep Web source. The second phase of the sample groups in response to individual groups of pages to pages that are related to their common control flow dependence, each corresponding to one type of site to answer: whether it is road games pages, page one game, does not match the pages except for pages. The third stage determines the QA-Page-lets high position on the page of each group according to the separation of sub-tree

in a set of clusters on a single page list of common sub-tree sets ranked by their actual diversity. Every single sub-tree corresponds to one set of content-type of the region set by the control-flow dependent answer pages. Then, using the internal cluster of indicators common to filter out content, and enhance the KU-Page-lets. At the end of the third stage, Thor recommends ranking list QA-Page-lets. The fourth phase of the bulkhead is the place to Page-lets KU-KU-detailed objects, which, in turn, add to the other Thor Deep Web information platform.

In our case, those 4 phases are suable for the project to analysis the website.

Tag Tree



Sample Tag Tree from IBM.com

Using the known variations of a Document Object Model, their system changed the website as a tag tree composed of tags and text. Under the tag, it is all the characters between the opening bracket "<" and a closing bracket ">", where each tag in the tag name (eg, BR, TD), and set attributes. The sequence of text characters between the sequence tags.

To translate the website into a tree tag requires that the page be formed. Requirements page is well formed, only the following: start tag including standalone guidelines, must have a corresponding end tag, all attribute values must be in quotes; tags are strictly slot. Pages which do not meet these criteria are automatically converted into well-formed using the Tidy [http://tidy.sourceforge.net/]. Properly developed site can be modeled as a guideline the tree T consists of tag nodes and content nodes. Tag node consists of all

characters from in particular to initiate the appropriate tag and end tag is marked with the name of the start tag. Content of the node consists of all characters between the start tag and the corresponding end tag or between the end tag and the start of the next tag. They mark the node to its content. All content nodes leave the tag tree.

They have made some definition of Tag Tree in a Web Objectization:

Definition 1 (Tag Tree): A tag tree of a page p is defined as a directed tree $\mathcal{T} = (V, E)$ *where* $V = V_T \cup V_C$, V_T *is a finite set of tag nodes and* V_C *is a finite set of content nodes;* $E \subset (V \times V)$, *representing the directed edges.* $\mathcal{T}$ *satisfies the following conditions:* $\forall (u, v) \in E, (v, u) \notin E$; $\forall u \in V, (u, u) \notin E$; *and* $\forall u \in V_C$, $\nexists v \in V$ *such that* $(u, v) \in E$.

Definition 2 (Subtree): Let $\mathcal{T} = (V, E)$ *be the tag tree for a page d, and* $\mathcal{T}' = (V', E')$ *is called a subtree of* $\mathcal{T}$ *anchored at node* u, *denoted as* $\text{subtree}(u)$ $(u \in V')$, *if and only if the following conditions hold: (1)* $V' \subseteq V$, *and* $\forall v \in V, v \neq u$, *if* $u \Longrightarrow^* v$ *then* $v \in V'$; *and (2)* $E' \subseteq E$, *and* $\forall v \in V', v \neq u, v \notin V_C, \exists w \in V', w \neq v$, *and* $(v, w) \in E'$

Definition 3 (Minimal Subtree with Property P): Let $\mathcal{T} = (V, E)$ *be the tag tree for a page p, and* $subtree(u) = (V', E')$ *be a subtree of* $\mathcal{T}$ *anchored at node* u. *We call* $subtree(u)$ *a minimal subtree with property* P, *denoted as* $subtree(u, P)$, *if and only if* $\forall v \in V, v \neq u$, *if* $\text{subtree}(v)$ *has the property* P, *then* $v \Longrightarrow^* u$ *holds.*

Definition 4 (QA-Pagelet): A QA-Pagelet is a minimal subtree that satisfies the following two conditions: (1) A QA-Pagelet is dynamically-generated in response to a query; and (2) it is a page fragment that serves as the primary query-answer content on the page.

Condition 1 the definition does not cover all the static parts of a page that is common to many Deep Web sites, such as navigation bars, the standard explanation, Standard, etc. However, not all regions of dynamically generated content, these definitions are designed to be direct answers to the query. Condition 2 is necessary to exclude from the definition of those regions, such as advertising, which are dynamically generated but is of secondary importance. The subtree corresponding to the QA-Pagelet is in dashed box. KU-Page-let roots are shaded in black and an assembly table.

