

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2005

Utility computing: Certification model, costing model, and related architecture development

Saif Ahmed Faruqi

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Databases and Information Systems Commons](#)

Recommended Citation

Faruqi, Saif Ahmed, "Utility computing: Certification model, costing model, and related architecture development" (2005). *Theses Digitization Project*. 2756.

<https://scholarworks.lib.csusb.edu/etd-project/2756>

This Thesis is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

UTILITY COMPUTING: CERTIFICATION MODEL, COSTING
MODEL, AND RELATED ARCHITECTURE DEVELOPMENT

A Thesis
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Saif Ahmed Faruqui


June 2005

UTILITY COMPUTING: CERTIFICATION MODEL, COSTING
MODEL, AND RELATED ARCHITECTURE DEVELOPMENT

A Thesis
Presented to the
Faculty of
California State University,
San Bernardino

by
Saif Ahmed Faruqui
June 2005


Approved by:



Dr. Yasha J. Karant, Chair, Computer
Science



Dr. Keith E. Schubert



Dr. David A. Turner

8 June 2005

Date

© 2005 Saif Ahmed Faruqi

ABSTRACT

One of the many issues in the nascent field of utility computing is identification, on-the-fly, of available resources at different participating resource centers. Another fundamental issue is that of quantifying the available assets and maturity of a resource center organization with a view to compare different centers and select the centers best matching a user organizations requirements. This research addresses the issues of assessment, certification, and costing of resource capabilities at utility computing resource centers. The various technical and business elements of a utility computing resource center are identified. With each of these elements a list of related factors is identified, that contribute to the cost of the element or can be used to assess and certify the capability of that resource. The certification factors are published in a certificate that a user can use to identify a center. The costing factors identified are placed in a matrix, and mathematically manipulated to arrive at a block diagonal matrix. This matrix can then be used to arrive at a costing model. This model is flexible enough to accommodate different configurations of resource requirements by a user organization, different service levels and availability

requirements. Based on the set of resources required, the duration, service level, and configuration a price or pricing model can be arrived at for that user.

ACKNOWLEDGMENTS

I would like to acknowledge the support of my adviser, Dr. Yasha J. Karant. His guidance, understanding and patience along the course of my research and M.S. program have been invaluable. His technical inputs have made this research possible.

I would also like to thank Dr. Keith E. Schubert for his support during my research and for patiently hearing me out and guiding me whenever I needed advice.

I would like to thank Dr. David A. Turner for his guidance and technical inputs.

Thanks are also due to Dr. Arturo Concepcion, Chair of the Computer Science Department and Dr. Josephine Mendoza, Graduate Adviser in the department. Both of them helped me in administrative matters and university requirements related to my research.

The support of the National Science Foundation under award 9810708 is gratefully acknowledged.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
CHAPTER ONE: BACKGROUND	
1.1 Introduction	1
1.2 Purpose of the Thesis	1
1.3 Context of the Problem	4
1.4 Significance of the Thesis	6
1.5 Assumptions	7
1.6 Limitations	9
1.7 Organization of the Thesis	9
CHAPTER TWO: LITERATURE REVIEW	
2.1 Introduction	11
2.2 Anatomy of the Grid	11
2.3 Hewlett Packard Pricing Model: The Computon	11
2.4 Open Grid Services Architecture	12
2.5 Summary	13
CHAPTER THREE: METHODOLOGY	
3.1 Introduction	14
3.2 Identification of Factors	14
3.3 Costing and Optimization Equation	14
3.4 Utility Computing Architecture	15
3.5 Summary	15

CHAPTER FOUR: RESULTS

4.1 Introduction	17
4.2 Utility Computing Certification	17
4.2.1 Physical Assets	18
4.2.2 Human Assets	23
4.2.3 Software and Licenses	25
4.2.4 Security	26
4.2.5 Overall Authority to Certify 1, 2, 3, 4, and 5.	28
4.2.6 Publication of Certificate	29
4.3 Resource Optimization and Costing Method	29
4.4 Abstract Architecture for Utility Computing Marketplace	35
4.5 Summary	37

CHAPTER FIVE: VALIDATION

5.1 Introduction	39
5.2 Validation of Costing Model	39
5.3 Summary	41

REFERENCES	42
------------------	----

LIST OF FIGURES

Figure 1. A Model for a Utility Computing Network: Including Utility Computing Centers, Local Registries, Regional Registries, and User Applications	35
---	----

CHAPTER ONE

BACKGROUND

1.1 Introduction

The contents of Chapter One present an overview of the thesis. The purpose of the thesis is discussed followed by the context of the problem, significance of the thesis, and assumptions. Next, the limitations that apply to the thesis are reviewed. Finally, definitions of terms are presented.

