Analysis and Study of Load Balancing in Cloud

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

under the Supervision of

Mr. Punit Gupta

By

Nikhil Gupta | 111330

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "***Analysis and Study of Load Balancing in Cloud***", submitted by ***Mr. Nikhil Gupta*** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Supervisor's Name: ***Mr. Punit Gupta***

Designation: ***Assistant Professor (Grade-I)***

Signature:

Date: May, 2015

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Punit Gupta** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude towards faculty member of **Jaypee University of Information Technology** for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to staff persons for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Date: Nikhil Gupta

Table of Content

List of Figures

List of Tables

Abstract

Cloud Computing is a way of delivering service through Internet. It has caused a paradigm shift in the world of computing. Cloud Services not only provides on-demand service but also are dynamically scalable and highly available. Its emergence has impacted the business world and has brought momentous enhancements in IT infrastructure. Although it provides a great opportunity to both service provider and end user, but still there are Security and Privacy issues that hinder the acceptance of Cloud services on a larger scale. The functionalities such as virtualization, multitenant and openness of cloud computing bring prospective security issues to these services such as unreliability, uncertainty and discrepancy. Data confidentiality and trust formation are the major security concerns in cloud. Therefore, there is a firm need to establish an Inherent trust on cloud service provider in order to assure the cloud behavior, data protection and make cloud technology globally acceptable.

Data protection deals with protecting the individual or organization's private data which is shared over the cloud. It is possible only when the security and trustworthiness of both the service provider and user is ensured. Therefore there is a firm need to establish trust between both and hence we need to develop a trust management model.

The Trust Model we are proposing here will work for Private, Public and Hybrid Cloud. It will take into the consideration both Direct and Indirect Trust or recommended trust. For the Calculation of Trust Value we will take Memory (RAM), MIPS (Million Instruction per Cycle), Frequency (Frequency of data center) and Fault Rate. We will initially calculate only Direct Trust and as the time will evolve we will also consider the indirect trust

Chapter 1
Cloud Computing

1. Cloud Computing

1.1 Introduction

Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rackspace, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

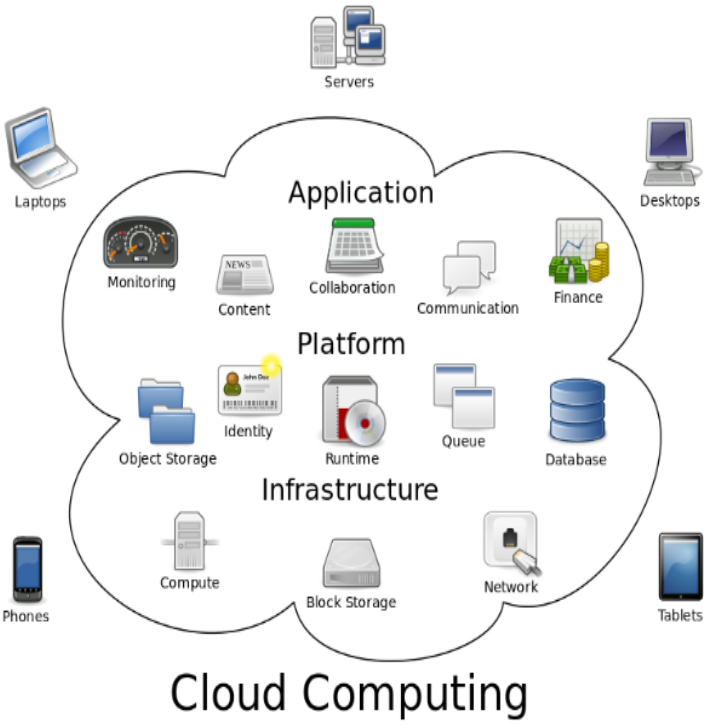


Figure 1.1 Cloud Computing

The term "moving to cloud" also refers to an organization moving away from a traditional CAPEX model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX model (use a shared cloud infrastructure and pay as one uses it).

Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing.

Cloud vendors are experiencing growth rates of 50% per annum.

1.2 Service Models

Cloud computing providers offer their services according to several fundamental models:

Proponents claim SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS is that the users' data are stored on the cloud provider's server. As a result, there could be unauthorized access to the data. For this reason, users are increasingly adopting intelligent third-party key management systems to help secure their data.

Infrastructure as a service (IaaS)

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, , KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-

demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

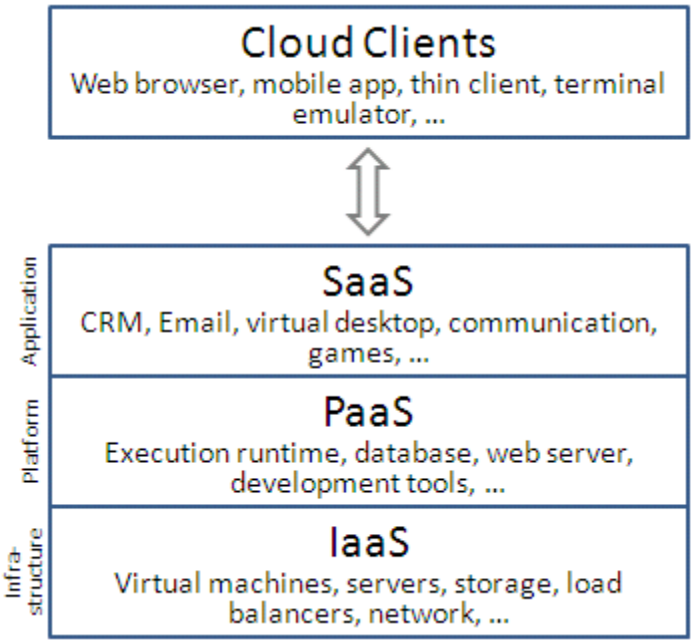


Figure1.2 Cloud Service Architecture

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed. Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Platform as a service (PaaS)

In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.

Figure 1.3 PaaS

With some PaaS offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The

latter has also been proposed by an architecture aiming to facilitate real-time in all type of cloud environments.

Platform as a service (PaaS) provides a computing platform and a key chimney. It joins with software as a service (SaaS) and infrastructure as a service (IaaS), model of cloud computing.

Software as a service (SaaS)

In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis. SaaS providers generally price applications using a subscription fee. Cloud users do not manage the cloud infrastructure and platform where the application runs

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support.

This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications are different from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Cloud users do not manage the cloud infrastructure and platform where the application runs.

Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization.

Figure 1.4 SaaS

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.

Proponents claim SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS is that the users' data are stored on the cloud provider's server. As a result, there could be unauthorized access to the data. For this reason, users are increasingly adopting intelligent third-party key management systems to help secure their data.

1.3 Deployment Models

Cloud computing types

Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free or offered on a pay-per-usage model. Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet. AWS and Microsoft also offer direct connect services called "AWS Direct Connect" and

"Azure ExpressRoute" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

Hybrid cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources.

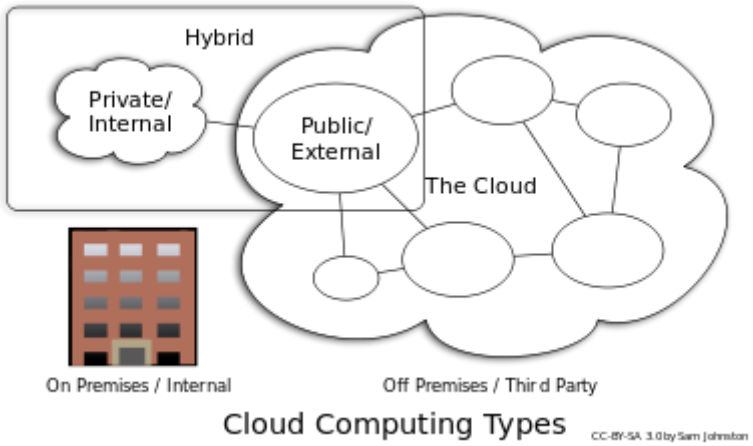


Figure 1.5 Types of Cloud

Gartner, Inc. defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

Varied use cases for hybrid cloud composition exist. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service. This example of hybrid cloud extends the capabilities of the enterprise to deliver a specific business service through the addition of externally available public cloud services.

Another example of hybrid cloud is one where IT organizations use public cloud computing resources to meet temporary capacity needs that cannot be met by the private cloud. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization only pays for extra compute resources when they are needed. Cloud bursting enables data centers to create an in-house IT infrastructure that supports average

workloads, and use cloud resources from public or private clouds, during spikes in processing demands.

Community cloud

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party, and either hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

Distributed cloud

Cloud computing can also be provided by a distributed set of machines that are running at different locations, while still connected to a single network or hub service. Examples of this include distributed computing platforms such as BOINC and Folding@Home. An interesting attempt in such direction is Cloud@Home, aiming at implementing cloud computing provisioning model on top of voluntarily shared resources

Inter cloud

The Intercloud is an interconnected global "cloud of clouds" and an extension of the Internet "network of networks" on which it is based. The focus is on direct interoperability between public cloud service providers, more so than between providers and consumers (as is the case for hybrid- and multi-cloud).

Multi cloud

Multi cloud is the use of multiple cloud computing services in a single heterogeneous architecture to reduce reliance on single vendors, increase flexibility through choice, mitigate against disasters, etc. It differs from hybrid cloud in that it refers to multiple cloud services, rather than multiple deployment modes (public, private, and legacy).

1.4 Architecture

Cloud computing sample architecture

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over a loose coupling mechanism such as a

messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others.

Cloud engineering

Cloud engineering is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high-level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems. It is a multidisciplinary method encompassing contributions from diverse areas such as systems, software, web, performance, information, security, platform, risk, and quality engineering.

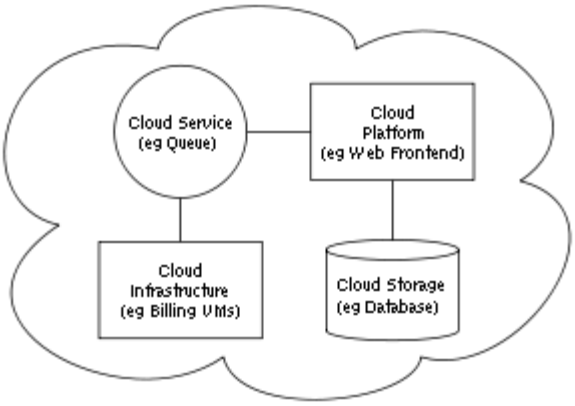


Figure 1.6 Cloud Architecture

1.5 Characteristics

Cloud computing exhibits the following key characteristics:

- ***Agility*** improves with users' ability to re-provision technological infrastructure resources.
- ***Application programming interfac***e (API) accessibility to software that enables machines to interact with cloud software in the same way that a traditional user interface (e.g., a computer desktop) facilitates interaction between humans and computers. Cloud computing systems typically use Representational State Transfer (REST)-based APIs.
- ***Cost reductions*** claimed by cloud providers. A public-cloud delivery model converts capital expenditure to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained, with usage-based options and fewer IT skills are required for implementation (in-house). The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.

- ***Device and location independence*** enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- ***Maintenance*** of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

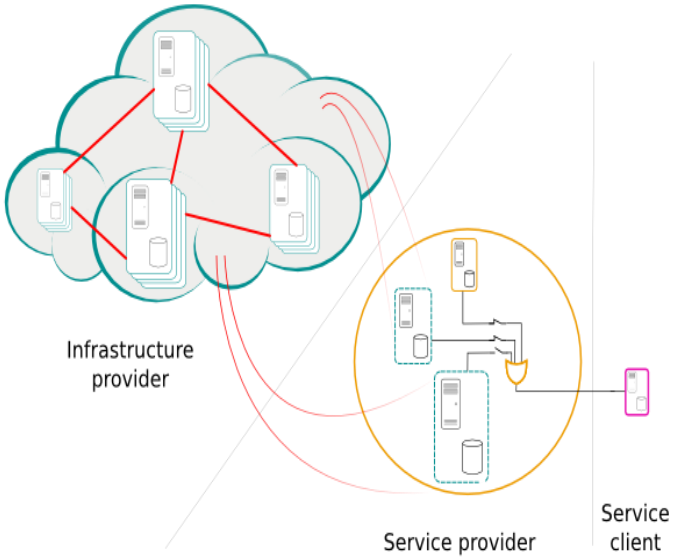


Figure 1.7 Cloud Model

- ***Multitenancy*** enables sharing of resources and costs across a large pool of users thus allowing for:

    1. Centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
    2. Peak-load capacity increases (users need not engineer for highest possible load-levels)
    3. Utilization and efficiency improvements for systems that are often only 10–20% utilized.

- ***Performance*** is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.
- ***Productivity*** may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.
- ***Reliability*** improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery.
- ***Scalability and elasticity*** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time (Note, the VM startup time varies by VM type, location, OS and cloud providers), without users having to engineer for peak loads.

- ***Security*** can improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford to tackle. However, the complexity of security is greatly increased when data is distributed over a wider area or over a greater number of devices, as well as in multi-tenant systems shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":

- ***On-demand self-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.
- ***Broad network access***. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- ***Resource pooling***. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- ***Rapid elasticity***. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.
- ***Measured service***. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Chapter 2
Cloud Infrastructure as a Service

2. Cloud Infrastructure as a Service

2.1 Introduction

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, Oracle VirtualBox, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

Figure 2.1 Cloud Iaas

To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

Infrastructure as a Service (IaaS) is one of the three fundamental service models of cloud computing alongside Platform as a Service (PaaS) and Software as a Service (SaaS). As with all cloud computing services it provides access to computing resource in a virtualised environment, "the Cloud", across a public connection, usually the internet. In the case of IaaS the computing resource provided is specifically that of virtualised hardware, in other words, computing infrastructure. The definition includes such offerings as virtual server space, network connections, bandwidth, IP addresses and load balancers. Physically, the pool of hardware resource is pulled from a multitude of servers and networks usually distributed across numerous data centers, all of which the cloud provider is responsible for maintaining. The client, on the other hand, is given access to the virtualised components in order to build their own IT platforms.

In common with the other two forms of cloud hosting, IaaS can be utilized by enterprise customers to create cost effective and easily scalable IT solutions where the complexities and expenses of managing the underlying hardware are outsourced to the cloud provider. If the scale of a business customer's operations fluctuate, or they are looking to expand, they can tap into the cloud resource as and when they need it rather than purchase, install and integrate hardware themselves.

