

Ontology Framework for Multi-Core based Cloud Computing

*Thesis submitted in partial fulfillment of the requirements
for the award of degree of*

**Master of Engineering
in
Software Engineering**

Submitted By
**Abhishek Kaushik
(Roll No. 801031002)**

Under the supervision of:
Dr. Inderveer Chana
Associate Professor



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2012

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Ontology Framework for Multi-Core based Cloud Computing*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Inderveer Chana* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Abhishek Kaushik
(Abhishek Kaushik)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Inderveer
18/06/12
(Dr. Inderveer Chana)

Associate Professor,
Computer Science and Engineering Department

Countersigned by

MS
(Dr. Maninder Singh)
Head
18/06/12
Computer Science and Engineering Department
Thapar University
Patiala

SKM
(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings throughout my research work to complete the research successfully.

I am indebted to my supervisor Dr. Inderveer Chana for her assistance, ideas, and feedbacks during the process in doing this dissertation. Without her guidance and support, this dissertation can not be completed on time.

I am also thankful to Dr. Maninder Singh, Head, Computer Science and Engineering Department for his kind help and cooperation. I express my gratitude to all the staff members of Computer Science and Engineering Department for providing me all the facilities required for the completion of my thesis work.

I extend my thanks to Ms. Nidhi, and Ms. Pankajdeep Kaur for sharing their expertise and time to help me accomplish this work.

I would like to say thanks to all my friends especially Jaspreet, Ravinder, Ashish, Deep and Mukesh for their support.

Lastly, I wish to express my sincere gratitude to my family for their encouragement and moral support.

Cloud Computing is the most recent trend in computing today. Though it is considered being a “new” technology, but the concept was given by John MacCarthy in 1960, that states “computation may organize as Public Utility someday”. Cloud computing facilitates the users what they need, as they need it. Business applications to IT infrastructure including virtual servers and storage are all parts of cloud service model. Cloud systems cannot be just considered as another form of resource provisioning infrastructure but infact it provides multiple opportunities to host and also offers various kind of cloud applications. With the help of cloud computing the development and provisioning time of different services reduces to minimum.

Despite numerous benefits, cloud usability is hindered by Interoperability of different cloud environments across the globe. The problem of interoperability and standardization between different cloud providers is responsible for issue like “vendor-locking”. It becomes difficult for users to switch between cloud service providers. Another factor impeding cloud computing is single-core processors. Many of the cloud providers have already altered their services from single-core to multi-core processors. Though multi-core processors in cloud computing comes with its own flaws; yet one of the major flaw is resource scheduling. Resources in cloud computing are limited and hence underutilization of these resources is not acceptable in any business organization.

In this thesis, existing resource scheduling methodologies for multi-core processors have been compared. Different cloud interoperability standards have also been discussed and a framework has been designed and implemented which addresses the problems of interoperability and multi-core resource scheduling. To address interoperability, ontology has been used for representing cloud platform. The experimental results gathered through cloudsim. The results clearly demonstrate that the proposed framework has better performance as compared to the FCFS (First Come First Serve) approach.

Table of Content

Certificate	i
Acknowledgement	ii
Abstract	iii
Contents	iv
List of Figures and Tables	vi
Chapter 1 Introduction	1
1.1 Cloud Computing	1
1.2 Character of Cloud Computing	3
1.3 Challenges of Cloud Computing	4
1.4 Motivation	7
1.5 Organization of Thesis	7
Chapter 2 Literature Survey	8
2.1 Standardization and Interoperability	8
2.1.1 Open Cloud Computing Interface	8
2.1.2 Unified Cloud Interface	9
2.1.3 Open Virtualization Format	10
2.2 Ontology based work in Cloud Computing	11
2.3 Virtualization	11
2.4 Challenges of Multi-Core Processors	12
2.4.1 Cache Coherence	12
2.4.2 Multithreading	13
2.4.3 Virtual Resource Utilization	13
2.5 Resource Scheduling in Multi-Core Systems	13
2.5.1 Underlying Layout Aware Scheduling	14
2.5.2 Forecasting and Time Delay Subtraction	15
2.5.3 Friendly Virtual Machines	17
2.5.4 Task Aware Virtual Machine Scheduling	19
Chapter 3 Problem Analysis	20

3.1 Gap Analysis	20
3.2 Objectives of the proposed work	22
Chapter 4 Ontology Based Cloud Framework	23
4.1 Proposed Frame Work	23
4.2 Design of Cloud Ontology	24
4.2.1 Hierarchal View of cloud ontology	25
4.2.2 Cloud View	26
4.3 Iaas Cloud Portal	28
4.3.1 Flow Chart	28
4.3.2 Data Flow Diagram	29
4.3.3 UML Diagram	30
4.4 Tools	34
4.4.1 Protégé'	34
4.4.2 MySQL	35
4.4.3 Web Protégé'	35
4.4.4 CloudSim	36
4.4.5 Eclipse	36
Chapter 5 Experimental Results	37
5.1 Implementation	37
5.1.1 Ontology Implementation	37
5.1.2 Iaas Cloud portal	48
Chapter 6 Conclusions and Future Scope	59
6.1 Conclusions	59
6.2 Thesis Contribution	59
6.3 Future Scope	59
References	61
Research Publications	65
Appendix	66

List of Figures and Tables

Figure 1.1: NON-Exhaustive View on the main aspect forming a Cloud System	1
Figure 1.2: SaaS, PaaS and IaaS	2
Figure 1.3: Moore's Law	5
Figure 2.1: OCCI class diagram	9
Figure 2.2: Unified Cloud Interface architecture	10
Figure 2.3: Standard deviation Comparison of four mapping Scheme	14
Figure 2.4: Block Diagram of FTDS	15
Figure 2.5: Performance comparison kernel compilation with Static, AISD and FTDS scheme	16
Figure 2.6: Time taken by NPB application on different configuration	17
Figure 2.7: VCT comparison with and with FVM strategy	18
Figure 2.8: Task Aware scheduling architecture	19
Figure 4.1: Ontology Based Cloud Frame Work	24
Figure 4.2: Hierarchal view of cloud ontology	25
Figure 4.3: Cloud View	27
Figure 4.4: Ontology Classes	27
Figure 4.5 Object Properties	28
Figure 4.6: Flowchart of IaaS Cloud Portal	29
Figure 4.7: 0 Level DFD of the Cloud Portal	30

Figure 4.8: Use Case Diagram of the Cloud Portal	31
Figure 4.9: Sequence Diagram of user and server	32
Figure 4.10: Sequence Diagram of user and server registration	32
Figure 4.11: Sequence Diagram of full interaction between user and server	32
Figure 4.12: Activity diagram of User performing query	33
Figure 4.13: Activity Diagram of User Registration	34
Figure 4.14 WebProtege' Architecture	35
Figure 5.1: "SaasCloud" Class	37
Figure 5.2: "High" Class	38
Figure 5.3: Instances of the "Individual" Class	38
Figure 5.4: Different developed Database	39
Figure 5.5: Protégé' server Starting	40
Figure 5.6: Apache Tomcat	41
Figure 5.7: Web Protégé' Ontologies	41
Figure 5.8: Base Ontology Loading	42
Figure 5.9: Base Ontology in Web Protégé'	43
Figure 5.10: Protégé' rich Client	44
Figure 5.11: Protégé' 3.4.6 opened with help of JNLP using Web Protégé'	45
Figure 5.12: Saas Cloud Class Query	46
Figure 5.13: High Class Query	47
Figure 5.14: Home Page of the Multi-Core Cloud Portal	48

Figure 5.15: Multi-Core Cloud Portal Login Error Page	49
Figure 5.16: Registration Form	50
Figure 5.17: Password Filed Empty Error	51
Figure 5.18: Address Field Empty Error	51
Figure 5.19: Cloud_Portal Database	52
Figure 5.20: Login_detail table in Cloud_portal databse	52
Figure 5.21: Users detail in Cloud Portal database	53
Figure 5.22: VM's mips and Cloudlet's length page	53
Figure 5.23: VM_MIPS and Cloudlet_Length fields with data	54
Figure 5.24: Execution terminal of Cloudsim	55
Figure 5.25: Second Example	56
Figure 5.26: Second Example execution	56
Figure 5.27: Ideal Scenario	57
Figure 5.38: General scenario	57
Tables	
Table 3.1: Multi-Core Scheduling Techniques Comparison	21

Chapter 1

Introduction

This chapter introduces Cloud computing and explains the important characteristics of the cloud computing along with its challenges.

1.1 Cloud Computing

With the development of Computer and Internet, new challenges are arising day by day. One of those challenges is increase in demand of connectivity and data handling. Due to globalization of every IT infrastructure, dynamic data and access is becoming mandatory. To overcome these challenges, a very elastic infrastructure is needed that can be managed according to increase or decrease in demand.

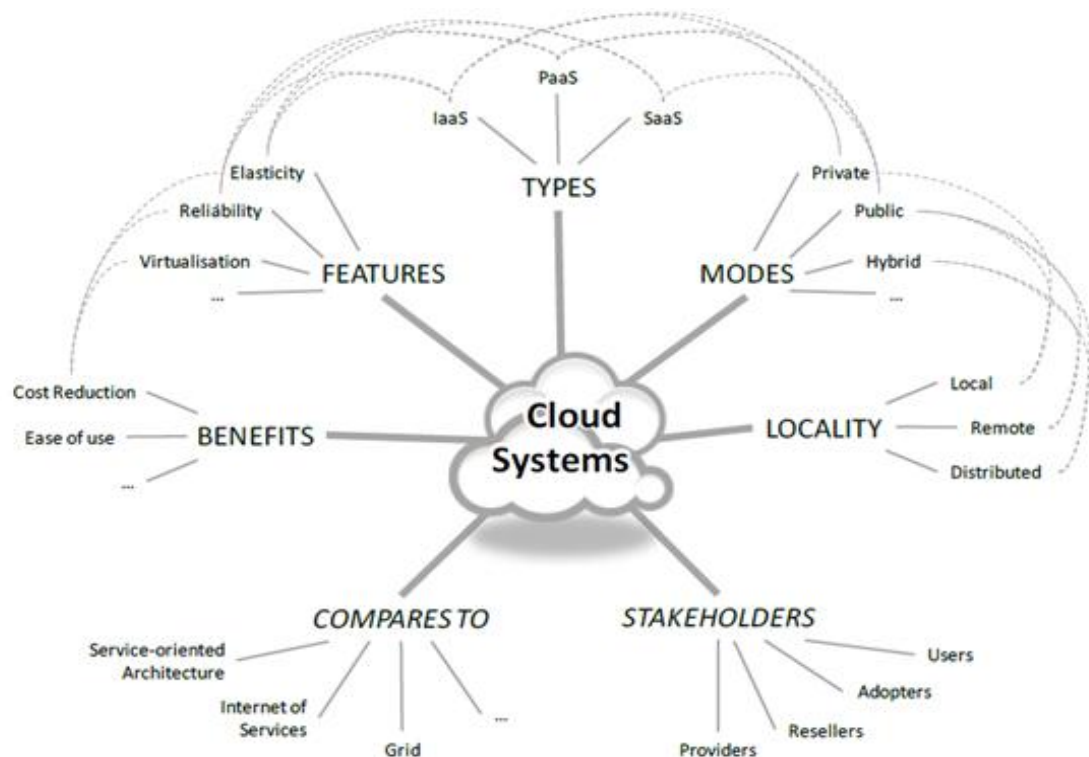


Figure 1.1: NON-Exhaustive View on the main aspect forming a Cloud System [1]

This necessity introduced the concept of Cloud. Though it has been said that "Cloud" is a new technology in Computer era but its market performance shows a totally different picture. Figure 1.1 [1] give us an abstract view of Cloud Systems.

Figure 1.1 shows how Cloud Computing is touching most of the aspects of the market and every level of consumers.

Cloud computing is a new model of consuming and delivering IT and business services. It enables users to get what they need, as they need it from advanced analytics and business applications to IT infrastructure and platform services, including virtual servers and storage. It can provide significant economies of scale and greater business agility, while accelerating the pace of innovation [2].

Cloud basically provides three types of services. These are SAAS, PAAS and IAAS (Figure 1.2) [3].

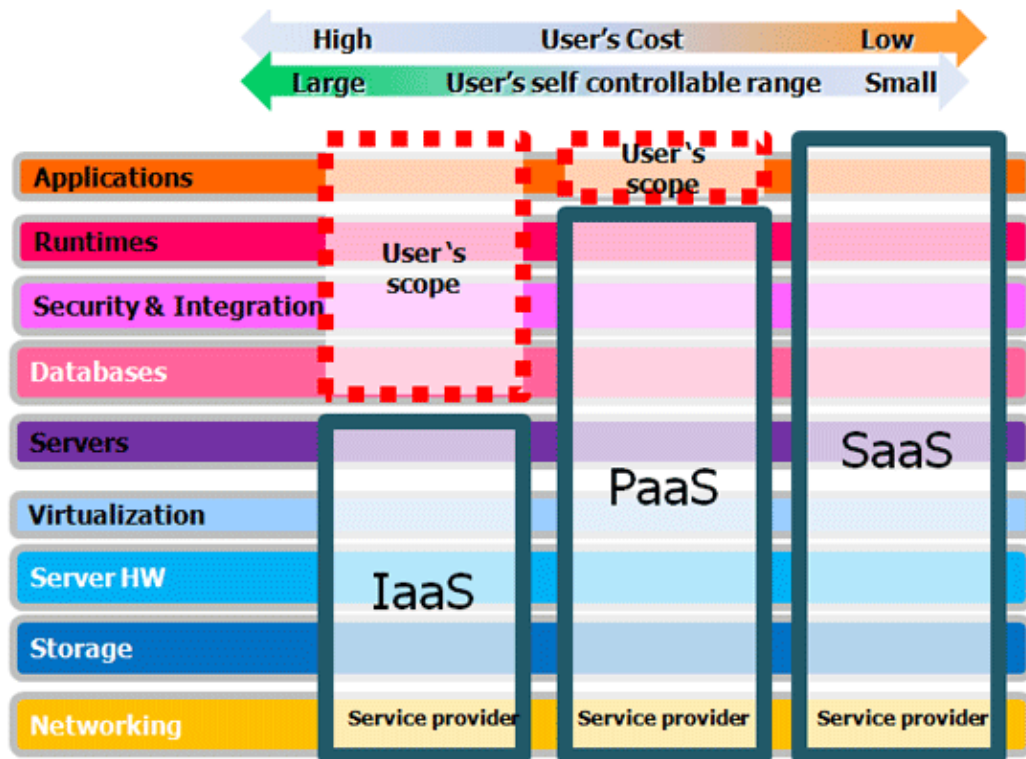


Figure 1.2: SaaS, PaaS and IaaS [3]

- Software as a Service – Software-as-a-Service (SaaS) [4] has been heralded by IT providers as a serious alternative to on-premises software. Cloud Computing provides

software as a service on internet. Individuals don't need to buy the proprietary rights of software they want to use. They can borrow those software from cloud providers according to their need. Some of the biggest name in market that provide software as a service are salesforce.com providing it's CRM application as service also Google web based calendar application , Microsoft online Sharepoint etc. These services are charges either pay per use or on monthly basis.

- Platform as a Service – PaaS offers a complete set of technologies which are required to develop and to operate SaaS applications [5]. This Application Development mostly done by with the help of Web Browsers itself. PAAS examples include Microsoft Azure Services Platform [6] ,Google App engine [7] and Salesforce.com internet application development platform etc.
- Infrastructure as a Service – Due to fluctuations in market demand of hardware and software, resources also vary [8]. To fulfill this constantly changing demand Infrastructure is provided on cloud as a service. Customer can order as many resources as he need and return them back to the provider when his/her goal is achieved. This kind of flexibility is provided in IAAS. One of the examples for IAAS is Amazon which offers s3 for storage and EC2 for computing power.

1.2 Characteristics of Cloud Computing

Cloud Computing emphasize on following five characteristics defined by National Institute of Standard and Technology (NIST) [9]:-

- I. On-demand self service: A consumer can unilaterally provision computing capabilities.
- II. Broad Network access: Services available on all over the network and standard mechanism can be used to access them.
- III. Resource Pooling: Resources of Cloud service provider are pooled to serve different customers by allocating and de-allocating physical and virtual resources to them according to their demand.

- IV. **Rapid elasticity:** Resources and services provided to the customer can be increased or decreased.
- V. **Measured service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

1.3 Challenges of Cloud Computing

This section provides a detailed overview about the challenges of cloud computing.

1.3.1 Interoperability and Standardization

“Standardization of Cloud Computing” [10] is becoming an extremely important research problem. Cloud computing enables a dispersed workforce to work effectively and allows easy collaboration with partners. Many organizations have workers based systems around the globe. These employees often need access to their systems and a hosted desktop service based on cloud computing technology. This facilitates them to access their desktop from any location across the globe. It radically improves the organizations efficiency but simultaneously degrades cloud flexibility. The degradation in flexibility leads to difficult cloud standardization.

Today’s cloud platform and infrastructure providers are providing much of the same services but require applications to interact with them in completely different ways. As more companies shifting applications on the cloud and private clouds interoperability becomes critical, organizations need to ensure that the clouds can talk to one another and workloads can be easily moved among clouds. Currently, each cloud provider has its own set of rules; having all providers follow a single set of standards would encourage even more organizations to make the shift. So if a user wants to move his applications between to two cloud providers he doesn’t have to re-architect the whole application.

1.3.2 Single - Core Systems in Cloud Computing

The computer industry is driven by pursuit of ever increasing performance. From high-end customized special-purpose computing in networking, telecommunications, and avionics to low-power embedded computing in desktop computing, portable computing

A computer cluster is a single logical unit consisting of multiple computers that are linked through a LAN. The networked computers essentially act as a single, much more powerful machine. A computer cluster provides much faster processing speed, larger storage capacity, better data integrity, superior reliability and wider availability of resources. Clusters basically are developed to provide Parallel processing, Load balancing and Fault tolerance. A cloud deals with the consumer directly or consumer depend on providers completely make these things mandatory. In case of cloud, cluster size is very large having thousands processors and if all these processors are single core, the time taken for sending data from one processor to another becomes significantly large and ruins the service because cores on the same machine can communicate faster than cores on different machines [14]. Instead of single-core systems, processors embedded on a single chip thereby, reduce latencies.

- High Processing-

Everyone needs high processing power and its solution in case of Single Core processors is to increase the no. of CPU cycles. This is not a completely feasible solution because it has its own limit. Number of cycles can be increased to a level only because after that the power consumption becomes very high [15] and that cause problems like increase in cost factor for increased power consumption

$$\text{Power} = \text{capacitance} \times \text{voltage}^2 \times \text{frequency}$$

- i. Heat- If the no. of CPU cycles increased the power consumption also increases hence increasing the amount of heat generated that can be dangerous to other equipments. Even if cooling mechanism is applied in case of Single Core processors for managing this, heat problem leads to an overwhelming cost.
- ii. Less support for parallel processing in single processor system in processing system is boosting the size of those Multi-core processors.

1.3.3 Cyber Security

Although cloud provides services to most of areas like educational, research, government etc. it also exposes data to risk. Companies depend on the cloud providers for their most sensitive data and in such cases cloud providers need to take care of security and disaster recovery issues.

1.4 Motivation

Cloud computing is a technology that uses the Internet and central remote servers to maintain data and applications. It facilitates consumers and businesses to use applications without installation and access their personal files at any computer with Internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Though there are numerous benefits of cloud computing, it needs to address issues like “Interoperability” and “Multi-Core compatibility”. Interoperability needs to be dealt with due to the increase in the number of cloud vendors. Similarly, as the whole computer community is adopting multi-core processors, cloud computing also needs to make itself compatible by including semantic representation.

1.5 Organization of Thesis

Rest of the chapters in this thesis are organized as:

Chapter 2 – This chapter describes in detail the literature survey on Cloud Computing Standards and Multi-Core Resource Scheduling Technique.

Chapter 3 – This chapter explains the problem statement of this thesis.

Chapter 4 – This chapter describes the design of the solution for the problem explained in this thesis.

Chapter 5 – This chapter describes the implementation of the proposed solution and the deduced results.

Chapter 6 – This chapter describes the conclusion and future research possible.

This chapter presents the literature survey on Standardization and Interoperability in cloud computing and along with the significance and challenges of adopting Multi-Core processors in cloud computing.

2.1 Standardization and Interoperability

Cloud computing can be defined as accessing third party software and services on web and paying as per usage. It facilitates scalability and virtualized resources over internet as a service, providing cost effective and scalable solution to customers. Still being in the early stage of development, cloud computing presents the classical problem of too many different approaches. Practically every new cloud provider comes with its own solution of interfacing with its resources or services. This variety is a reflection, not only of different potential angles of approaching the concepts of cloud computing, but it is also a reflection of the variety of underlying offer on the market in what concerns the models of storage, networking, processes licensing or even integration of own resources in a cloud. Every new cloud service provider has their own way on how a user or cloud application interacts with their cloud leading to cloud API propagation [16]. This kills the cloud ecosystem by limiting cloud choice because of vendor locking, portability, ability to use the cloud services provided by multiple vendors including the ability to use an organization's own existing data center resources seamlessly.

There are several different kinds of standards under development today, some of these are discussed here.

2.1.1 Open Cloud Computing Interface (OCCI)

Open Cloud Computing Interface [17] is an effort driven by a working group in the standards track of the Open Grid Forum. OCCI attempts to create an open interoperable

protocol and API for the Cloud. The proposed by OCCI can be used for integration, validating interoperability and portability between different Cloud providers. In cases where features other than those of OCCI need to be maintained, provider specific APIs can be used. OCCI is broken into different modules in attempt of making these specifications to be very easy, flexible and extensible. First module describes the core model. How the proposed model can be mapped and submitted using a HTTP/REST approach is shown in second module. The third module describes the infrastructure entities and shows how these entities are related to the first module that is core. Figure 2.1 shows the class diagram of the OCCI.