1.2 Purpose of the Thesis

The purpose of the thesis was to propose one set of solutions to some of the challenges that are delaying the adoption of utility computing on a wider scale. I develop, as part of this research, a set of three components needed for effective deployment of the utility computing infrastructure. These components enable efficient look-up, and comparison of service offerings of different utility computing resource centers connected to the utility computing network. A certificate is developed that contains a comprehensive set of attributes associated with a resource center, which accurately describes the center's technical and managerial capability. Another component of the research combines a given set of resource

requirements, mathematically with weights assigned to each resource, to arrive at a concise equation. The process is flexible; it allows different combination of weights to be applied to arrive at equations tailored to specific combinations and prices of resources. This equation can be used for resource use optimization and cost estimation.

The first utility computing infrastructure component developed as part of this thesis is the capability certificate. This certificate contains a set of basic resources, and factors, identified after a thorough literature survey as essential in measuring the capability of a particular resource center. The elements of the certificate are exhaustive enough to allow meaningful comparison of resources at different centers. Besides resources, it contains metrics for measuring management maturity, technical staff training, security measures deployed, etc. This component provides the means of auditing utility computing resource centers, based on the resources and factors that are part of the certificate. It also enables manual or electronic comparison, in real-time, of different resource centers, with a view to selecting one of them for use.

The second component developed in this research is a method for combining resource requirements of a user to

arrive at an equation of total resources needed. This equation will include weighted measures of each component needed. The purpose of this equation is to provide a flexible and easily configurable method for predicting the cost of using a set of resources. This equation is needed by a resource provider to quote a price for using resources to the resource user, and this equation is also required by the resource user to optimize resource use with a view to minimizing the cost to him. In an open-market for computing resources, similar to a commodities market, different vendors can have different prices attached to each of the resources required by a buyer. In this situation various sellers can list their resource offerings and quote their price, for the buyer's required set of resources, using this equation. It is like when a company puts out a tender for a set of supplies. A group of competing vendors of these supplies, in the market, quote their price for the supplies. This price is based on the price of each individual supply, which is determined by market forces. The only difference in the utility computing model is that a set of computing resource in combination with another may be worth more than its individual cost. For example, for a bandwidth heavy software application, a combination of high

bandwidth and high processing power is more valuable than just high processing power.

The third component of this research is a proposed abstract architecture that makes use of the first two components of the research to implement a utility computing market. This architecture will enable a utility computing marketplace connecting grid based utility computing centers of multiple vendors. Future work needs to address the implementation of this architecture in a test environment.

1.3 Context of the Problem

The context of the problem was to address the need for a standard method of quantifying and comparing resource offerings of utility computing resource centers. Many companies including Sun Microsystems, HP, and IBM are developing and selling new technology aimed at the utility computing market. Utility computing refers to the concepts, technologies, and architectures developed to convert computing power into a utility, just like electricity or water. Farms of computing resources will be interconnected, for e.g. using grid middleware, and deployed on the Internet as a computing utility. Resource buyers will be able to search for, acquire, and use these

resources as and when they need. And they will only pay for what they use. This powerful concept will eventually free many types of IT users from the need to buy, deploy and maintain dedicated IT infrastructures. Just like very few electricity users actually maintain captive power plants.

But the true potential of this model of computing will only be realized when utility computing resources will be sold and bought in a competitive open environment. One model being proposed is that of a commodity trading market, where computing resources will be sold and bought as a commodity. In this setting price of the resources will be set by the market. Buyers will be able to bid for the resource they need, for the lowest price.

Already Sun and Archipelago Holdings have planned a pilot electronic exchange where users can sell and buy computing power. Sun has developed a unit for this power - the CPU-hour, or the amount of work a processor can do in one hour. For e.g., if a computing task is distributed over many processors running in parallel, the total amount of processing time is measured in CPU-hours and the customer is charged according to the cost per CPU-hour.

Similarly the storage space required by a customer over a period of time can be modeled using a modeling

application. The corresponding storage resources can be bought from a utility computing resource center.

1.4 Significance of the Thesis

The significance of the thesis was because even though various companies have developed their individual utility computing technologies and related processes, they have not generalized them for variable resource configurations. Particularly, there is no method of publishing resource capabilities of a center to allow search and comparison by a resource buyer. Besides, there is no standard method for optimizing resource requirements and cost, when choosing to buy resources from amongst a set of competing resource centers.

To realize the dream of computing as a utility, there is a need to develop standards. First, there need to be standards to certify resources and capabilities at resource centers by authorized auditing authorities. These certificates will assure a buyer that the center does indeed have the resources and capability that it claims it has. It will also allow comparison of different resource centers.

Secondly, there is the need for a method to combine a given set of resources required, mathematically, to arrive

at an equation for requirement optimization and cost estimation. For example, if a buyer was to use a modeling software (not part of present research) to arrive at an estimated set of resources for a specific performance requirement, he could then use the developed method and equation to predict total cost of all resources over the entire period of use. He could then adjust the set of resources to see how the total cost changes, in the end choosing the optimum combination of cost and performance.