Website Clustering

It notified about the page clustering problem, the explanation of Concrete similarity metrics is telling us how to select a suitable clustering algorithm to do a job. It analyzed

URL-based, Link-based, Content-based and the Size-based. It also introduces the two well-known clustering algorithms, Simple K-Means and Bisecting K-Means :

SimpleKMeans(Number of Clusters k, Input Vectors $\mathcal{D}$)
Let $\mathcal{D} = \{d_1, ..., d_n\}$ denote the set of n page vectors
Let N denote the total number of distinct tags in $\mathcal{D}$
Let $d_j = < (tag_1, w_j 1), \ldots, (tag_N, w_{jN}) >$ denote a page vector of N elements, w_{jl} is the TFIDF weight of the tag_l in page j $(l = 1, \ldots, N)$
Let $\mathcal{C} = \{C_1, ..., C_k\}$ denote a clustering of $\mathcal{D}$ into k clusters
Let μ_i denote the center of cluster C_i
foreach cluster C_i
  Randomly pick a page vector, say d_j from $\mathcal{D}$
  Initialize a cluster center $\mu_i = d_j$, where $d_j \in \mathcal{D}$
repeat
  foreach input page vector $d_j \in \mathcal{D}$
    foreach cluster $C_i \in \mathcal{C}$ $i = 1, \ldots, k$
      compute $\delta_i = sim(d_j, mu_i)$
    if δ_h is the smallest among $\delta_1, \delta_2, \ldots, \delta_k$
      mu_h is the nearest cluster center to d_j
    Assign d_j to the cluster C_h
  // refine cluster centers using centroid of each cluster
  foreach cluster $C_i \in \mathcal{C}$
    foreach tag l in d_j $(l = 1, \ldots, N))$
      $cw_{ij} \leftarrow \frac{1}{|C_i|} \sum_{l=1}^{N} w_{jl}$
    $\mu_i \leftarrow < (tag_1, cw_{i1}), \ldots, (tag_N, cw_{iN}) >$
until cluster centers no longer change
return $\mathcal{C}$

Tag Tree Signature Simple K-Means Page Clustering Algorithm

BisectingKMeans(Number of Clusters k, Input Vectors $\mathcal{D}$, Iterations I)
Define a clustering $\mathcal{C} = \{C_1\}$
foreach input vector $d_j \in \mathcal{D}$
  Assign d_j to C_1
for $i = 1$ **to** $k - 1$
  Select a cluster $C_i \in \mathcal{C}$
  Let $\mathcal{D}_i$ denote the set of page vectors in C_i
  Define a set of candidate clusterings $Candidate = \{Candidate_1, ..., Candidate_I\}$
  for $j = 1$ **to** I
    $Candidate_j \leftarrow$ SimpleKMeans(2, $\mathcal{D}_{C_i}$)
  $\hat{C} \leftarrow$ BestClustering($Candidate$)
  $\mathcal{C} \leftarrow \{\mathcal{C} \cup \hat{C} \setminus C_i\}$
return $\mathcal{C}$

Tag Tree Signature Bisecting K-Means Page Clustering Algorithm

And it recommended a better way for selecting the page clustering algorithm:

Average Fanout: Clusters that have pages with higher average fanout may be more likely to contain QA-Pagelets. The average fan-out for a *Cluster*$_i$ can be computed by the average of the largest fanout of a node in each page of the cluster. Namely,

$$\frac{1}{|Cluster_i|} \sum_{p \in Cluster_i} \max_{u \in p.V} \{fanout(u)\}$$

The *p.V* denotes the set of nodes in page *p*.

Average Page Size: Larger pages may tend to be more likely to contain QA-Pagelets. We define the average page size for a $Cluster_i$ as

$$\frac{1}{|Cluster_i|}\sum_{p \in Cluster_c} Size(p)$$

The *Size(p)* denotes the size of page *p* in bytes.

An excellent Algorithm can make the calculation rapidly. It tells that when splitting a thread to Cluster Server, we should use domain based, not page based. Because the thread too small, that will increase the Server loading to collect, filtering and sorting the distributed threading from different Node Servers.

The researchers have made a detail experimental to prove their theory. Test in 50 websites and within 100 pages for each site to create data sets of 55,000 pages (1,100 pages per site), 550,000 pages (11,000 pages per site), and 5,500,000 pages (110,000 pages per site).

1.5. Web Services in Grid Computing

This chapter introduces relevant paper regarding "Experiences with GRIA – Industrial applications on a Web Services Grid" that is written by Mike Surridge and Steve Taylor, IT Innovation Centre, IEEE.