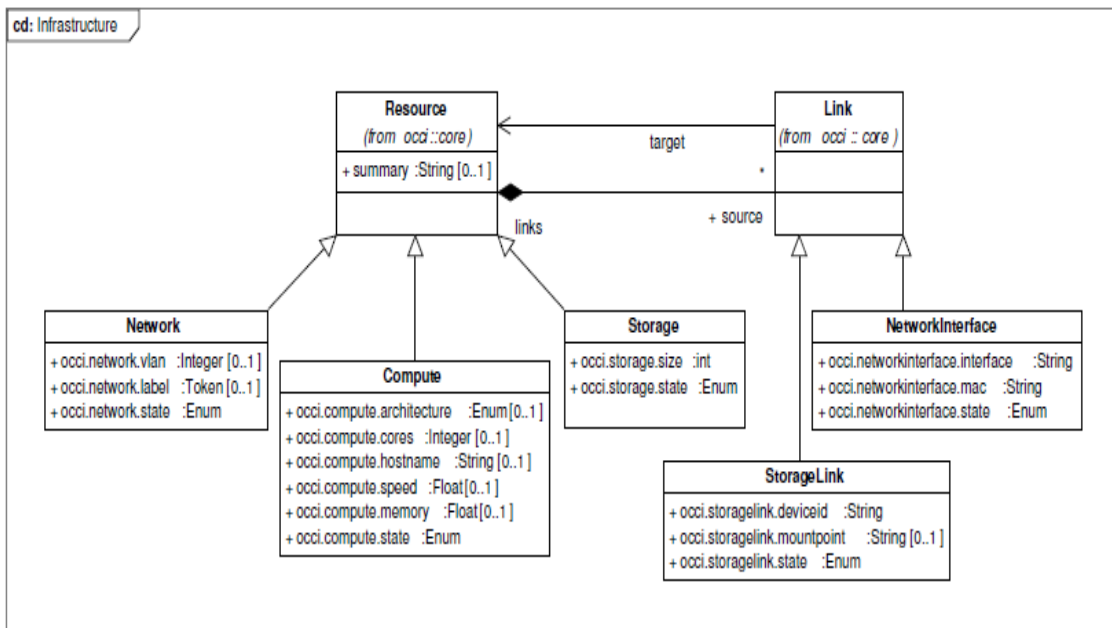


Figure 2.1: OCCI class diagram [17]

2.1.2 Unified Cloud Interface (UCI)

Unified Cloud Interface [18] is the project of The Cloud Computing Interoperability Forum (CCIF). Unified Cloud interface proposes that, an API can be developed for all cloud APIs of different cloud providers. The approach which is used is RDF (Relational Database Framework) based on ontology language for enabling interoperability. The rationale behind using RDF is that it helps in providing semantics to the information that is implemented by web browsers. The unified cloud model addresses both Platform as a Service as well as infrastructure cloud platforms. This model enables a decentralized but

still extensible hybrid cloud computing environment. This architecture focuses on secure worldwide asynchronous communication. Figure 2.2 shows the entire architecture of the Unified Cloud Interface architecture.

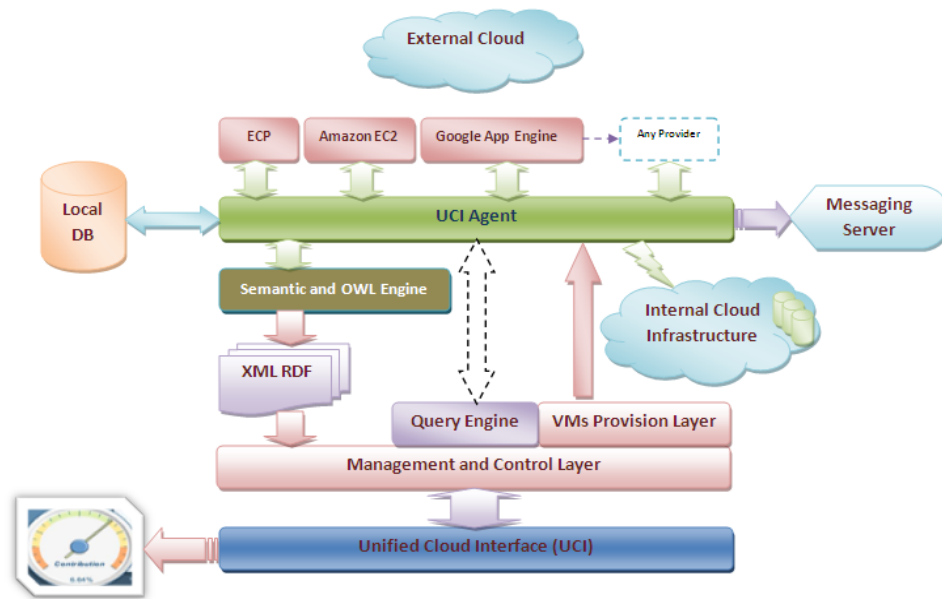


Figure 2.2: Unified Cloud Interface architecture [18]

2.1.3. Open Virtualization Format (OVF) [19]

With the hasty adoption of virtualization, it seems obligatory to find a standard way to package and distribute virtual machines. VMware and other leaders in the virtualization field have created the Open Virtualization Format (OVF) [19], a platform independent, efficient, extensible, and open packaging and distribution format for virtual machines. What OVF does is, it facilitates efficient, flexible, and secure circulation of enterprise software. It also ensures the independency of customers and vendor by providing platform independent virtual. OVF facilitates customers by giving them choice to deploy virtual machine on the virtualization platform according to their requirement. OVF provide following features:

- Improve user experience with simplified installations
- Offer customers virtualization platform independence and flexibility
- Efficiently deliver enterprise software through portable virtual machines

- Offer platform-specific enhancements and easier adoption of advances in virtualization through extensibility

The portability and interoperability inherent in OVF will enable the growth of the virtual appliance market as well as virtualization as a whole.

2.2 Ontology: Cloud Computing Perspective

Lamia Youseff et al. [20] gave a thorough interpretation of cloud in terms of ontology in their work, covering all the theoretical points but did not mention any technical implementations. Similarly Yong Beom Ma, Sung Ho Jang, and Jong Sik Lee [21] proposed resource management algorithm in cloud using ontology. In their work they suggest that user's request for resources is not processed only on the basis of CPU size, operating system or storage but also using agreed SLA's (service level agreement).

Takeshi Takahashi, Youki Kadobayashi and Hiroyuki Fujiwara [22] developed a cyber security ontology using actual cyber security for cloud computing. Taekgyeong Han and Kwang Mong Sim [23] proposed a cloud service discovery system which consults ontology when retrieving information about Cloud services. Haytham Tawfeek al Feel and Mohamed Helmy Khafagy [24] suggested an ontology based file system that can store and retrieve in cloud based on the content of the information.

2.3 Virtualization

Virtual Machine (VM) technology has recently emerged as an essential building-block for data centers and cluster systems, mainly due to its capabilities of isolating, consolidating and migrating workload [25]. Altogether, these features allow a data center to serve multiple users in a secure, flexible and efficient way. Consequently, these virtualized infrastructures are consider a key component to drive the emerging Cloud Computing paradigm [26]. Virtualization helps in creating virtual versions of resources like servers, operating system, networks and storages. Virtualization basically enables a single user to access multiple physical devices virtually, either its one operating system using multiple computers to evaluate database or a single computer controlling several

machines. Cloud provides physical and software resources to users and virtualization make it possible for user to have single computer experience even if all those resources exit remotely from each other.

Virtualization Types:

- **Full Virtualization:**

It allows applications to run on a unmodified guest operating system. One of the examples of this type of virtualization is VMware virtualization software. It uses a combination of direct execution and binary translation techniques to achieve full virtualization of an x86 system. In this case, Virtualization software itself is responsible for converting the data or resource in application compatible format.

- **Para-Virtualization:**

In case of Para-virtualization modifies the guest operating system to prevent binary translation. Therefore it adds certain performance advantage for workloads but sometime require modified operating system kernels [27]. Xen an open source project is developed for providing Para-Virtualization.

2.4 Challenges with Multi-Core Processors

Single core processors problems like low latencies between clusters, heat etc. degrading the performance of cloud computing. This is one of the reasons that use of multi-core processors in cloud computing exponentially increasing. Still there are some problems with Multi Core Systems which need to address before utilizing the full potential of multi-core systems in the cloud.

2.4.1 Cache Coherence - A cache system is *coherent* if, whenever an object is read, the returned value is the one that is most recently written [28]. Main memory is slow that's why cache memory is used to cope up with the speed of processors, to enhances the overall speed of the system. Whenever processor needs some data, it first checks its cache memory then request goes to main memory. In all architectures of Multi-Core Processors "L1 Cache" is dedicated that's the cause of the problem. It is not always necessary that all L1 cache memories have updated data all the time. So every processor may give different result for same problem.

2.4.2 Multithreading - Multithreading is also one of the most important issues holding the power of Multi-Core systems from unleashing. “With the possible exception of Java, there are no widely used commercial development languages with multithreaded extensions.”[29] Most of the software used are not totally compatible with Multi-Core systems. Programmers need to do lot of rework to make these program to work on Multi-core systems efficiently. If one core is being used much more than another, the programmer is not taking full advantage of the multi-core system. Many companies have new designed new products based on multi-core capabilities.

2.4.3 Virtual Resource Utilization- As Multi-Core is as important as virtualization for cloud. This leads to a new set of problem. These are:-

- Good Resource Utilization- When we distribute a virtual machine on several cores it is important that VMs utilize all the cores efficiently.
- Coordination- This problem talks about the even distribution of resource amongst all the running virtual machines. So that none of VMs starves or consume more than sufficient resources

2.5 Resource Scheduling in Multi-Core Systems

Virtual machines in case of Multi-Core systems are much more like threaded applications running simultaneously to complete a job and fighting for resources. It is necessary to handle these demands efficiently and quickly to avoid starvation and wastage of resources.

Resource scheduling in case of virtual machine can be done in two ways:-

- Hardware based Scheduling-
The virtual machine manager is embedded in the circuits of a hardware component instead of being called up from a third-party software application [30]. Hardware itself is intelligent enough to share itself in an efficient way and no under or over utilization occur. It is given by AMD-V and Intel VT technologies in X-86 processor itself.
- Software based Scheduling-
This type depends on Software for scheduling purpose. It also occurs in two ways. Firstly, host operating system modified in such a way that it can manage resources of

virtual machine also that are running on it. It going to add more pressure on host operating system and that can lead to degrade host operating system's performance and ultimately the whole system goes down. Also this is not particularly defined which OS can run as guest on which host operating system. The option left is to add scheduling technique for every type of guest in OS in every Host OS. This is not feasible

Either make Virtual Machines itself in a way that they can use available resources evenly (Friendly virtual Machine) or develop separate techniques to schedule their resources.

2.5.1 Underlying Layout Aware Scheduling

In Underlying Layout Aware Scheduling (ULAS) [31] resources are dynamically adjusted according to the workload generated on the virtual machines by the applications running on them or by the mapping of Virtual CPUs to the Physical CPUs, to get maximum throughput.

In Figure 2.3 there are four mapping schemes of Virtual CPUs to Physical CPUs. In first, none of the Virtual CPUs is mapped (pinned) to any Physical CPUs.

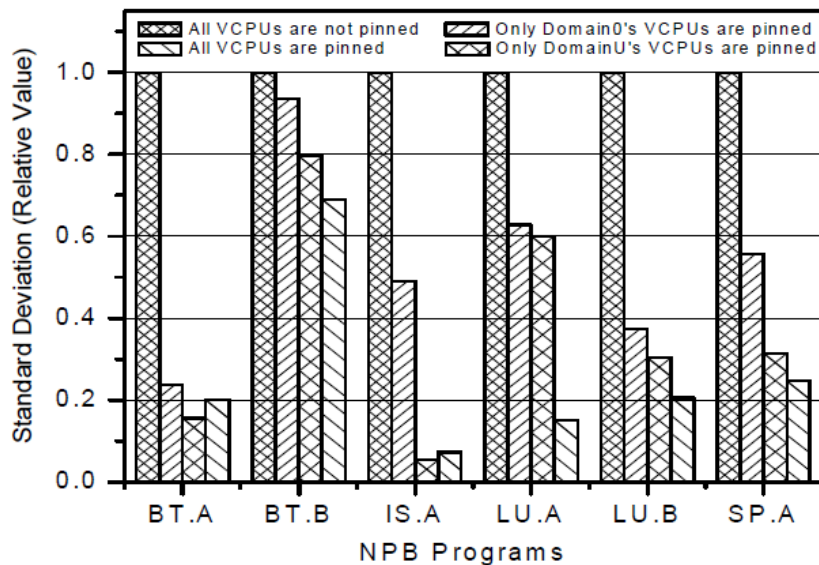


Figure 2.3: Standard deviation Comparison of four mapping Scheme [31]

In second, all of them are pinned to specific Physical CPUs and in last, there are two domain in which specific Virtual CPUs mapped to random Physical CPUs. Using a NPB (Network Parallel Benchmark) performance of all the combinations is detected. The case

in which all Virtual CPUs are pinned to specific Physical CPUs has lowest standard deviation. It clearly shows that if mapping of Virtual CPUs done on the basis Underlying Architecture in gives better performance rather than random allocation.

2.5.2 Forecasting and Time Delay Subtraction

FTDS [32] is Forecasting and Time Delay Subtraction. AISD (Additive Increase Subtractive Decrease) [32] is a traditional scheme for allocating resources. In some cases where allocation and de-allocation of resources works very rapidly (like in kernel compilation) , Threshing [32] adds more overhead to the AISD. In FTDS (Figure 2.4) [32] using PID (Proportional Integral Derivate) [33] resources are allocated on the basis of past history of static allocation of resources and on current utilization.

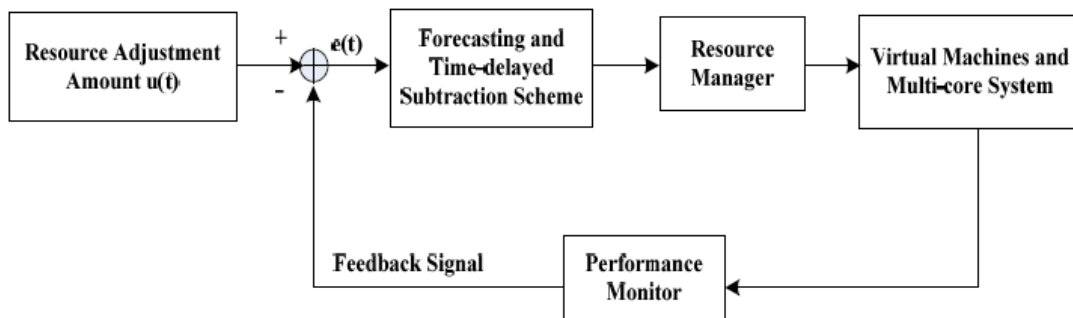


Figure 2.4: Block Diagram of FTDS [32]

Figure 2.5 shows the performance comparison of Static Scheduling, AISD and FTDS (Xendd) in case of Kernel Compilation with different comparisons.

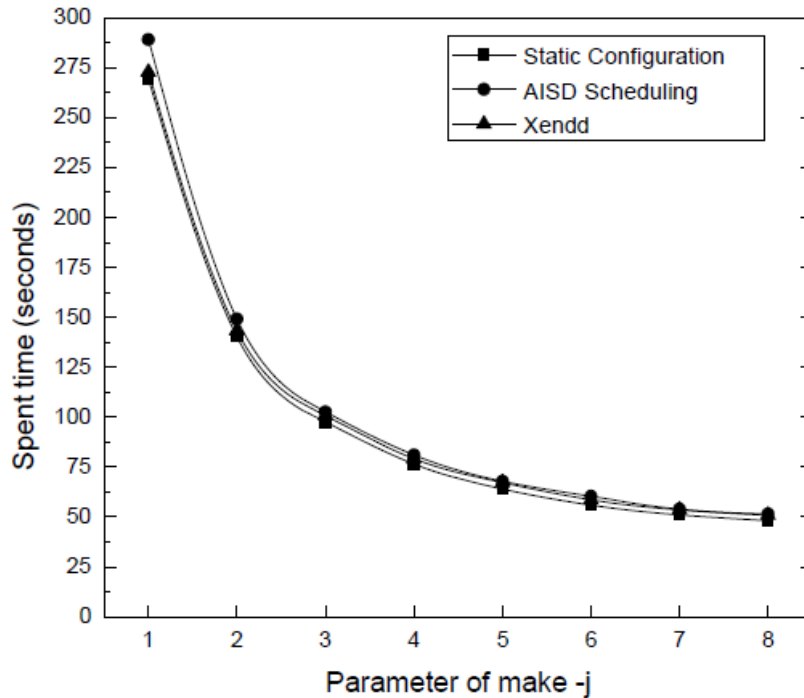


Figure 2.5: Performance comparison kernel compilation with Static, AISD and FTDS scheme [32]

Similarly, in case of NPB (Network Parallel Benchmark) comparison (Figure 2.6) of all these three techniques Static priority, AISD and FTDS. FTDS shows better results in comparison to AISD and Static Priority.

2.5.3 Friendly Virtual Machine

Friendly Virtual Machine (FVM) [34] is based on the concept that the virtual machine itself manages or share resources with other Virtual Machines evenly without causing any resource congestion. FVM [34] framework uses Additive Increase Multiplicative Decrease (AISD) policy.

It consists of three stages:

1. Overload Detection
2. Controlling Resource Consumption
3. Adaption Strategy

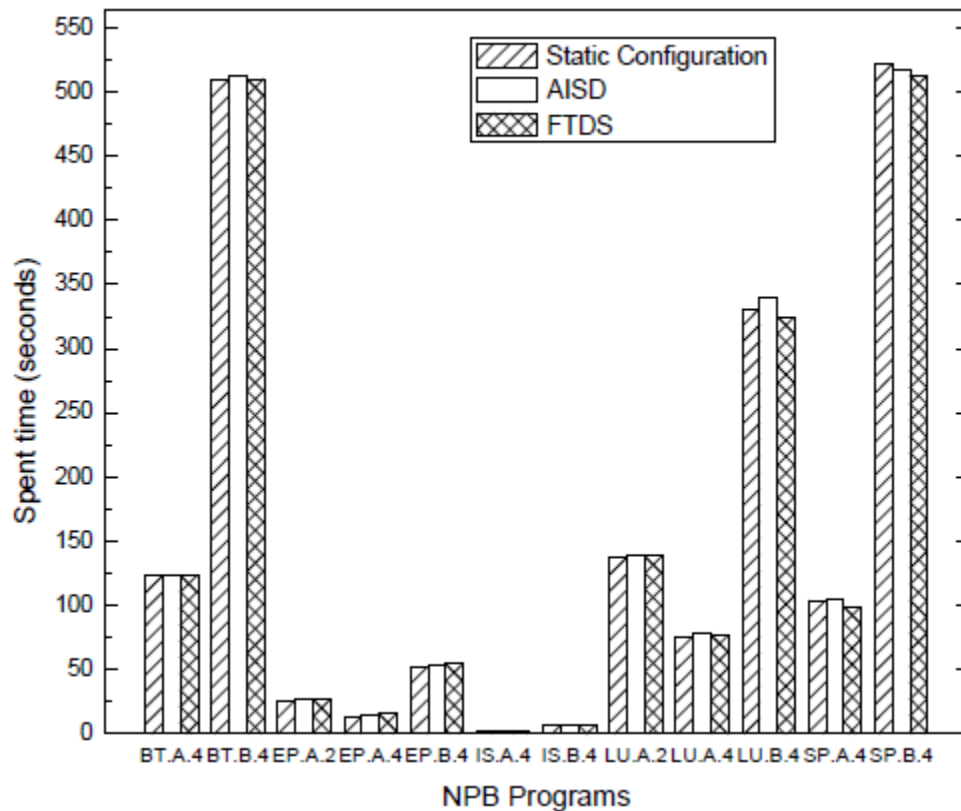


Figure 2.6: Time taken by NPB application on different configuration [32]

- Overload detection- Different types of metrics are used for overload detection like response time and throughput metrics. Evaluation of the values from these metrics helps in detecting that “is system in an overload state?” Difference between two VCT (Virtual Clock Time) [34] is used to detect the overload. Also a congestion control signal is developed using EWMA [35].
- Controlling Resource Consumption- If system is in overloaded state then either by making Virtual Machines sleep periodically or suspending and resuming the threads of different applications helps in controlling the resource consumption.
- Adaption Strategy- For optimizing the resource consumption AIMD [34] (additive increase multiplicative decrease) is used as an adaptive strategy. In AIMD allocation of resources is additive but di-allocation of resources done multiplicatively to deal with congestion.

Experiment done for detecting “Fairness of resource sharing”, Throughput and VCT of virtual machines with FVM and without FVM (Figure 2.7) [34] shows significant improvement in case of “4With FVM” strategy . A benchmarking application is used. This application reads data from the memory perform some computations and write it back very rapidly.

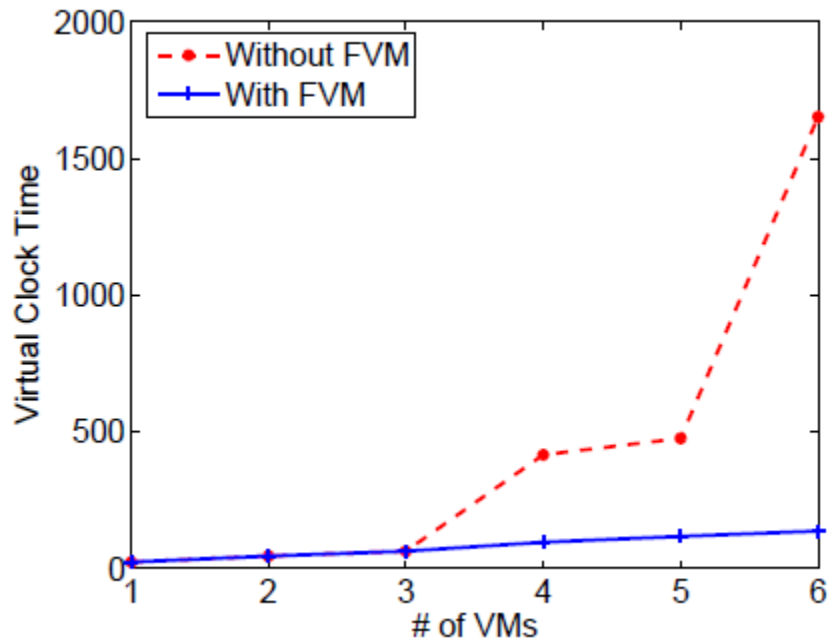


Figure 2.7: VCT comparison with and with FVM strategy [34]

2.5.4 Task-Aware VM Scheduling

In Task Aware VM scheduling method [36] guest OS utilize the current resource utilization data and feed it to the virtual machine monitor (VMM) (Figure 2.8).

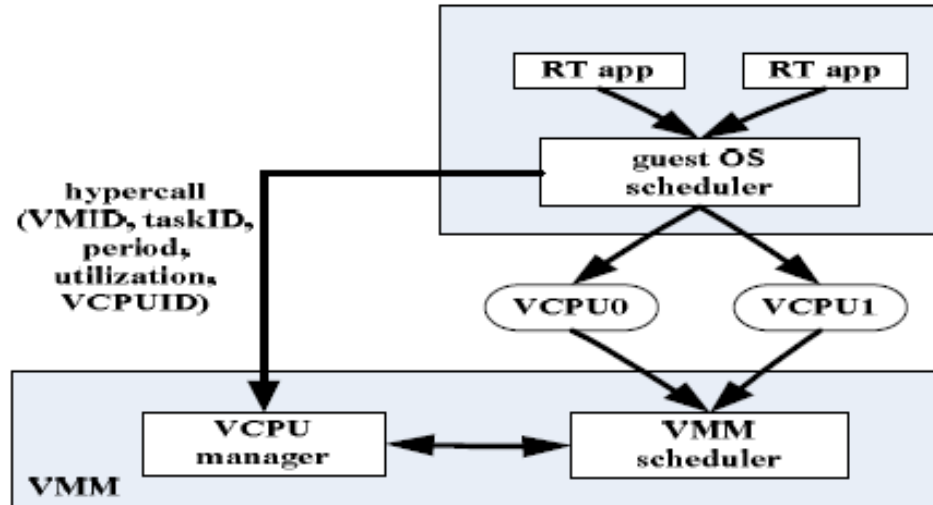


Figure 2.8: Task Aware scheduling architecture [36]

In Task Aware scheduling guest operating systems scheduler determine the resource utilization by using deadline and execution time of the process & then send the information to virtual machine monitor that send info. Further to VCPU manager. VCPU manager use that info. to assign a physical cpu to a virtual cpu.