Finally, I propose an abstract architecture to deploy these and other related utility computing components (not a part of this research) into an implement able form. But, I do not implement this architecture; I only propose it as a model for a detailed implementation in the future.

1.5 Assumptions

The following assumptions were made regarding the thesis:

1. The utility computing resource centers, that are part of the network, deploy their resources interconnected with a grid middleware. Example of popular industrial strength grid middleware is the Globus Toolkit, from USC ISI, ANL.

2. The utility computing resource centers have an accurate mechanism in place to measure available resources, at any point in time. This mechanism should update the list of available resources dynamically. This mechanism is built into the Globus Toolkit.
3. The resource requirements of the resource buyer can all be satisfied at one resource center (except in the case of data stored in a location different from the center).
4. The resource buyer is willing to hire computing resources not located on his premises to carry out his computing task.
5. The resource buyer has a method or tool to model the resources he requires during the entire life cycle of his computing task. This model or toll should take into consideration service quality levels and model the resource requirement accordingly.
6. The resource center operator has information about the unit cost of each factor that we take into consideration to arrive at final cost to buyer

1.6 Limitations

During the development of the project, a few of limitations were noted. These limitations are presented here.

First, dynamic resource center selection, by an agent, is only possible if the list of available resources at the center is updated dynamically and regularly. Another limitation is that the abstract architecture developed to create a utility computing network only works for a group of grid middleware based resource centers. Though, it can be extended for other kinds of middleware.

Another limitation of this research is due to its dependence on external methods and tools to model resource requirements for a user. This limits the resource optimization and cost reduction to the efficiency of the method or tool used.

1.7 Organization of the Thesis

The thesis was divided into five chapters. Chapter One provides an introduction to the context of the problem, purpose of the thesis, significance of the thesis, limitations, and definitions of terms. Chapter Two consists of a review of relevant literature. Chapter Three documents the Methodology used in this thesis. Chapter

Four presents the results from the thesis. Chapter Five presents the validation from the thesis. The Appendices for the Thesis follows Chapter Five. Finally, the references for the Thesis are presented.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Chapter Two consists of a discussion of the relevant literature. Specifically, I discuss the various materials available on the pricing models of the different utility computing vendors. I also discuss literature dealing with grid and utility computing, specifically from the Globus project [4].

2.2 Anatomy of the Grid

This paper starts with a discussion of the "Grid problem". The Grid is defined, and related problems arising out of the unique Grid architecture are discussed. The paper further discusses the technologies developed to solve these Grid related problems. It goes on to develop a Grid architecture, including protocols, services, APIs, and SDKs. The paper also describes requirements that any Grid system must satisfy.

2.3 Hewlett Packard Pricing Model: The Computon

This article discusses HP's research to develop a new pricing model for its outsourced capacity-on-demand computing services. Under HP's scheme, prices would vary based on factors such as the overall demand placed on

servers, storage devices and other IT resources. A new unit-of-computing metric, called a "computon", similar to pricing models that utilities use to charge customers for kilowatt-hours of electricity, is being developed. The article goes on to discuss how potential customers have mixed feelings about this new pricing model. Some of them feel that the new model is too complicated, and that what is really needed is an easy way to buy computing power, in small inexpensive increments. Some wonder if the model will allow buyers to measure their usage accurately or if it will just be a way of hiding the cost behind a complicated model. But some analysts see it as a positive evolutionary step in the development of a utility-based computing.

2.4 Open Grid Services Architecture

This article discusses the Open Grid Services Architecture (OGSA) model developed as part of the Globus project. The OGSA enables the integration of services and resources across distributed, heterogeneous, dynamic virtual organizations. The article describes how the Globus toolkit can be used to deploy grid services based on the OGSA. It describes the features of the OGSA in detail and how it fits into the requirements of the

organization trying to implement a grid infrastructure, within the organization or across organizations. In the end it describes the architecture of a virtual organization (utility computing center) which uses the Globus toolkit and OGSA.

2.5 Summary

The literature important to the thesis was presented in Chapter Two. The literature discussed included that describing a pricing model for utility-based computing being developed by HP. It also discussed two papers from the Globus project describing the computing Grid and the OGSA grid services architecture. This literature was primarily used as reference for this research.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

Chapter Three documents the Methodology used in this thesis. Specifically, the steps used to arrive at the certification factors, and the costing and optimization model. It also discusses the steps used to arrive at the abstract architecture for a grid computing network.

3.2 Identification of Factors

Based on literature and experience we first identified the various factors involved in the operation of a utility computing resources center. These factors included resources that a center user buys directly from the center, and the hidden cost factors that contribute to the cost of these resources.

3.3 Costing and Optimization Equation

Once we identified these factors, we looked at methods of combining them in different ways to arrive at a costing and optimization equation. We found that the best method was when we grouped factors together based on how closely they are related to each other. These groups of factors gave us the starting point for development of the costing model. We also discovered that factors that may be

specially required by a particular user can be grouped together into a new group and can be priced separately. After reviewing various methods for combining these groups we arrived at the block diagonal matrix method, since it provides us the flexibility to combine the different factors in different ways based on the requirement.

