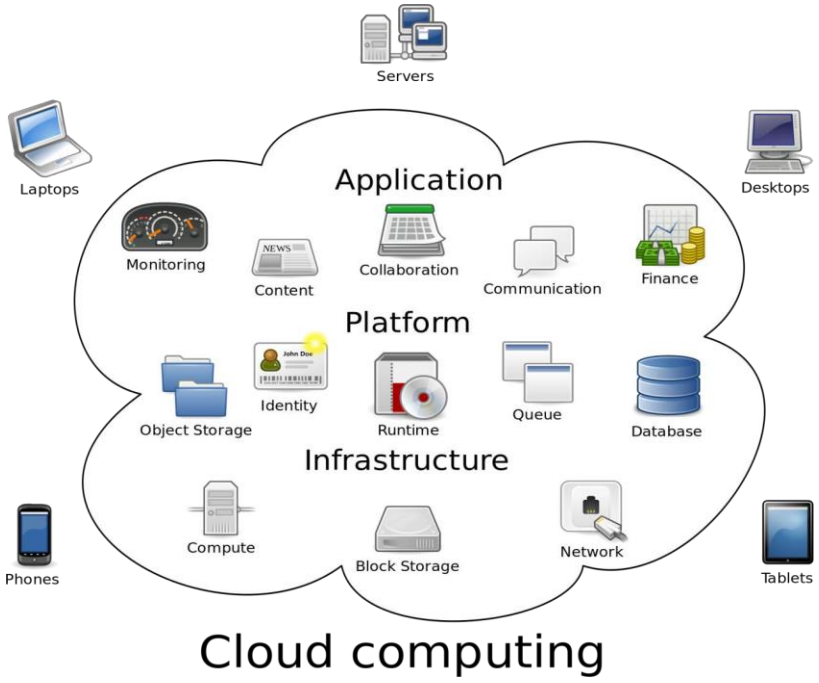
Department of Computer Sci. &Engineering

LAB MANUAL



MCA

High Performance Computing Paradigms and Applications

SSBT'S COLLEGE OF ENGINEERING & TECHNOLOGY
BAMBHORI

SSBT'S COLLEGE OF ENGINEERING & TECHNOLOGY

BAMBHORI

Laboratory Manual

High Performance Computing Paradigms and Applications

For

Final Year Students MCA

Dept: Computer Science & Engineering

DOs and DON'Ts in Laboratory:

1. Make entry in the Log Book as soon as you enter the Laboratory.

2. All the students should sit according to their roll numbers starting from their left to right.
3. All the students are supposed to enter the terminal number in the log book.

4. Do not change the terminal on which you are working.

5. All the students are expected to get at least the algorithm of the program/concept to be implemented.
6. Strictly observe the instructions given by the teacher/Lab Instructor.

Instruction for Laboratory Teachers:

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

COLLEGE OF ENGINEERING & TECHNOLOGY

BAMBHORI

Department of Computer Science and Engineering

Vision of CSE Department

To emerge as the leading Computer Engineering department for inclusive development of students.

Mission of the CSE Department

To provide student-centered conducive environment for preparing knowledgeable, competent and value-added computer engineers.

SUBJECT INDEX

13	Case Study: Amazon Web Services.	

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 1

Aim: Study and do the Configuration of CLoudSim. Also execute & check the performance of existing
algorithms.

Theory

1. Background

Cloud computing is a pay as you use model, which delivers infrastructure (IaaS), platform (PaaS) and software (SaaS) as services to users as per their requirements. Cloud computing exposes data centers capabilities as network virtual services, which may set of required hardware, application with support of database as well as user interface. This allows the the users to deploy and access application across the internet which is based on demand and QoS requirements. As Cloud computing is a new concept and is still in a very early stage of its evolutions, so researchers and system developers are working on improving the technology to deliver better on processing, quality & cost parameters but most of the research is focused on quantifying the performance of provisioning policies. And to test such research on real cloud environment like Amazon EC2, Microsoft Azure, Google App Engine for different applications models under transient conditions is extremely challenging as:

1. Clouds exhibit varying demands, supply patterns, system sizes, and resources (hardware,software, and network).
2. Users have heterogeneous, dynamic, and competing QoS requirements.
3. Applications have varying performance, workload, and dynamic application scalingrequirements.

Benchmarking the application performance on the real public cloud infrastructure like google cloud, Microsoft azure etc., is not suitable due to their multitenant

nature as well as variable workload fulfillments. Also, there are very few admin level configurations that a user/researcher could be able to perform. Hence, this makes the reproduction of results that can be relied upon, an extremely difficult undertaking. Further, even if the we do, it is tedious and time consuming to re-configure benchmarking parameters across a massive-scale Cloud computing infrastructure over multiple test runs. therefore, it is impossible to undertake benchmarking experiments as repeatable, dependable, and scalable environments using real-world public Cloud systems.