2.6 Conclusion

Different cloud standards and Multi-core resource scheduling techniques have been discussed in this chapter. Use of Ontology in cloud computing has also been surveyed. These standards and techniques provide an insight of the ongoing work the concern area. Next chapter focuses on the analysis of the problem based on literature review.

Chapter 3

Problem Analysis

Based on the literature survey various gaps have been identified. This chapter presents the gap analysis and the objective of this research work.

3.1 Gap Analysis

3.1.1 Current Standards for Cloud Interoperability

In literature survey chapter three different Interoperability standards were explored. These all standards provide solution for “vendor-locking” in their own way. Unified Cloud Interface focuses on building an API for all cloud APIs available from different vendors using RDF, based on ontology language and thus enables interoperability. Open Cloud Computing interface talks about providing an API specification for remote management of IaaS services across vendors and Open Virtualization Format focuses on building an industry standard format for portable virtual machines. VMs thus built can interoperate with any other VMs.

Out of these three, two standards focus on developing vendor free APIs or in other words API for vendor APIs. There is no need to change the whole configuration every time user changes his/her vendor. Unified Cloud Interface conceptualizes an Ontology [38] based approach for developing the Interoperable Cloud standards.

As the existing web is moving towards semantic web [39] and can help in developing interoperable cloud standards, it seems natural that cloud computing also needs semantic description because of its dependency on web. Developing the cloud ontology also helps in increasing the knowledge level expertise of the cloud infrastructure. As in ontology everything is represented as real world entities, it becomes easy to understand them and modify them according to the future requirement.

3.1.2 Multi-Core Scheduling Techniques

The current available scheduling methodologies discussed in the literature survey are not totally compatible with the present cloud environment. Table 3.1 shows some of these techniques. Some of these techniques focus on the limited number of cores in a given scenario but as in case of cloud, the numbers of the cores can be thousands. In such “high number of cores” case these techniques are questionable.

Table 3.1: Multi-Core Scheduling Techniques Comparison

Technique	Test Bed	In Cloud Computing
Underlying Layout Aware Scheduling & Domain based Static priority scheduling [31]	Red Hat Enterprise Linux Server 5.1 Xen 3.4 Unstable	Need to deal with flexibility problem before implementing this technique in Cloud Computing
Forecasting and Time Delay Subtraction Scheme [32]	Red Hat Enterprise Linux Server 5.1 Xen 3.4-unstable	Due to huge number of virtual machines in Cloud Computing, storage of all Past resource utilization data is not entirely possible
Friendly Virtual Machine [34]	User Model Linux Xen	FVM comes with an extra overhead to VM's and in cloud computing the no. of VM's is high enough to decrease overall performance
Task-Aware VM Scheduling [36]	Linux 2.4.20 Xen VMM 4.0.1	Cloud customer can demand for any type of operating system but this techniques limits to open source operating system and virtualization software due to modifications required

Some of the cloud computing specific problems that these techniques don't address are as follows:

- Modify Guest Operating System

Modifying guest OS is not flexible enough for cloud computing. Cloud provides every type of operating system to its customers and not all of them are open source. So this technique cannot be applicable in case of heterogeneous operating systems environment because operating systems like windows are already available.

- **Modify Virtual Machines**

Making modification in virtual machines, so that instead of fighting for resources they themselves “divide them evenly” is also an incompatible technique for cloud.

At a single time there are thousands of virtual machine running on a cloud and it is not easy to modify every single one of them and also the machines that are generated every second on customers demand. And even if it is done, it will be going to add a huge extra overhead to the cloud.

- **History of resource utilization**

In a live Cloud computing environments there are numbers of resources and there are also numbers of application running on them. It is not possible for a cloud provider to store resource utilization data for every machine and every application due to the enormous amount of data.

3.2 Objectives of the proposed work

Following are the objectives of proposed work

- Analysis of interoperability and standardization techniques in cloud computing
- Exploring the potential of Multi-Core processors in cloud computing
- Design an Ontology Based framework for cloud
- Develop an effective resource scheduling mechanism for multi-core processors cloud environment.

3.3 Conclusion

This chapter discussed the problem statement. Next chapter discusses the design of the solution.

This chapter provides the design of the solution for the problem statement explained in this thesis through proposed Frame Work. The frame work has been divided into two layers. First layer concerns with Cloud Ontology and second layer concern with IaaS cloud portal. Also it gives an introduction of the tools used to design this frame work.

4.1 Proposed Frame Work

Need of interoperability and resource scheduling in Cloud Computing can be achieved using the proposed frame work (Figure 4.1). Proposed framework is divided in two separate layers. First layer is ontology based, that deals with interoperability problem. It comprises of following units:-

- Browser -Web Browser acts as an interface for both User and developer. User can query using Manchester OWL syntax [40] and developer can use SPARQL [41] and Manchester OWL syntax both.
- Database - All the information and queries are stored in database. The stored information is used to answer the queries. Changes done by developer simultaneously updates in the database.
- OWL Ontology - In OWL Ontology layer the ontology rules are developed. These rules define relations between different ontology classes.
- Management Services - In management services rules are divided in different categories like billing, data usage and payment. Answer of user queries and to management of entire cloud infrastructure is depends upon these rules.

Second layer of the Frame Work deals with the Multi-Core Resource Scheduling Problem. After the Service Level Agreement based scheduling of applications, further scheduling of applications on hardware level is done by this layer. It can comprises of:

- Guest Operating System- Guest os are the operating systems offered by the cloud providers to their customers.
- Hypervisor- Hypervisor is the part which acts as a communication tunnel between guest operating system and hardware. Any direct communication is forbidden for security purposes. Also all the application and resource scheduling done by hypervisor. The multi-core resource scheduling technique proposed in this thesis is also an part of hypervisor.
- Hardware- All the hardware parts like hard disks, processors etc. lie here. In this framework hardware part concentrate on Multi-Core processors.

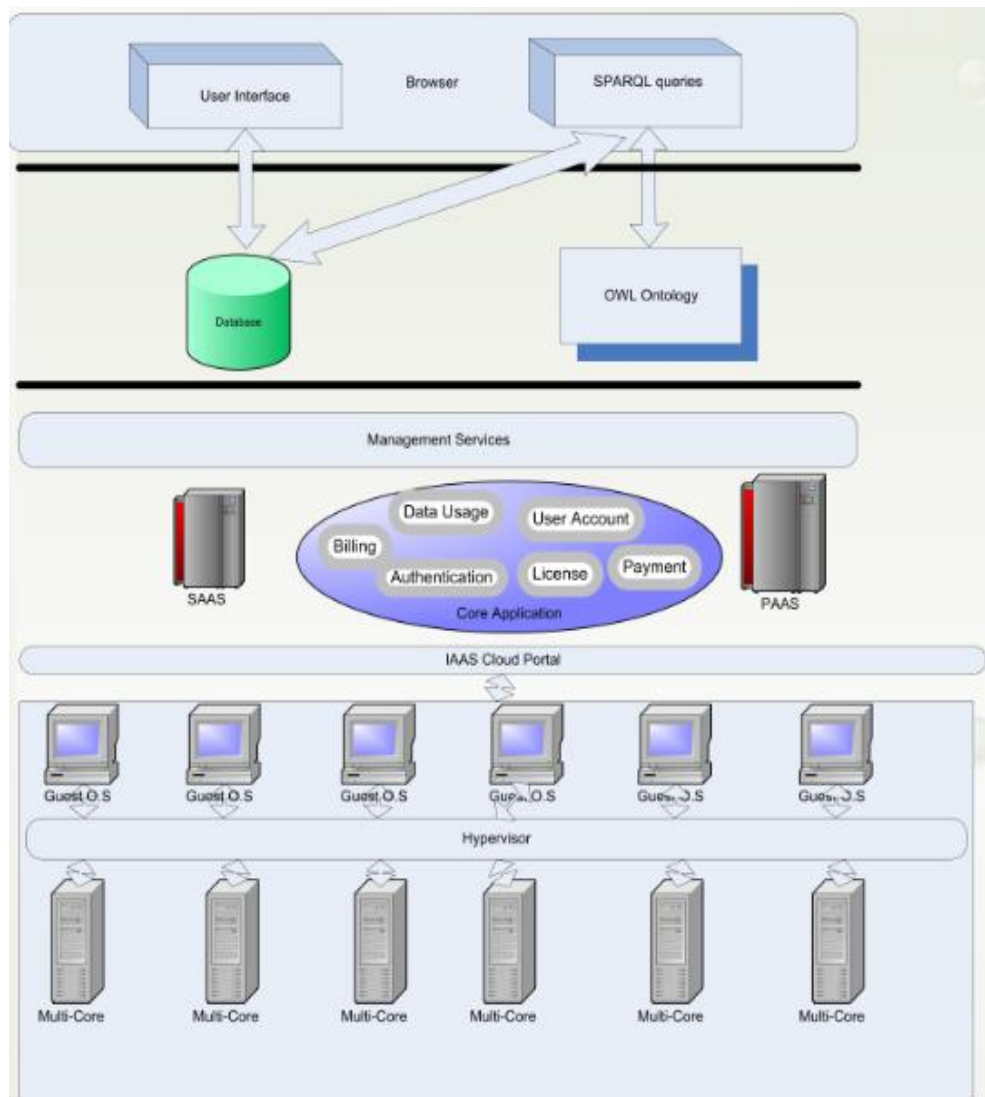


Figure 4.1: Ontology Based Cloud Frame Work

4.2 Design of Cloud ontology

Ontology can be defined as a “formal explicit specification of a shared conceptualization of a domain” [42]. In ontology real world entities can be described as they are and their relationship with other entities. This representation is done using their real names. This kind of architecture helps in sharing common structure of information among developers.

Also because of the clear understanding of the developed architecture, reuse of the obtained knowledge is easy. Ontology based solution is designed for the first layer of the Frame Work. The design of the developed ontology is explained using two representations.

4.2.1 Hierarchal view of developed cloud ontology – Figure 4.2 shows a hierarchal view of ontology. In this view a descending architecture of the different developed ontology classes is shown. Here “Thing” class is a root class and all other classes follow it. Similarly “Preference” class is followed by “High”, “Medium” and “Low” subclasses.

A brief explanation of above classes is given below.

- Cloud - This class represents the services of cloud. It has three subclasses which are: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a service).
- Hypervisor - Hypervisor Class shows different hypervisors that are available on cloud. Hypervisor class has two sub classes: Full Virtualization and Para Virtualization. These sub classes are further sub divided in sub classes like VMware and Xen-hypervisor.
- Operating System - This particular class deals with operating systems choices available on the cloud for the users. It has sub classes like Windows, Ubuntu, Fedora, Linux etc.
- Preference - This is a special class that deals with the SLA (service level agreement) part. Here services are divided in three different categories which are

sub classes of preference class and these categories are “High”, “Medium” and “Low”. “High” category services are those services for which user pay highest amount of fee. These services are come with no or very little downtime. Similarly “Medium” category provides same services with little bit high downtime or fewer options with fewer fees. Same hierarchy is followed in “Low” category.

- User Account - As the name shows this particular class takes care of different costumer’s personal data. Sub classes like billing, data usage and payment which are user specific are also the part of User Account class.
- Machine- This class defines two sub classes Hardware Machine and Virtual Machine.
- Storage - This class represents the storage type available for users to choose on cloud.

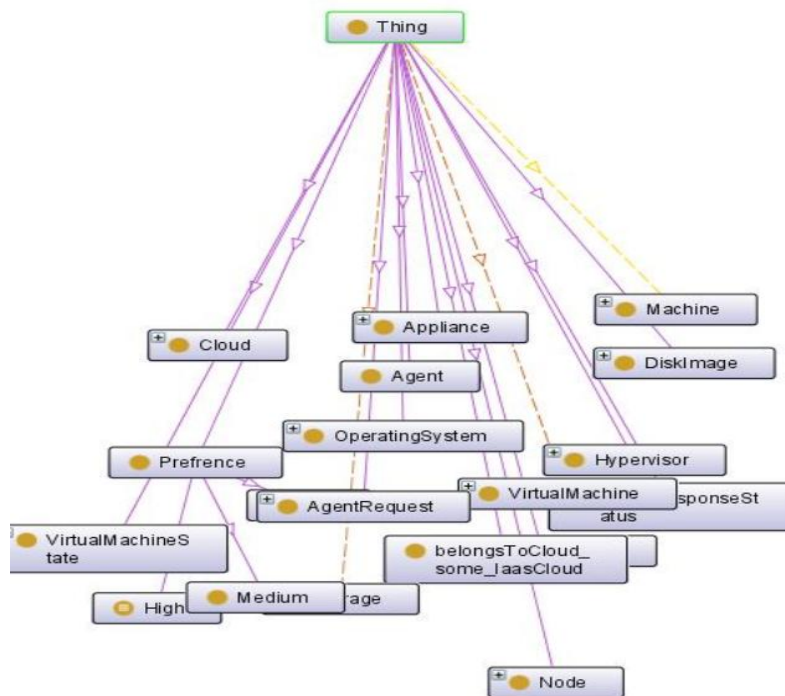


Figure 4.2: Hierarchical view of cloud ontology

4.2.2 Cloud View

Cloud view is visualizing an ontology as a tag cloud. This view provides the insight of the developed cloud ontology. A tag cloud is a visualization of word frequencies. Tag cloud provides a view that shows how frequently words appear in a given text. Same strategy used in case of cloud view.

Cloud view provides the most frequently used classes in a developed ontology. These classes are represented using bold and comparatively large font sizes with other classes. In above figure classes like “Operating System”, “Cloud”, “Virtual Machines” etc. are some of those classes which are being used most frequently in comparison to the other classes like “Agent Request”, “Low” and “Fedora” etc.

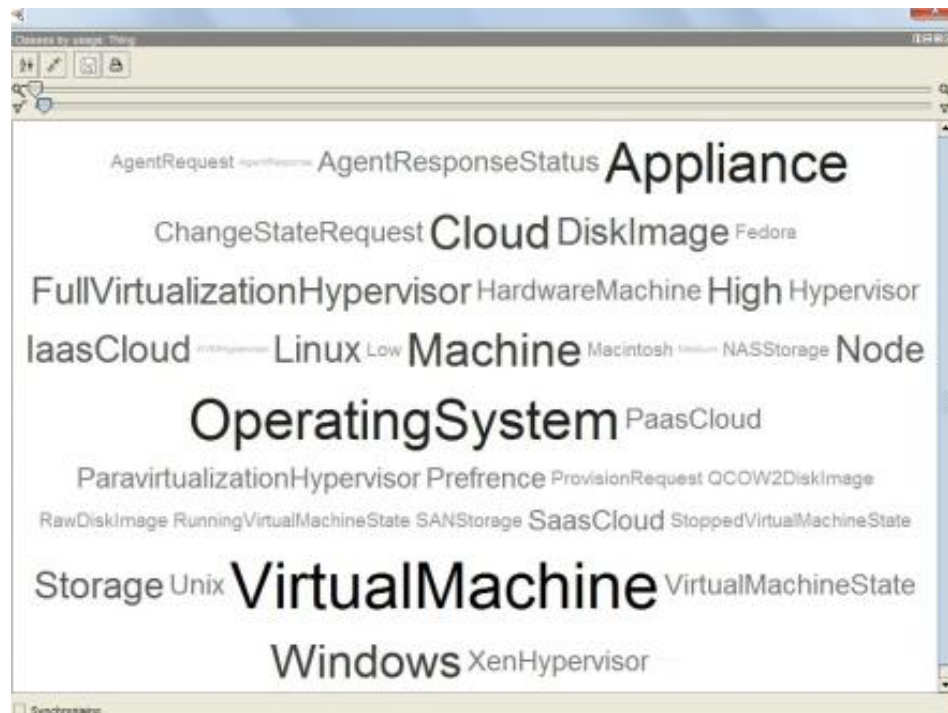


Figure 4.3: Cloud View

Development of the ontology is done in the Protégé 4.0. Developed Ontology Classes shown in Figure 4.4.

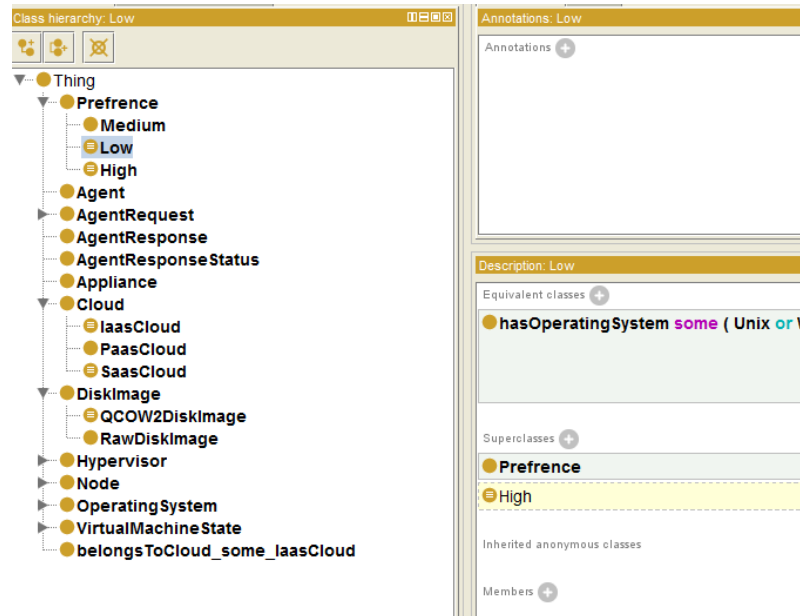


Figure 4.4: Ontology Classes

These Classes are interrelated with each other. These interrelations between these classes are forged using object properties that define how classes are related to each other (Figure 4.5).

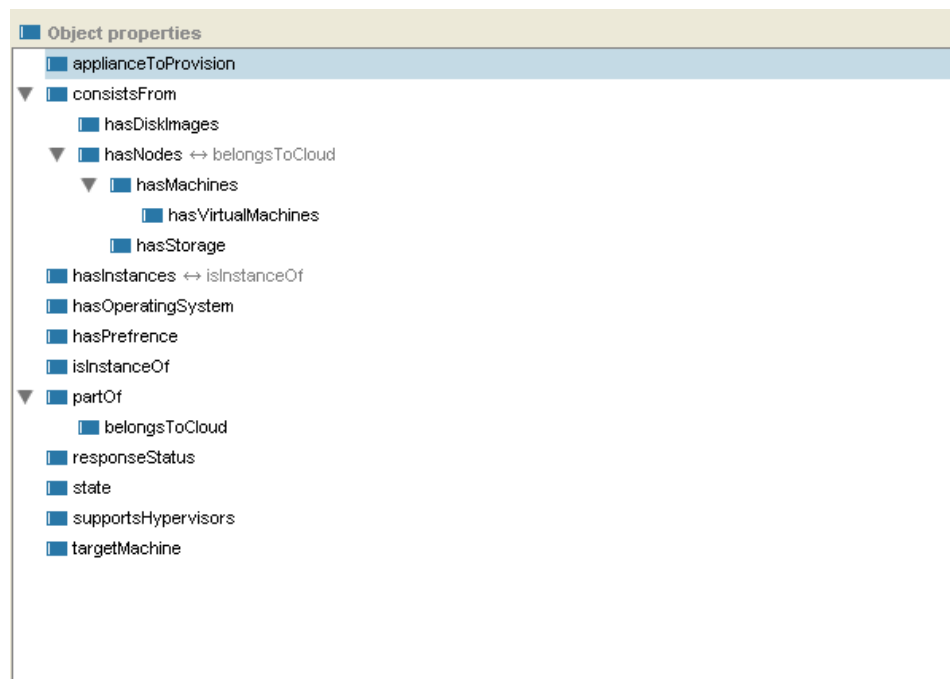


Figure 4.5: Object Properties

4.3 IAAS Cloud Portal

To solve problems discussed in the gap analysis section of multi-core techniques, a length based scheduling mechanism is developed. This mechanism receives the Length of the application and VM configuration from user and uses that information for the best possible combination of allocation of the application on the generated virtual machines.

4.3.1 Flowchart

A flowchart is a diagram that depicts the “flow” of a program. Flowchart of the IaaS cloud portal architecture is shown in Figure 4.6.

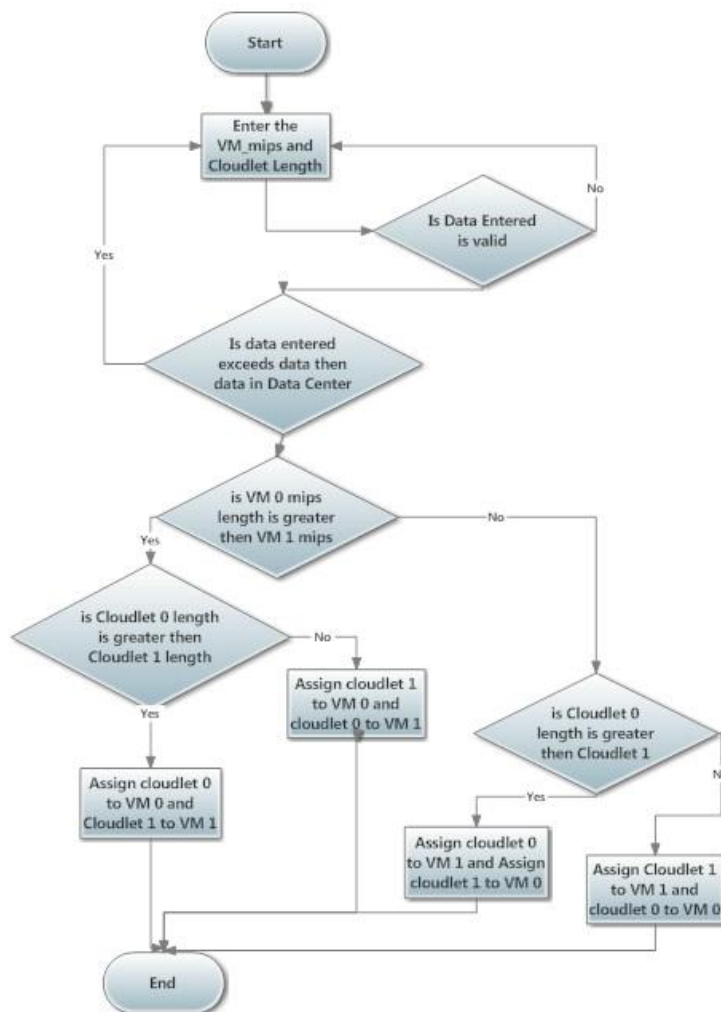


Figure 4.6: Flowchart of IaaS cloud Portal

4.3.2 Data Flow Diagrams

Data Flow Diagrams known as DFDs are a way to model real world situation They are the interface between the real world activities and an understanding of how this can be converted into a computer system. It is a way of taking the physical view and converting it into a logical view.