3.4 Utility Computing Architecture

The architecture was developed keeping in mind the requirements of the utility computing stake holders. It aims to develop an open market for utility computing services. It accomplishes that by providing means to the resource buyers to search for the best price/performance match for their requirements, available on the utility computing network. It also provides a mechanism for negotiation to try and match offers to requirements. The architecture uses the OGSA architecture [5] as a basic building block.

3.5 Summary

This chapter discusses the methodology used to arrive at the various deliverables of this research. It discusses the method used to develop the certificate for utility computing resource centers. It also describes the steps followed in the development of the costing and

optimization model. Finally, it describes the principles and steps used to arrive at the utility computing network architecture.

CHAPTER FOUR

RESULTS

4.1 Introduction

Included in Chapter Four is a presentation of the results of the thesis. The chapter starts with discussion of the factors identified to form part of the certificate to be awarded to utility computing resource centers after they are audited. This is followed with a discussion of the method developed to arrive at a mathematical equation for resource requirement optimization and cost optimization. Finally, the abstract architecture model which implements a marketplace for utility computing is presented.

4.2 Utility Computing Certification

This thesis proposes a model of certification, to certify utility computing resource centers -- outsourced and in-house utility computing resources available on demand. I propose a set of categories, and specific factors within each category, on which utility computing resource centers can be assessed. I also give specific examples under each category, but, the choice of which factors and categories to use in the final certificate rests with the standardization body for this certificate.

Another piece that will fit with my research to complete the picture is the prediction for resources required by a set of tasks submitted for simultaneous execution to a resource center, based on performance requirements. This will require the separate development of metrics for the same. The pricing for resources used by a specific job will be determined by "market" forces.

4.2.1 Physical Assets

The computing resource capacity and performance of a particular resource center will be quantified, for all the following resources. This includes listing the different kinds of resources available along with their numbers and specifications and performance predictions at a particular point in time, resource wise. This dimension is relevant for both long term and on-the-fly decision making, for resource center users. For a long term resource requirement the buyer will check to see that the center has the hardware assets his application will need throughout its lifetime. In the case of resource acquisition at short notice, the buyer will use this dimension of the certificate to decide for or against submitting a task to that center. For example, if a academic research lab or corporate department suddenly needs excess computing capacity, over its existing utility

the architecture they were tested on, and the performance figures of each package/architecture combination. For example the certificate for a specific center may include the performance numbers for the different architectures it has deployed for use, say a cluster of Sun Fire dual core processors running the Sun Sparc OS, and the packages run most often on that cluster, say the parallel BLAST bioinformatics package, and the HMMR bioinformatics package.

4.2.1.2 Database Transactions per Second. For a job that needs to access a database to complete execution, the job assigner will need a measure of the maximum number of database transactions possible for use by his application for each available kind of database. For this we propose to include a list of available databases and the maximum number of transactions possible for each database per application, in our metric. After a decision to use a resource center has been taken, the user can store his data in the databases there. Subsequently, whenever a particular job tries to execute, it will be guaranteed a certain number of transactions, on the database of interest. The available number of transactions will change dynamically as jobs are added and removed from the current list of the resource center. But, the data center operator

will have to guarantee that each job is allowed its maximum number of transactions throughout its life cycle. The entry under this head in the certificate will include the information on the types and versions of databases available at the resource center, and the maximum number of transaction that a new user can expect to be available for his application.

4.2.1.3 Data Communications Throughput. This measure of the available resources of a resource center depends on network topology and congestion in the network at a resource center, and between the resource center and the job assigner. It is a dynamically changing quantity. Bandwidth requirements of all the presently executing jobs at a resource center will be predicted to calculate availability of bandwidth for an incoming resource. These predicted values will be modeled over time to check for congestion in the network on the resource center to decide if the incoming job can execute giving required performance. History of bandwidth availability will be maintained to predict bandwidth available between the allocation location and the resource center. This measure of the resource center also needs to provide a guarantee of minimum bandwidth. Once a user has been sold a service giving him this guarantee, it the responsibility of the

center operator to provide the minimum promised bandwidth over the life cycle of the user's application.

4.2.1.4. Mass Storage Capacity and Mass Storage Rate to Transactions per Second Engine. Total available secondary storage at a resource center will be a measure used to allocate jobs. The storage available could also include storage accessible over a WAN to the resource center. In the latter case available bandwidth would play a role in deciding which center to allocate a job to. A utility computing customer will model his resource needs for the entire life of the engagement with the utility computing center. One of the resources he will take into consideration is the available storage at the center. If he wants to use data stored at a location outside the utility computing center he is considering for his computing needs, the buyer will also need information about the mass storage rate to the TPS engine. The overall performance of his application in this situation will depend on the rate at which data can be transferred back and forth between the remote storage location and the center. Metrics published for this factor will be the total available storage, the storage per transaction, and the storage per kilobit/sec.