2.2 Salient Features

- Enterprise infrastructure; by internal business networks, such as private clouds and virtual local area networks, which utilize pooled server and networking resources and in which a business can store their data and run the applications they need to operate day-to-day. Expanding businesses can scale their infrastructure in accordance with their growth whilst private clouds (accessible only by the business itself) can protect the storage and transfer of the sensitive data that some businesses are required to handle.
- Cloud hosting; the hosting of websites on virtual servers which are founded upon pooled resources from underlying physical servers. A website hosted in the cloud, for example, can benefit from the redundancy provided by a vast network of physical servers and on demand scalability to deal with unexpected demands placed on the website.
- Virtual Data Centers (VDC); a virtualised network of interconnected virtual servers which can be used to offer enhanced cloud hosting capabilities, enterprise IT infrastructure or to integrate all of these operations within either a private or public cloud implementation.

Characteristics and components of IaaS include:

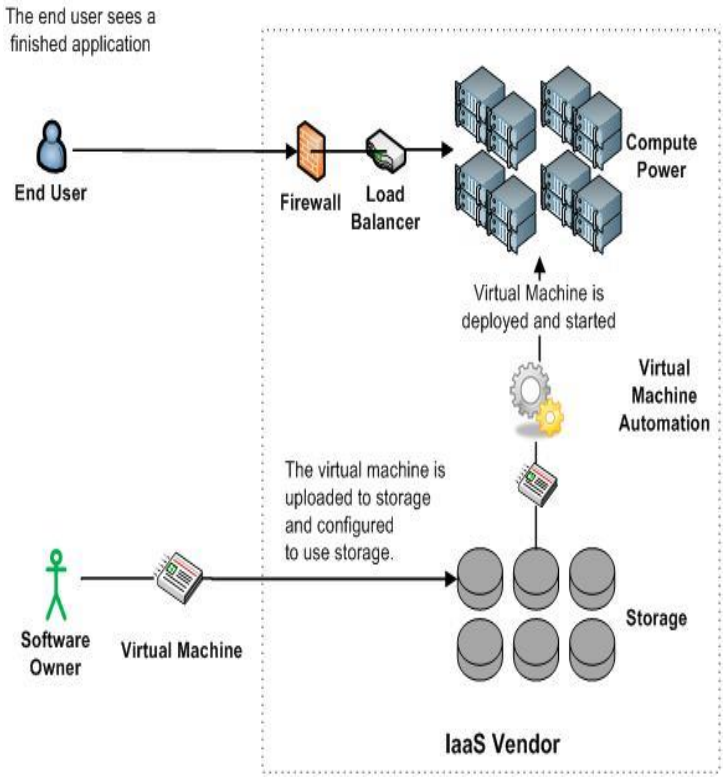
- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Desktop virtualization.

- Policy-based services.
- Internet connectivity.

2.3 Features and Benefits

A typical Infrastructure as a Service offering can deliver the following features and benefits:

- Scalability; resource is available as and when the client needs it and, therefore, there are no delays in expanding capacity or the wastage of unused capacity
- No investment in hardware; the underlying physical hardware that supports an IaaS service is set up and maintained by the cloud provider, saving the time and cost of doing so on the client side
- Utility style costing; the service can be accessed on demand and the client only pays for the resource that they actually use
- Location independence; the service can usually be accessed from any location as long as there is an internet connection and the security protocol of the cloud allows it
- Physical security of data center locations; services available through a public cloud, or private clouds hosted externally with the cloud provider, benefit from the physical security afforded to the servers which are hosted within a data center
- No single point of failure; if one server or network switch, for example, were to fail, the broader service would be unaffected due to the remaining multitude of hardware resources and redundancy configurations. For many services if one entire data center were to go offline, never mind one server, the IaaS service could still run successfully.



Fiaure 2.2 Cloud Iaas Model

Chapter 3
Load Balancing in Cloud

3. Load Balancing in Cloud

3.1 Introduction

Load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.

In computing, load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process.

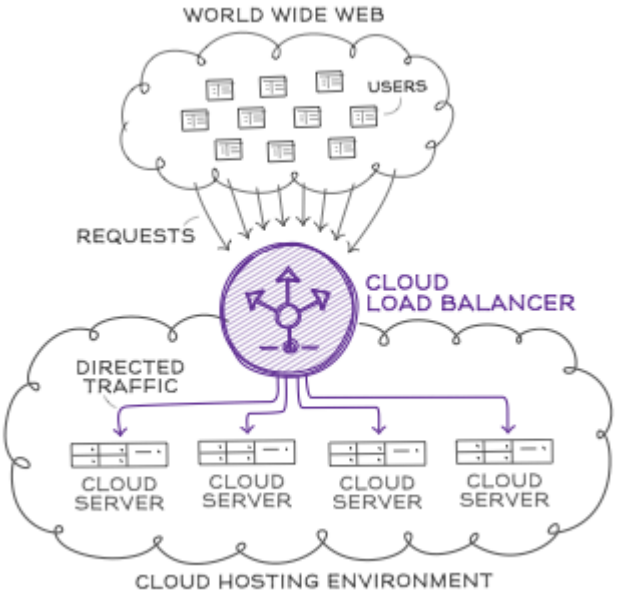


Figure 3.1 Load Balancing

Load balancing differs from channel bonding in that load balancing divides traffic between network interfaces on a network socket (OSI model layer 4) basis, while channel bonding implies a division of traffic between physical interfaces at a lower level, either per packet (OSI model Layer 3) or on a data link (OSI model Layer 2) basis.

One of the most commonly used applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly load-balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers and Domain Name System (DNS) servers. Lately, some load balancers have evolved to support databases; these are called database load balancers.

For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards

requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting back-end servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

Some load balancers provide a mechanism for doing something special in the event that all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying a message regarding the outage. Load balancing gives the IT team a chance to achieve a significantly higher fault tolerance. It can automatically provide the amount of capacity needed to respond to any increase or decrease of application traffic.

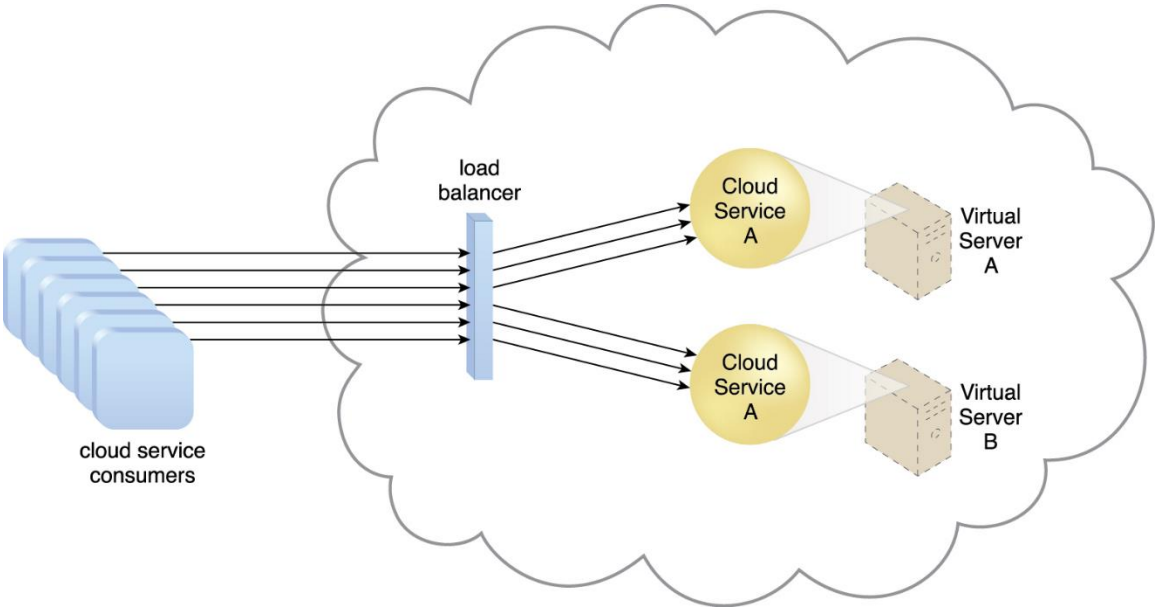


Figure 3.2 Load Balancing in action

It is also important that the load balancer itself does not become a single point of failure. Usually load balancers are implemented in high-availability pairs which may also replicate session persistence data if required by the specific application.

3.2 Load balancer features

Hardware and software load balancers may have a variety of special features. The fundamental feature of a load balancer is to be able to distribute incoming requests over a

number of backend servers in the cluster according to a scheduling algorithm. Most of the following features are vendor specific:

- ***Asymmetric load****:* A ratio can be manually assigned to cause some backend servers to get a greater share of the workload than others. This is sometimes used as a crude way to account for some servers having more capacity than others and may not always work as desired.
- ***Priority activation****:* When the number of available server's drops below a certain number, or load gets too high, standby servers can be brought online.
- ***SSL Offload and Acceleration****:* Depending on the workload, processing the encryption and authentication requirements of an SSL request can become a major part of the demand on the Web Server's CPU; as the demand increases, users will see slower response times, as the SSL overhead is distributed among Web servers. To remove this demand on Web servers, a balancer can terminate SSL connections, passing HTTPS requests as HTTP requests to the Web servers. If the balancer itself is not overloaded, this does not noticeably degrade the performance perceived by end users. The downside of this approach is that all of the SSL processing is concentrated on a single device (the balancer) which can become a new bottleneck. Some load balancer appliances include specialized hardware to process SSL. Instead of upgrading the load balancer, which is quite expensive dedicated hardware, it may be cheaper to forgo SSL offload and add a few Web servers. Also, some server vendors such as Oracle/Sun now incorporate cryptographic acceleration hardware into their CPUs such as the T2000. F5 Networks incorporates a dedicated SSL acceleration hardware card in their local traffic manager (LTM) which is used for encrypting and decrypting SSL traffic. One clear benefit to SSL offloading in the balancer is that it enables it to do balancing or content switching based on data in the HTTPS request.
- ***HTTP*** *compression:* reduces amount of data to be transferred for HTTP objects by utilizing zip compression available in all modern web browsers. The larger the response and the further away the client is, the more this feature can improve response times. The tradeoff is that this feature puts additional CPU demand on the Load Balancer and could be done by Web servers instead.

- ***TCP offload**:* different vendors use different terms for this, but the idea is that normally each HTTP request from each client is a different TCP connection. This feature utilizes HTTP/1.1 to consolidate multiple HTTP requests from multiple clients into a single TCP socket to the back-end servers.

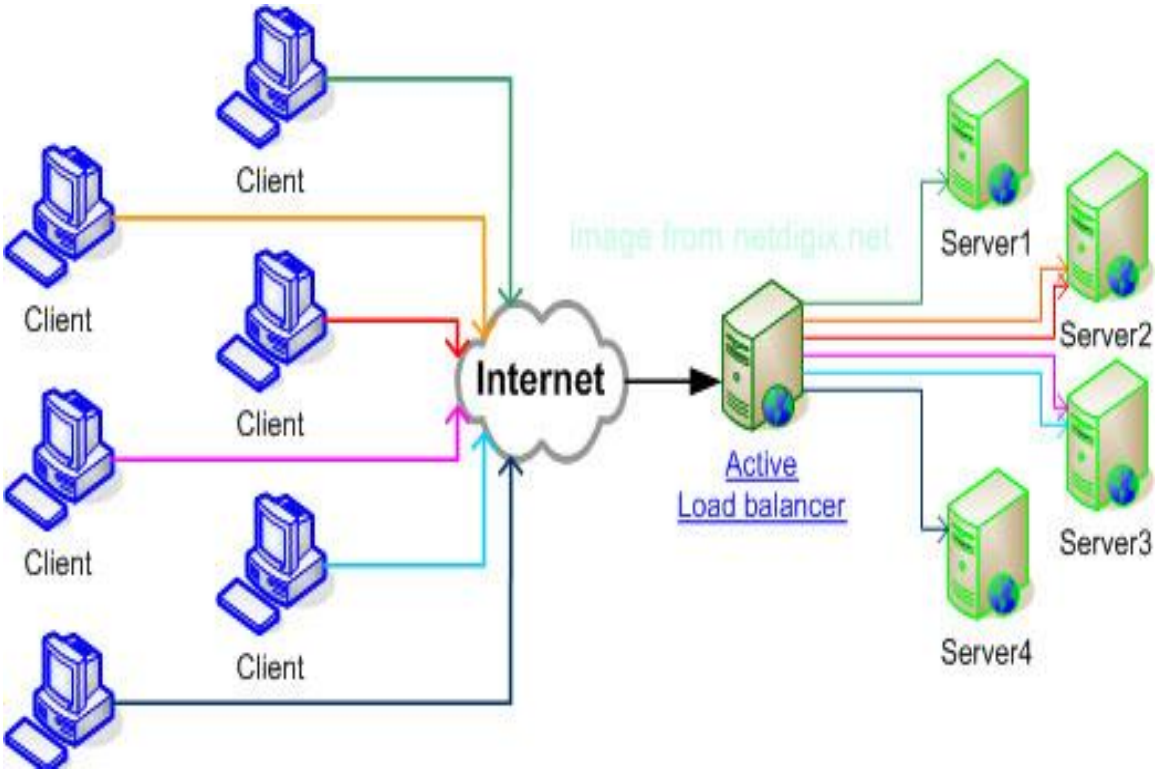


Figure 3.3 Active Load Balancer

- ***TCP buffering**:* the load balancer can buffer responses from the server and spoon-feed the data out to slow clients, allowing the web server to free a thread for other tasks faster than it would if it had to send the entire request to the client directly.
- ***Direct Server Return**:* an option for asymmetrical load distribution, where request and reply have different network paths.
- ***Health checking**:* the balancer polls servers for application layer health and removes failed servers from the pool.
- ***HTTP caching**:* the balancer stores static content so that some requests can be handled without contacting the servers.

- ***Content filtering****:* some balancers can arbitrarily modify traffic on the way through.
- ***HTTP security****:* some balancers can hide HTTP error pages, remove server identification headers from HTTP responses, and encrypt cookies so that end users cannot manipulate them.
- ***Priority queuing****:* also known as rate shaping, the ability to give different priority to different traffic.
- ***Content-aware switching****:* most load balancers can send requests to different servers based on the URL being requested, assuming the request is not encrypted (HTTP) or if it is encrypted (via HTTPS) that the HTTPS request is terminated (decrypted) at the load balancer.
- ***Client authentication:*** authenticate users against a variety of authentication sources before allowing them access to a website.
- ***Programmatic traffic manipulation:*** at least one balancer allows the use of a scripting language to allow custom balancing methods, arbitrary traffic manipulations, and more.
- ***Firewall:*** direct connections to backend servers are prevented, for network security reasons Firewall is a set of rules that decide whether the traffic may pass through an interface or not..