Their main purpose is to communicate with the user and the analyst understands of the scope of the required system. Figure 4.7 shows a 0 LEVEL DFD of the proposed IaaS cloud portal.

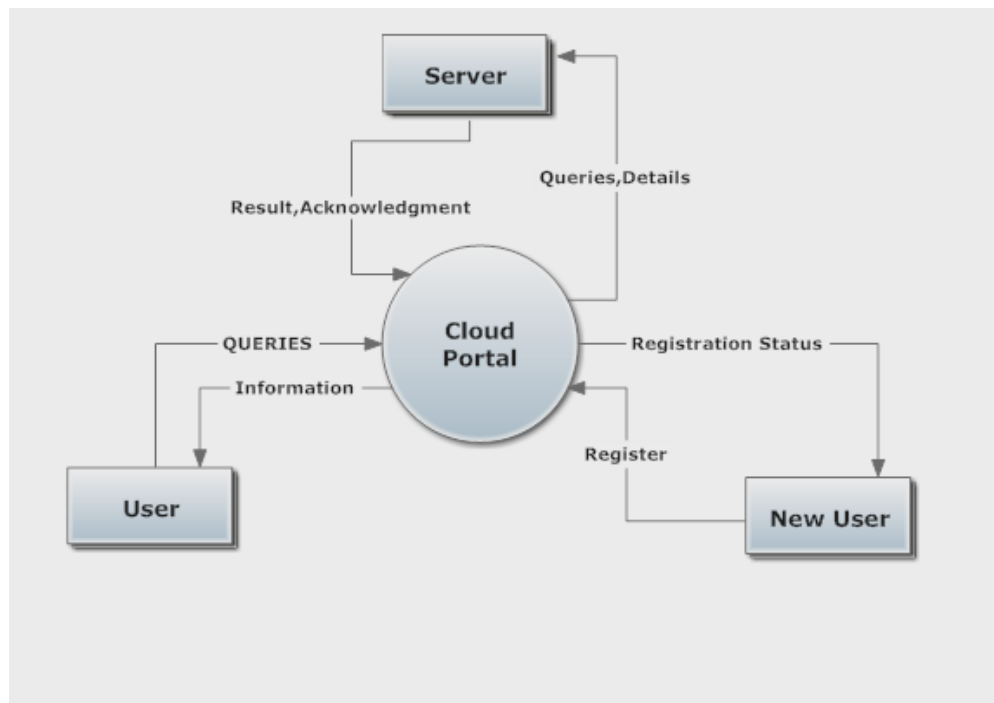


Figure 4.7: 0 Level DFD of the IaaS Cloud Portal

4.3.3 UML Diagrams

Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, business modeling and other non- software systems. The UML uses mostly graphical notations to express the design of software projects. These graphical notations are combined to construct UML diagram.

i. Use Case Diagram

Use case Diagram describes the behavior of the target system from an external point of view. Use cases describe "the core" of the actual requirements.

- Use Case - A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- Actors - An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- Associations - Associations between actors and use cases are indicated by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

Figure 4.8 shows the Use Case diagram of the proposed technique of Iaas Cloud Portal.

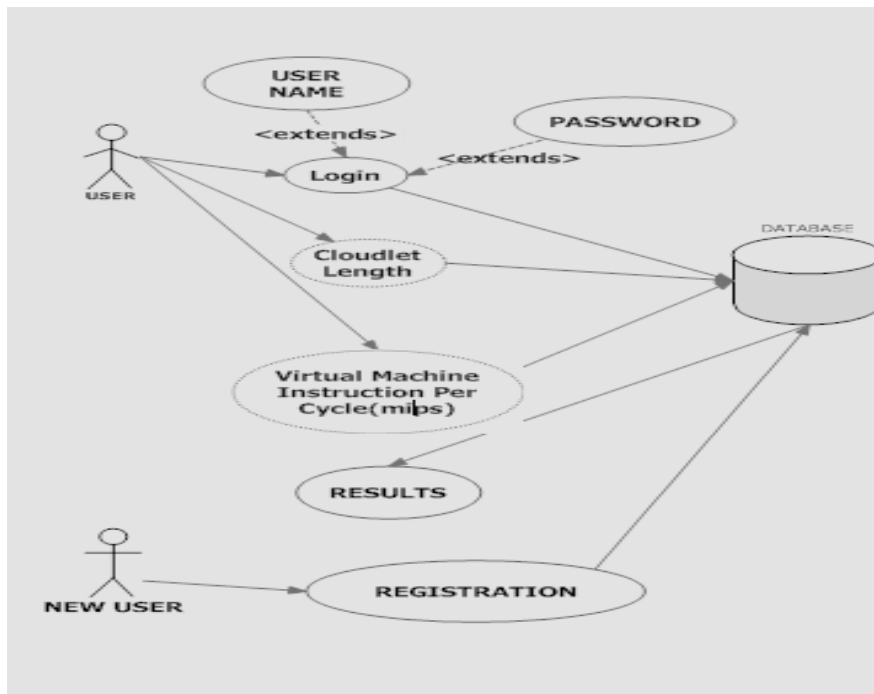


Figure 4.8: Use Case Diagram of the Iaas Cloud Portal

ii. Sequence Diagram

Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. Figure 4.9, 4.10

and 4.11 shows the sequence diagram of the proposed technique. Figure 4.9 shows the sequence diagram of the interaction between User and Server. Figure 4.10 shows the sequence diagram between User and the Server registration. Figure 4.11 shows the full interaction of user and the server.

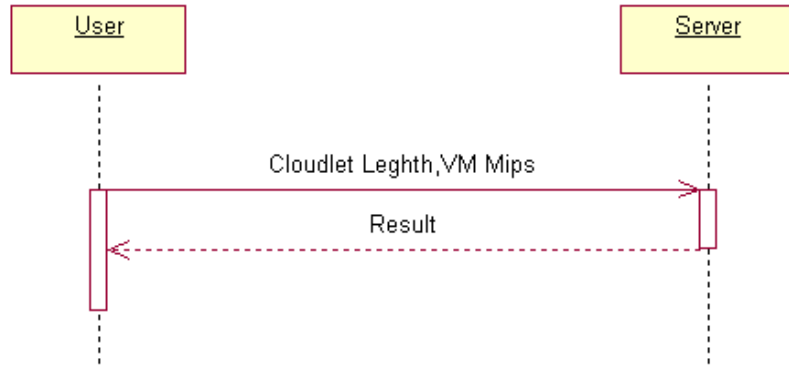


Figure 4.9: Sequence Diagram of user and server



Figure 4.10: Sequence Diagram of user and server registration

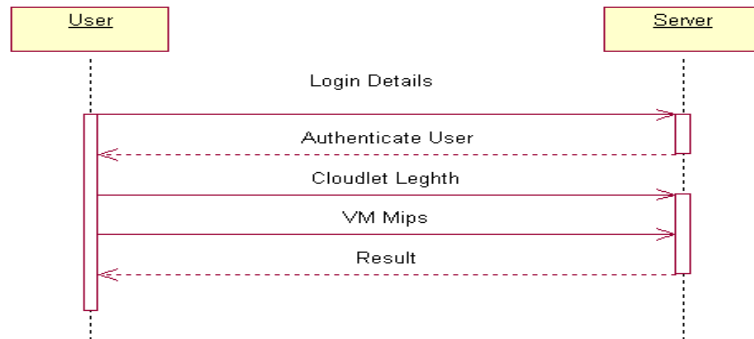


Figure 4.11: Sequence Diagram of full interaction between user and server

iii. Activity Diagram

Activity diagrams help to describe the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process. It is object-oriented equivalent of flow charts and data-flow diagrams (DFDs). Figure 4.12 and Figure 4.13 shows the various activities of the IaaS cloud portal.

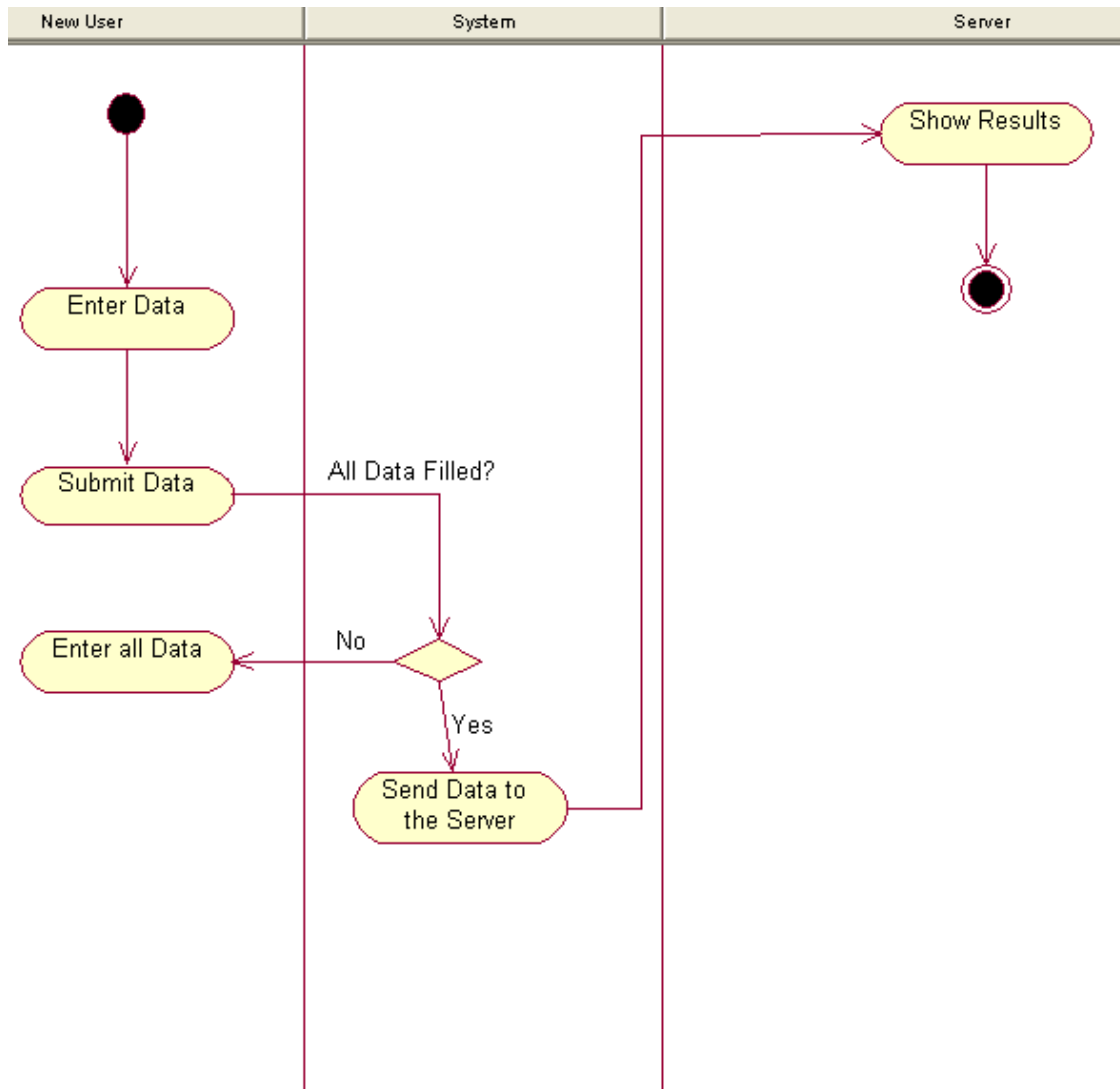


Figure 4.12: Activity diagram of User performing query

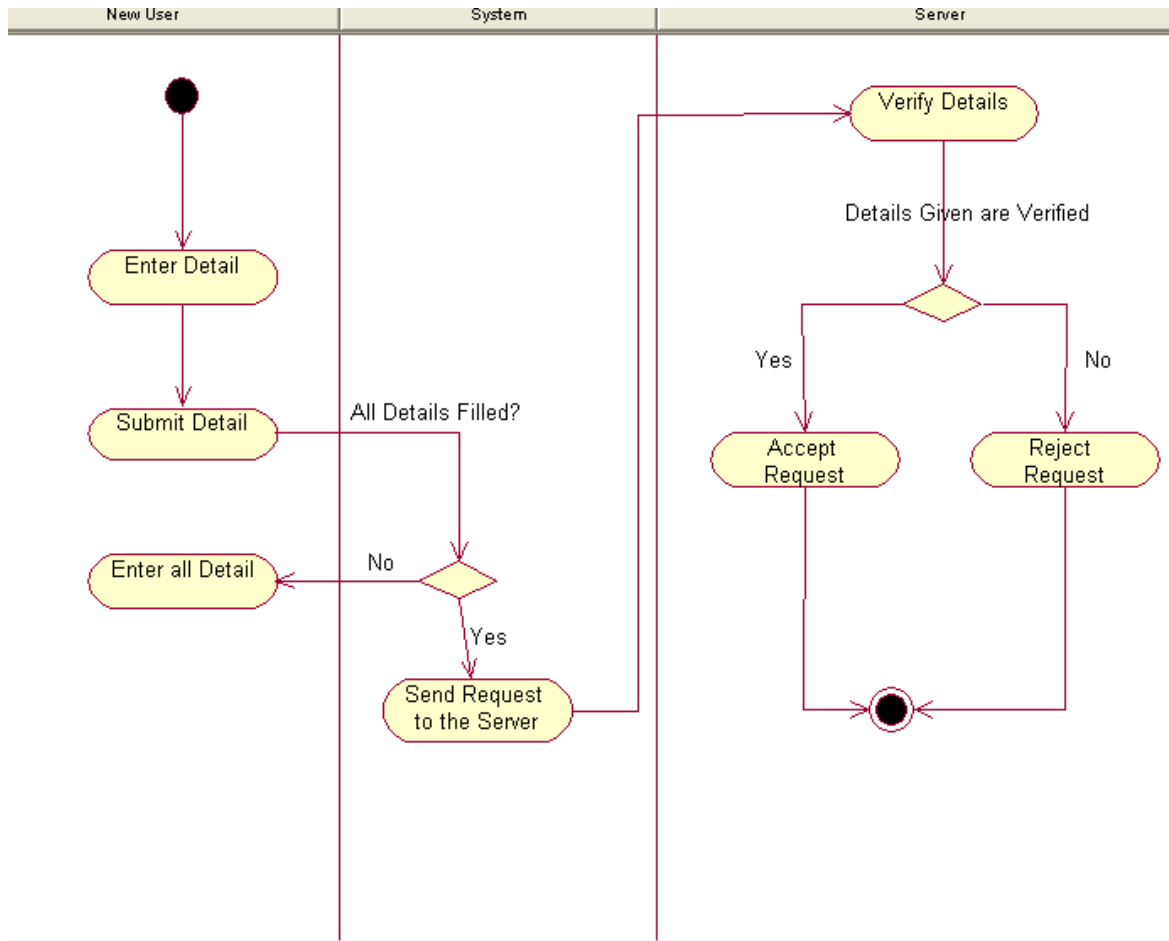


Figure 4.13: Activity Diagram of User Registration

4.4 Tools

A brief introduction of tools, which are used in designing and implementation of the Cloud Ontology Framework is given below.

4.4.1 Protégé'

Protégé' [43] is open source tool developed in Stanford University. It is an Ontology editor. Protégé supports ontology development using Ontology Web Language. Protégé' is java based tool. It helps user in:

- Creating, loading and saving both Ontology Web Language (OWL) [44] and Relational Database Framework (RDF) [45] ontologies.
- Provide functionality to develop and edit the relationships between different classes.
- Different Queries can be executed on the developed ontology.
- Provide reasoners like Fact++ for checking the Ontology Logic
- Define individuals and rules.

There are several versions of the protégé are available like Protégé 4.2 beta, Protégé 4.1 3, Protégé 3.5 beta, Protégé 3.4.8 etc. Protégé 4.2 is the latest version available.

4.4.2 MySQL

MySQL [46] is a Free, Open Source, and High Performance Database Engine. It works great with PHP. Based on SQL Standards, it is probably at least as compliant as other Databases. There are several versions of MySQL, from Version 3.19 in 1996 to Version 5.5, which is the current stable version available to download for general public under GNU licensee.

4.4.3 Web protégé'

WebProtege [47] is an open source, lightweight, web-based ontology editor. The goal behind development of web protégé is to support the web based ontology development. WebProtege uses Google Web Toolkit (GWT) for the user interface, and Protégé' for supporting ontology services.

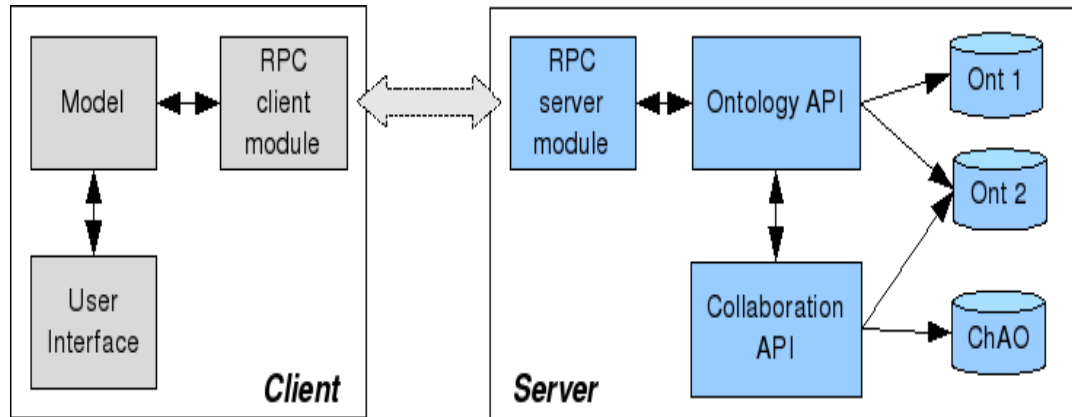


Figure 4.14: WebProtege Architecture

4.4.4 CloudSim

Clouds enable platform for dynamic and flexible application provisioning, by exposing data center's capabilities as a network of virtual services. So users can access and deploy applications from anywhere in the Internet driven by demand and QoS requirements. There is a lot of room for improvement in present cloud environments. It is difficult to perform experiments on live cloud environments because of their complexity to setup and distribute changes. Also it is possible to perform benchmarking experiments in repeatable, dependable, and scalable environment using real-world Cloud. Considering that none of the current distributed system simulators offer the environment that can be used for modeling Cloud, CloudSim present itself as only choice.

CloudSim [48] is a holistic software framework for modeling Cloud computing environments and performance testing application services. it is an extensible simulation toolkit. Described as a framework for modeling and simulation of both single and inter-networked (federation of clouds) clouds. HP and other leading organizations and also many universities around the world are using CloudSim for:

- Cloud resource provisioning,
- Energy-efficient management of data center resources,
- Optimization of Cloud computing
- Research activities.

4.4.5 Eclipse

Eclipse [49] is an open source project. A consortium of companies including IBM launched in November 2001 for the development of eclipse. Eclipse designed to help developers with specific development tasks. It is basically an Integrated Development Environment (IDE) which Provide open platform for application development tools. Eclipse runs on wide range of operating systems. Latest version of Eclipse available is Eclipse Indigo (3.7.2).

4.5 Conclusion

This chapter focused on design of the proposed framework. Also the different tools used in designing of the solution were explained in this chapter. Next chapter discusses the implementation and the experimental results.

Chapter 5

Experimental Result

This chapter focuses on the implementation of the proposed frame work in two separate parts. First part focuses on the implementation of the ontology layer of the framework and second part deals with resource allocation in Multi-Core systems. Ontology development and implementation is done using Protégé’ 3.4.6, Protégé’ 4.0 and Web Protégé’. Multi-Core resource scheduling is implemented using Cloudsim 2.0. Installation of mentioned softwares is shown in appendix part of the thesis.

5.1 Implementation

5.1.1 Ontology Implementation

Figure 5.1 and Figure 5.2 shows that how the object properties connects the different developed classes. Figure 5.1 shows “SaasCloud” relation with “HardwareMachine”, “SaasStorage” and “NaasStorage” classes. Similarly Figure 5.2 shows “High” class relations with different operating systems classes like Fedora, windows, Ubuntu.

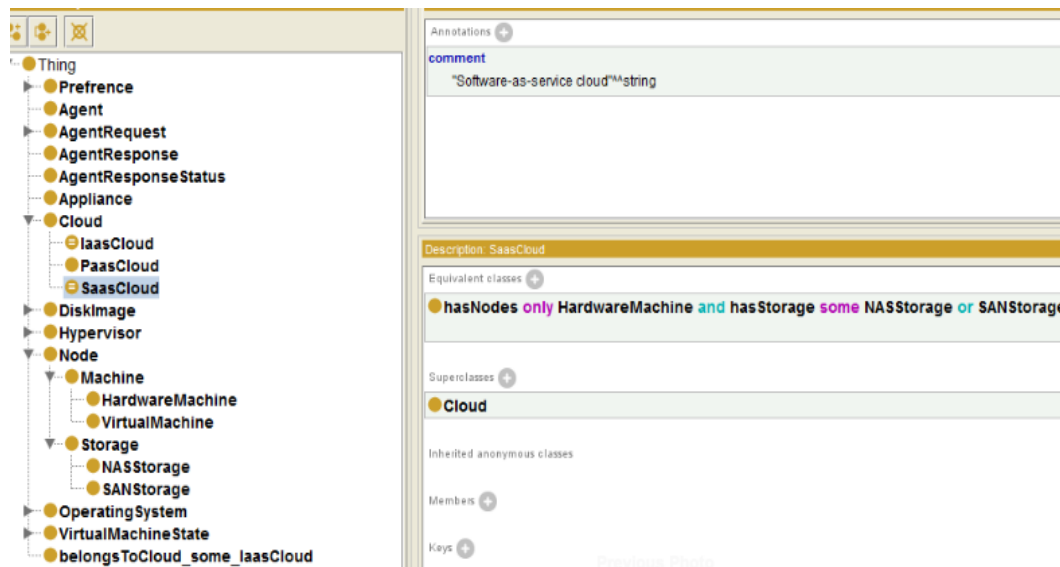


Figure 5.1: “SaasCloud” Class

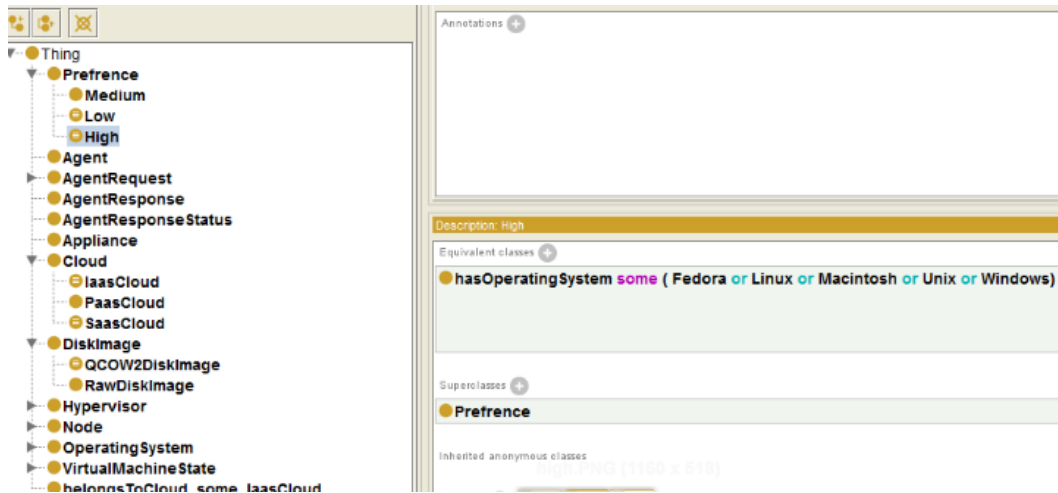


Figure 5.2: “High” Class

There can be particular individuals of a particular class. Figure 5.3 shows the “Individual” class with its instances.