4.2.1.5 Specific Architecture (Processor, etc.). Some jobs may require the availability of specific hardware architecture to execute. The required architecture could include the processor type, bus speeds, memory latency, type and bandwidth, and cache latency, type and bandwidth. This information will form another one of the parameters to decide on a resource center. All available hardware configurations will be published.

4.2.2 Human Assets

This factor is useful for overall decision making, for an organization to decide whether to include a resource center in its set of centers. Important attributes of the human assets are maturity, experience and quantifiable capability. A critical skill required in these human assets is the ability to forecast resource requirements for multiple tasks, running simultaneously, for individual performance requirements. An important measure of the human resource of a utility computing organization will be the SEI P-CMM certification [3]. If the organization does have the certification then the level of P-CMM it has achieved will be the metric used for human resource measurement. In the absence of a P-CMM certificate, the following categories and corresponding

factors will be used to measure the expertise of the human assets of the resource center.

4.2.2.1 Management. Under this head the expertise of the utility computing resource center management in running a mission critical center will be measured. This will serve as a measure of the confidence in management a potential client can have. The attributes used will be educational qualifications, relevant work experience, training, leadership experience, and technical skills. A formula to assign an overall rating to the management of a resource center, on a scale will be used. This will consolidate the values assigned to each of the attributes above into a single representative number of the quality of the center management. The scale will have expertise level from inexperienced to expert, with various intermediate levels.

4.2.2.2 Systems Administration. The most important human element of a resource center is the systems administrator. A method for assessing the capabilities of these professionals will assign an overall rating to this factor. It will take into account relevant certifications, experience, past record and education. A standardized questionnaire will be given to each of the system administrators in the organization to assess them followed

by group interviews and personal interviews. A value will be assigned, to the center, which will indicate the mix of administrators of different expertise levels.

4.2.2.3 Development Programmers. The development programmers will also be assessed in a manner similar to that used for the systems administrators. They will be assessed on experience, past record, education, and relevant certifications. A value to indicate the mix of programmers of different expertise levels will be assigned.

4.2.3 Software and Licenses

This dimension measures the software capability of the resource center. It also addresses legal compliance issues, both national and international. It is a factor in both overall and on-the-fly decision making. This is because license possession may be an issue for on-the-fly decisions. For example, if a given user of the resource center needs to use more instances of a software package than he has done before, he will need information on licensing before taking a decision.

4.2.3.1 OS Vender (HP-UX, AIX, MS Windows Server, etc.). The OS used should allow multitasking and parallel execution. At any given time multiple applications could be running on one instance of the OS. Since the user will

decide on a given center based on performance guarantees, the OS should assure a minimum level of performance on different hardware configurations. An important consideration is the license the OS vendor gives for its use in a utility computing environment. It should allow for flexibility in the number of users and the duration of use of the OS. The center operator should only be charged for the number of users and total duration of use of the OS. Therefore, the license should be flexible and allow for variability of users and the time it is used.

4.2.3.2 ISV Vendor (e.g., Oracle, SAP, PeopleSoft, Adobe, Quark, etc.). The licensing mechanism used by software vendors will need to take into consideration varying levels of usage of their software package over a period of time. The license agreement should allow as many instances as needed by the user and charge the resource center accordingly. This requirement is the same as that for the OS.

4.2.4 Security

Security is a central issue in the utility computing model. Tasks will be submitted to resource centers after matching the security requirement of each task with the security rating of that center. This can be further refined to allocate a security rating to sub-parts of a

resource center, thus further differentiating individual sets of resources at a site. Security will also be an important factor in billing rates. The security rating of a resource center will be decided by grading it on a predefined set of parameters. The following parameters will be used for our initial set:

1. *Completion of a thorough Security Policy*
2. *Implementation of a complete Incident Management procedure*
3. *Completion of a Risk Assessment report*
4. *Completion of a Threat / Vulnerability Analysis*
5. *Development of an audited Security Architecture*
6. *Appropriate deployment of Network Intrusion Detection systems*
7. *Anti Viral Software Policy & Implementation*
8. *Network Architecture and Configuration policy*
9. *Establishment & Conduction of rigorous Auditing procedures*
10. *Staff screening*
11. *Authentication mechanisms*
12. *Authorization mechanisms*
13. *Repudiation mechanisms*

4.2.4.1 Indemnification. Each task submitted to a center will need to have a monetary value attached to it. This will enable tasks to be insured against loss or theft. The specific value of a task or of data can be decided at the time of drawing up a contract with the resource center. For real-time resource acquisition, automated negotiation mechanisms will need to be devised for this purpose. An important factor here will be past experience, as empirical data will come in handy when deciding on assigning a value to tasks.