Chapter 4
Trust in Cloud (Related Work)

4. Trust in Cloud (Related Work)

4.1 Introduction

As a new computing mode, cloud computing can provide users with virtualized and scalable web services, which faced with serious security challenges, however. Access control is one of the most important measures to ensure the security of cloud computing. But applying traditional access control model into the Cloud directly could not solve the uncertainty and vulnerability caused by the open conditions of cloud computing. In cloud computing environment, only when the security and reliability of both interaction parties are ensured, data security can be effectively guaranteed during interactions between users and the Cloud. Therefore, building a mutual trust relationship between user and cloud platform is the key to implement new kinds of access control method in cloud computing environment.

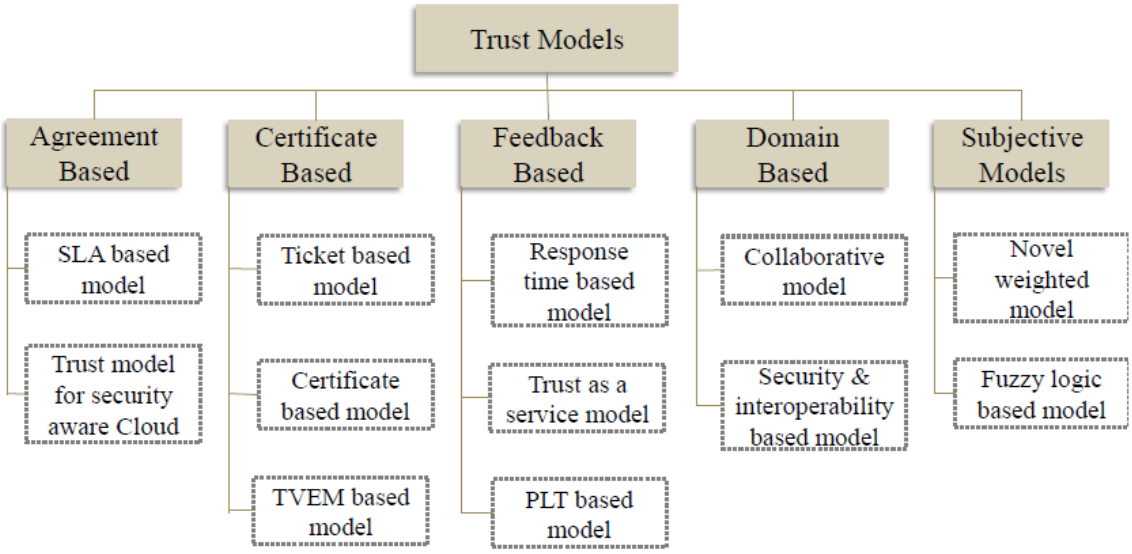


Figure 4.1 Trust Models

4.2 Related Work on Trust Models

Numerous trust models have been proposed in cloud.

Title: Data Provenance Trusted Model in Cloud Computing

Summary: Mohda Izua Mohd Saad proposed a novel data provenance trusted model to provide secured access to data provenance via a secured communication channel. This

model also propose a consolidation log storage for virtual and physical layer in cloud environment. Transparency and confident towards cloud provider are some of the prominent issues in cloud today. In order to solve these problems, cloud service providers should have a high level of assurance and accountability in order to maintain trust between them and the users. This trust can be achieved through data provenance. Data provenance provides historical data from its original resources and can facilitate trust between cloud providers and users. They have discussed the overview of data provenance in cloud computing and significant approach in provenance logging system. This is then followed by discussion of provenance challenges in cloud environment. Finally, this paper propose a novel data provenance trusted model to provide secured access to data provenance via a secured communication channel. This model also propose a consolidation log storage for virtual and physical layer in cloud environment.

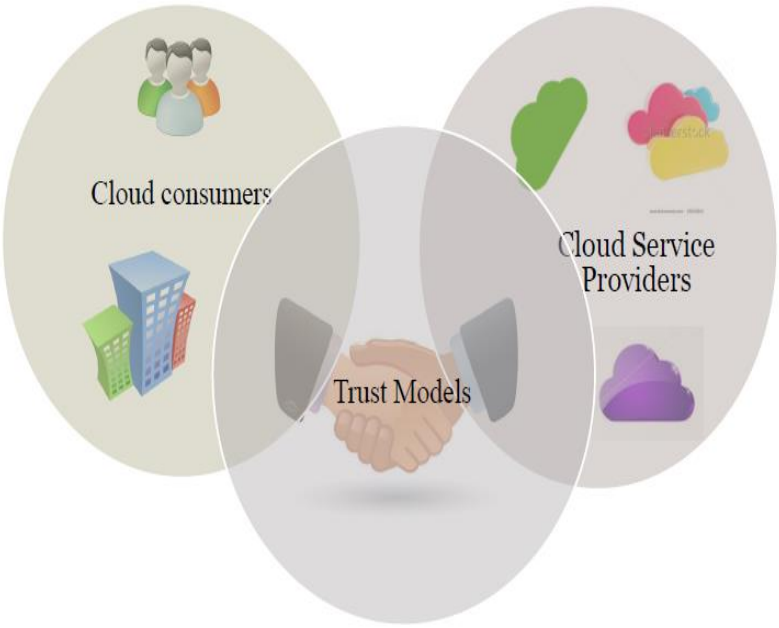


Figure 4.2 Trust Models

Title: A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing

Summary: WenAn Tan proposed a trust service-oriented workflow scheduling algorithm. The scheduling algorithm adopts a trust metric that combines direct trust and recommendation trust. In addition, we provide balance policies to enable users to balance different requirements, including time, cost, and trust. Cloud services have been utilized in large-scale distributed environments. As an effective service aggregation methodology, workflow technology has been used to construct composite services. Efficient and dependable workflow scheduling (WFS) is crucial for integrating enterprise systems. While WFS has been widely studied, WFS-related algorithms are mainly focused on optimizing execution time or cost. However, in cloud computing environment, WFS is up against the threats of the inherent uncertainty and unreliability to the applications. Therefore, trust service-oriented strategies must be considered inWFS. As a result, this paper proposes a trust service-oriented workflow scheduling algorithm. The scheduling algorithm adopts a trust metric that combines direct trust and recommendation trust. In addition, we provide balance policies to enable users to balance different requirements,

including time, cost, and trust. A case study was conducted to illustrate the value of the proposed algorithm. The experimental results show that the proposed approach is effective and feasible.

Title: Trust Model for a Cloud Computing Application and Service

Summary: Rizwana A.R. Shaikh propose a trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service. The trust value acts as a means for selecting any cloud service. Cloud computing has attracted great attention due its various features. There are various rising concerns in its growth, one of which is security. Trust between provider and a user is the most important factor to be considered for achieving security, which is also essential to promote the reputation of various cloud providers and their offered services. Here we present a trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service. The trust value acts as a means for selecting any cloud service.

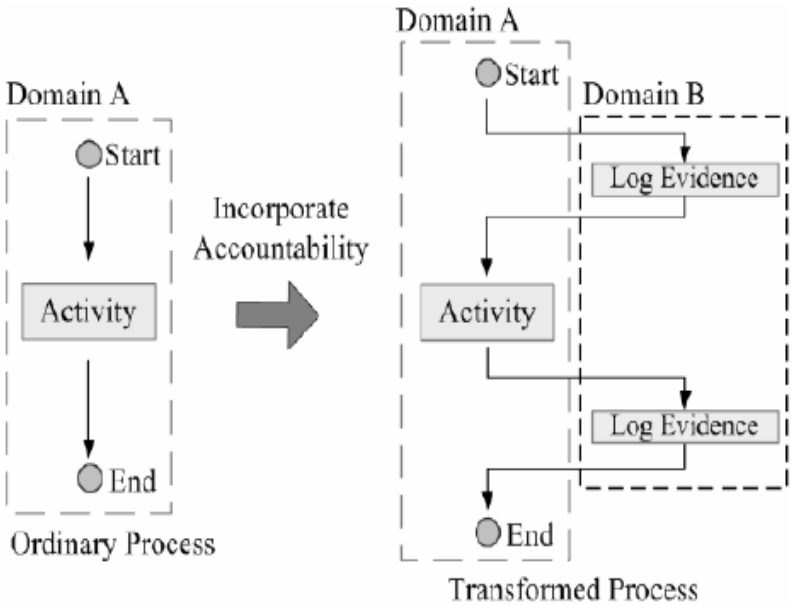


Figure 4.3 Incorporating Accountability

Title: A Trust Management Model to enhance security of Cloud Computing Environments

Summary: Xiaodong Sun introduces a trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust similarity has been addressed to prevent the behavior of associated cheat of middle nodes. And this model is geared toward the cloud users who are making their decision on whether to use services of some cloud computing providers by giving them trust evaluation sets about providers and then building reasonable trust relationship between them. With the proliferation of cloud computing, the way of reasonable establishment of trust relationship among entities, as a vital part for forming security mechanism in cloud computing environments, is attracting increasing attention. This article introduces a trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust similarity has been addressed to prevent the behavior of associated cheat of middle nodes. And this model is geared toward

the cloud users who are making their decision on whether to use services of some cloud computing providers by giving them trust evaluation sets about providers and then building reasonable trust relationship between them. Our motivation is trying to propose a new idea and method on trust management in cloud computing and further studies are still required to justify the rationality and practicability of this model.

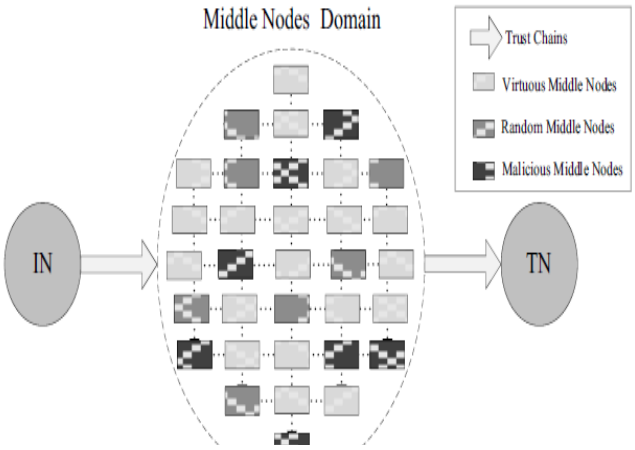


Figure 4.4 Simulation Environment

Title: Modeling and Evaluation of Trust in Cloud Computing Environments

Summary: Qiang Guo introduced a definition of trust in cloud systems and the properties of trust are analyzed. Based on the properties and semantics of trust, an extensible trust evaluation model named ETEC is proposed, which includes a time-variant comprehensive evaluation method for expressing direct trust and a space-variant evaluation method for calculating recommendation trust. To compute trust in cloud systems, an algorithm based on the ETEC model is given. Simulation and analysis shows that this model can calculate the trust degree effectively and reasonably in cloud computing environments. Today, one of the most important factors for the success of cloud computing is to create trust and security.

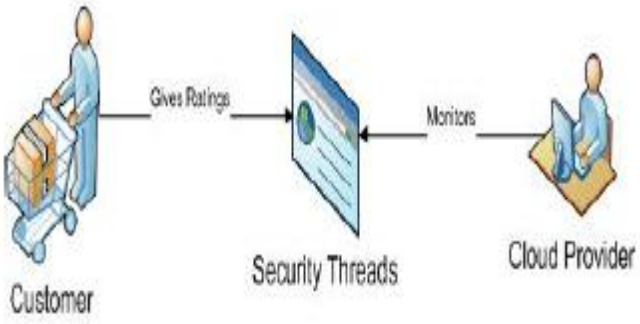


Figure 4.5 Trust Relationship defined

Cloud computing will face a lot of challenges when the key element trust is absent. There are no special trust evaluation models for cloud c omputing environment. In this paper, the definition of trust in cloud systems is introduced and the properties of trust are analyzed. Based on the properties and semantics of trust, an extensible trust evaluation model named
ETEC is proposed, which includes a time-variant comprehensive evaluation method for expressing direct trust and a space-variant evaluation method for calculating recommendation trust. To compute trust in cloud systems, an algorithm based on the ETEC model is given. Simulation and analysis shows

that this model can calculate the trust degree effectively and reasonably in cloud computing environments.

Title: A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model

Summary: Xiaoqiong Yang also proposed A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model for statistics-based behaviors

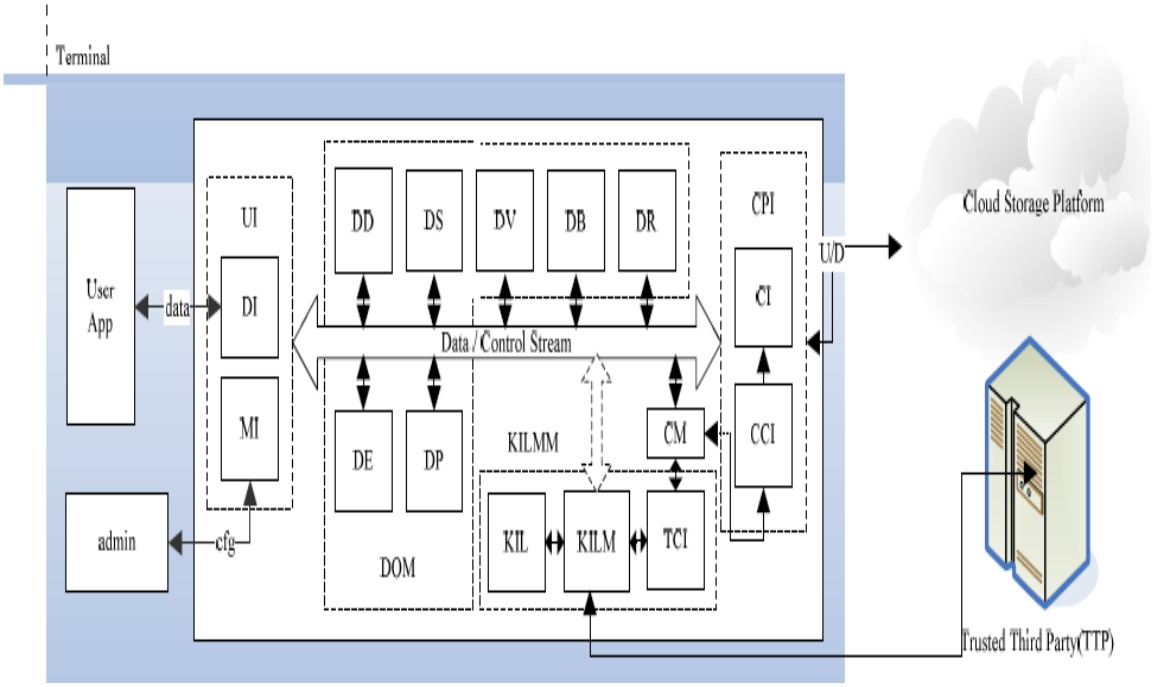


Figure 4.6 Structure of TCMCS

which generates the threshold for each type of behaviors and each user's membership degree between the user's performance and the calculated threshold. And then the membership degree and the behavior weight will be together sent to calculate the user's behavior trust evaluation value using a simple normalized method.