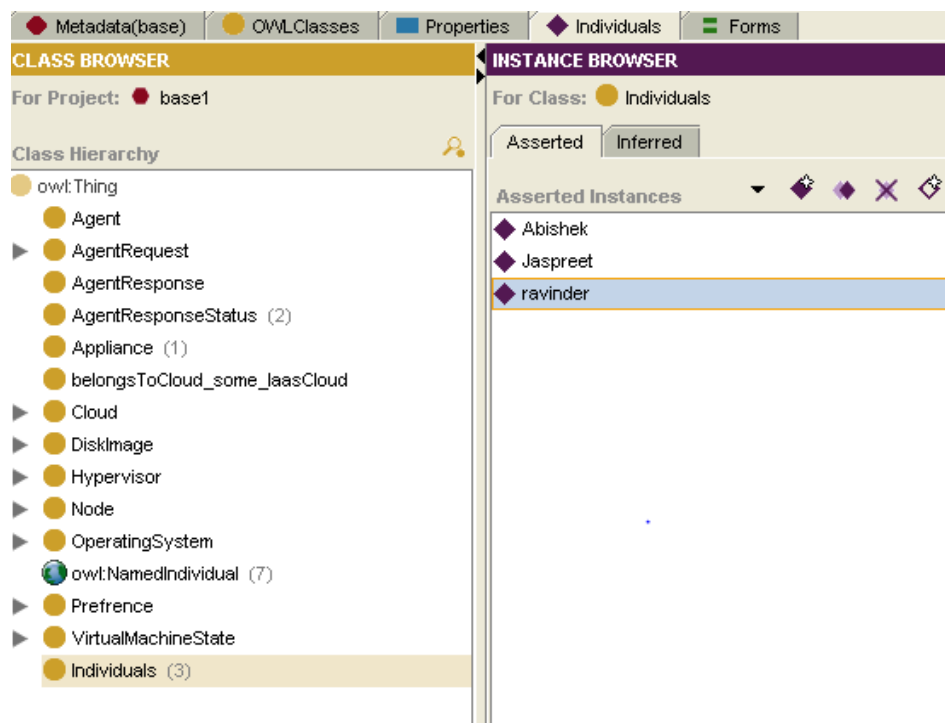


Figure 5.3: Instances of the “Individual” Class

Protégé 3.4.6 can be connected to a database (Figure 5.4) in case of the developed ontology is too big but database connection functionality is still not available in the protégé’ 4.0.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| ashish |
| cloud |
| cloud1 |
| cloud2 |
| cloud_portal |
| cloudportal |
| mysql |
| newspaper |
| performance_schema |
| periodic |
| pizza |
| protege |
| protege1 |
| protegee |
| test |
| testdb |
+-----+
7 rows in set (0.05 sec)
```

Figure 5.4: Different developed Database

As mentioned in the starting of this chapter that there is some compatibility issues between different versions of Protégé. Ontology is created in Protégé’ 4.0, but this version of protégé’ is not compatible with the Web Protégé’. So as a solution Protégé 3.4.6 is used instead of Protégé 4.0 in further implementation.

The reason behind incompatibility between protégé 4.0 and web protégé’ is the “server” functionality. Web protégé’ needs “server” to load the ontology but protégé 4.0 still doesn’t have it. Since protégé’ 3.4.6 support this functionality it is used in the proposed framework. For loading the ontologies protégé’ server needs to be starting first as shown in the Figure 5.5.

```
C:\Windows\system32\cmd.exe
/wines.pprj
Loading project base from file:/D:/ontologies/base/base1.pprj
Loading triples for: file:/D:/ontologies/base/base1.owl.owl
Completed triple loading after 55 ms
Postprocess: Process entities with incorrect Java type (0 entities) ... 0 ms
Postprocess: Process meta-classes (3 meta-classes) ... 0 ms
Postprocess: Process subclasses of rdf:List (1 classes) ... 0 ms
Postprocess: Instances with multiple types (33 instances) ... 1 ms
Postprocess: Add inferred superclasses ... 0 ms
Postprocess: Process orphan classes (41 classes) ... 1 ms
Postprocess: Generalized Concept Inclusion (0 axioms) ... 0 ms
Postprocess: Abstract classes... 0 ms
Postprocess: Domain and range of properties... 3 ms
Postprocess: Possibly typed entities (1 resources) ... 0 ms
Updating underlying frames model in 2 ms
Loading project plugins for project base
Loading project plugins for project Collaborative Newspaper Annotations
Loading project plugins for project Collaborative Pizza Annotations
Loading project plugins for project Collaborative Newspaper
Loading project plugins for project Newspaper
Loading project plugins for project Wines
Loading project plugins for project Pizza
Loading project plugins for project Collaborative Pizza
Protege server ready to accept connections...
```

Figure 5.5: Protégé’ server Starting

Above figure shows when the protégé' server loaded all available ontologies before accepting the connections, Web protégé' needs Apache Tomcat for full functioning. The steps are following:

- i. Install apache tomcat
- ii. Download web protégé.war file
- iii. Put web protégé.war file in "webapp" folder of apache tomcat installation directory.
- iv. Start apache tomcat
- v. Run localhost:8080//webprotege/WebProtege.html in web browser.

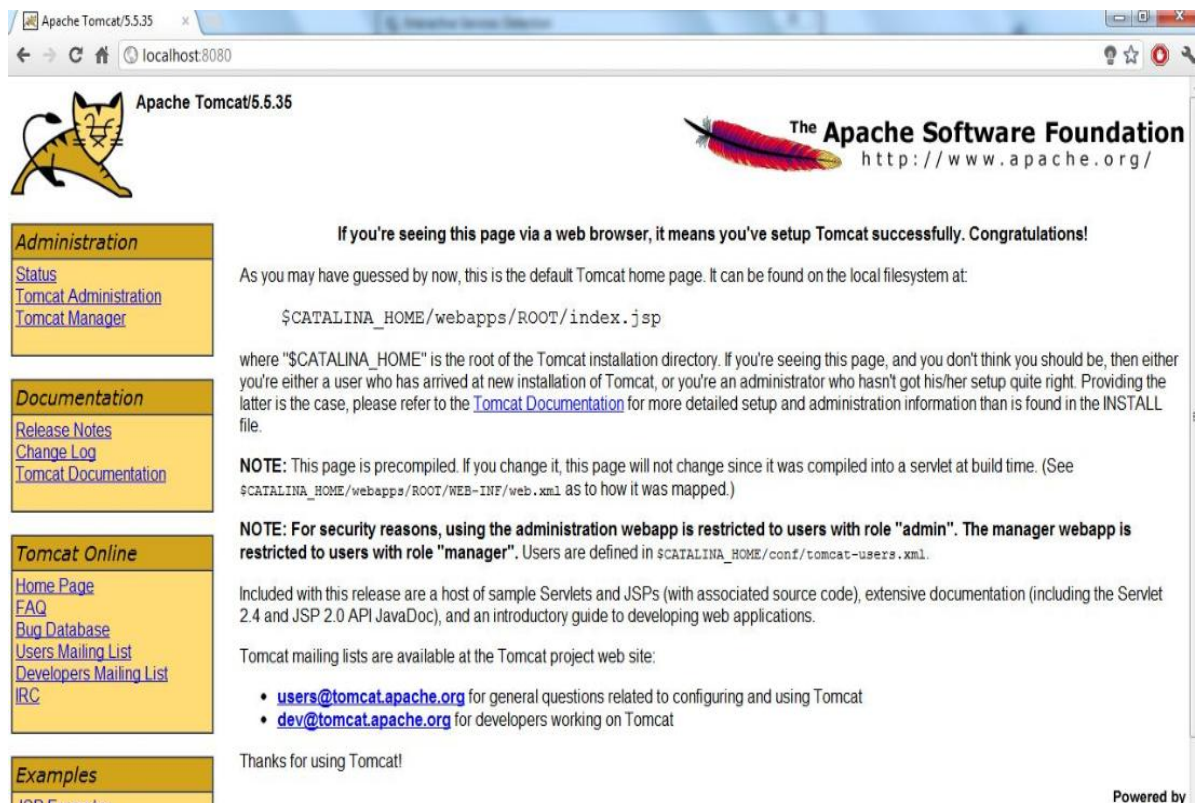


Figure 5.6: Apache Tomcat

Web protégé' shows all the ontologies loaded by the protégé' server and it also shows the name of their owners (Figure 5.7).

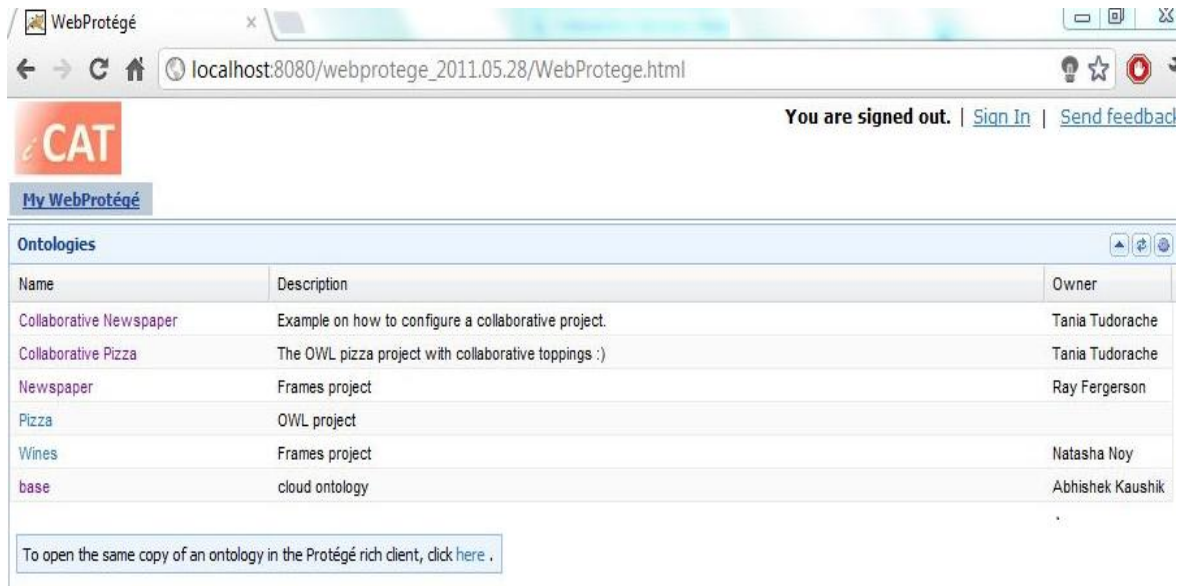


Figure 5.7: Web Protégé' Ontologies

There are several ontologies present in the web protégé' like collaborative newspaper, pizza, and wines etc. which are inbuilt in the protégé. The last ontology in web protégé' diagram in "base". This is the cloud ontology proposed in this thesis.

Figure 5.8 shows the loading of the "base" ontology in web protégé'.

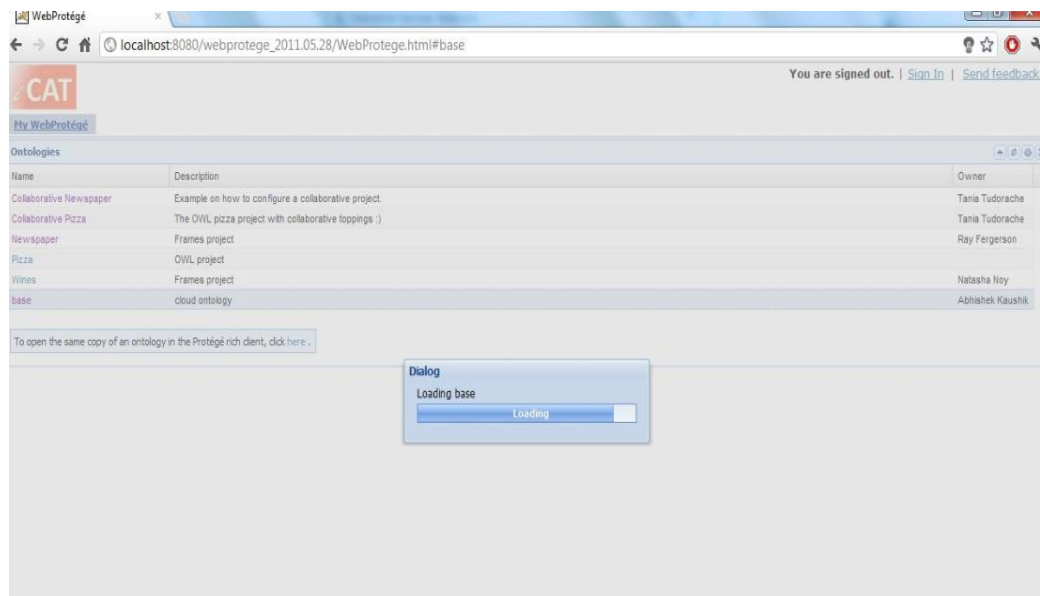


Figure 5.8: Base Ontology Loading

Figure 5.9 shows the full view of the developed cloud ontology in web protégé’.

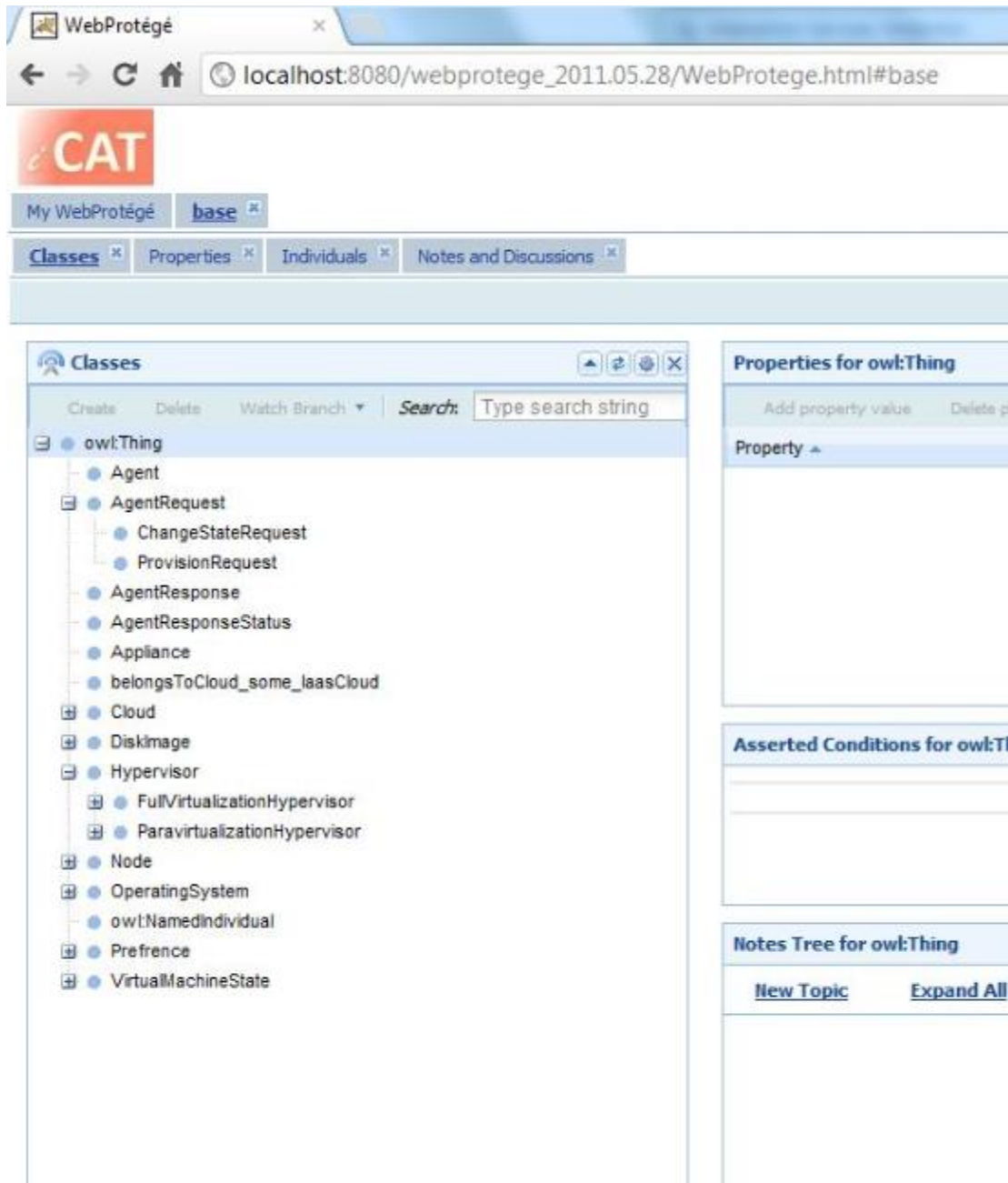


Figure 5.9: Base Ontology in Web Protégé’

The purpose of using Web Protégé’ for the implementation of the developed ontology is to show the web based development and use of the ontology for customer purpose. The implementation shown in this thesis is a standalone implementation of the Web Protégé’.

Remote host of implementation is also possible using Web Protégé'. Still Web Protégé' is in its developing phase, there are lots of functionalities that are required and that can be added in the Web Protégé'. Web Protégé' used in this thesis is not compatible with Protégé' version 4.0 which is the latest version of the Protégé' family. Therefore Protégé' 3.4.6 is used with the Web Protégé'. Also the query tab is still not available in Web Protégé' but the work is in progress. To provide the query tab functionality in Web Protégé' there is an option of Protégé Rich Client is available, as shown in the Figure 5.10 with black rectangle.

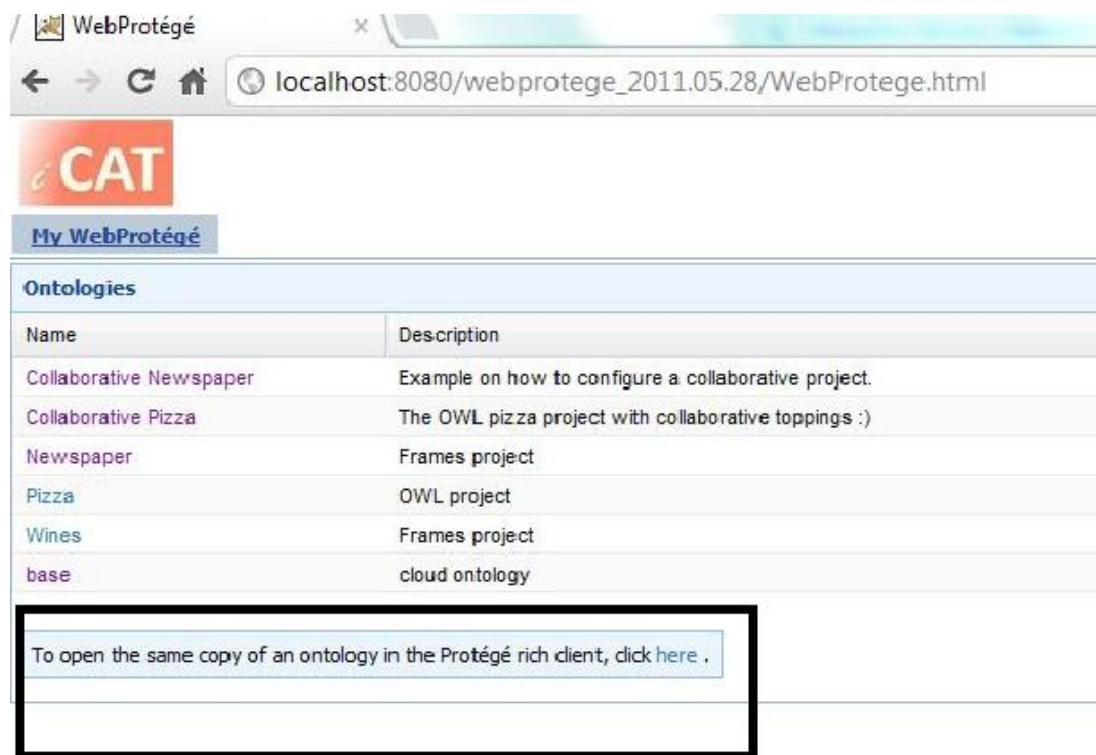


Figure 5.10: Protégé' rich Client

When Protégé' rich client is selected "JNLP" executes. Java Network Launch Protocol (JNLP) is an XML-based protocol that can be used to deploy Java and JavaFX applications on the Internet. As mentioned in the tools part of the chapter four that protégé' is a java based application. So with help of the JNLP, protégé' 3.4.6 starts.

Protégé 3.4.6 gives the option for editing the available ontologies and to execute the SPARQL queries (Figure 5.11).

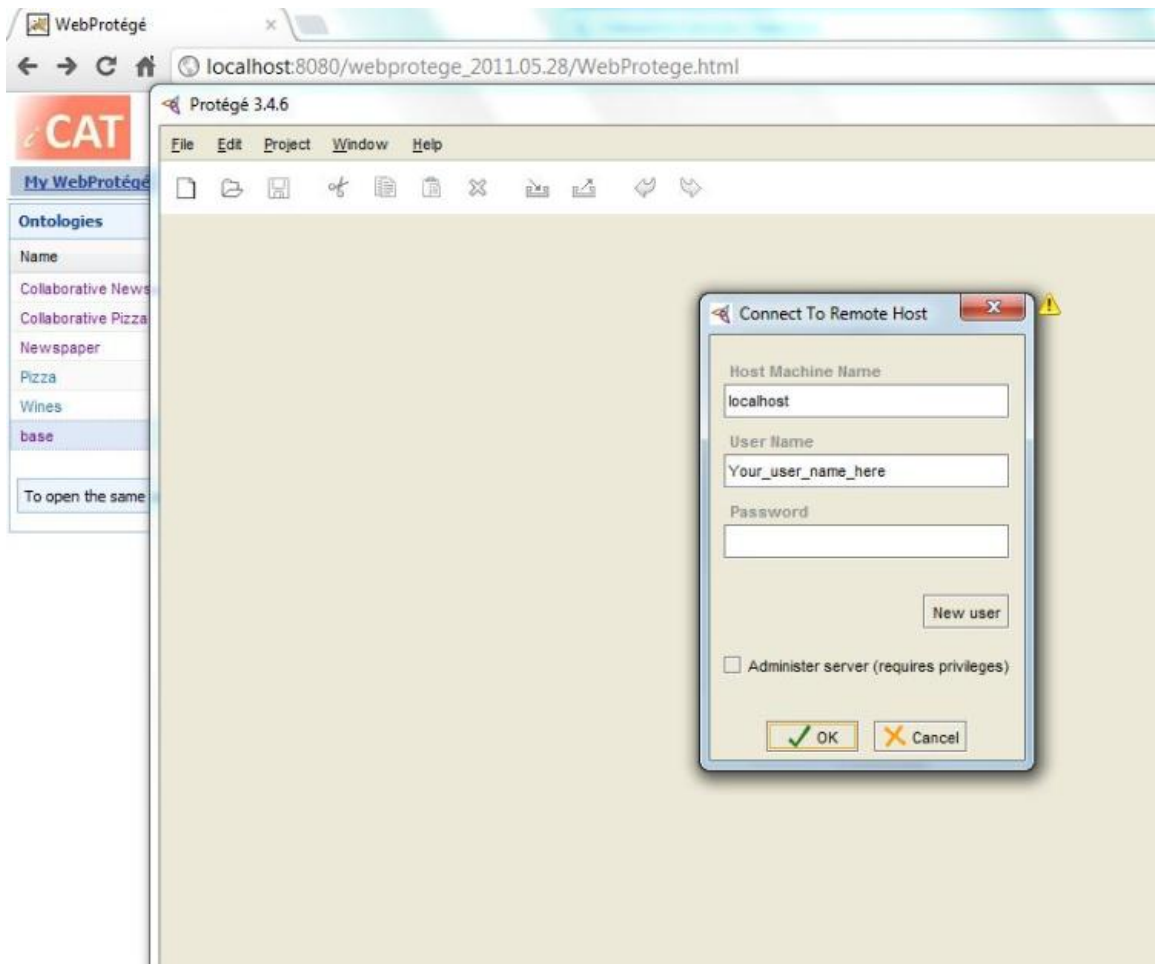


Figure 5.11: Protégé' 3.4.6 opened with help of JNLP using Web Protégé.

There is one problem with the Protégé' 3.4.6 that it only supports the SPARQL queries. The main purpose behind developing this ontology is to provide semantic based queries. Such queries that seem equivalent with natural language like English.

This is the reason why Protégé 4.0 is used to show the query part. The first query is shown in the "SaaS cloud" class whose relations with other classes are already shown in Figure 5.1 for making the query more understandable.

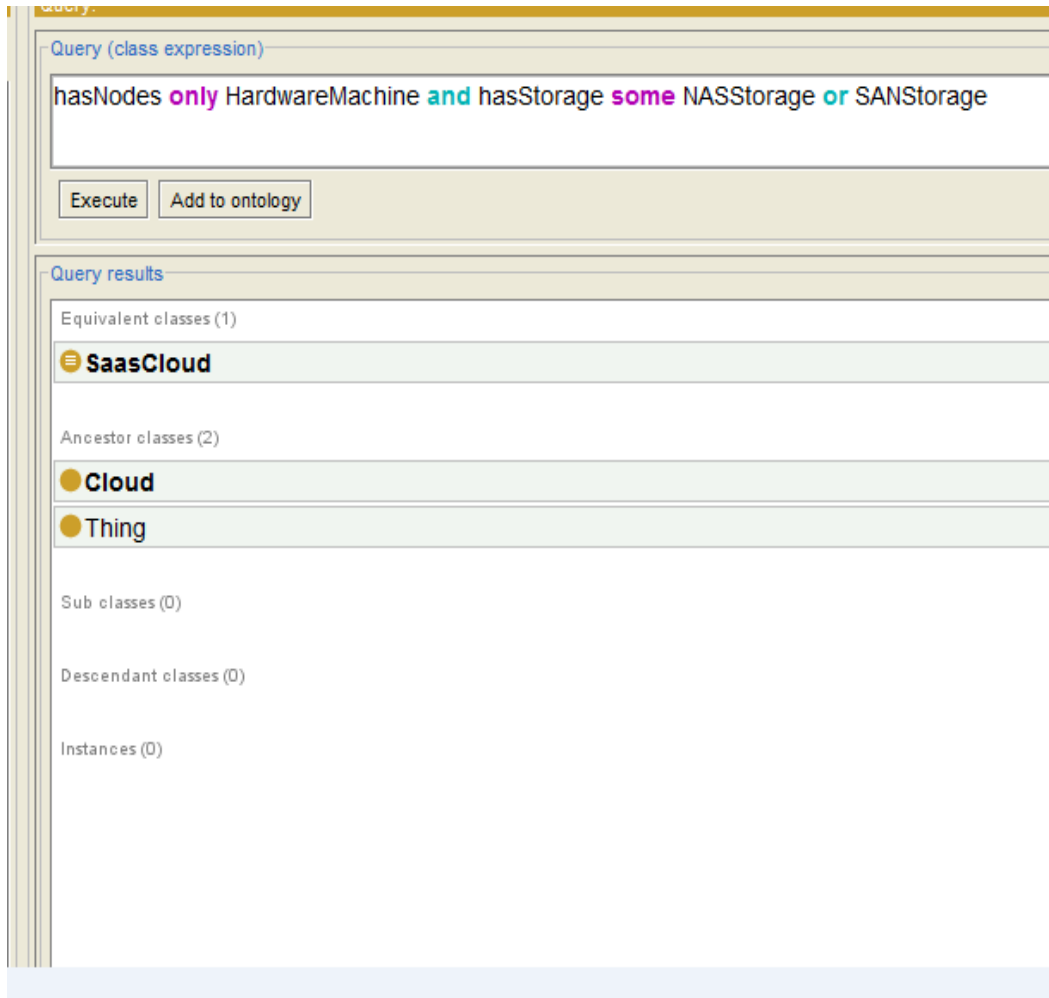


Figure 5.12: Saas Cloud Class Query

The query shown in Figure 5.12 is:

hasNodes only HardwareMachine and hasStorage some NASStorage or SANStorage.

It is similar to the natural language query:

Cloud type that has nodes only Hardware Machine and has storage either NAS storage or SAN storage or both.

The answer is **Saas Cloud class** because of the pre defined relation of this class with classes mentioned in the query.

Similarly there is this second query (Figure 5.13) performed on “High” class whose relations with other classes are already shown in the Figure 5.2.

The screenshot shows a query interface with a yellow header bar labeled "Query:". Below it, a text box contains the query: "hasOperatingSystem some (Fedora or Linux or Macintosh or Unix or Windows)". Below the text box are two buttons: "Execute" and "Add to ontology".

Below the query section is a "Query results" section. It contains several categories of results:

- Equivalent classes (1):** A single result, "High", with a hamburger menu icon to its left.
- Ancestor classes (1):** A single result, "Thing", with a yellow circle icon to its left.
- Sub classes (1):** A single result, "Low", with a hamburger menu icon to its left.
- Descendant classes (1):** A single result, "Low", with a yellow circle icon to its left.
- Instances (1):** A single result, "DummyUbuntuIntrepidIbexAppliance", with a purple diamond icon to its left.

At the bottom of the results section, there is a "Play Slideshow" button.

Figure 5.13: High Class Query

Figure 5.13 shows the query and the result. The query is

hasOperatingSystem some (Fedora or Linux or Macintosh or Unix or Windows).

It can be read as:

Preference which has operating system either Fedora or Linux or Macintosh or Unix or Windows or all of them.

The result is “High” class because this is the only class that allowed using all the operating system.

5.1.2 IaaS Cloud portal

For implementing the proposed technique a IaaS cloud portal is designed in java in association with Cloudsim 2.0. MySQL is used for the backend database. Figure 5.14 shows the Login form of the portal.



Figure 5.14: Home Page of the IaaS Cloud Portal

There are three different buttons are available on the portal home page. These are

- First button is login button, if user already has the login id and password.
- Second button is cancel button for canceling the login process.
- Third button is for that user who doesn't already have login id and password and need a registration.

Page shown in Figure 5.15 will appear for the following three conditions.

- If users leave both User Name and Password filed empty.
- If user enter wrong User Name or Password
- If User enter information in only one field.

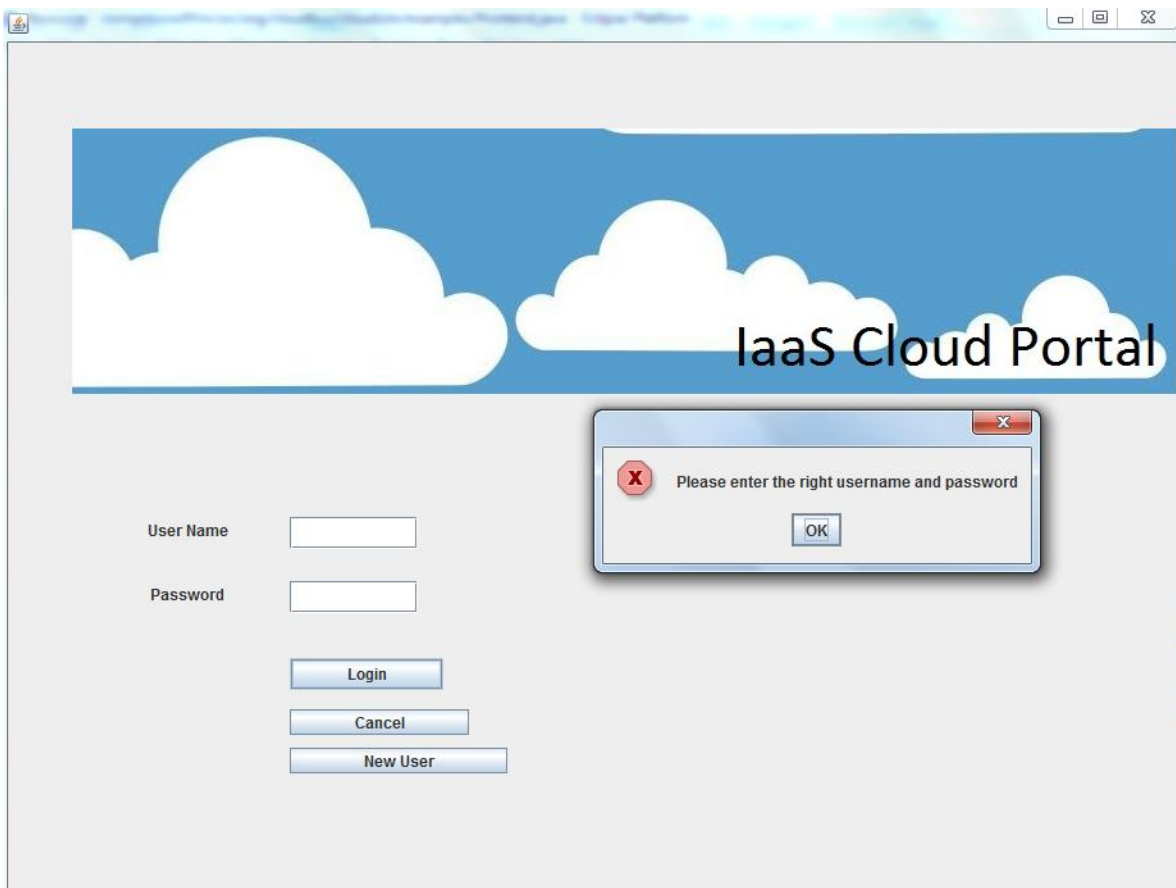


Figure 5.15: IaaS cloud Portal Login Error Page

When new users try to use the system she/he must register first. There is new user button on the home page of the cloud portal. Page shown in Figure 5.16 appear for the new user once he/she chose the New User button. This page is Registration Form. There are several fields shown on this page like First Name, Last Name, Email id, Confirm Email ID, Password, Address and Contact Number.

The image shows a registration form window with the following fields and buttons:

- First Name*
- Last name*
- Email Id*
- Confirm Email Id*
- Password*
- Address*
- Contact NO.*
- Save
- Clear

Figure 5.16: Registration Form

All the fields shown in Figure 5.16 have an asterisk with their name. The purpose of the asterisk is to show that every field is mandatory to fill. None of the field must be left unfilled. If so, then registration will fail. So, for the successful registration new user must provide all the required information.

Followed by the fields there are two buttons. One is save button and another is clear. New user must press save button after filling all the fields, so that his/her information will be stored in database successfully. Second button is clear button. If new user filled some wrong information or need to change the information he/she can use clear button to erase the filled data and start fresh again.

As for the case when new user leaved any of the required filled empty, pages shown in Figure 5.17 and Figure 5.18 will appear.

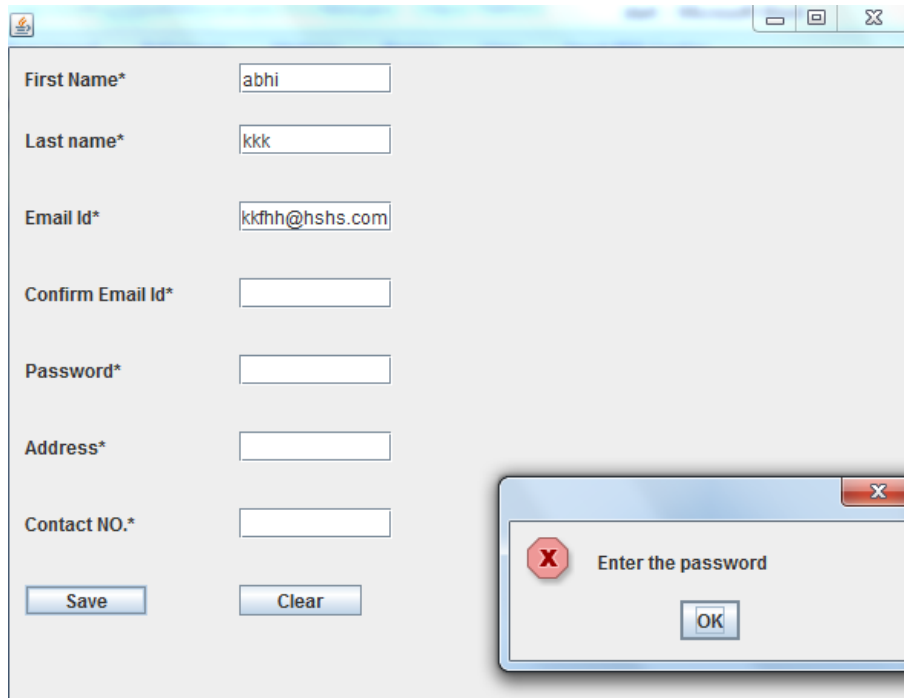


Figure 5.17: Password Filed Empty Error

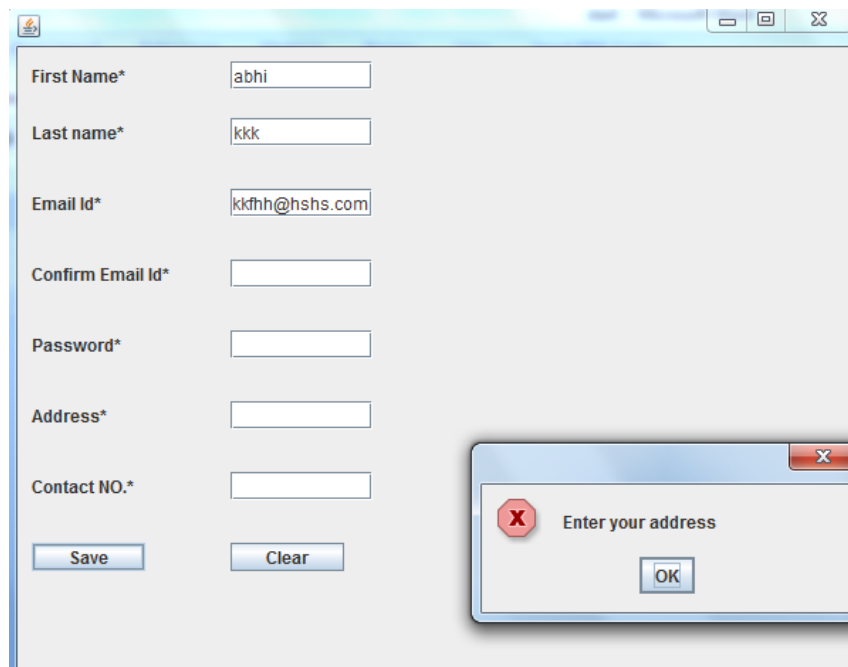


Figure 5.18 – Address Field Empty Error

Figure 5.19 shows the different database created in the process. The database used for this cloud portal is Cloud_Portal circled with red outline.