4.2.5 Overall Authority to Certify 1, 2, 3, 4, and 5.

The entity certifying a given resource center should have the authority to certify utility computing resource centers. This authority could be granted by the body administering the certification standards and processes. An analogy is the authority granted by the Software Engineering Institute (SEI) of the Carnegie Mellon University, to organizations like KPMG to audit software companies to certify them at different CMM levels. Companies that are existing auditors for CMM, and Six Sigma, could also certify utility computing centers.

4.2.6 Publication of Certificate

Once a utility computing resource center has been audited under the various categories and factors identified above, a combined certificate will be published. This certificate will include values for all the heads, some of which are static, and others dynamic. For example, storage, TPS, and bandwidth available at a center will change over the time, whereas a factor like the level of expertise of management would be constant for a reasonable duration of time. For this reason when selecting a resource center on a real-time basis, a search agent will need to look at the variable factors to decide which of them, or which subset of them, best suits the requirements of the resource buyer. The agent will then list the top few best fitting centers for a human decision.

4.3 Resource Optimization and Costing Method

A method for costing of services at the utility computing resource center has been developed. This method can be used by the utility computing resource buyer to model his specific requirement of resources, including the relative weight of each resource within a group of related resources, and relative weights of the different resource

groups. Individual resource weights will be determined by the minimum needed quantity or quality of that resource, the duration that resource will be used for, the need to upgrade that resource during the life cycle of the contract, and the cost of that resource. For example, if a resource buyer requires higher guarantee of service for the number of database transactions and the quality of database administration, this group of resources will be assigned a higher weight. Taking another example, if the buyer requires a higher level of security, the security related group of resources will be weighed higher. By varying the different weights, according to his quality of service requirements, of factors within resource groups, and groups within the set of groups, the buyer can optimize his resource requirement. Similarly, a resource center operator can use the quality of service required, and minimum quantity of dedicated resources required to assign weights to each resource and resource group. He can then use the method described to arrive at the cost to him of these resources. He can also use market prices, with markup, of each resource to arrive at the price he will charge the buyer.

The costing factors can be divided into groups of factors that are closely related to each other, with each

factor assigned a code. The following groups have been identified:

1. Database transactions:
 - a) Cost/Transaction/Database (c00); b) Database administration (c01); c) Database software (c02).
2. Data communication: a) Bandwidth (c10);
 - b) Redundancy (c11); c) Network administration (c12); d) Network software (c13); e) Security software (c14).
3. Mass storage: a) Security software (c20);
 - b) Storage (c21); c) Storage software (c22); d) Human resources (c23); e) Redundancy (c24).
4. Common costs: a) Common human assets (c30);
 - b) Common management and overhead (c31); c) Common security (c32); d) Common indemnification (c33).
5. Hardware architecture: a) Unit cost of each configuration of similar processors (c40);
 - b) Power requirements (c41).
6. OS and software: a) Licensing (c50); b) System administration (c51).
7. Indemnification: a) Indemnification cost per job (c60).

All these cost factors are considered at a fixed point in time. Each of them can be specified in the form $c(t)$, which is a particular resource cost, as a function of time. The individual resource costs are calculated as the average of the costs incurred on that resource over the period of a year, as a factor of unit time. Empirical data for the previous few years is used to arrive at the unit cost. For a new resource center, this data can be obtained from older centers. These costs will all be placed in a matrix, called the cost matrix. This matrix is a single dimensional column matrix. The costs from each group of factors are stacked on top of each other in the cost matrix. The resulting matrix is of the form:

$$C: = \begin{bmatrix} c_{i0} \end{bmatrix} \quad (i=1:22)$$

with the i representing the factors above.

Each cost factor within a group will be assigned a relative weight, w . The weight assigned to a cost factor within a group is determined by the relative importance to the resource buyer of that resource. For example, suppose a resource buyer requests that within the database group, he wants a standard database software, but a higher than normal level of database administration expertise. In this case, the resource center owner will assign a higher weight to DB administration, than he would to the database

software. A higher level of service will translate to a higher weight for that factor.

These weights will be decided by the resource center owner, based on the requirement of the buyer for that resource, as explained in the beginning of this section. Besides the relative weights of each resource within a group, each of the resource groups will be assigned individual weights. These weights will be decided based on the quality of service and quality of product requirements for the group. The group weights will also reflect the mix of resources required. For example, if two groups of resources can be provided together conveniently, their weights will be lower, when they appear together. The weights of the individual resources, w_{ij} , will be placed in a weight matrix, W , with each group placed in an individual row. The matrix will be zero filled.

Since the range over which the weights are distributed can be different for different groups of factors, we need to normalize all the weights. The summation of all weights in each row together with the group weight assigned to that row will give us the normalization factor for that row. For example, the normalization factor for the first row of weights, N_0 , is:

$$N_0 = \sum_{i=0}^4 w_{i0} + g_0,$$

where g_0 is the weight assigned to the first group of resources, as a group weight. Each weight in the weight matrix is then divided by the normalization factor for that group of resources. The resulting values in the matrix are then all inverted. These two steps are performed to factor in the weights of each resource group into the weight of individual resources.