GRIA Project

The GRIA project is designed to be used by the Net industry. The GRIA middleware is based on Web Services, and aims to meet the needs of industry for security and business-to-business (B2B) service procurement and operation. This offers a well-defined B2B models for accounting and QoS agreement, and proxy-free delegation's support for account management and service federation. The GRIA v3 software is currently used in industry. A business-oriented approach, irrespective of the Open Grid Services Architecture proposals for changing the Global Grid Forum, GRIA has demonstrated the need for a wider understanding of Virtual Organizations (Vos). The traditional academic Vos are continual, resourceful, and, logically centralized, membership-oriented management structures. In contrast, the GRIA experience has been that the business is likely to project focused Vos and distributed process-oriented management structures.

Starting with the seemingly more modest goal of a business support existing Grid system, the GRIA project, new software is fully Web Services, and focused on the beginning of commercial business applications and business models. This includes the off the-side Web Services technologies, security add-ons, and the model-based access control process, which is in the business processes. It is also a stimulus for the development and standardization in the field of B2B negotiation, mediation and resource methods. GRIA stresses the need for a wider range of different VO models, including the fast and agile B2B models, as well as the large, long-VO models feature a number of large-scale scientific research cooperation. GRIA also highlights the need for the Semantic Grid to support open markets and processes. These will be addressed in future work using the EC IST GRIA middleware project Next GRID and SIMD.

1.6. Advanced Grid Computing Control

This chapter introduces relevant paper regarding "Self Adaptivity in Grid Computing".