```
mysql> create database cloud_portal;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| ashish |
| cloud |
| cloud1 |
| cloud2 |
| cloud_portal |
| cloudportal |
| mysql |
| newspaper |
| performance_schema |
| periodic |
| pizza |
| protege |
| protege1 |
| protegee |
| test |
| testdb |
+-----+
17 rows in set (0.05 sec)

mysql>
```

Figure 5.19: Cloud_Portal Database

Figure 5.20 shows the login_detail table created in the cloud_portal database.

```
mysql> create table login_detail
-> (
-> first_name VARCHAR(60),
-> last_name VARCHAR(60),
-> email VARCHAR(60),
-> password VARCHAR(60),
-> address VARCHAR(200),
-> phone_number INT
-> )
-> ;
Query OK, 0 rows affected (0.13 sec)

mysql>
```

Figure 5.20: Login_detail table in Cloud_portal database

Figure 5.21 shows the different users information stored in cloud portal database.

```
MySQL Command Line Client
mysql> use cloud_portal;
Database changed
mysql> show tables;
+-----+
| Tables_in_cloud_portal |
+-----+
| login_detail            |
+-----+
1 row in set (0.02 sec)

mysql> select * from login_detail;
+-----+-----+-----+-----+-----+-----+
| first_name | last_name | email | password | address | phone_number |
+-----+-----+-----+-----+-----+-----+
| a          | a        | a     | a        | a        | 98898080    |
| b          | b        | b     | b        | b        | 8888        |
| ashish    | narang   | jgjfj | ashish   | hhh     | 99887755    |
| ashish    | narang   | jgjfj | ashish   | hhh     | 99887755    |
| c          | c        | c     | c        | c        | 88888      |
| f          | f        | f     | f        | f        | 7889       |
| jassi     | j        | j     | jassi    | j        | 779808     |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

Figure 5.21: Users detail in Cloud Portal database.

Once user successfully registered and logged in the system page shown in figure 5.22 will appear.

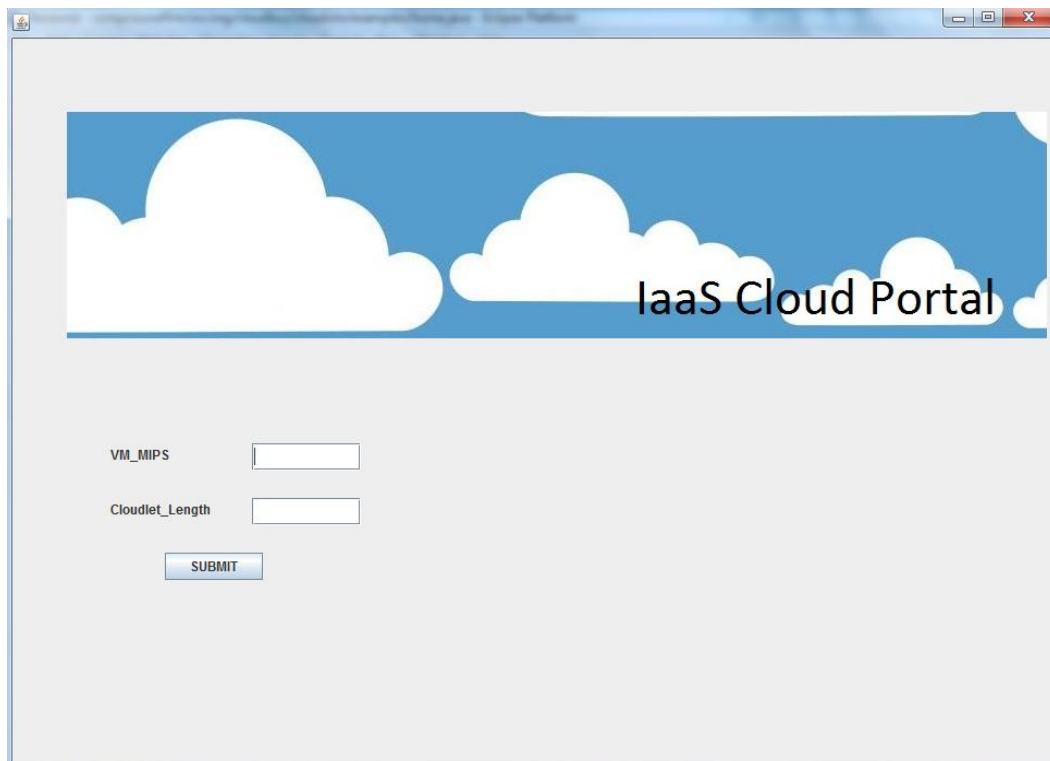


Figure 5.22: VM's mips and Cloudlet's length page

Page shown in the Figure 5.22 has two fields which are explained below:

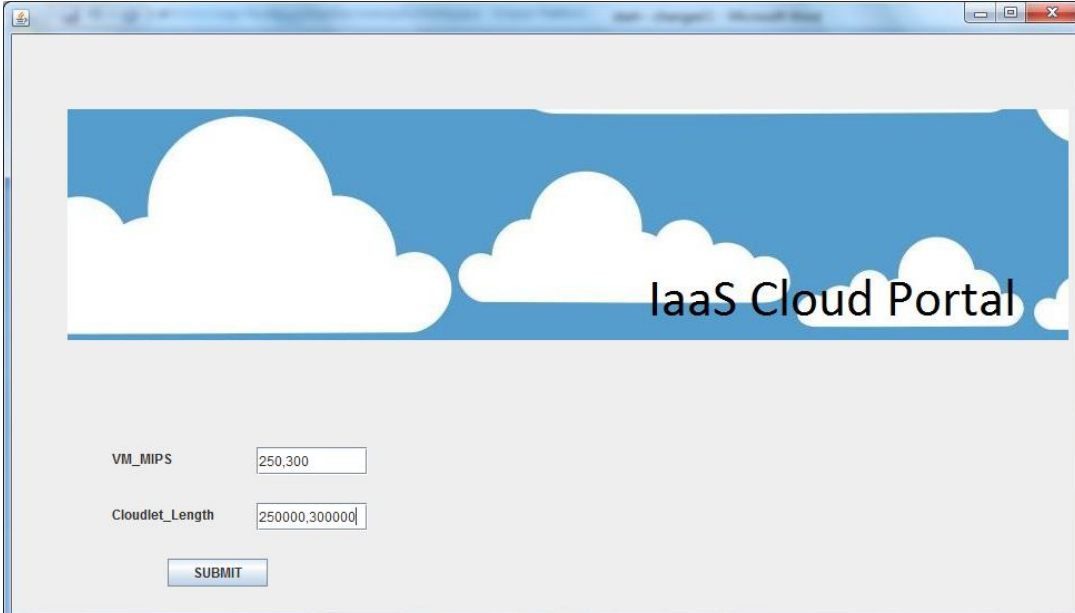
- VM_MIPS – This field is for providing the Machine Instruction Per Cycle for the desired virtual machines by user. For the simplicity, only two virtual machines mips can be given here separated by a comma in the same tab.
- Cloudlet_Length – This field is for defining the length of the cloudlets. Similarly as VM_MIPS only two cloudlets length can be provided at a time separated by comma in the same tab.