Next, the normalized weights for each group are stacked on top of each other, in a single dimensional column matrix. This gives the final weight matrix for cost calculation.

Finally, to arrive at the final cost, we multiply the transpose of the cost matrix with the weight matrix. The product of cost and weight matrices will give us

$\zeta = (c^T * w)$. This matrix, ζ , contains the final costs for each group of resources along its diagonal. The summation of the values of this matrix gives us the final cost to the buyer for a given set of resources.

4.4 Abstract Architecture for Utility Computing Marketplace

I propose a two level architecture for discovery, acquisition, and use of utility computing resources. This architecture builds on the utility computing center level architecture proposed in [5].

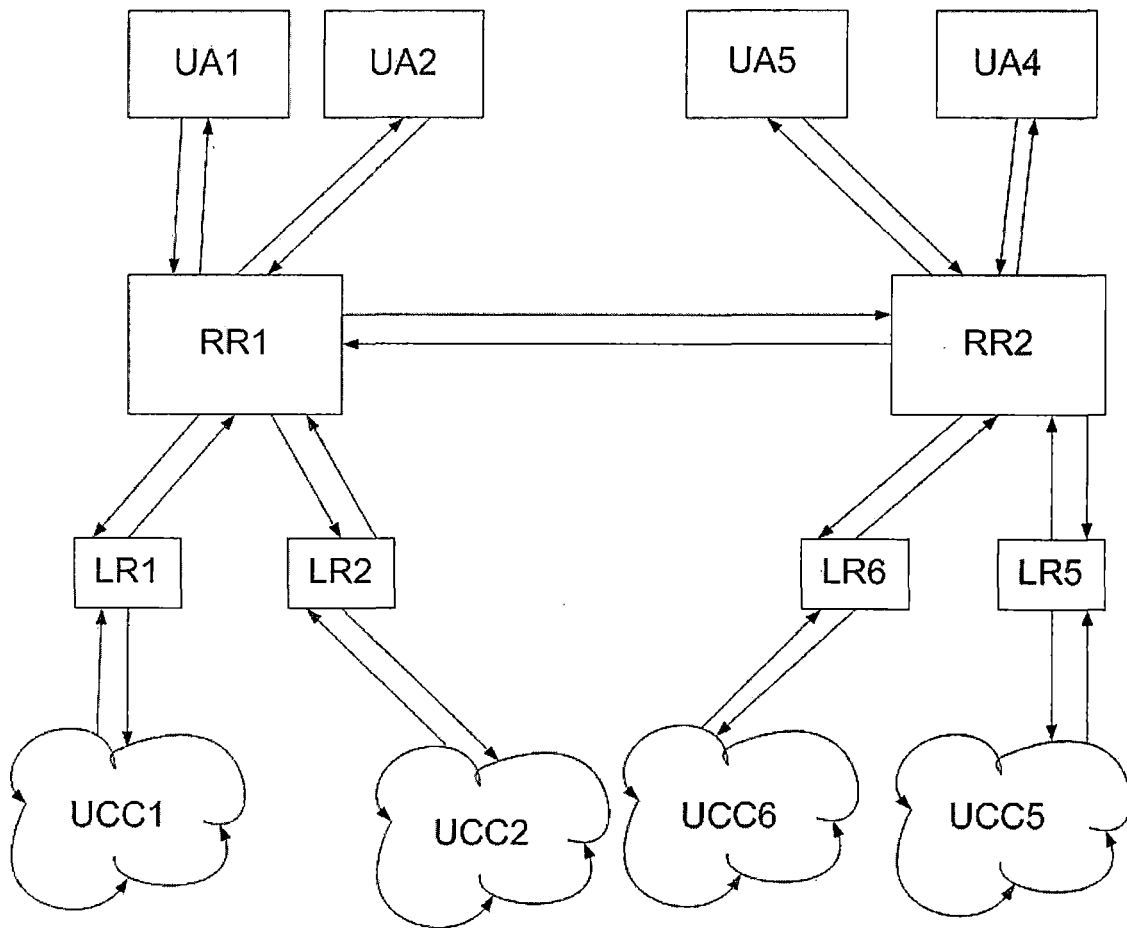


Figure 1. A Model for a Utility Computing Network:
Including Utility Computing Centers, Local Registries,
Regional Registries, and User Applications

In this model each center, called a Virtual Organization (VO), maintains a registry of available services in that center. The format and content of the registry is not specified. I propose to use the certificate I have developed as part of this research as the content of the registry at each center (Fig. 1). This registry will be updated dynamically as the status and availability of each resource changes. Each of these local registries will in turn be connected to a regional registry. The regional registry will contain all the certificates of the utility computing centers connected to it directly. The regional registry will also change as the contents of the local certificates change. All regional registries will be connected to other regional registries closest to them. In this way the interconnections of all the regional registries will form one big utility computing network. The users of the utility computing network will also be connected to this network, at the regional registry closest to them.