Finally as the basis of intra-domain trust and recommendation trust, behavior trust could help for users' further dynamic authorization of access control.

Title: A Trusted Control Model of Cloud Storage

Summary: Junfeng Tian proposed a Trusted Control Model of Cloud Storage with access control (TCMCS) to handle all the interactions between a client and CSS to ensure the transparency of the data manipulation.

It is TCMCS that shield the differences among different cloud storage platforms, which can separate operations that protected both security and integrity of data from users' applications. Since users only need to care about their own business logic and the development of application program is greatly simplified.

Title: QoS Based Trust Management Model for Cloud IaaS

Summary: Punit Gupta proposed a QoS Based Trust Management Model for Cloud IaaS that is suitable for trust value management for the cloud IaaS parameters. Based on the above achieved trust values, a scheduling is done for better allocation of resources and further enhance the QOS of services been provided to the users. In this paper, an approach for managing trust in Cloud IaaS is proposed.

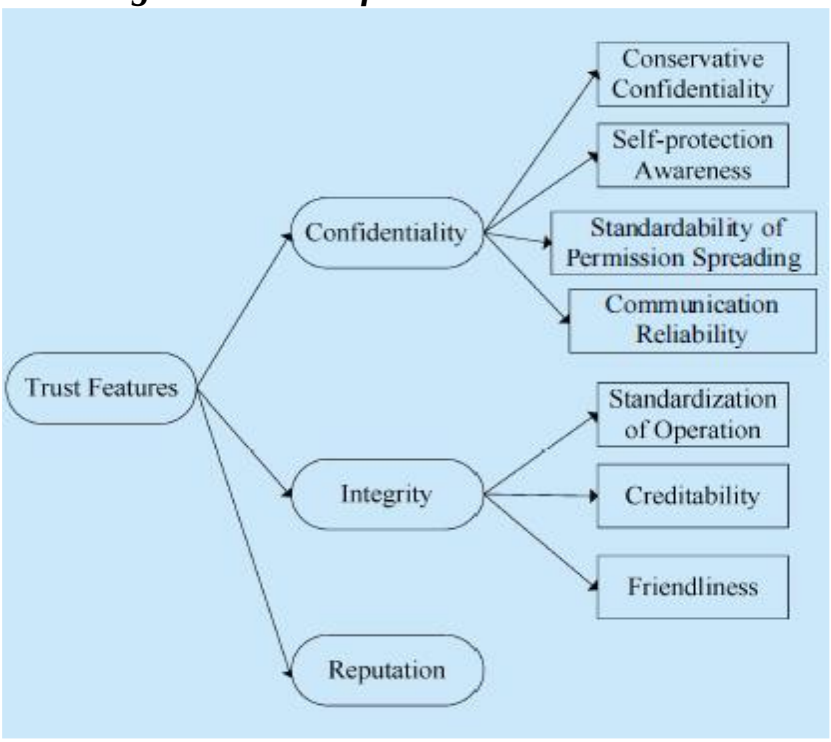


Figure 4.7 Trust Features

As a new computing mode, cloud computing can provide users with virtualized and scalable web services, which faced with serious security challenges, however. Access control is one of the most important measures to ensure the security of cloud computing. But applying traditional access control model into the Cloud directly could not solve the uncertainty and vulnerability caused by the open conditions of cloud computing. In cloud computing environment, only when the security and reliability of both interaction parties are ensured, data security can be effectively guaranteed during interactions between users and the Cloud. Therefore, building a mutual trust relationship between implement new kinds o f access control method in cloud computing environment. Combining with Trust

Management (TM) , a mutual trust based access control (MTBAC) model is proposed in this paper. MTBAC model take both user's behavior trust and cloud services node's credibility into consideration. Trust relationships between users and cloud service nodes are established by mutual trust mechanism. Security problems of access control are solved by implementing MTBAC model into cloud computing environment. Simulation experiments show that MTBAC model can guarantee the interaction between users and cloud service nodes.

4.3 Trust in Cloud

Cloud computing is a service delivering mode based on the Internet. It can provide users with scalable services as required through the Internet and has been widely recognized and applied. To utilize computing resources more effectively and safely, people begin to pay close attention to hidden security problems in the Cloud. The features of virtualization, multitenant and openness of cloud computing bring potential security issues to cloud services such as unreliability, insecurity and inconsistency. Security issue is an important aspect of cloud computing which cannot be ignored.

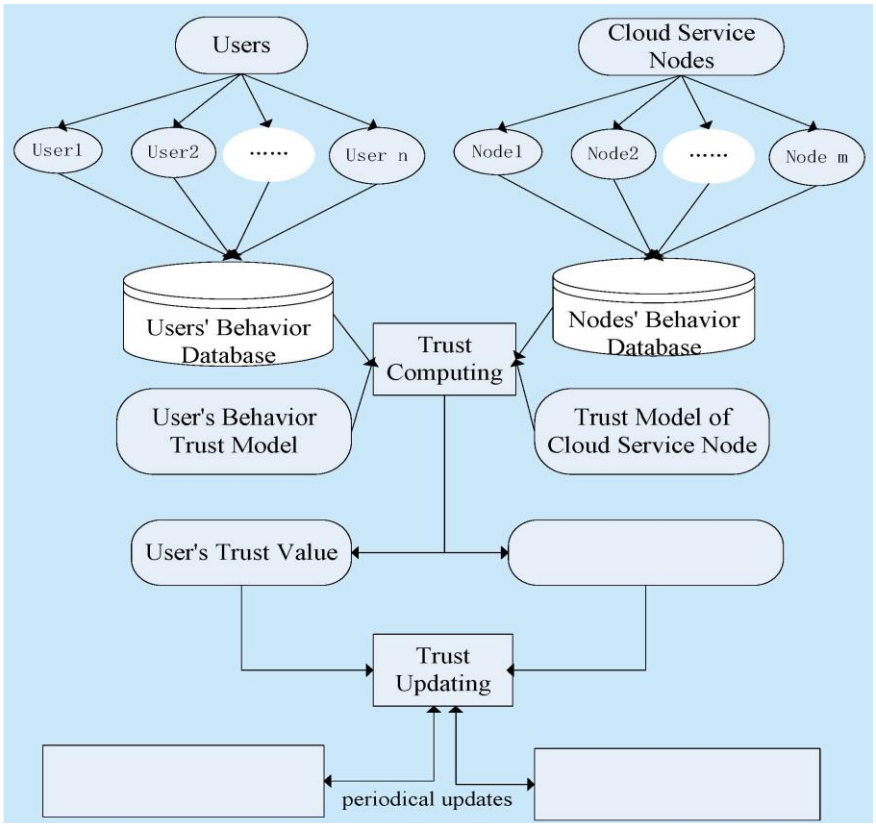


Figure 4.8 Trust in Cloud

Access control is one of the most important measures to ensure the security of cloud computing. Early access control technology can not only ensure normal access requirements of valid users, prevent invasions of unauthorized users, but it can also solve security problems caused by valid users' disoperation. Cloud computing environment is a typical distributed environment; hence the distribution, dynamism, and anonymity of

information resources and services are remarkable features of cloud computing environment.

TABLE I: Comparative Analysis of Trust Models in Cloud Computing

Assessment Features	Agreement based Trust Models		Certificates based Trust Models			Feedback based Trust Models			Domain based Trust Models		Subjective Trust Models	
	SLA based trust model [25]	Trust model for security aware Cloud [26]	Ticket based trust model [27]	Certificates based trust model [34]	(TVEM) based trust model [28]	Trust evaluation model based on response time [35]	Trust as a service model [23]	PLT based trust model [29]	Collaborative Trust Model [30]	Security and interoperability centered trust model [31]	A novel weighted trust model [32]	Fuzzy comprehensive based trust model [33]
Data Integrity	Medium	Medium	High	Medium	High	Low	Low	Medium	Low	Low	Low	Low
Data Control and Ownership	High	High	High	High	High	Low	Low	Low	Low	Low	Low	Low
Model Complexity	Low	Low	Medium	High	High	Medium	High	Medium	Low	Low	High	High
Detection of Untrusted Entities	High	High	High	Low	High	Low	High	High	High	Medium	High	Medium
Process Execution Control	Low	High	Low	Low	High	Low	Low	Low	Low	Low	Low	Low
Quality of Service Attributes	Medium	Low	Low	Low	Low	High	medium	medium	Low	Low	Low	Low
Dynamic Trust Update and Logging	Medium	Low	Low	Medium	Medium	High	Medium	Low	High	High	Medium	High

Chapter 5
Proposed Trust Based Model

5. Proposed Trust Based Model

5.1 Introduction

Cloud Computing is a way of delivering service through Internet. It has caused a paradigm shift in the world of computing. Cloud Services not only provides on-demand service but also are dynamically scalable and highly available. Its emergence has impacted the business world and has brought momentous enhancements in IT infrastructure. Although it provides a great opportunity to both service provider and end user, but still there are Security and Privacy issues that hinder the acceptance of Cloud services on a larger scale. The functionalities such as virtualization, multitenant and openness of cloud computing bring prospective security issues to these services such as unreliability, uncertainty and discrepancy. Data confidentiality and trust formation are the major security concerns in cloud. Therefore, there is a firm need to establish an Inherent trust on cloud service provider in order to assure the cloud behavior, data protection and make cloud technology globally acceptable.

Figure 5.1 Trust Relationship Defined

Data protection deals with protecting the individual or organization's private data which is shared over the cloud. It is possible only when the security and trustworthiness of both the service provider and user is ensured. Therefore there is a firm need to establish trust between both and hence we need to develop a trust management model.

5.2 Types of Trust

Types of Trust

Blind trust: This is default trust before any event on the system, and which would include agent to initiate relationship with unknown entities.

Conditional trust: This is a classical state of trust during the life of the agent .This condition trust is likely to evolved ,and can be subject to some sets of constraints or condition .

Unconditional trust: Such a trust is the probability be configured directly by an administrator, and would not be sensitive to successful/unsuccessful interaction and external recommendation of any other sources of evolution of the conditional trust.

Direct Trust: Direct trust relationship is built through direct experience of interactions between the user and entity. At time t, user u's direct trust towards cloud service node c is formulized as DtJt).

Recommend trust: Recommend trust, or Rt, is recommended by some intermediate entity. At time t, recommend trust which the intermediate entity k gives the user u about cloud node c can be described as R(t).
The following formula shows a calculation method of recommended trust. R(t) = DtuJt) x Dtk,(t) (4) Actually, there is more than one intermediate entity which could provide the user with recommend trust. Different intermediate entities have different significance on trust values.

5.3 Related Work

Numerous trust models have been proposed in cloud. Mohda Izua Mohd Saad proposed a novel data provenance trusted model to provide secured access to data provenance via a secured communication channel. This model also propose a consolidation log storage for virtual and physical layer in cloud environment.

WenAn Tan proposed a trust service-oriented workflow scheduling algorithm. The scheduling algorithm adopts a trust metric that combines direct trust and recommendation trust. In addition, we provide balance policies to enable users to balance different requirements, including time, cost, and trust.

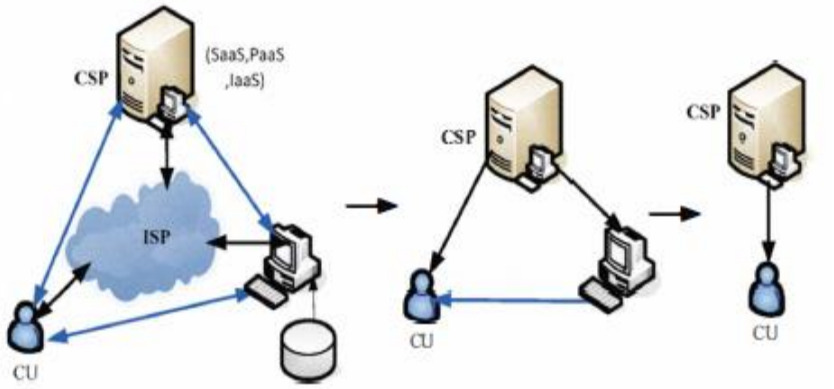


Figure 2. simplified trust model in cloud computing

Rizwana A.R. Shaikh propose a trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service. The trust value acts as a means for selecting any cloud service.

Xiaodong Sun introduces a trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust similarity has been addressed to prevent the behavior of associated cheat of middle nodes. And this model is geared toward

the cloud users who are making their decision on whether to use services of some cloud computing providers by giving them trust evaluation sets about providers and then building reasonable trust relationship between them.

Qiang Guo introduced a definition of trust in cloud systems and the properties of trust are analyzed. Based on the properties and semantics of trust, an extensible trust evaluation model named ETEC is proposed, which includes a time-variant comprehensive evaluation method for expressing direct trust and a space-variant evaluation method for calculating recommendation trust. To compute trust in cloud systems, an algorithm based on the ETEC model is given. Simulation and analysis shows that this model can calculate the trust degree effectively and reasonably in cloud computing environments.

Xiaoqiong Yang also proposed A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model for statistics-based behaviors which generates the threshold for each type of behaviors and each user's membership degree between the user's performance and the calculated threshold. And then the membership degree and the behavior weight will be together sent to calculate the user's behavior trust evaluation value using a simple normalized method. Finally as the basis of intra-domain trust and recommendation trust, behavior trust could help for users' further dynamic authorization of access control.

Junfeng Tian proposed a Trusted Control Model of Cloud Storage with access control (TCMCS) to handle all the interactions between a client and CSS to ensure the transparency of the data manipulation. It is TCMCS that shield the differences among different cloud storage platforms, which can separate operations that protected both security and integrity of data from users' applications. Since users only need to care about their own business logic and the development of application program is greatly simplified.

Punit Gupta proposed a QoS Based Trust Management Model for Cloud IaaS that is suitable for trust value management for the cloud IaaS parameters. Based on the above achieved trust values, a scheduling is done for better allocation of resources and further enhance the QOS of services been provided to the users. In this paper, an approach for managing trust in Cloud IaaS is proposed.