After filling these two fields, user must press the submit button for the final execution.

Two different examples are presented here:

1. Example 1

Figure 5.23 shows VM_MIPS and Cloudlet_Length fields with the data.



The screenshot shows a web browser window titled "IaaS Cloud Portal". The page features a header with a blue background and white clouds. Below the header, there are two input fields: "VM_MIPS" with the value "250,300" and "Cloudlet_Length" with the value "250000,300000". A "SUBMIT" button is located below the input fields.

Figure 5.23: VM_MIPS and Cloudlet_Length fields with data.

After executing this given data, cloudsim the cloud simulator provides the execution summary shown in Figure 5.24.

```

Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: VM #0 has been created in Datacenter #2, Host #0
0.0: Broker: VM #1 has been created in Datacenter #2, Host #0
0.0: Broker: Sending cloudlet 0 to VM #0
0.0: Broker: Sending cloudlet 1 to VM #1
1000.0: Broker: Cloudlet 0 received
1000.0: Broker: Cloudlet 1 received
1000.0: Broker: All Cloudlets executed. Finishing...
1000.0: Broker: Destroying VM #0
1000.0: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
    0         SUCCESS       2           0     1000         0         1000
    1         SUCCESS       2           1     1000         0         1000
-----

```

Figure 5.24: Execution terminal of Cloudsim

As according to the proposed technique in this thesis, the cloudlet with more length will be assigned to the virtual machine with higher number of machine instruction per cycle (mips). Figure 5.24 shows that cloudlet 0 whose length is 250000 assigned to virtual machine whose mips is 250 and cloudlet 1 whose length is 300000 is assigned to virtual machine whose mips is 300.

2. Example 2

Similarly there is a second example to further analyze the proposed technique. Figure 5.25 shows a different dataset than Figure 5.23. This time the machine instruction per cycle for first virtual machine is 400 and second virtual machine has 300 machine instructions per cycle. As for the cloudlet the lengths remain same as 250000 for first cloudlet and length is 300000 for second cloudlet.

As per the given data it is shown in Figure 5.26 that cloudlet 0 with length is 250000, assigned to the virtual machine who's mips is 300 and the cloudlet 1 with length 300000 is assigned to the virtual machine with mips 400.

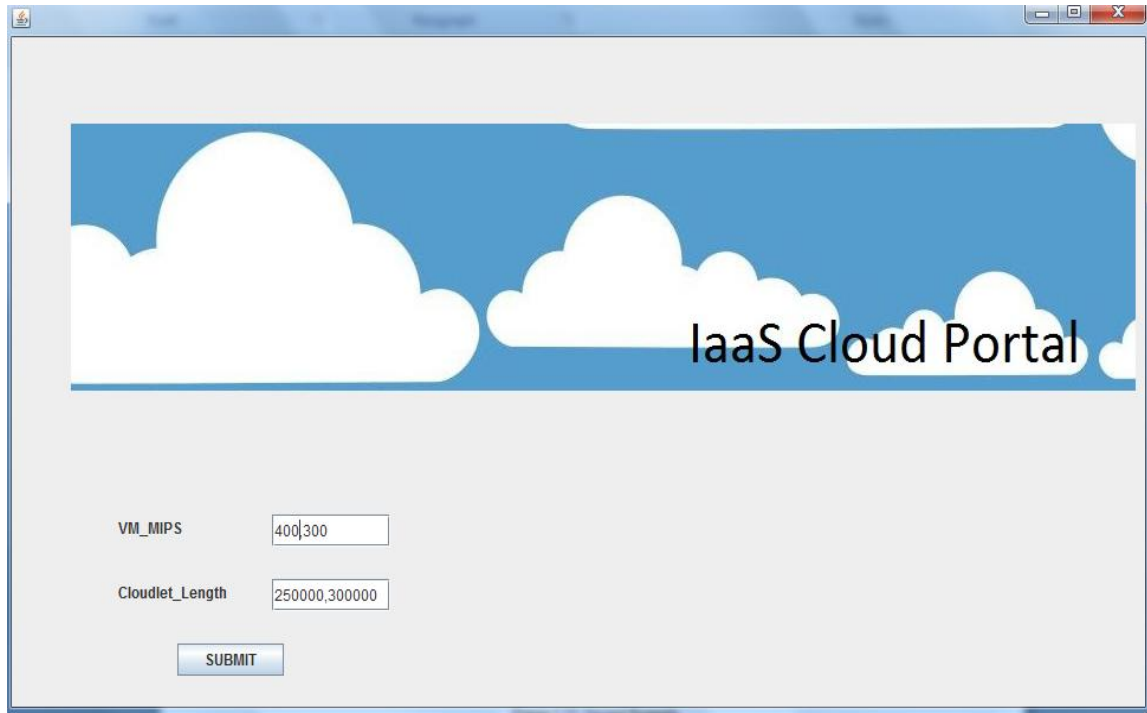


Figure 5.25: Example 2

Proposed technique is compared with the First Come First Come (FCFS) technique of the cloudsims simulator

```

Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: VM #0 has been created in Datacenter #2, Host #0
0.0: Broker: VM #1 has been created in Datacenter #2, Host #0
0.0: Broker: Sending cloudlet 0 to VM #1
0.0: Broker: Sending cloudlet 1 to VM #0
750.0: Broker: Cloudlet 1 received
833.3333333333334: Broker: Cloudlet 0 received
833.3333333333334: Broker: All Cloudlets executed. Finishing...
833.3333333333334: Broker: Destroying VM #0
833.3333333333334: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM ID  Time  Start Time  Finish Time
1           SUCCESS    2              0      750      0           750
0           SUCCESS    2              1     833.33      0           833.33

```

Figure 5.26: Example 2 execution

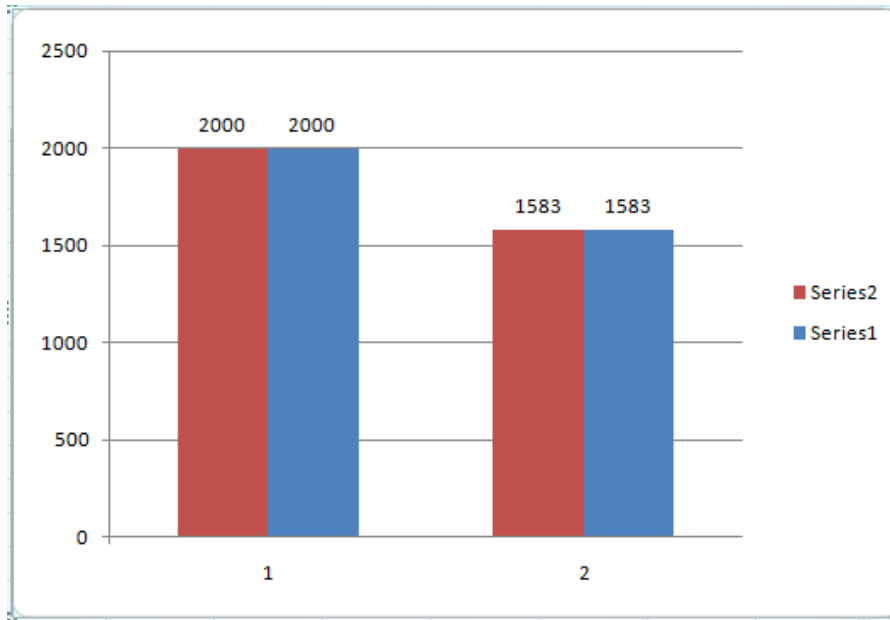


Figure 5.27: Ideal Scenario

Figure 5.27 shows an ideal scenario similar to one show in Figure 5.23. In such case both the techniques proposed and the FCFS shows same total time of execution. Figure 5.27 shows two cases, in first both the techniques give total execution time 2000 and in another 1583 for the same configuration. Red line represents FCFS and Blue Line represents proposed technique.

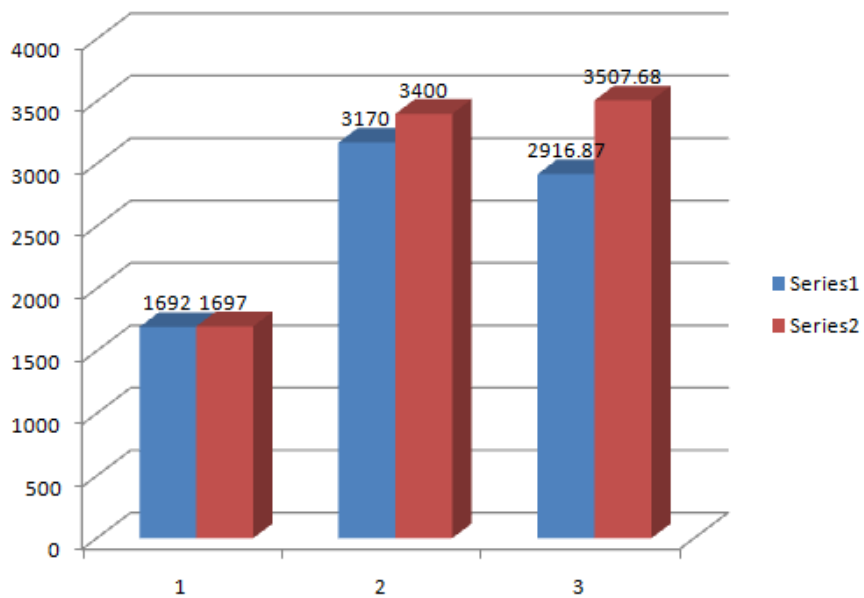


Figure 5.28: General scenario

An ideal scenario is being presented which does not occur very often. There can be a shuffling of data and in that case performance of proposed technique is far better than FCFS (Figure 5.28).

Figure 5.28 shows a general scenario when requests reach in a random order. It is clear from the graph that in all the three cases that total execution time in case, of FCFS is more expensive than the proposed scheme in terms of total execution time. Blue line represents proposed technique and Red line represents FCFS.

5.2 Conclusion

This chapter presented the implementation of the both layers of the framework and the comparison of the proposed technique with the First Come First Serve technique for the resource scheduling in multi-core systems. Next chapter will discuss the conclusion and the future scope.

Conclusion and Future Scope

This chapter concludes the work presented in this thesis. At the end of this chapter, some future directions have been proposed which can be considered to improve the Frame Work proposed in this thesis.

6.1 Conclusion

Cloud computing and its important characteristics have been discussed in this work. This thesis focuses on the interoperability and standardization challenges that cloud computing is facing today. Several cloud interoperability standards are compared with respect to the ontology as an answer for the standardization. This thesis also gives an insight about the problem of using multi-core processors in cloud computing. Drawbacks of the single core processors, need of the multi-core processors, virtualization's types and importance are also discussed. Also current resource scheduling techniques in multi-core processors are compared with respect to the cloud computing. Further, an ontology based cloud frame work is proposed and implemented in this work.

6.2 Thesis Contribution

- a) In this thesis, currently available interoperability cloud standards like CCFI, OVF and UCI are discussed.
- b) Currently available multi-core resource scheduling techniques like ULAS, FVM, FTDS and Task Aware Scheduling are discussed and compared.
- c) An Ontology based Cloud Framework is proposed and designed.
- d) Ontology layer of the framework is implemented using Protégé.
- e) A IaaS cloud portal is designed and implemented using cloudsim for the hardware layer of the proposed framework.

6.3 Future Scope

- a) Cloud Ontology developed in this thesis can be extended further to include maximum functionalities of live cloud.
- b) More comparison factors can be included in cloud portal for the much better resource allocation.
- c) Further functionalities can be included in Web Protégé to decrease its dependency on the standalone protégé' versions.
- d) Standardization and Multi-Core Resource Scheduling parts of the proposed framework can be combined to give a unified solution.

References

- [1] Cloud Computing [online] Available: <http://www.ibm.com/cloud-computing/us/en/>.
- [2] Schubert, L., The Future of Cloud Computing Opportunities for European Countries, pp8.
- [3] Saas, Paas and Iaas [online] Available: http://www.nec.co.th/asp/Sol_contact_sol_service_delivery.asp.
- [4] Chou, W., “Web Services: Software-as-a-Service (SaaS), Communication”, IEEE International Conference on Web Services, pp 1, 23-26 Sept. 2008.
- [5] “PAAS,” Available: www.salesforce.com/paas.
- [6] Azure [online]. Available: www.microsoft.com/azure/data.msp.
- [7] Google app engine [online]. Available: code.google.com/appengine.
- [8] Wiedemann, D. , “Organizational Determinants of Corporate IaaS Usage”, in proceedings of IEEE 13th Conference on Commerce and Enterprise Computing (CEC), Proventa AG, Frankfurt, Germany, pp 191-196, 5-7 Sept. 2011.
- [9] Peter, M., Grance, T., “The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, vol. 53 , issue 6 ,pp 50, 2009.
- [10] Borenstein, N. and Blake, J., ”Cloud Computing Standards: Where's the Beef?”, Internet Computing, IEEE, vol 15, issue 3, pp 74-78, May-June 2012
- [11] Video Transcript, “Excerpts from a Conversation with Gordon Moore: Moore’s Law”, Intel Corporation, 2005.
- [12] Moores Diagram [online] Available at: <http://www.creopoint.com/profiles/blogs/>
- [13] Peng et al, “Memory Performance and Scalability of Intel’s and AMD’s Dual-Core Processors: A Case Study”, IEEE, 2007.
- [14] Bahulkar, K., Hofmann, N., Jagtap, D., Abu-Ghazaleh, N., Ponomarev, D. , “Performance Evaluation of PDES on Multi-core Clusters”, in Proceedings of 2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Fairfax, VA , pp 131-140, 17-20 Oct. 2010.

- [15] Julian, B., Chenguang , X., Gurumurthi, S., “Understanding Performance Issues on both Single Core and Multi-core Architecture”, Department of Computer Science. University of Virginia. Charlottesville, 2007.
- [16] Cloud API Propagation and the Race to Zero (Cloud Interoperability) [online] Available at: [http://www .elasticvapor.com/2009/01/cloud-APIpropagation- and-race-to-zero.html](http://www.elasticvapor.com/2009/01/cloud-APIpropagation-and-race-to-zero.html).
- [17] Open Grid Forum. (2010) Open cloud computing interface core & models. . [Online]. Available at: <http://forge.ogf.org/sf/wiki/ do/viewPage/projects.occi-wg/wiki/CoreAndModels>.
- [18] Unified Cloud Interface [online] Available at: <http://code.google.com/p/unifiedcloud/>.
- [19] Open Virtualization Format Specification, Distributed Management Task Force, Jan. 12, 2010, dSP0243 Version 1.1.0.
- [20] Youseff, L., B9utrico M. and Silva, D.D., “Towards United Ontology of Cloud Computing” Grid Computing Environments Workshop. GCE’08. pp 1-10.
- [21] Ma, Y.B., Jang S. and Lee, J.S., “Ontology-Based Resource Management for Cloud Computing” In proceedings of ACIIDS(2). PP 343-352.
- [22] Takahashi, T., Kadobayashi, Y. and Fujiwara, H.,” Ontological Approach toward Cybersecurity in Cloud Computing” In proceeding SIN '10 Proceedings of the 3rd international conference on Security of information and networks, ACM, 2010, New York, NY, USA.
- [23] Han, T. and Sim, K. M. 2010,” An Ontology-enhanced Cloud Service Discovery System” In proceedings of International MultiConference of Engineers and Computer Scientists vol. 1. IMECS 2010, March 17-19, 2010, Hong Kong.
- [24] Feel, H. T., Khafagy, M. H., “OCSS: Ontology Cloud Storage System” First International Symposium on Network Cloud Computing and Applications (NCCA). pp 9-13, 2011.
- [25] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I.,War eld, A, “ Xen and the art of virtualization.” In: SOSP '03:

- Proceedings of the 19th ACM Symposium on Operating Systems Principles, New York, NY, USA, ACM, pp.164-177, 2003.
- [26] Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I. "Cloud computing an emerging IT platforms", Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25(6), pp. 599-616, June 2009.
- [27] K., Adams and O., Agesen, "A Comparison of Software and Hardware Techniques for x86 Virtualization. ", Proceedings of ASPLOS, San Jose, CA, USA, pp. 12, October 2006.
- [28] Ken, C.A., "Cache Coherence in Distributed Systems" Western Research Laboratory, ,pp 79, December 1987.
- [29] M. Creeger, "Multicore CPUs for the Masses", *QUEUE*, ACM, NY, USA, vol. 3, issue 7, September 2005.
- [30] Hardware Virtualization [online]. Available: <http://searchvmware.techtarget.com/definition/hardware-virtualization>.
- [31] Zhiyuan, S., Jian, H., Hai, J., Kan, H., "XenMVM: in proceedings of Exploring Potential Performance of Multi-Core System in Virtual Machine Environmen"t, *Ubiquitous Information Technologies and Applications (CUTE)*, 2010 Proceedings of the 5th International Conference on , December 2010.
- [32] Jian, H., Hai, J., Kan H., Zhiyuan, S.," Adjusting Virtual Computing Resources in Threshing Cases "in Proceedings of Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on Real Time Systems, Pisa, April 2010 .
- [33] Ortega , R., Kelly R., "PID Self-Tuners: Some Theoretical and Practical Aspects", *IEEE Transactions on Industrial Electronics*, Nov. 1984.
- [34] Zhang, Y., Bestavros, A., Guirguis, M., Matta, I. and West, R. , "Friendly Virtual Machines: Leveraging a Feedback-Control Model for Application Adaptation", Proceedings of the 1st ACM/USENIX Conference on Virtual Execution Environments (VEE'05), Chicago, Illinois, pp.2-12, June 2005.
- [35] EWMA. Avialable [online]: http://www.bionicturtle.com/how-to/article/exponentially_weighted_moving_average_ewma/.

- [36] Yong, P., Dongkun S., "Task-aware Virtual Machine Scheduling for Multi-core Real-time Systems" School of Information and Communication Engineering, Sungkyunkwan University, Korea, June 2011.
- [37] Gruber, T.R.. Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies vol.43, pp 907-928, 1995.
- [38] Semantic Web Available at: <http://www.w3.org/standards/semanticweb/>.
- [39] Patel, P., Ranabahu, A. and Sheth, A., "Service Level Agreement in Cloud Computing" Wright State University , Sept. 5, 2009.
- [40] Horridge, M., Drummond, N., Goodwin, J., Rector, A., Robert Stevens, R. and Wang, H.H. , "The Manchester OWL Syntax" CEUR-WS.org, vol. 216, pp 10-11, 2006.
- [41] SPARQL [online] Available at: <http://www.w3.org/TR/rdf-spa\rql-query/>
- [42] Zhihong, Z., Mingtian, Z., "Web Ontology Language OWL and its Description Logic Foundation", Proceedings of The Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT, pp 57-160, 2003.
- [43] Protégé [online] Available at: <http://protege.stanford.edu/>.
- [44] Aref, M.M. and Zhou, Z. 2005. The Ontology Web Language (OWL) For A Multi-Agent Understating system.In proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems. PP 586-591.
- [45] Ying,D. 2005. The on data semantic description framework using RDF/XML, First International Conference on Semantics, Knowledge and Grid. SKG '05(Nov.-2005) . pp 138-138, 27-29.
- [46] MySQL [online] Available at: <http://www.mysql.com/>.
- [47] Tudorache, T., Vendetti, J. and F Noy N. 2008. Web-Protégé' A Lightweight OWL Ontology Editor for the Web. CEUR-WS.org vol. 8 issue. 10. PP 569.
- [48] Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C. and Buyya, R., "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, issue. 1, pp. 23–50, 2011.

[49] Eclipse [online] Available at: <http://www.eclipse.org/>.

List of Papers

Published

- [1] Kaushik, A. and Chana, I., "Multi-Core: A cloud Perspective," International Conference on Recent Trends in Computing, Mechatronics and Communication, at Om Institute of Technology and Management, Hisar, page no. 417 - 420, ISBN 978-81-923446-0-7, February, 2012.

Accepted

- [2] Kaushik, A. and Chana, I., "Ontology Based Cloud Framework", Third International Conference on Advanced Computing and Communication Technologies for High Performance Applications, FISAT, Cochin, Kerala, India, June 2012.

Communicated

- [3] Kaushik, A. and Chana, I., "Multi-Core Scheduling in Cloud Computing", International Journal of Computer Applications, July 2012.

Environmental Setup for Ontology Based Cloud Framework

The Experimental set up for the Ontology Based Cloud Framework is given as follows:

A.1 Installation of Protégé' 4.0

Protégé' 4.0 can be downloaded from <http://protege.stanford.edu/>. Installation steps are shown in Figure A.1 to Figure A.5.