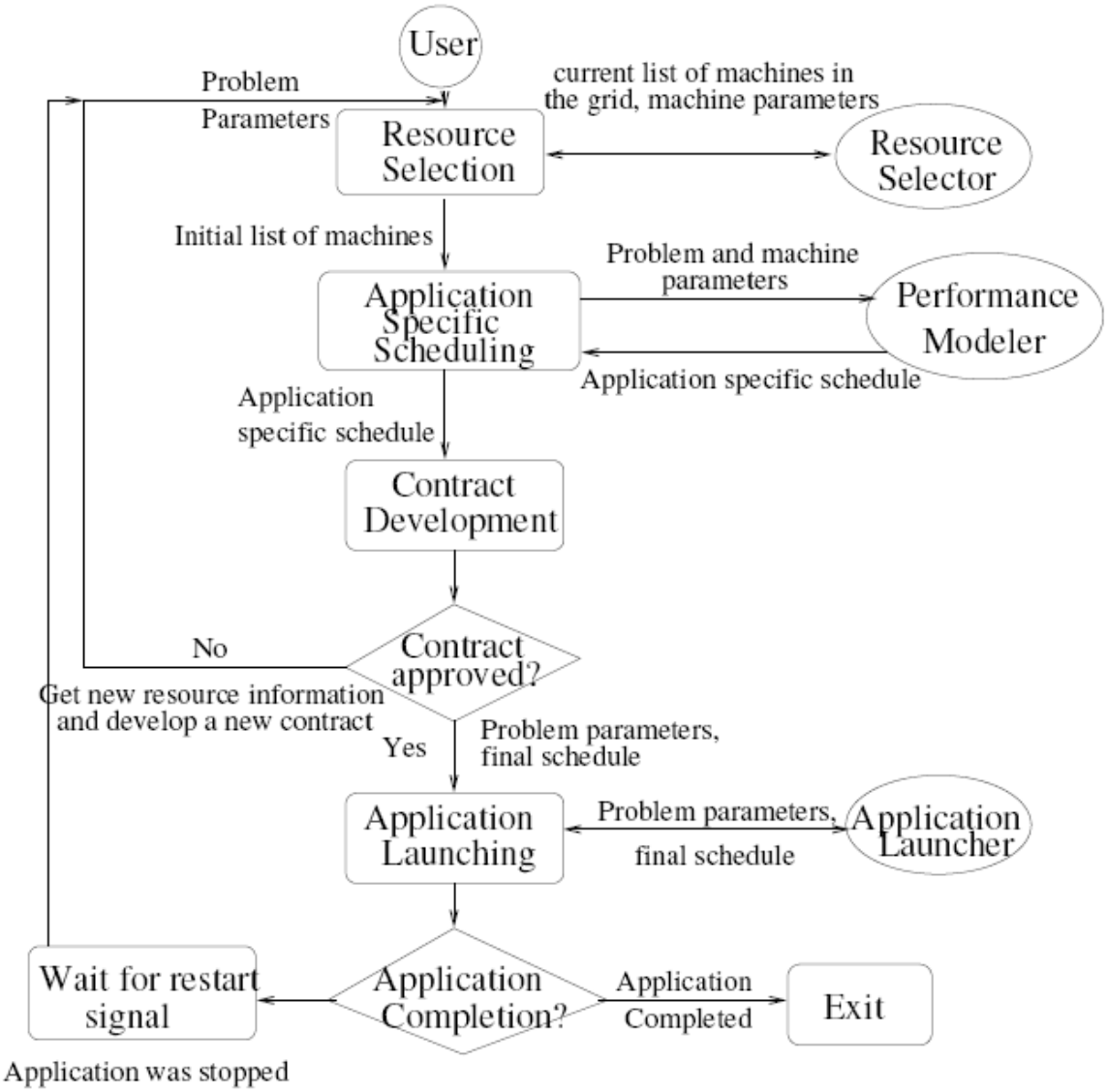
Introduction

This document is an "Self Adaptivity in Grid Computing" that is written by Sathish S. Vadhiyar and Jack J. Dongarra; Supercomputer Education and Research Centre Indian Institute of Science, Computer Science Department, University of Tennessee Knoxville, Computer Science and Mathematics Oak Ridge National Laboratory. It talks about how the Grid Computing has a methodology for dynamically balance work-load in Grid Environment. It's an advanced topic for a large Grid Computing Application. They found few self-adaptive software systems to do a comparison and deeply analyses them, all systems that dynamically adapt to changes in load characteristics, resources, computational Grids. Computational Grids include the dynamics of large stocks, so the opportunity to migrate executing programs in the different resources assumes great importance. Specifically, the main reasons of migration programs and grid systems to provide fault tolerance and to adapt to changes in system load. In this paper, we focus on executing the migration of applications and the Grid systems in order to adapt to the dynamics of the load of resources. Two disadvantages found in these systems, First, the individual policies of those working in the migration system of suspension and migration of applications to carry out programs for different systems, applications, may experience a long waiting time between when they are suspended, if they are new on the new system. Second, due to the use of the predefined conditions for suspension and migration and due to lack of knowledge about the remaining execution time of programs, applications may be suspended and moved, even if they intend to complete in a short period of time. This, of course, less desirable results of the network-oriented systems, where a large load dynamics can lead to the frequent satisfaction of predefined conditions and therefore may lead to the frequent invocations of suspension and migration decisions.