5.4 Review Work

Title	Improved	Proposal
A Cloud Trust Model in a Security Aware Cloud	- Multiple stakeholder problem - Open space security problem - Mission critical data	- In addition to conventional trust models, consider both internal trust and contracted trust that controls cloud service providers under contracts and related documents. - Two additional layers of trust.

	handling problem	
Evaluation of User Behavior Trust in Cloud Computing	- Trust supervision	- Discusses evaluation importance of user behavior trust and evaluation strategy in the cloud computing, including trust object analysis, principle on evaluating user behavior trust, basic idea of evaluating user behavior trust, evaluation strategy of behavior trust for each access, and long access, which laid the theoretical foundation of trust for the practical cloud computing application.
Modeling and Evaluation of Trust in Cloud Computing Environments	- Trust evaluation	- A time-variant comprehensive evaluation method for expressing direct trust. - A space-variant evaluation method for calculating recommendation trust. - To compute trust in cloud systems, an algorithm based on the ETEC model is given.
Research on Trust-Based Access Control Model in Cloud Computing	- Dynamic and secure access control.	- A trust-based dynamic access control model for cloud computing environment inspired by the GTRBAC model, where the users can validate their legal identities and acquire their access control privileges for the resources according to the role information and the trust-degree in the lightweight certificates.
A Trust Management Model to enhance security of Cloud Computing Environments	- Trust management - Trust relationship	- A trust management model based on fuzzy set theory and named TMFC including direct trust measurement and computing, connecting, and trust chain incorporating where the issue of recommended trust similarity has been addressed to prevent the behavior of

		associated cheat of middle nodes.
A Trusted Control Model of Cloud Storage	- Security and Efficiency in data sharing.	- A trusted control model of cloud storage (TCMCS) is presented. - With the using of cipher-text access control and integrity verification, TCMCS makes the users' data safe. - Considering the security and efficiency in data sharing, we introduced the trusted third party (TTP).
A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model	- Trust Evaluation	- A statistical user-behavior trust evaluation algorithm based on cloud model for statistics-based behaviors which generates the threshold for each type of behaviors and each user's membership degree between the user's performance and the calculated threshold. - As the basis of intra-domain trust and recommendation trust, behavior trust could help for users' further dynamic authorization of access control.
QoS Based Trust Management Model for Cloud IaaS	- Scheduling of resources	- A suitable trust model based on the existing model that is suitable for trust value management for the cloud IaaS parameters. - Based on the above achieved trust values, a scheduling is done for better allocation of resources and further enhance the QOS of services been provided to the users. - In this paper, an approach for managing trust in Cloud IaaS is proposed.
Adaptive and attribute based trust model for service level agreement guarantee in cloud computing	- Selecting more trustworthy service provider.	- An innovative computing method for GTD based on the direct trust attributes which can well satisfy two social properties of trust: the dynamic nature and the real-time nature.

Trust Model Engines in cloud computing	- Trust management	- A trust management system can be achieved in cloud computing by implementing multiple trust models that define what a user can do and what type of protection he has. - By identifying the static and dynamic trust models, the cloud provider and the cloud user can have a proper control of the cloud security.
Trust Model for a Cloud Computing Application and Service	- Trust value	- A trust based solution in terms of a trust model that can be used to calculate the security strength of a particular cloud service. The trust value acts as a means for selecting any cloud service.
A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing	- Balance different requirements including time cost and trust.	- A trust service-oriented workflow scheduling algorithm. - The scheduling algorithm adopts a trust metric that combines direct trust and recommendation trust. - In addition, balance policies are provided to enable users to balance different requirements, including time, cost, and trust.
Assessment Criteria for Trust Models in Cloud Computing	- Trust models in cloud	- An assessment criterion for the evaluation of trust models, containing the essential features that are mandatory for trust establishment in Cloud environment. - Also presented a detailed analysis of existing trust models and analyzed them with respect to our proposed assessment criteria.
Data Provenance Trusted Model in Cloud Computing	- Accountability - Trust	- A novel data provenance trusted model to provide secured access to data provenance via a secured communication channel. - This model also propose a consolidation log storage for virtual and physical layer in cloud environment.

MTBAC: A Mutual Trust Based Access Control Model in Cloud Computing	- Security - Trust	- A mutual trust based access control (MTBAC) model is proposed in this paper. - MTBAC model take both user's behavior trust and cloud services node's credibility into consideration. - Trust relationships between users and cloud service nodes are established by mutual trust mechanism. Security problems of access control are solved by implementing MTBAC model into cloud computing environment.
A Centralized Trust Model Approach for Cloud Computing.	- Real Time values for trust.	- A new trust model that involves all the cloud stakeholders such as CSU's, CSP's and third party auditors.

5.5 Required Parameters

- ***Memory***: RAM
- ***MIPS***: Million Instruction per Cycles
- ***Fault Rate***: Number of task failed to the total tasks or the number of faults over a period of time.
- ***Response Time***: Time taken to accomplish a standard request
- ***Direct Trust*:** Direct trust relationship is built through direct experience of interactions between the user and entity. At time t, user u's direct trust towards cloud service node c is formulized as DtJt).
- ***Recommend trust*:** Recommend trust, or Rt, is recommended by some intermediate entity. At time t, recommend trust which the intermediate entity k gives the user u about cloud node c can be described as R(t).The following formula shows a calculation method of recommended trust. R(t) = DtuJt) x Dtk,(t) (4) Actually, there is more than one intermediate entity which could provide the user with recommend trust. Different intermediate entities have different significance on trust values.

- ***Trust Degree***: Trust degree represents the tendency which entity the user would choose to interact. At time t, trust degree of cloud service node c is expressed as Tc (t), where Tc (t) E [0, 1]. Tc (t) = I represents that user u trusts node c completely. Tc (t) =0 means that user u does not trust node c at all. Trust degree is composed of direct trust and recommended trust.

5.6 Proposed Trust Model

The Trust Model we are proposing here will work for Private, Public and Hybrid Cloud. It will take into the consideration both Direct and Indirect Trust or recommended trust.

For the Calculation of Trust Value we will take Memory (RAM), MIPS (Million Instruction per Cycle), Frequency (Frequency of data center) and Fault Rate.

We will initially calculate only Direct Trust and as the time will evolve we will also consider the indirect trust.

We will initiate the parameter MIPS and Fault with zero and Memory and MIPS will be according to the datacenter. With these we will calculate the Trust Value and initial load balancing will be done according to that.

As the Time will evolve we will calculate the Indirect or Recommended Trust also using Fault Rate which we initially considered zero and Response Time.

Now since the request from both public and private cloud will start arriving we assume that some of them will not be accomplished due to technical faults and hence the parameter Fault Rate will have some value other than zero.

After Calculating Values for both we will rate the datacenters according to the trust value calculated, This Trust value will be combined of Direct and Indirect Trust.

And hence allocate the private request to a server with high trust value since the request has lower chances of going down and allocate the public request to a server with low trust value since the request has higher chances of going down.

We will keep this dynamic process going in order to ensure a Trust Based Load Balancing.

5.7 Pseudo Code

Load Balancing Algorithm

1: Start

2: Initialize servers

3: Calculate individual trust values of each data center

4: CheckDirect _TrustValue ( )

5: Now according the size of request start dividing them into public or private

6: If (Data Center Value > Threshold)
Send the private cloud requests to since they are less likely to fail.

7: Else If (Data Center Value < Threshold)

Send the public cloud requests to since they are more likely to fail.
8: Else

Keep Searching

9: CheckIndirect _Value ( )

CheckDirect _TrustValue ( )

- **Take these factors into consideration for calculation of trust value for Direct Trust:**
  - **RAM (Memory)**
  - **MIPS (Microprocessor without Interlocked Pipeline Stage)**
  - **Frequency of data center**
  - **Fault Rate (initially 0)**
- **Where T = α1 * RAM + α2 * MIPS + α3 * Frequency**
- **And α1 + α2 + α3 =1**
- **Also α1 + α2 = (proportion in which these effect the trust value)**

CheckIndirect _Value ( )

- **Calculate the Indirect Trust which is based on:**
  - **Fault Rate(number of task failed/total tasks)**
  - **Response Time (Time taken to accomplish a standard request.)**

Chapter 6
Experiments and Results

6.1 Simulation on Cloudsim

Figure 6.1: A simple example showing how to create a datacenter with one host and run one cloudlet on it

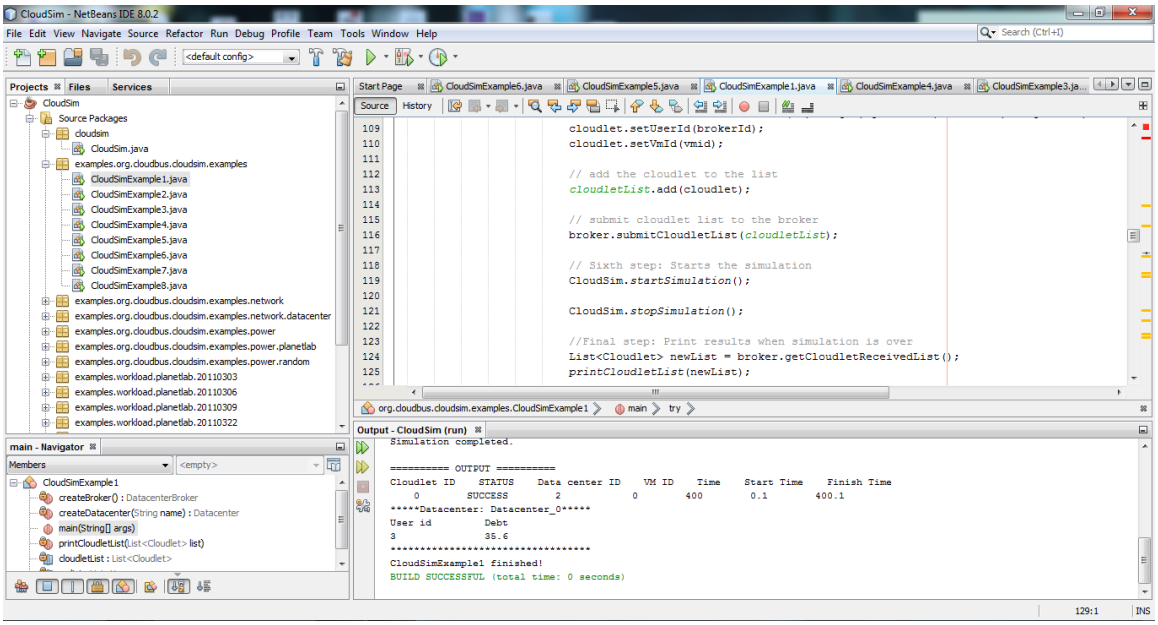


Figure 6.2: A simple example showing how to create a datacenter with one host and run two cloudlets on it. The cloudlets run in VMs with the same MIPS requirements. The cloudlets will take the same time to complete the execution.

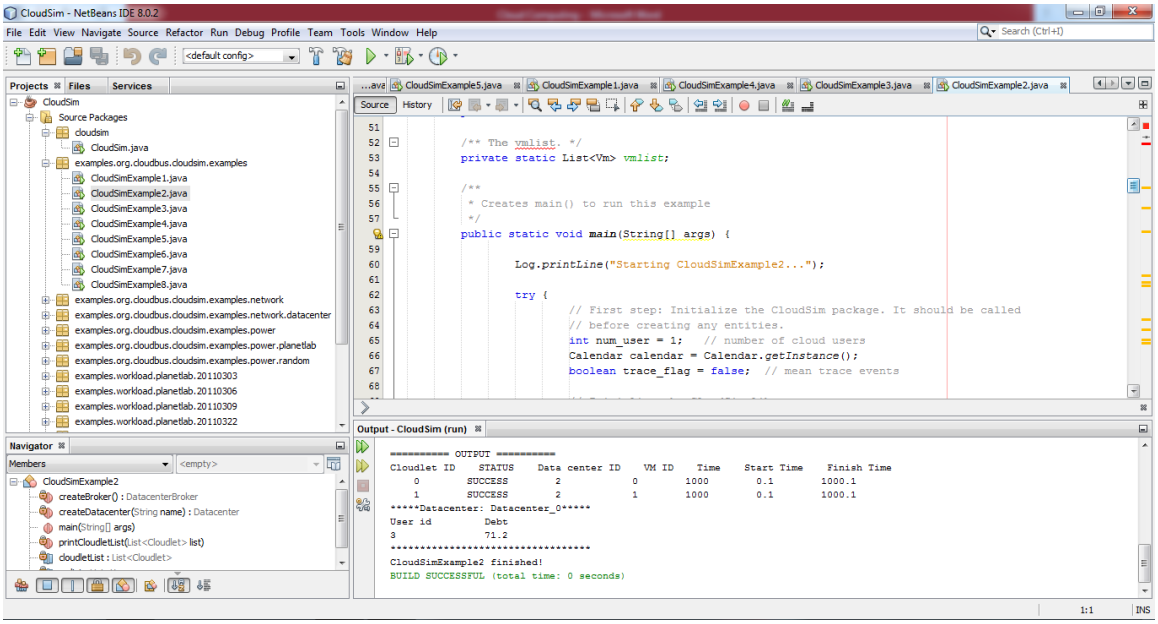


Figure 6.3: A simple example showing how to create a datacenter with two hosts and run two cloudlets on it. The cloudlets run in VMs with different MIPS requirements. The cloudlets will take different time to complete the execution depending on the requested VM performance.

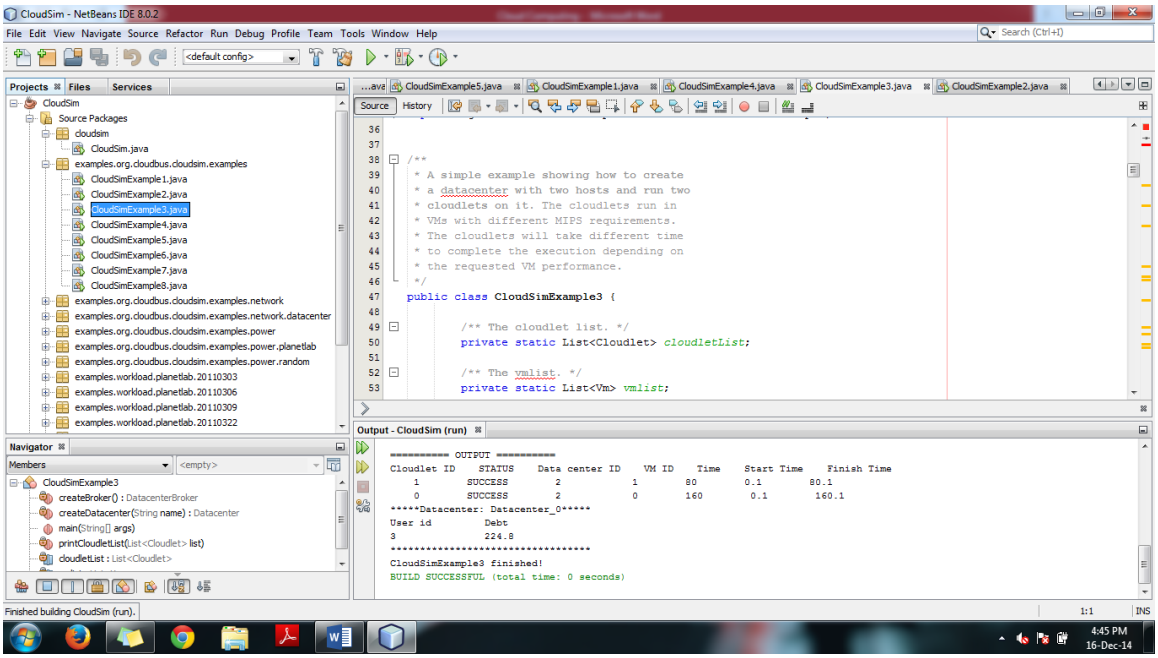


Figure 6.4: A simple example showing how to create two datacenters with one host each and run two cloudlets on them.