When a user program has a resource requirement, it will use a spider program to search the utility computing network for it. The spider program will be passed resource requirements as parameters, along with information about how widely to search (maximum number of hops), negotiation

parameters, etc. The spider will start with the closest regional registry. It will try to match the requirements with the certificates available with that registry. If it finds one or more matches, it will return that information to the user program. If not the spider will fan out to the next set of regional registries connected to the first registry. It will follow this process till it exhausts the number of hops or finds appropriate resources available at a center. If it does not find the appropriate resources available at any of the centers it searched, it will negotiate with the ones closest to its requirements. If a deal is reached with a center as a result of this negotiation, the spider program returns the local registry location for this center to the user program.

The user program next requests the specific resources from the local registry. In return the user program receives a set of handles identifying the specific services requested. For more details on local utility center request life cycle management, please refer to [].

4.5 Summary

This chapter presented the results of my research. It started with the certificate developed for utility computing resource centers. The factors relevant to

measuring capabilities of resource centers were presented grouped under different categories. Their relevance to measuring resource center capability was discussed in detail. Another result presented in this chapter was the method developed to optimize the resource requirement by a user. This method can also be used to arrive at a total cost of providing this resource set, by the center operator. He can further use it to arrive at the price he will charge the buyer, for these resources. Finally, an abstract architecture is proposed to connect the utility computing centers, local registries, regional registries, and user applications in a network. This architecture will allow dynamic search and discovery of resources, their acquisition and use.

CHAPTER FIVE

VALIDATION

5.1 Introduction

Included in Chapter Five is a presentation of the validation of the thesis. The validation is for the method developed for calculation of total cost to the resource buyer at a utility computing resource center. Lastly, the Chapter concludes with a summary of the validation.

5.2 Validation of Costing Model

Assume that a resource buyer asks for the set of twenty-two resources listed in section 4.3 above. Let us assume that the average unit cost for each of these 22 resources is given by the following matrix:

$$C = [12, 32, 9, 53, 21, 4, 92, 80, 127, 17, 31, 73, 284, 34, 17, 4, 67, 290, 15, 21, 100, 4]$$

Let us next assume that the following matrix assigns weights to each of the resources:

$$W = [10, 4, 5, 0, 0; 21, 40, 10, 15, 10; 12, 15, 11, 5, 17; 95, 15, 45, 80, 0; 5, 9, 0, 0, 0; 15, 10, 0, 0, 0; 75, 0, 0, 0, 0]$$

And the following matrix assigns weights to the individual groups:

$$G = [5, 10, 2, 7, 20, 15, 7]^T$$

We normalize the matrix, W , following the method described above. We first add all the values in a row of the W matrix with the corresponding value in the G matrix, giving us the normalization factor. Next, we divide the non-zero values in that row of W by the normalization factor. And finally, we invert all the non-zero values in the matrix. This results in the following normalized matrix:

$$W_N = [2.4, 6, 4.8, 0, 0; 5.1, 2.65, 10.6, 7.07, 10.6; 5.17, 4.13, 5.64, 12.4, 3.65; 2.55, 16.13, 5.38, 3.03, 0; 6.8, 3.78, 0, 0, 0; 2.67, 4, 0, 0, 0; 1.09, 0, 0, 0, 0]$$

After transposing these rows and stacking them on top of each other, with removal of the zeros, we get the following matrix:

$$W_F = [2.4, 6, 4.8, 5.1, 2.65, 10.6, 7.07, 10.6, 5.17, 4.13, 5.64, 12.4, 3.65, 2.55, 16.13, 5.38, 3.03, 6.8, 3.78, 2.67, 4, 1.09]^T$$

Finally, to arrive at the total unit cost to the vendor of the utility computing service, we do matrix multiplication of the cost matrix, C , with the weight matrix, W_F . The result of this multiplication is: [6241.67]. This is the final unit cost of the set of resources requested by the buyer.

5.3 Summary

In this chapter I display, using hypothetical matrices, the model used to arrive at the unit cost price of a set of utility computing resources. This example shows how the weights can be used to model the importance of each individual resource.

REFERENCES

- [1] T. Hoffman. *HP takes new pricing path for utility-based computing*. Online Article, ComputerWorld, <http://www.computerworld.com/managementtopics/management/itspending/story/0,10801,81522,00.html>
- [2] Author Unknown. *Pay-Per-Use-Pricing*. Online Article, Sun Website, <http://www.sun.com/service/utility>
- [3] S. Faruqui, E. Gomez, Y. Karant, K. Schubert. A Model for Certifying and Costing of the resource capabilities of utility computing resource centers. *Proceedings of ICCSA-2004*, June 2004.
- [4] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.
- [5] I. Foster, C. Kesselman, J. Nick, S. Tuecke. Grid Services for Distributed System Integration. *Computer*, 35(6), 2002.