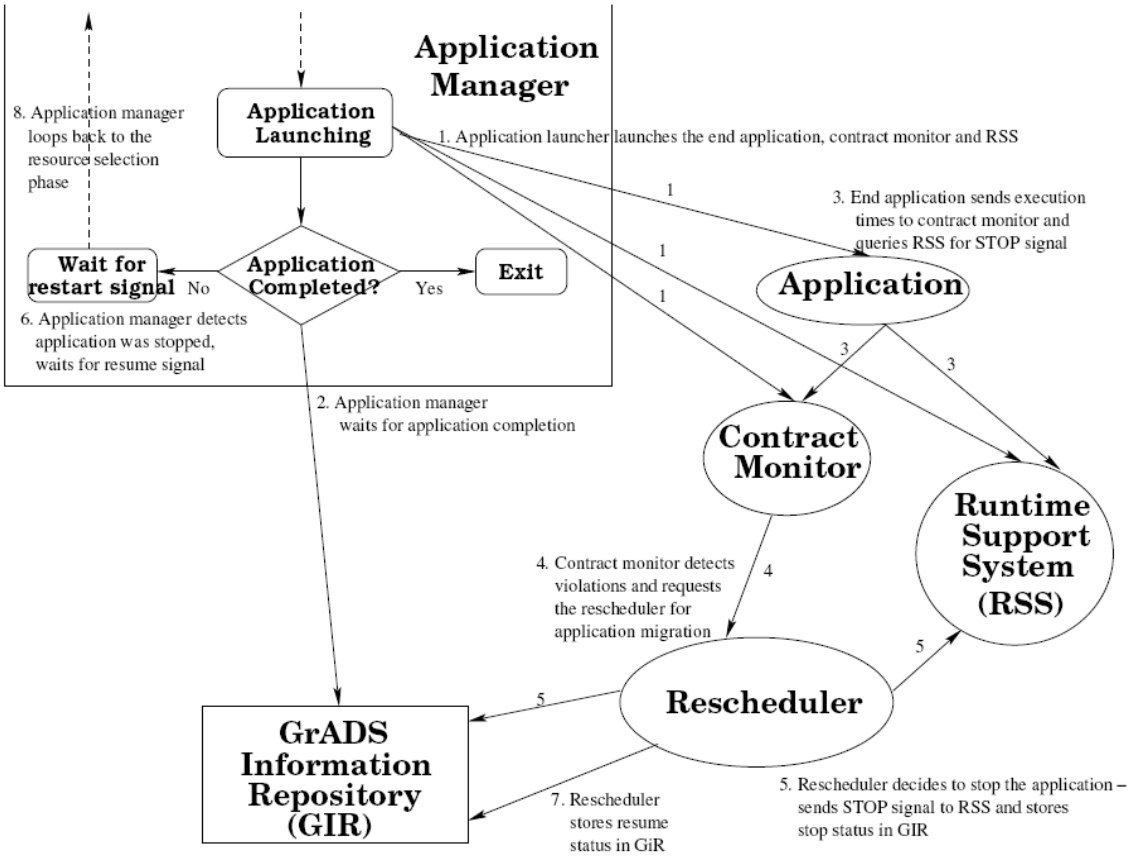
GrADS architecture

They introduce GrADS architecture. GrADS (Grid Application Development Software) is an ongoing research project involving many institutions and its aim is to simplify distributed heterogeneous computing in the same way that World Wide Web Simplified the exchange of information over the Internet. University of Tennessee examines issues related to the integration of libraries in the GrADS system. In his previous work, they have demonstrated ease with which the number of libraries as ScaLAPACK can be integrated into the Grid system and the ease with which the library can be used over the Grid. They also showed that some results demonstrate the benefits of Grid solution of a large number of problems. In the architecture of GrADS, a user wanting to solve through

the application of grid based on the GrADS manager. The life cycle and the manager were shown GrADS:



GrADS application manager



Interaction of Migration Framework

Many of the migration of existing systems, migrating applications are to the resources under the loading conditions of simple policies that cannot be applied to Grid systems. They implement the migration system, which takes into account both system load and application characteristics. Migrant decisions based on factors including the amount of resources, load, point, application of life, when the load is introduced, and from applications. They also implemented the system, that is opportunistically migrating executive applications to use the additional free resources. The experiments were performed and the results were presented to demonstrate the possibilities of migration system.

They aim to provide more reliable system and the SRS system, and provide cost effective Reschedule redistribution of data. In addition, instead of fix reschedule threshold is 30% of their future work will participate in the determination of the term limits dynamically based on the observation of dynamic load behavior of the system resources.

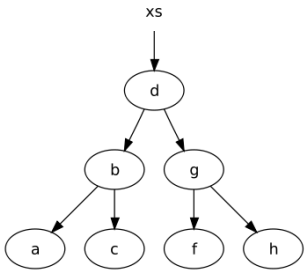
They offer their approach to examine the usefulness of complex applications involving multiple components and / or written in multi-programming languages similar to the

efforts of Mayes. Now, the average efficiency ratio is used when the track will be contacted reschedule migration. And in the future, their plan to investigate a more restrictive policy on contact with reschedule. Mechanisms to quantify the defects discovered in the implementation of the model to monitor and transmit information, the application developer must also be investigated.

Methodology

Binary Tree Object Model (BTOM)

Through the BTOM, system can analyze all the level of web structure regularly. It also provide easy node finding method to the program for seeking target web element.



Multithreading

System will apply **Amdahl's Law:** "…the performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used," (Hennessey and Patterson, 29) for increase each process performance.

$$\text{Speedup} = \frac{1}{\frac{\text{Fraction}_{\text{parallel}}}{\text{\# processors}} + (1-\text{Fraction}_{\text{parallel}})}$$

Regular Expression (RE)

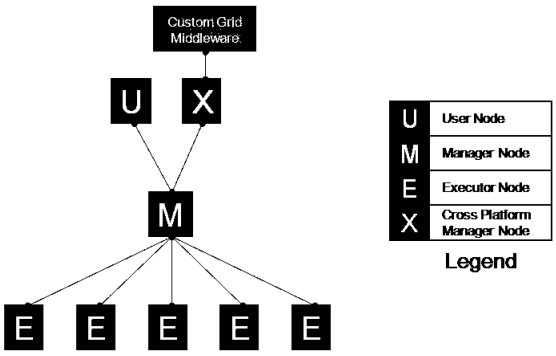
RE can filter the target text (string) rapidly. It is a standard method to examine string and identifies parts that match the provided specification. I will use this formula to capture all links from the web.

```
(href|src)(=|
= )[\\\"\\\'](http:\\/\\/|\\.\\/|\\/)?\\w+(\\.\\
w+)*(\\/\\w+(\\.\\w+)?)*(\\/|\\?\\w*=\\w
*(&\\w*=\\w*)*)?[\\\"\\\']
```

Grid Computing

Grid Computing allows the system split the job and distribute to other node servers for increasing the speed of analysis calculation.