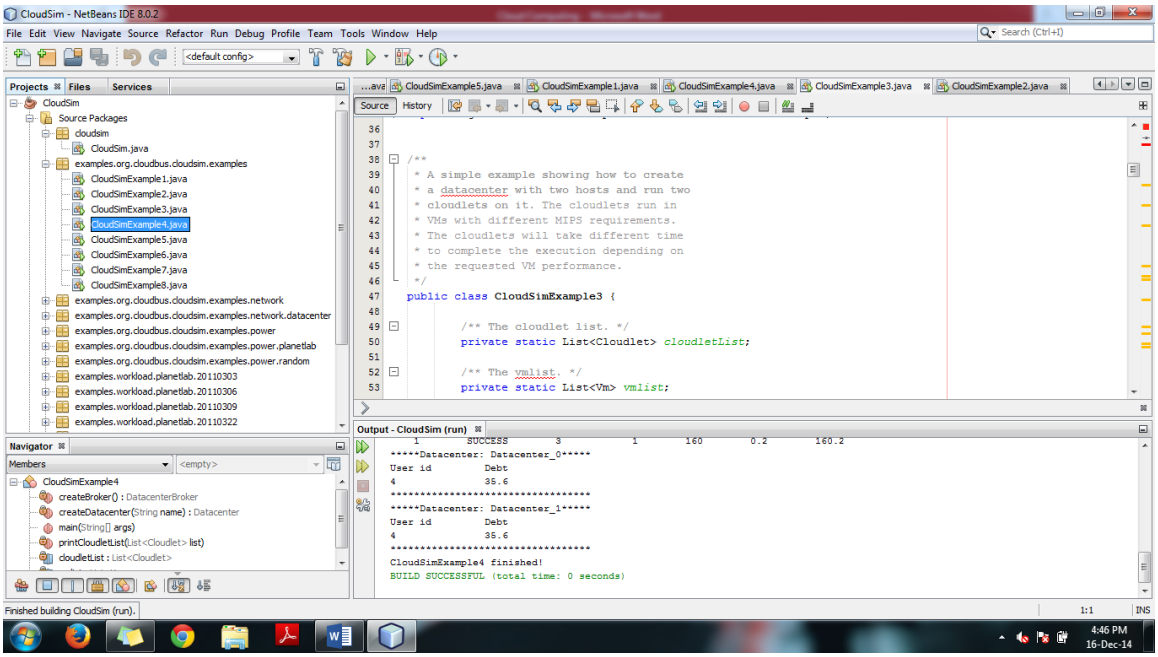


Figure 6.5: A simple example showing how to create two datacenters with one host each and run cloudlets of two users on them.

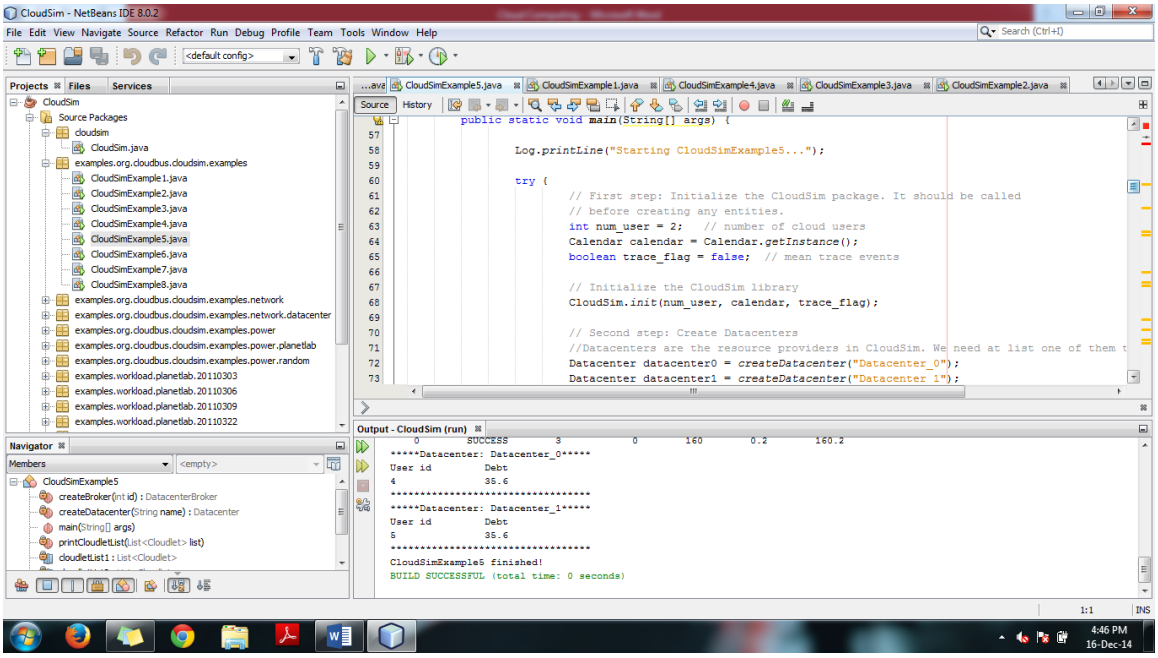
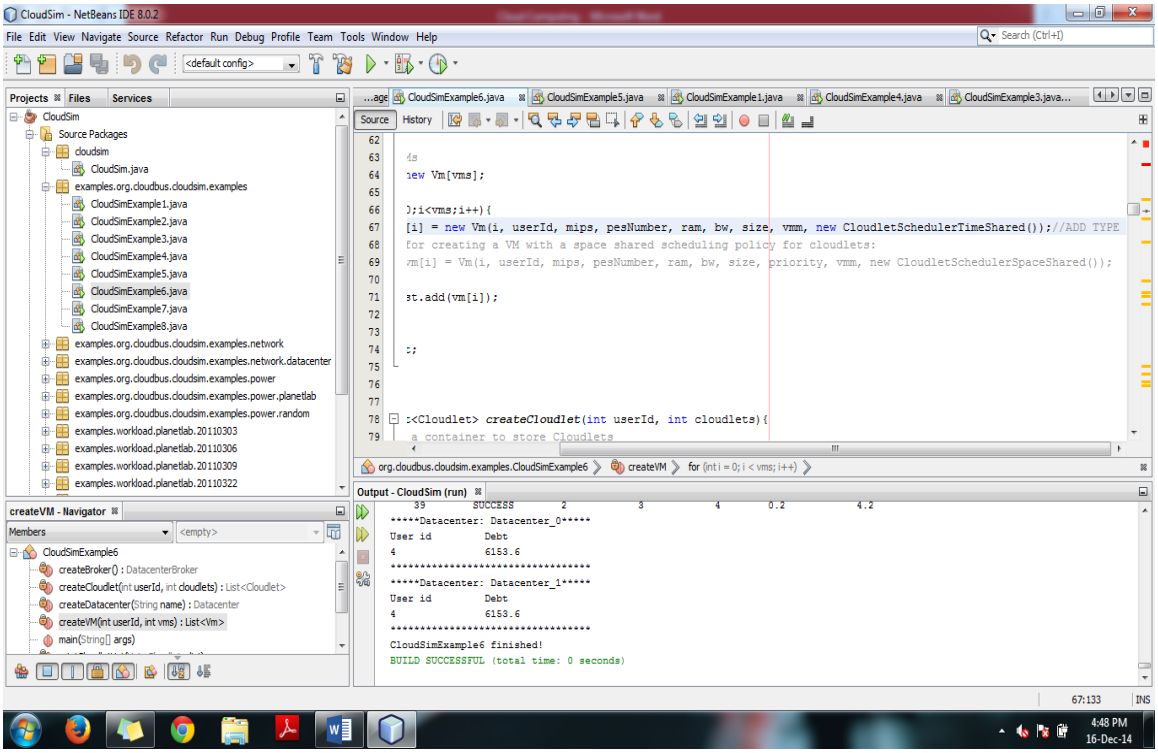
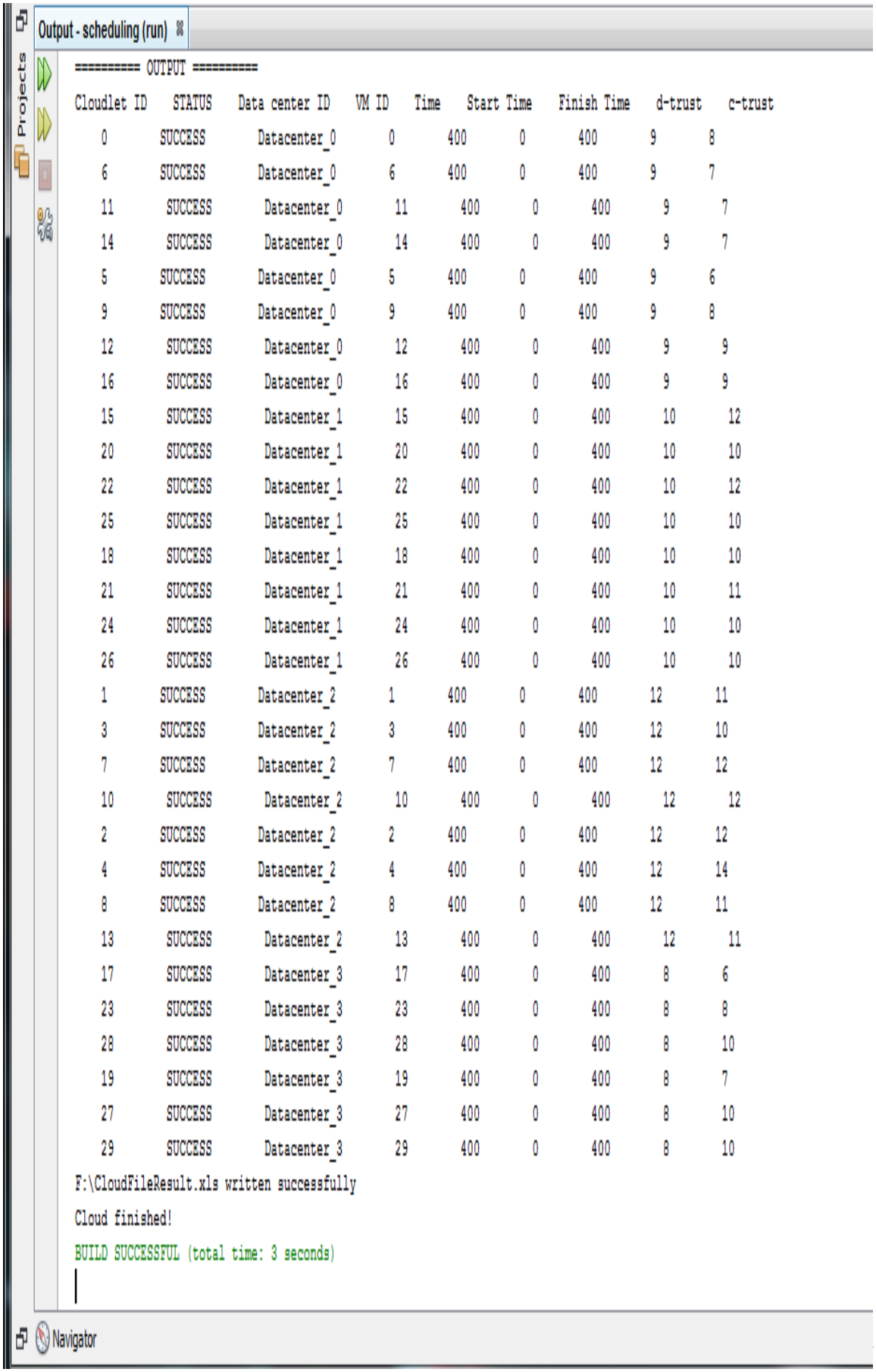


Figure 6.6: An example showing how to create scalable simulations.






```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time    d-trust    c-trust
    0        SUCCESS        Datacenter_0        0        400        0        400        9        8
    6        SUCCESS        Datacenter_0        6        400        0        400        9        7
    11        SUCCESS        Datacenter_0        11        400        0        400        9        7
    14        SUCCESS        Datacenter_0        14        400        0        400        9        7
    5        SUCCESS        Datacenter_0        5        400        0        400        9        6
    9        SUCCESS        Datacenter_0        9        400        0        400        9        8
    12        SUCCESS        Datacenter_0        12        400        0        400        9        9
    16        SUCCESS        Datacenter_0        16        400        0        400        9        9
    15        SUCCESS        Datacenter_1        15        400        0        400        10        12
    20        SUCCESS        Datacenter_1        20        400        0        400        10        10
    22        SUCCESS        Datacenter_1        22        400        0        400        10        12
    25        SUCCESS        Datacenter_1        25        400        0        400        10        10
    18        SUCCESS        Datacenter_1        18        400        0        400        10        10
    21        SUCCESS        Datacenter_1        21        400        0        400        10        11
    24        SUCCESS        Datacenter_1        24        400        0        400        10        10
    26        SUCCESS        Datacenter_1        26        400        0        400        10        10
    1        SUCCESS        Datacenter_2        1        400        0        400        12        11
    3        SUCCESS        Datacenter_2        3        400        0        400        12        10
    7        SUCCESS        Datacenter_2        7        400        0        400        12        12
    10        SUCCESS        Datacenter_2        10        400        0        400        12        12
    2        SUCCESS        Datacenter_2        2        400        0        400        12        12
    4        SUCCESS        Datacenter_2        4        400        0        400        12        14
    8        SUCCESS        Datacenter_2        8        400        0        400        12        11
    13        SUCCESS        Datacenter_2        13        400        0        400        12        11
    17        SUCCESS        Datacenter_3        17        400        0        400        8        6
    23        SUCCESS        Datacenter_3        23        400        0        400        8        8
    28        SUCCESS        Datacenter_3        28        400        0        400        8        10
    19        SUCCESS        Datacenter_3        19        400        0        400        8        7
    27        SUCCESS        Datacenter_3        27        400        0        400        8        10
    29        SUCCESS        Datacenter_3        29        400        0        400        8        10
F:\CloudFileResult.xls written successfully
Cloud finished!
BUILD SUCCESSFUL (total time: 3 seconds)
```