Thus the need ot use of simulation tool(s) arises, which may become a viable alternative to evauate/benchmark the test workloads in a controlled and fully configurable environment which can reapatable over multiple iterations and reproduce the results for analysis. This simulation based approach various benefits across the researchers community as it allow them to:

1. Test services in repeatable and controllable environment.
2. Tuning the system bottlenecks (performance issues) before deploying on real clouds.
3. Evaluating different workload mix and resource performance scenarios on simulatedinfrastructures for developing and testing adaptive application provisioning techniques.

There are number of simulators that can be used to simulate the working of new services, and CloudSim is the more generalized and effective simulator for testing Cloud computing related hypothesis. CloudSim is an extensible simulation framework that allows seamless modeling, simulation and Experimentation relate to emerging cloud computing infrastructures and application services. Using CloudSim researchers and industry-based developers can test the performance of a newly developed
application service in a controlled and easy to set-up environment. Based on the evaluation resultsreported by CloudSim, they can further fine-tune the service performance.

2. Features of CloudSim

1. Support for modeling and simulation of large scale Cloud computing environments, including data centers, on a single physical computing node.
2. A self-contained platform for modeling Clouds, service brokers, provisioning, and allocation policies.
3. Support for simulation of network connections among the simulated system elements.
4. Facility for simulation of federated Cloud environment that inter-networks resources from both private and public domains, a feature critical for research studies related to Cloud-Bursts and automatic application scaling.
5. Availability of a virtualization engine that aids in the creation and management of multiple, independent, and co-hosted virtualized services on a data center node.
6. Flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services. These compelling features of CloudSim would speed up the development of new application provisioning algorithms for Cloud computing.

3. CloudSim Architecture

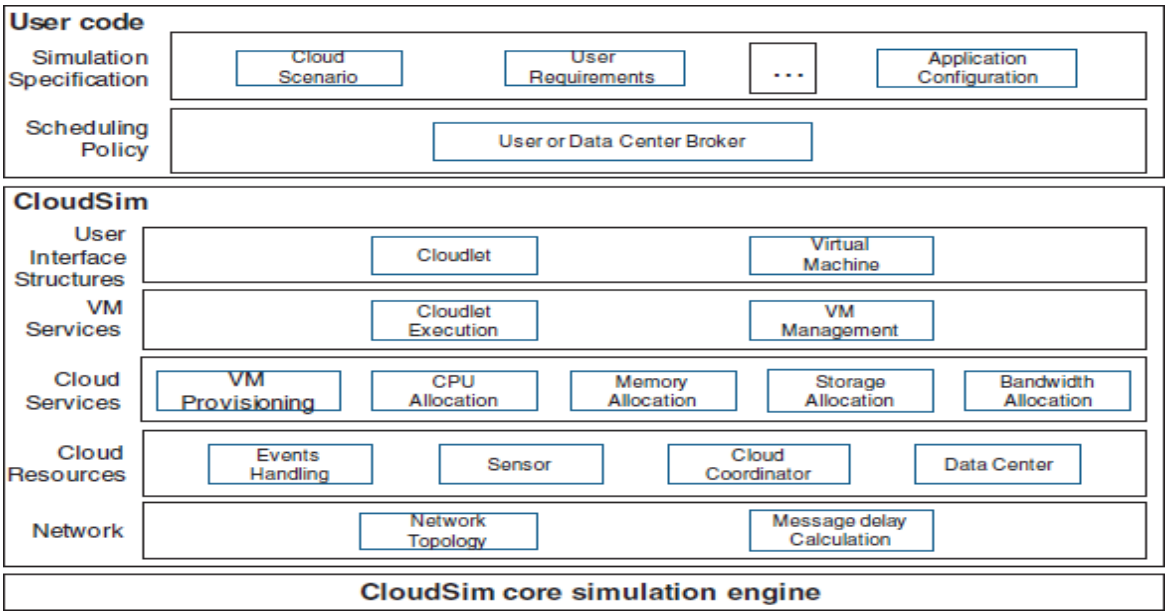


Figure 1: Layered CloudSim Architecture

Above diagram demonstrates about the layered architecture of CloudSim. The **CloudSim Core simulation engine** provides support for modeling and simulation of virtualized Cloud-based data center environments including queuing and processing of events, creation of cloud system entities (like data center, host, virtual machines, brokers, services etc.) communication between components and management of the simulation clock. The **CloudSim layer** provides the dedicated management interfaces for Virtual Machines, memory, storage, and bandwidth. Also it manages the other fundamental issues, such as provisioning of hosts to Virtual Machines, managing application execution, and monitoring dynamic system state(e.g. Network topology, sensors, storage characteristics etc) etc.