Grid Computing has been applied to different scientific problems through loosely-coupled computers, and it is used in commercial enterprises for data analysis and processing in back-end.



Resource Required

- Hardware
  - Existing Web Servers
- Software
  - Microsoft Windows / Linux
  - Microsoft .NET Framework 2.0 / Mono 2.0
  - Alchemi - .NET based Enterprise Grid Computing Framework
  - Microsoft Visual Studio 2008 Express Edition
- Programming Language
  - ISO/IEC 23270 C#

Project Plan with Gantt chart

Mission Item	Nov	Dec	Jan	Feb	Mar
Concept Understanding					
Methodology Study					
Comparing other Competitive Products					
System Analysis					
System Design					
System Coding					
System Testing					
Review					
Debug					
Writing Report					
Hand In					

References

[1] JIM WALDO, GEOFF WYANT, ANN WOLLRATH, SAM KENDALL. *"A NOTE ON DISTRIBUTED COMPUTING"*, SUN MICROSYSTEMS. (2004)

[2] AKSHAY LUTHER, RAJKUMAR BUYYA, RAJIV RANJAN, AND SRIKUMAR VENUGOPAL. *"ALCHEMI: A .NET-BASED ENTERPRISE GRID COMPUTING SYSTEM."* THE UNIVERSITY OF MELBOURNE, AUSTRALIA

[3] JAMES CAVERLEE AND LING LIU. *"QA-PAGELET: DATA PREPARATION TECHNIQUES FOR LARGE-SCALE DATA ANALYSIS OF THE DEEP WEB"*. GEORGIA INSTITUTE OF TECHNOLOGY. IEEE. 10.1109/TKDE.2005.151. VOLUME 17, ISSUE 9, PAGE(S): 1247 – 1262 (SEPT. 2005)

[4] PAUL STRONG. *"ENTERPRISE GRID COMPUTING"*. SUN MICROSYSTEMS. ACM. ISSN:1542-7730. VOLUME 3 , ISSUE 6 (JULY/AUGUST 2005)

[5] SURRIDGE, M., TAYLOR, S., DE ROURE, D., ZALUSKA, E. EXPERIENCES WITH *"GRIA – INDUSTRIAL APPLICATIONS ON A WEB SERVICES GRID"*. IEEE. 10.1109/E-SCIENCE.2005.38. VOLUME , ISSUE , PAGE(S):98 - 105 (JULY 2005)

[6] SATHISH S. VADHIYAR, JACK J. DONGARRA. *"SELF ADAPTIVITY IN GRID COMPUTING"*. ACM. ISSN:1532-0626. VOLUME 17 , ISSUE 2-4 (FEBRUARY 2005)

[7] LARRY SMARR, CHARLES E. CATLETT. *"METACOMPUTING"*. ACM. ISSN:0001-0782. VOLUME 35 , ISSUE 6 (JUNE 1992)

[8] BRAD ADELBERG. *"A TOOL FOR SEMI-AUTOMATICALLY EXTRACTING STRUCTURED AND SEMISTRUCTURED DATA FROM TEXT DOCUMENTS"*. ACM. ISSN:0163-5808. VOLUME 27 , ISSUE 2 (JUNE 1998)

[9] ARVIND ARASU, HECTOR GARCIA-MOLINA. "*EXTRACTING STRUCTURED DATA FROM WEB PAGES"*. STANFORD UNIVERSITY. ACM. ISBN:1-58113-634-X . (2003)

[10] RICARDO A. BAEZA-YATES, BERTHIER RIBEIRO-NETO. *"MODERN INFORMATION RETRIEVAL"*. ACM. ISBN:020139829X. (1999)

[11] DOUG BEEFERMAN, ADAM BERGER. *"AGGLOMERATIVE CLUSTERING OF A SEARCH ENGINE QUERY LOG. IN KNOWLEDGE DISCOVERY AND DATA MINING"*. ACM. ISBN:1-58113-233-6. (2000)