Figure 2 Trust Algorithm Simulation

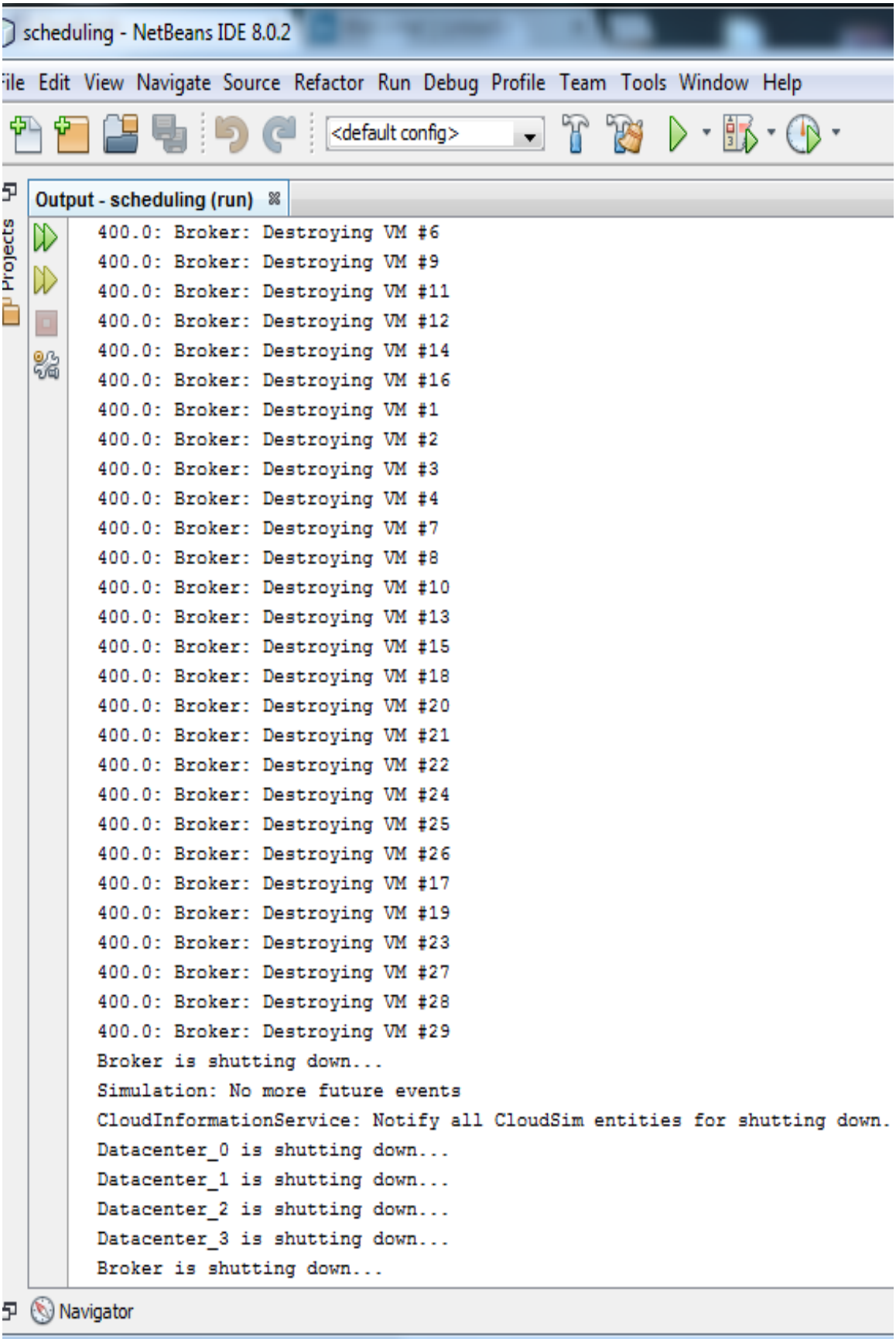


Figure 3Trust Algorithm Simulation

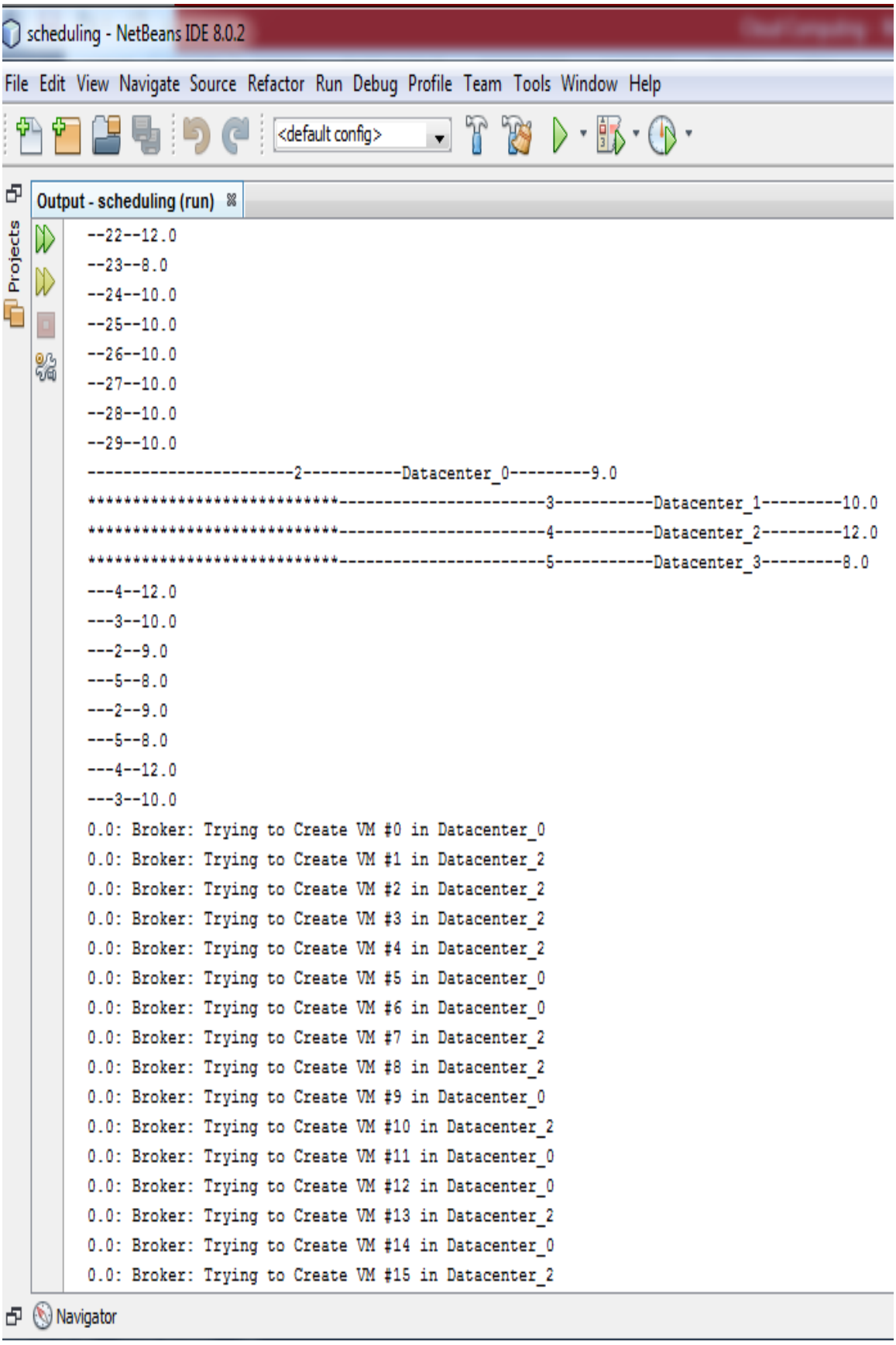


Figure 4Trust Algorithm Simulation

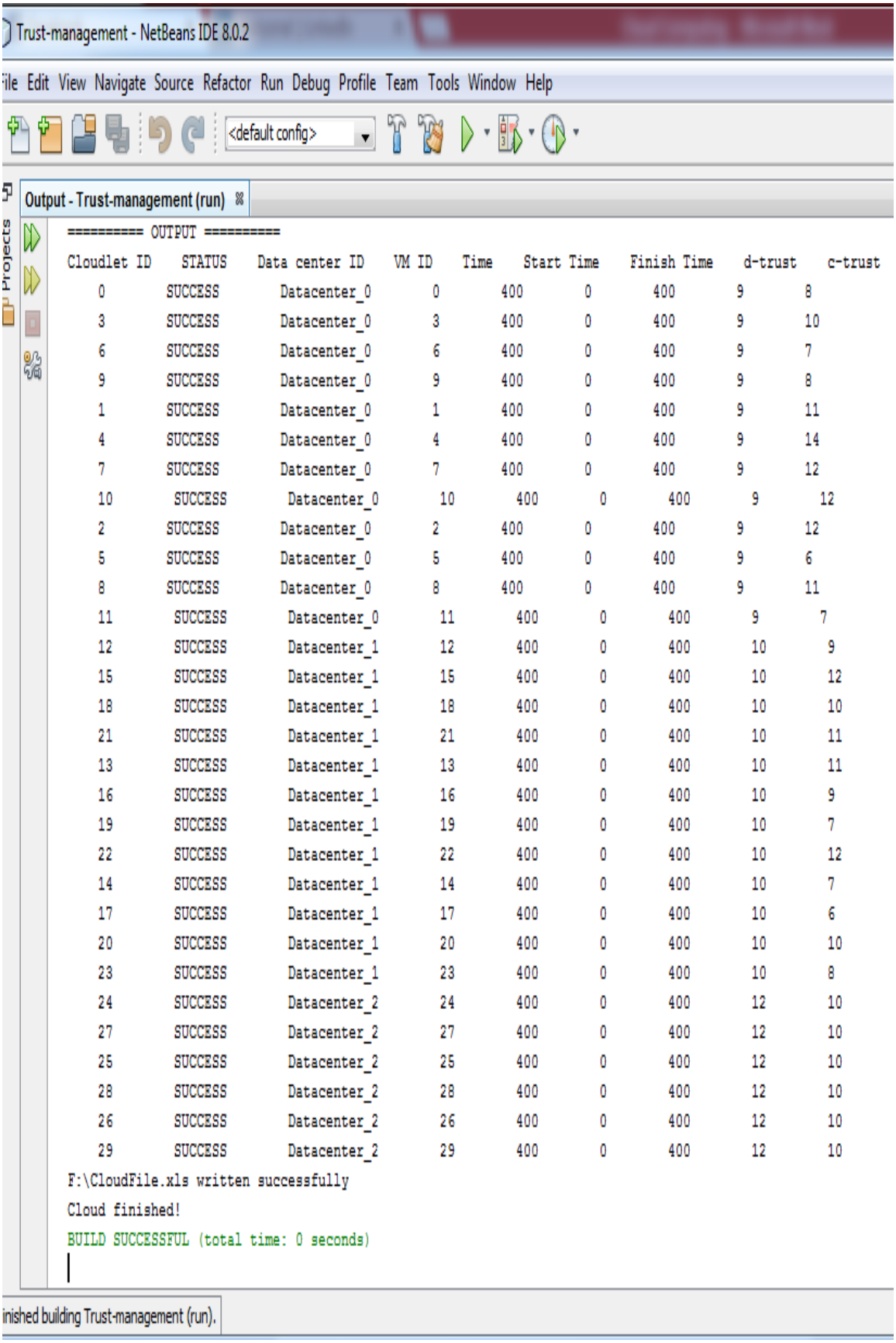
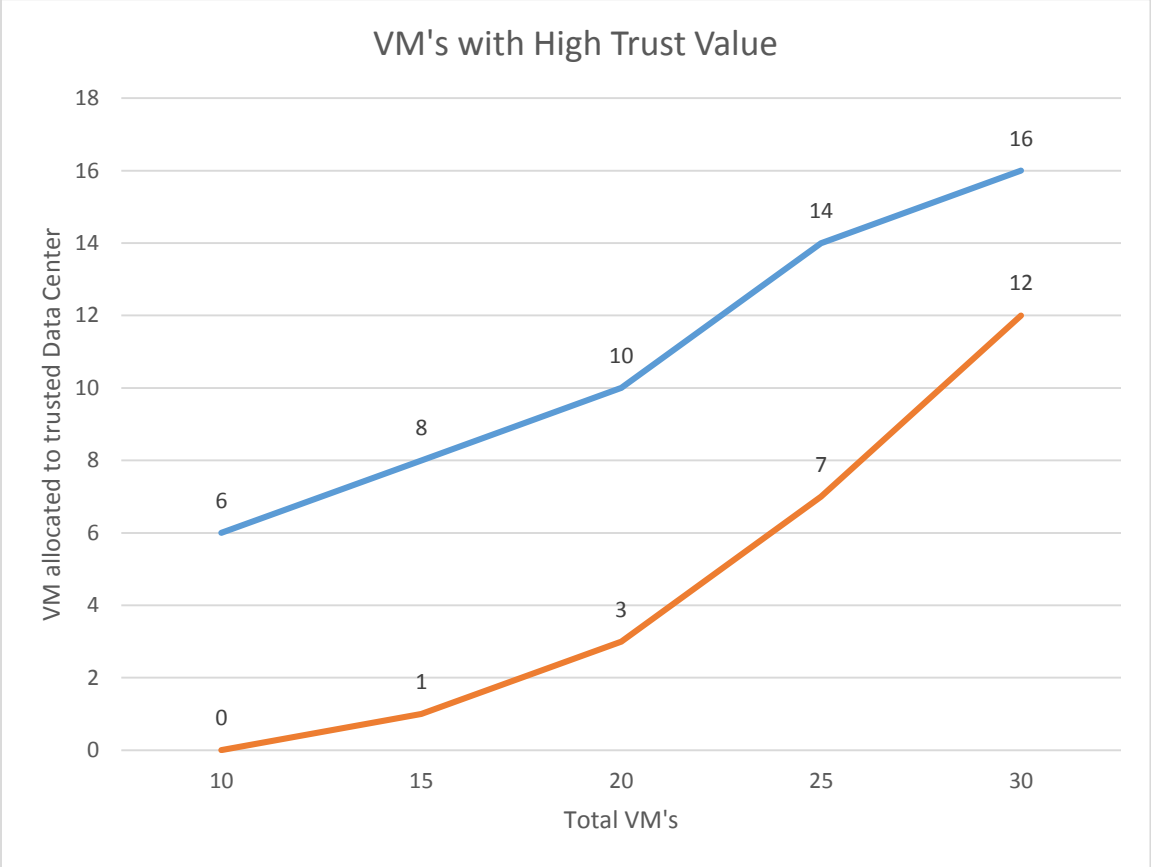