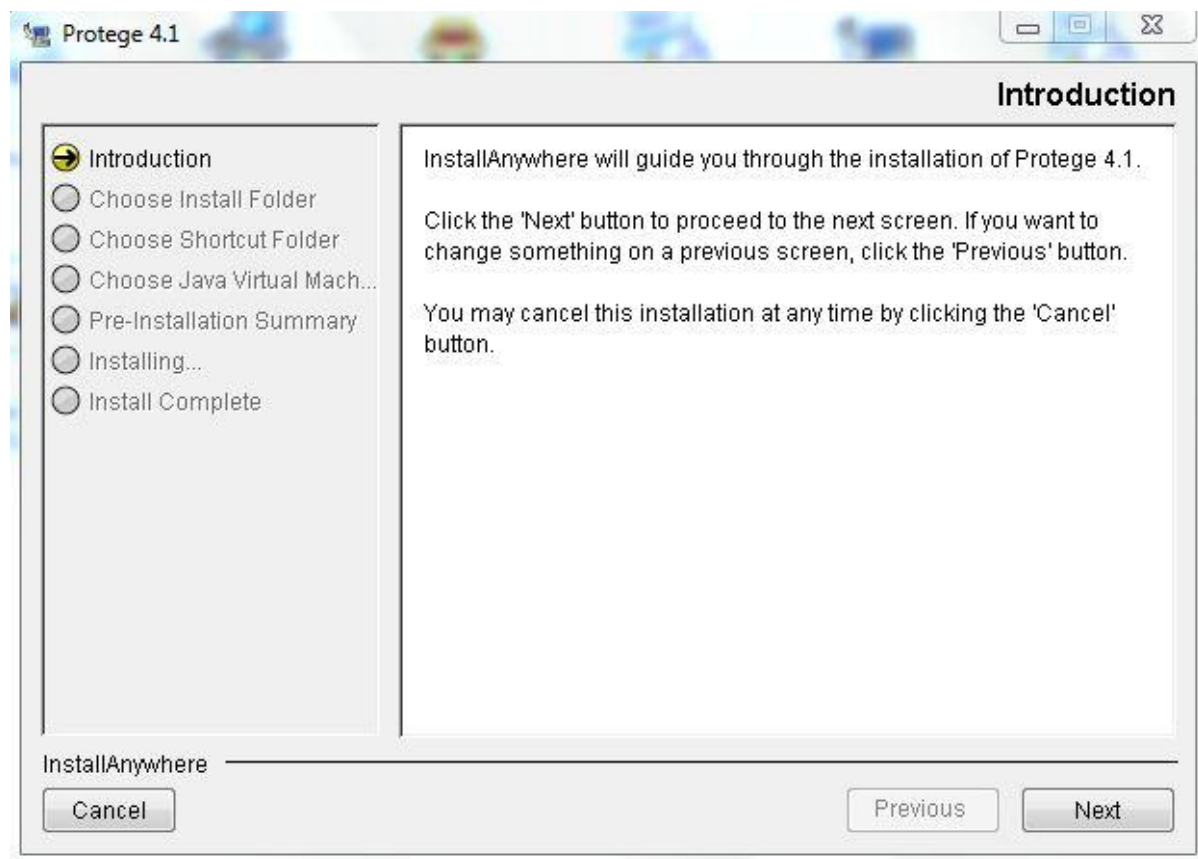


Figure A.1.1: Home Page of Protégé' 4.0

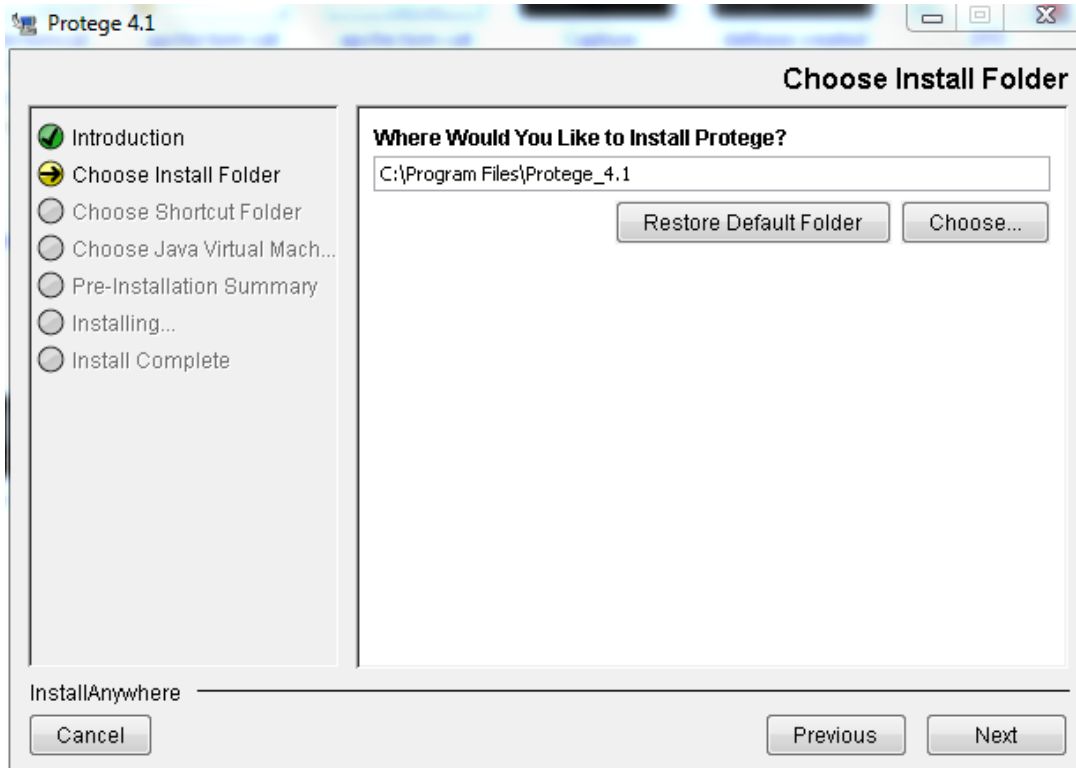


Figure A.1.2: Select Installation Directory

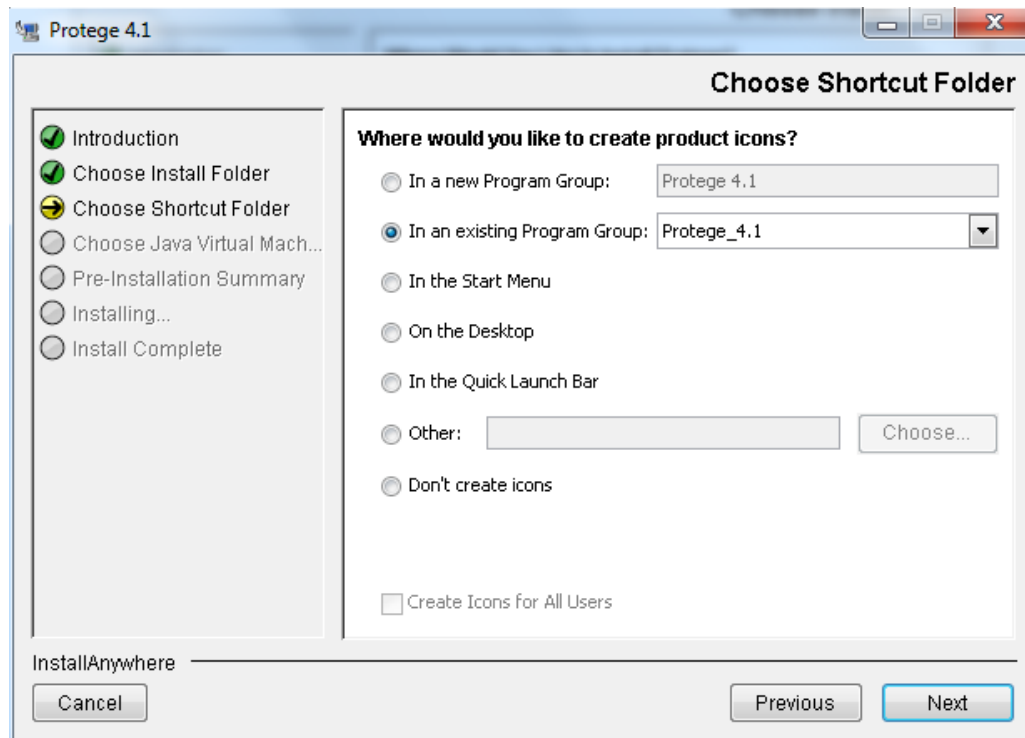


Figure A.1.3: Choose Shortcut Folder

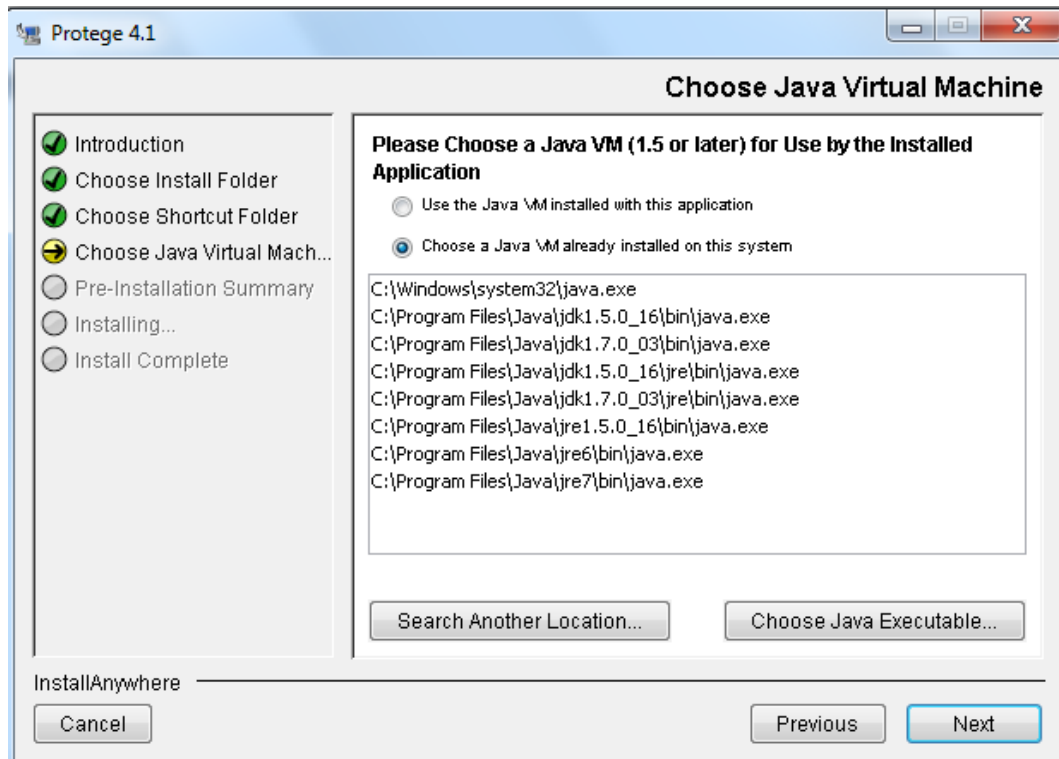


Figure A.1.4: Choose Java Virtual Machine

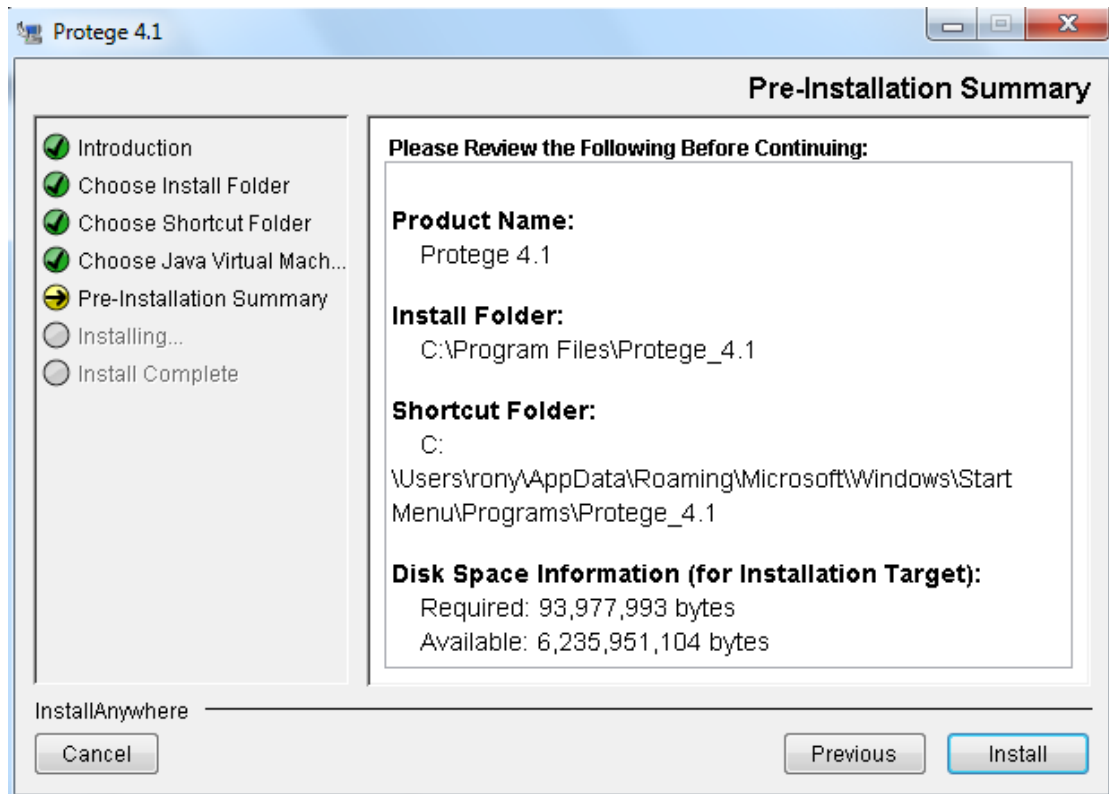


Figure A.1.5: Pre-Install Summary and Final Installation

A.2 Installation of Web Protégé

Webprotégé' .war file can be downloaded from <http://protegewiki.stanford.edu/wiki/WebProtege>. Apache tom cat installation is necessary for the Web protégé'. It can be downloaded from <http://tomcat.apache.org/download-55.cgi>. The Intermediate snapshots are as follows:

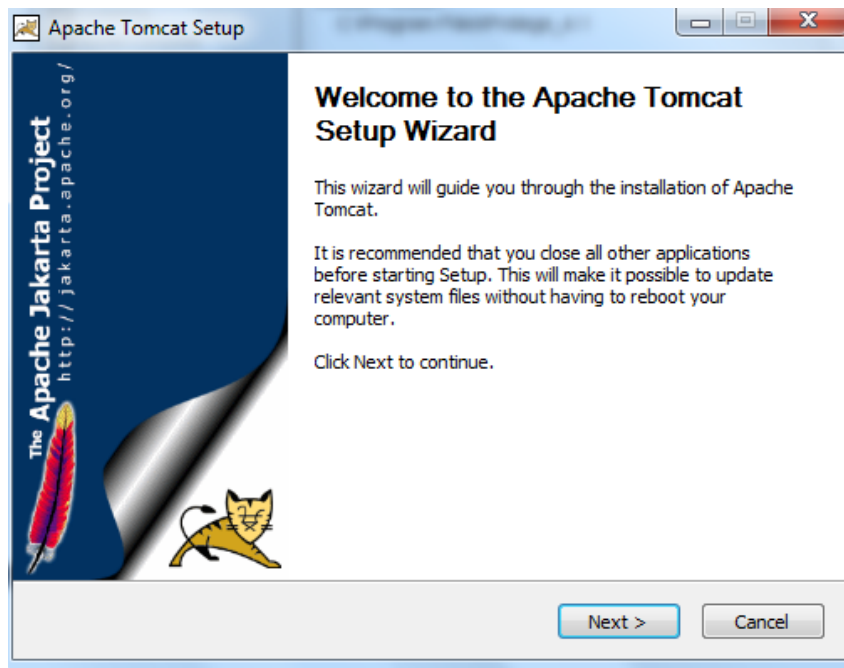


Figure A 2.1: Apache Tomcat Home Page

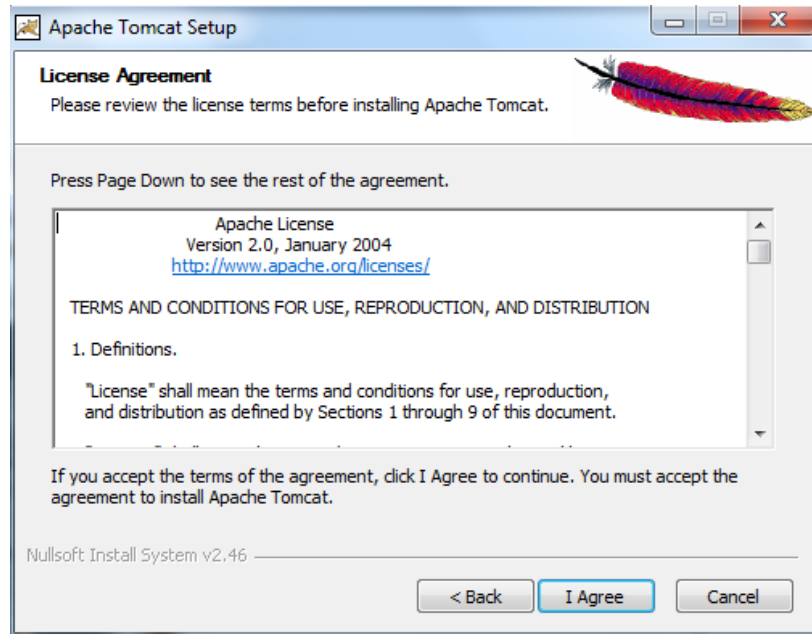


Figure A.2.2: End User License

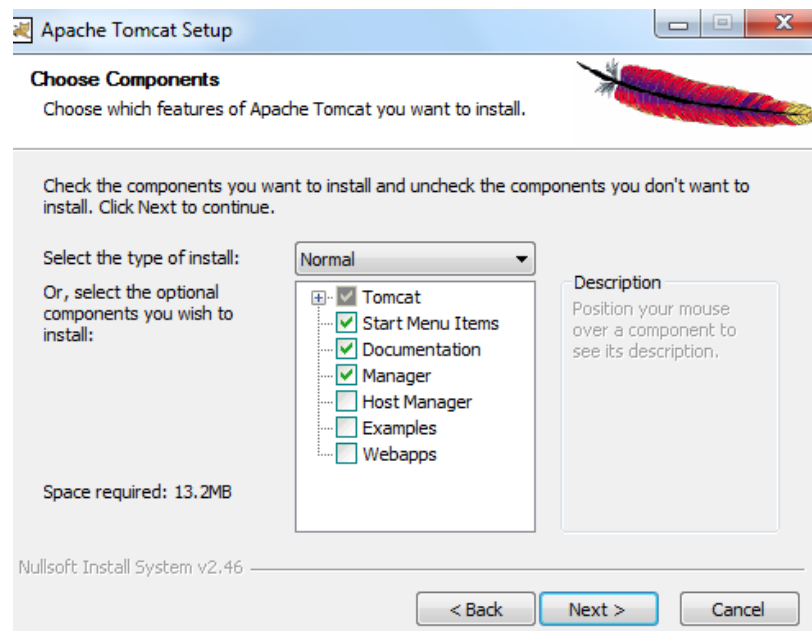


Figure A.2.3: Choose Components

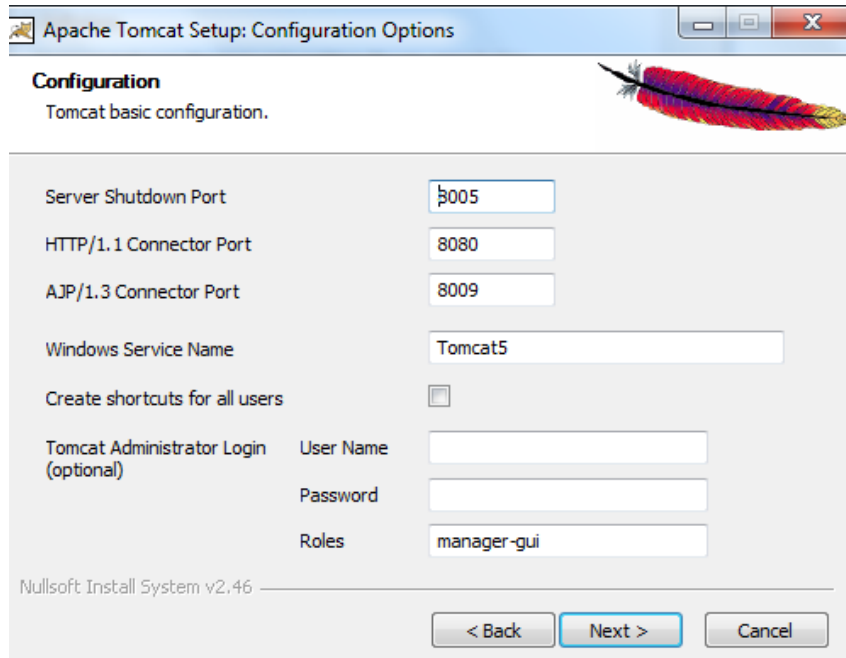


Figure A.2.4: Pre-Install Configuration

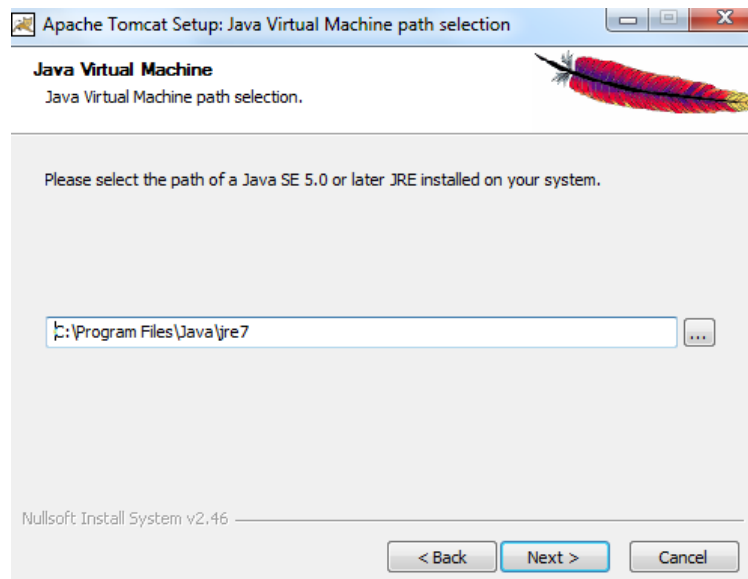


Figure A.2.5: Define Directory and Final Installation

After Installation of Apache Tomcat, put the downloaded file of Web Protégé' in the WebApp folder of the installed Apache Tomcat. Start Apache Tomcat server (Figure A.2.6).

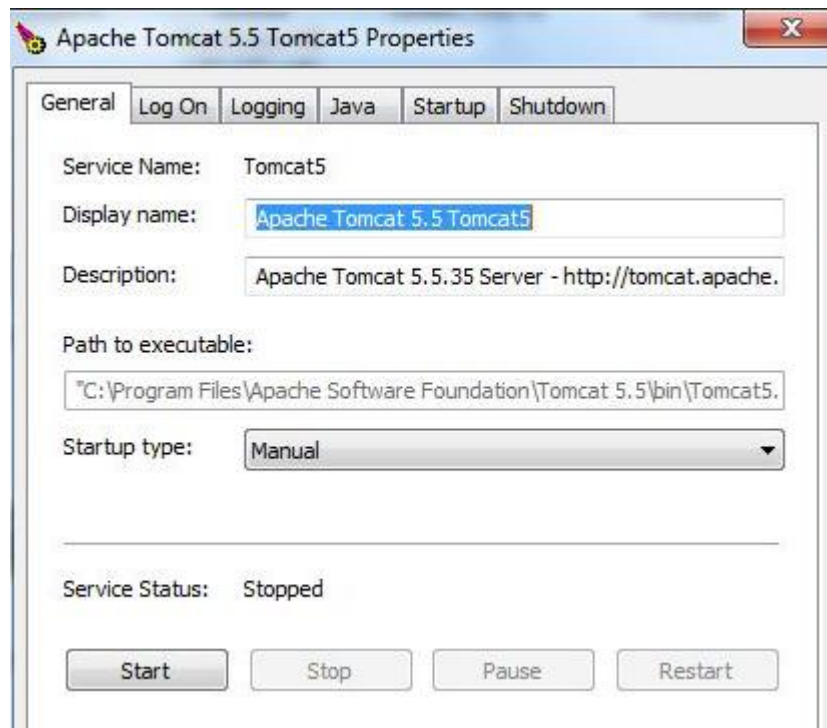


Figure A.2.6: Start Apache Tomcat Server

Execute http://localhost:8080/webprotege_2011.05.28/WebProtege.html in the web browser.