[12] WILLIAM W. COHEN. *"RECOGNIZING STRUCTURE IN WEB PAGES USING SIMILARITY QUERIES"*. AAAI-99. (1999)

[13] JON M. KLEINBERG. *"AUTHORITATIVE SOURCES IN A HYPERLINKED ENVIRONMENT"*. ACM. VOLUME 46, ISSUE 5. ISSN:0004-5411 (SEPTEMBER 1999)

[14] RAGHAVAN, S. RAJAGOPALAN, R. KUMAR, P AND A. TOMKINS. *"TRAWLING THE WEB FOR EMERGING CYBER-COMMUNITIES"*. IN WWW '99. (1999)

[15] LIU, L.; PU, C.; HAN, W. *"XWRAP: AN XML-ENABLED WRAPPER CONSTRUCTION SYSTEM FOR WEBINFORMATION SOURCES"*. IEEE. 10.1109/ICDE.2000.839475. VOLUME , ISSUE , 2000 PAGE(S):611 - 621 (2000)

[16] A. NIERMAN AND H. V. JAGADISH. *"EVALUATING STRUCTURAL SIMILARITY IN XML DOCUMENTS"*. IN PROC. OF THE 5TH INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES (WEBDB). PAGES 61--66. MADISON. WISCONSIN, (2002).

[17] YING ZHAO, GEORGE KARYPIS. *"CRITERION FUNCTIONS FOR DOCUMENT CLUSTERING: EXPERIMENTS AND ANALYSIS"*. TECHNICAL REPORT, UNIVERSITY OF MINNESOTA. UMN CS 01-040, 2001, (2002)

[18] *"COMMON OBJECT REQUEST BROKER: ARCHITECTURE AND SPECIFICATION."*,THE OBJECT MANAGEMENT GROUP. OMG DOCUMENT NUMBER 91.12.1 (1991).

[19] BLACK, A., N. HUTCHINSON, E. JUL, H. LEVY, AND L. CARTER. *"DISTRIBUTION AND ABSTRACT TYPES IN EMERALD."*,. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING SE-13, NO. 1, (JANUARY 1987).

[20] DASGUPTA, P., R. J. LEBLANC, AND E. SPAFFORD. *"THE CLOUDS PROJECT: DESIGNING AND IMPLEMENTING A FAULT TOLERANT DISTRIBUTED OPERATING SYSTEM."*, GEORGIA INSTITUTE OF TECHNOLOGY TECHNICAL REPORT GIT-ICS-85/29.(1985).

[21] MICROSOFT CORPORATION. *"OBJECT LINKING AND EMBEDDING PROGRAMMERS REFERENCE"*, VERSION 1. MICROSOFT PRESS, 1992.

[22] COOK, ROBERT. *"MOD- A LANGUAGE FOR DISTRIBUTED PROCESSING."*, PROCEEDINGS OF THE 1ST INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (OCTOBER 1979).

[23] ANDREW D. BIRRELL, BRUCE JAY NELSON. *"IMPLEMENTING REMOTE PROCEDURE CALLS"*. ACM. ISSN:0734-2071 . VOLUME 2 , ISSUE 1 (FEBRUARY 1984)

[24] TONY MASON. *"BOOK REVIEWS: NETWORK COMPUTING ARCHITECTURE AND NETWORK COMPUTING SYSTEM REFERENCE MANUAL"*. ACM. ISSN:0146-4833 . VOLUME 20 , ISSUE 3 (JULY 1990)

[25] VASSOS HADZILACOS, SAM TOUEG. *"FAULT-TOLERANT BROADCASTS AND RELATED PROBLEMS"*. ACM. ISBN:0-201-62427-3. (1993)

[26] *"FATHER OF THE GRID"*, THE UNIVERSITY OF CHICAGO MAGAZINE: APRIL 2004, [ONLINE], AVAILABLE: HTTP://MAGAZINE.UCHICAGO.EDU/0404/FEATURES/INDEX.SHTML ACCESSED 19 DEC,2008.

[27] BELL, MICHAEL. WILEY & SONS. *"INTRODUCTION TO SERVICE-ORIENTED MODELING"*. PP. 3. ISBN 978-0-470-14111-3. (2008).