Figure 5Trust Algorithm Simulation

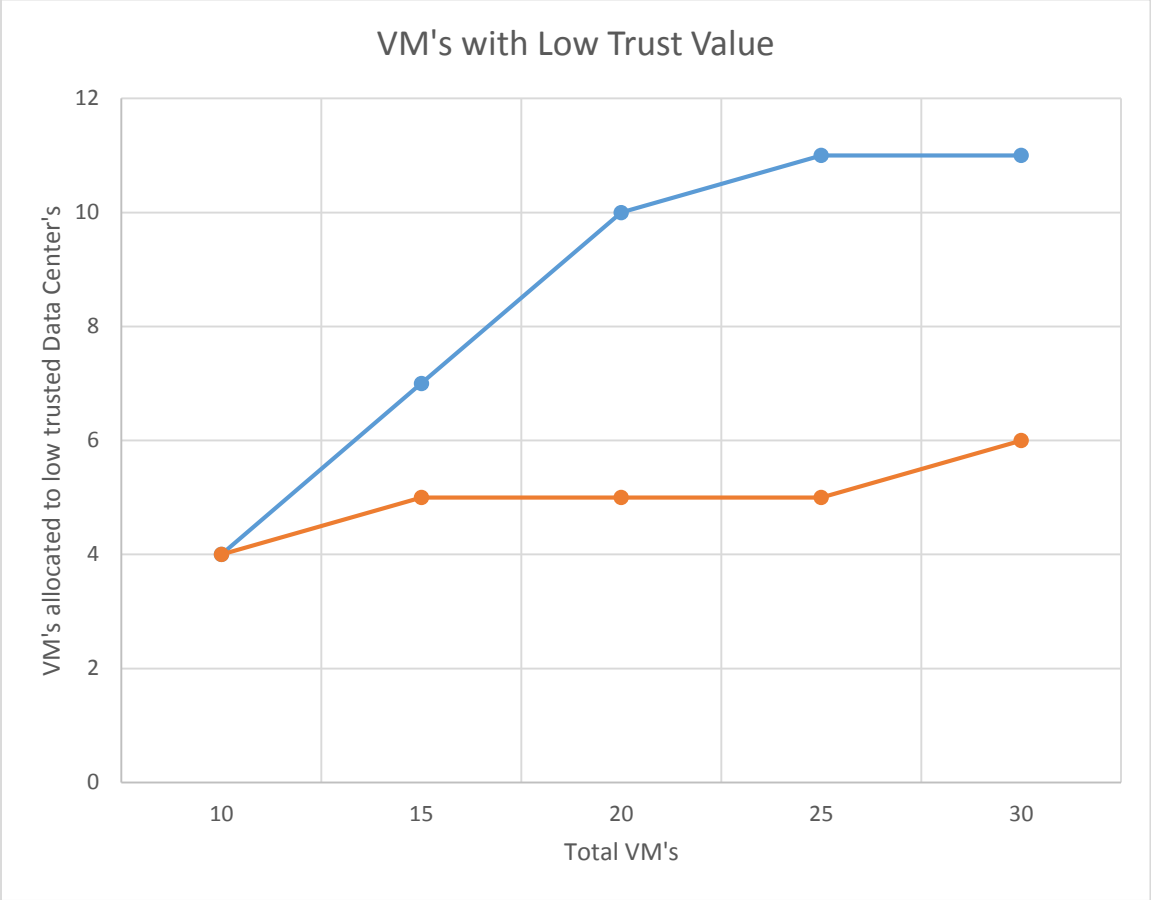
6.2 Results and Graphs

Total VM's	VM allocated (with trust model)	VM allocated (without trust model)
10	6	0
15	8	1
20	10	3
25	14	7
30	16	12



Series 1 represents trust based and Series 2 represents existing Algorithm

Total VM's	VM allocated (with trust model)	VM allocated (without trust model)
10	4	4
15	7	5
20	10	5
25	11	5
30	11	6



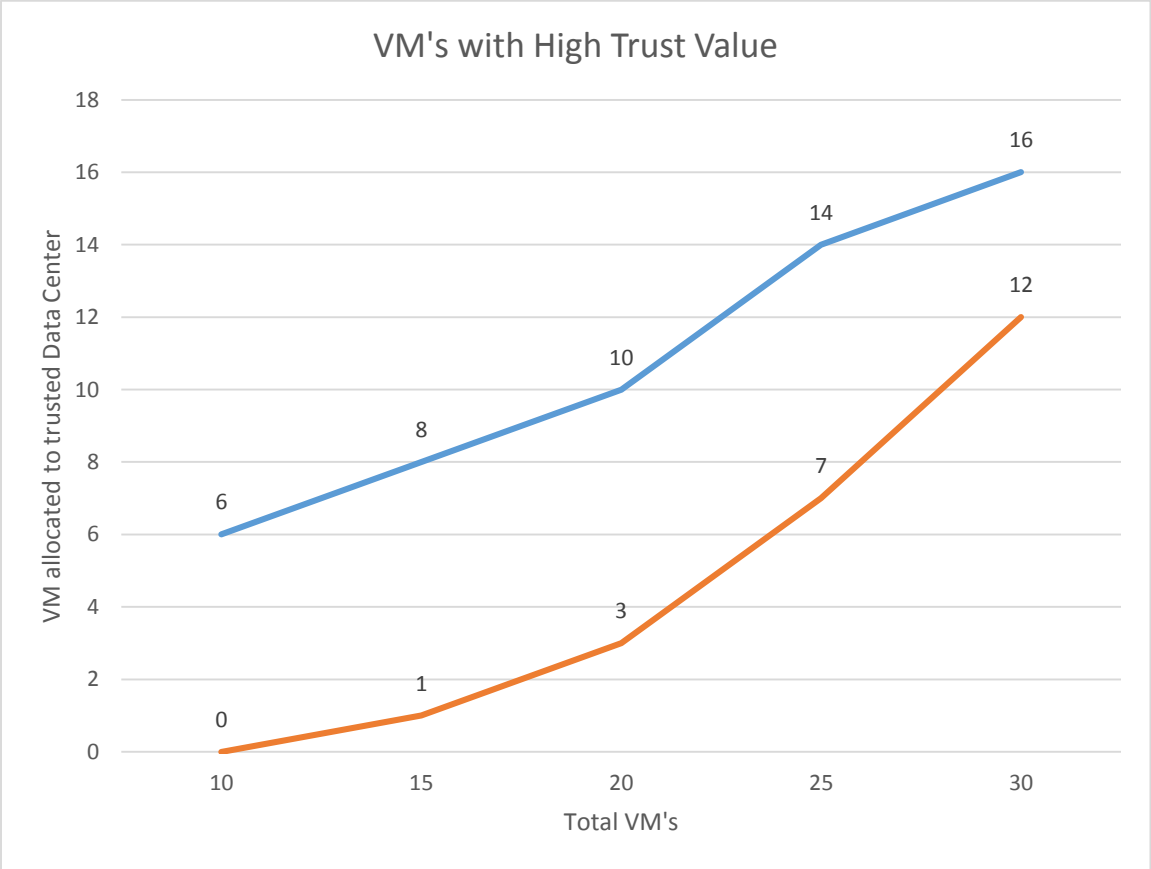
Series 1 represents trust based and Series 2 represents existing Algorithm

Chapter 7
Conclusion and Future Work

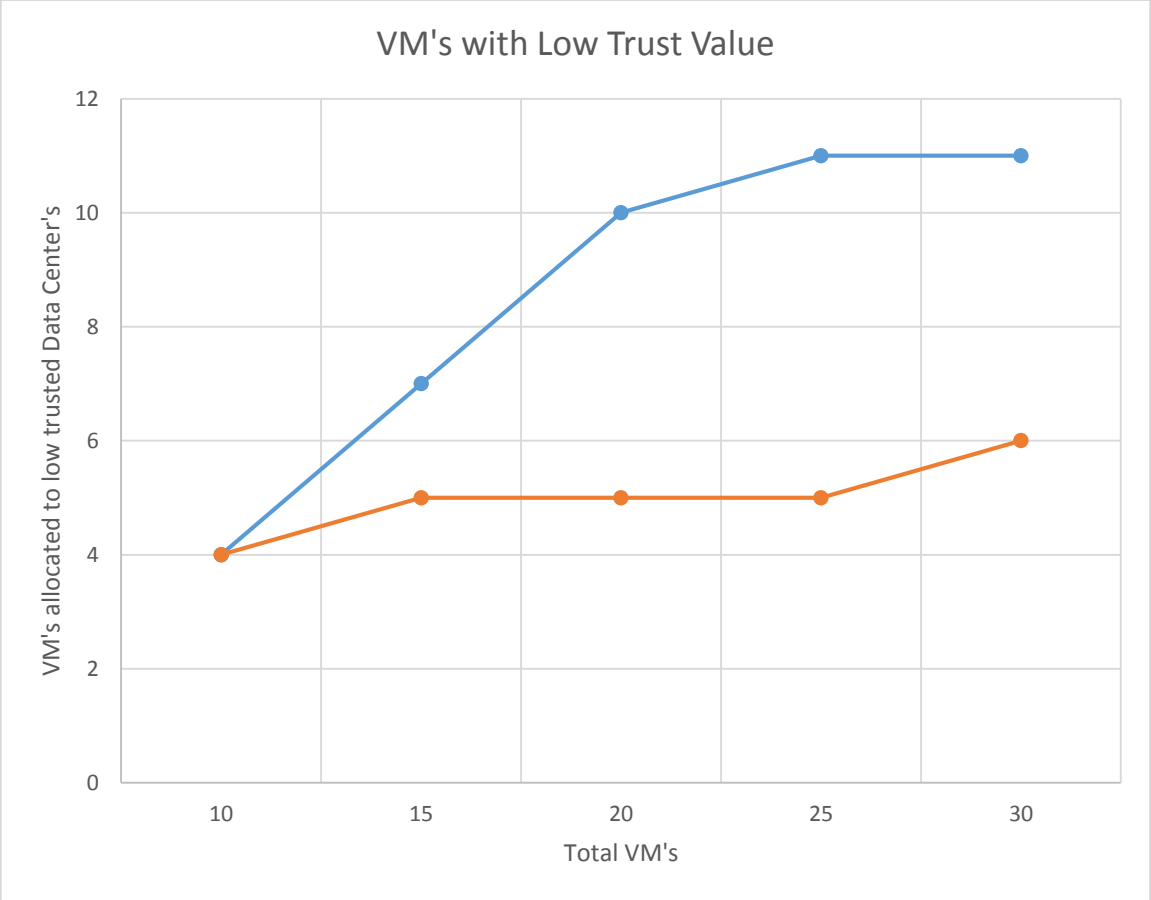
7.1 Conclusion

From the graphs showed in Experiments and Results it can be clearly concluded that the allocation of high trust VM's on data center with high trust value and low trust VM's on data center with low trust value is better from the trust based algorithm we have developed as compared to the existing algorithm.

The trust based algorithm we have developed is running efficiently and the results are quite comparable. This trust based algorithm will help VM's to run for maximum time without any fault rate for both private and public clouds.



Series 1 represents trust based and Series 2 represents existing Algorithm



Series 1 represents trust based and Series 2 represents existing Algorithm

7.2 Future Work

Since cloud IaaS is an emerging field. More and more algorithms can be further developed for the scheduling of VM's. Also more parameters like frequency, cost, reliability, fault rate, tolerance can be included for better analysis and scheduling of VM's. Also algorithms for Public, Private, Hybrid and Community clouds can be developed keeping in mind their individual specifications and needs.

In a nutshell there is a lot more scope for further development and making these algorithms 100 percent reliable and trustworthy developing a mutual trusted relation between client and the service provider.

Chapter 8
References

8.1 Research Papers

[1] H. Sato, A. Kanai, and S. Tanimoto, “A Cloud Trust Model in a Security Aware Cloud,” 2010.

[2] L. Ian, “Evaluation of User Behavior Trust in Cloud Computing,” no. Iccasm 2010, pp. 567–572, 2015.

[3] Z. Tan, Z. Tang, R. Li, A. Sallam, and L. Yang, “Research on Trust-Based Access Control Model in Cloud Computing.”

[4] X. Sun, “A Trust Management Model to enhance security of Cloud Computing Environments,” pp. 0–4, 2011.

[5] J. Tian and Z. Wu, “2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring A Trusted Control Model of Cloud Storage,” 2012.

[6] X. Yang, “A Statistical User-Behavior Trust Evaluation Algorithm Based on Cloud Model.”

[7] M. K. Goyal, “QoS Based Trust Management Model for Cloud IaaS,” pp. 843–847, 2012.

[8] X. Li and J. Du, “Adaptive and attribute-based trust model for service- level agreement guarantee in cloud computing,” no. February 2012, pp. 39–50, 2013.

[9] M. R. Farcasescu, “Trust Model Engines in cloud computing,” pp. 1–6, 2012.

[10] R. A. R. Shaikh and M. Sasikumar, “Trust Model for a Cloud Computing Application and Service,” 2012.

[11] W. Tan, Y. Sun, L. X. Li, G. Lu, and T. Wang, “A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing,” pp. 1–11, 2013.

[12] A. Kanwal, R. Masood, U. E. Ghazia, M. A. Shibli, and A. G. Abbasi, “Assessment Criteria for Trust Models in Cloud Computing,” 2013.

[13] M. Izuan, M. Saad, K. A. Jalil, and M. Manaf, “Data Provenance Trusted Model in Cloud Computing,” vol. 2013, 2013.

[14] L. I. N. Guoyuan, W. Danrul, B. I. E. Yuyul, and L. E. I. Min, “MTBAC : A Mutual Trust Based Access Control Model in Cloud Computing,” no. April, pp. 154–162, 2014.

8.2 Web URL's

- http://en.wikipedia.org/wiki/Cloud_computing
- http://cloudsim-setup.blogspot.in/
- http://www.interoute.com/what-iaas
- http://www.rackspace.co.uk/cloud/load-balancers
- http://searchnetworking.techtarget.com/definition/load-balancing
- http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS
- http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5636125&abstractAccess=no&userType=inst
- http://www.cloudbus.org/cloudsim/doc/readme.txt
- http://onlinelibrary.wiley.com/doi/10.1002/spe.995/abstract