[28] *"NOVELL MONO, OPEN SOURCE .NET DEVELOPMENT FRAMEWORK."* 03 FEB 2009, [ONLINE] AVAILABLE: HTTP://WWW.MONO-PROJECT.COM/MAIN_PAGE ACCESSED 03 FEB 2009

[29] MICHAEL DI STEFANO, JOHN WILEY & SONS, *"DISTRIBUTED DATA MANAGEMENT FOR GRID COMPUTING"*, Wiley-IEEE, (2005)

[30] IAN FOSTER. *"WHAT IS THE GRID? A THREE POINT CHECKLIST*", ARGONNE NATIONAL LABORATORY, MATHEMATICS & COMPUTER SCIENCE DIVISION, (MARCH 2007)

[31] DEPARTMENT OF COMPUTER SCIENCE, BALCI, O. "*SOFTWARE ENGINEERING LECTURE NOTES"*, VIRGINIA TECH, BLACKSBURG, VA, P. 24. (1998)

[32] *"LINK CHECKER COMPARISON"*, [ONLINE] AVAILABLE: HTTP://WWW.CRYER.CO.UK/RESOURCES/LINK_CHECKERS.HTM ACCESSED 03 FEB 2009

[33] STANLEY CHOW JEFF SMITH CHRISTOPHE GUSTAVE , GALASSO & ASSOCIATES, LP, *"VERIFYING AUTHENTICITY OF WEBPAGES"*, ORIGIN: AUSTIN, TX US, IPC8 CLASS: AH04L900FI, USPC CLASS: 713156

[34] *"HTML 4.01 SPECIFICATION"*, W3C RECOMMENDATION 24 DECEMBER 1999 [ONLINE], AVAILABLE: http://www.w3.org/TR/html401 ACCESSED 05 JAN 2009

[35] *"C# ISO/IEC 23270:2006",* ISO/IEC 23270:2006 SPECIFIES THE FORM AND ESTABLISHES THE INTERPRETATION OF PROGRAMS WRITTEN IN THE C# PROGRAMMING LANGUAGE. [ONLINE], AVAILABLE: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=42926 ACCESSED 10 FEB 2009

[36] DONALD KNUTH. "*THE ART OF COMPUTER PROGRAMMING"* VOL 1. FUNDAMENTAL ALGORITHMS, THIRD EDITION. ADDISON-WESLEY. ISBN 0-201-89683-4. SECTION 2.3, ESPECIALLY SUBSECTIONS 2.3.1-2.3.2 (PP.318-348). (1997)

[37] *"REGULAR EXPRESSIONS"*, THE SINGLE UNIX SPECIFICATION, VERSION 2, THE OPEN GROUP, (1997)

[38] RAY, ZHAO ZHANG. *"AN EFFICIENT ANONYMITY PROTOCOL FOR GRID COMPUTING"*, IEEE. 10.1109/GRID.2004.9. Volume , Issue , 8 Nov. 2004 Page(s): 200 - 207 (2004)

[39] ZHIGUO SHI, YEPING HE, XIAOYONG HUAI, HONG ZHANG, *"IDENTITY ANONYMITY FOR GRID COMPUTING COORDINATION BASED ON TRUSTED COMPUTING"*, IEEE, 10.1109/GCC.2007.77. Volume , Issue , 16-18 Aug. 2007 Page(s):403 – 410. (2007)

[40] DAVIES, ANTONY, *"COMPUTATIONAL INTERMEDIATION AND THE EVOLUTION OF COMPUTATION AS A COMMODITY",* APPLIED ECONOMICS 36: 1131. DOI:10.1080/0003684042000247334, (JUNE 2004)

[41] CARL KESSELMAN. *"THE GRID: BLUEPRINT FOR A NEW COMPUTING INFRASTRUCTURE"*. MORGAN KAUFMANN PUBLISHERS, FOSTER, IAN; ISBN 1-55860-475-8. (NOVEMBER 1998)

[42] BERMAN, FRAN, ANTHONY J. G. HEY, GEOFFREY C. FOX, "*GRID COMPUTING: MAKING THE GLOBAL INFRASTRUCTURE A REALITY"*, ACM. ISBN 0-470-85319-0. (2003)

[43] LI, MAOZHEN, MARK A. BAKER. *"THE GRID: CORE TECHNOLOGIES"*, WILEY. ISBN 0-470-09417-6. (MAY 2005)

[44] *"GRID COMPUTING: A BRIEF TECHNOLOGY ANALYSIS".* CTO NETWORK LIBRARY, SMITH, ROGER, 2005.

[45] STOCKINGER, HEINZ, *"DEFINING THE GRID: A SNAPSHOT ON THE CURRENT VIEW"*, SUPERCOMPUTING 42: 3. DOI:10.1007/S11227-006-0037-9, 2007