And the **User Code layer** is a custom layer where user write their own code to redefine thecharacteristics of the simulating environment as per their new research findings.

Design and Implementation of CloudSim

In this section, we provide finer details related to the fundamental classes of

CloudSim, which are also the building blocks of the simulator. The overall class design diagram for CloudSim is shown in Figure 2.

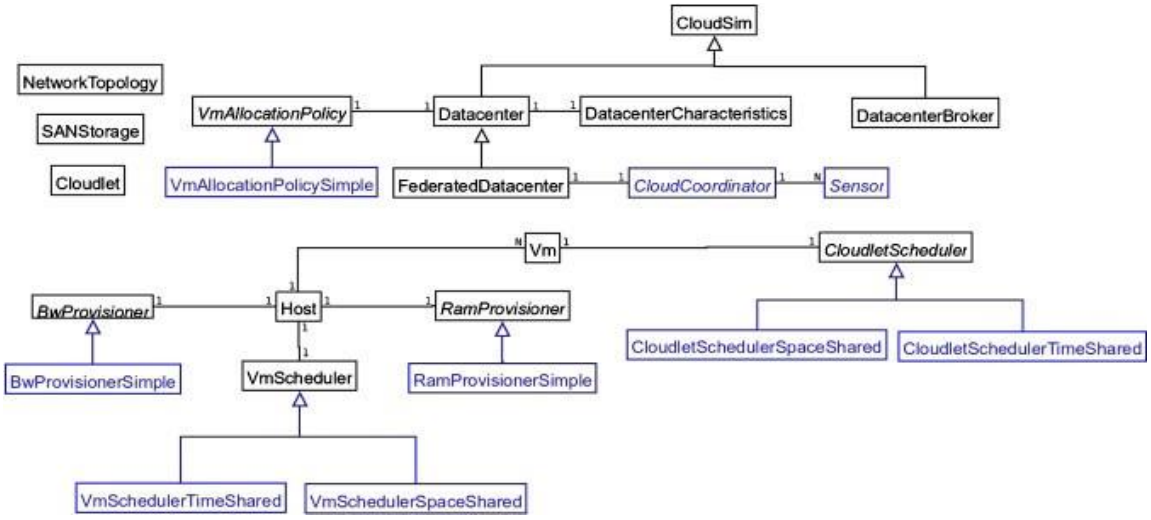


Figure 2: CloudSim Class Design Diagram

The description of all the major classes mentioned in class diagram above is described below:

1. **BwProvisioner**: The main role of this component is to undertake the allocation of network bandwidths to a set of competing VMs that are deployed across the data center. Cloud system developers and researchers can extend this class with their own policies (priority, QoS) to reflect the needs of their applications.
2. **CloudCoordinator**: It is responsible for periodically monitoring the internal state of data center resources and based on that it undertakes dynamic load-shredding decisions. Developers aiming to deploy their application services across multiple clouds can extend this class for implementing their custom inter-cloud provisioning policies.
3. **Cloudlet**: This class models the Cloud-based application services (program based tasks) such as content delivery, social networking, and business workflow. CloudSim mimics the complexity of an application in terms of its computational requirements. Every application service has a

pre- assigned instruction length and data transfer (both pre and post fetches) overhead that it needs to undertake during its life cycle.

4. **CloudletScheduler**: This is responsible for implementation of different policies that determine the share of processing power among Cloudlets in a VM. There are two types of provisioning policies offered: space-shared (using **CloudetSchedulerSpaceShared** class) and time-shared (using **CloudletSchedulerTimeShared** class).
5. **Datacenter**: This class models the core infrastructure-level services (i.e. hardware) that are offered by Cloud providers (Amazon, Azure, and App Engine). It encapsulates a set of compute hosts that can either be homogeneous or heterogeneous with respect to their hardware configurations (memory, cores, capacity, and storage). Also, every Datacenter component instantiates a generalized application provisioning component that implements a set of policies for allocating bandwidth, memory, and storage devices to hosts and VMs.
6. **DatacenterBroker or Cloud Broker**: This class models a broker, which is responsible for mediating negotiations between SaaS and Cloud providers; and such negotiations are driven by QoS requirements. The broker class acts on behalf of applications. It discovers suitable Cloud service providers by querying the CIS and undertakes online negotiations for allocation of resources/services that can meet the application's QoS needs. **Researchers and system developers** must extend this class for evaluating and testing custom brokering policies. The difference between the broker and the CloudCoordinator is that the broker represents the customer (i.e. decisions of these components are made in order to increase user-related performance metrics), whereas the CloudCoordinator acts on behalf of the data center, i.e. it tries to maximize the overall performance of the data center, without considering the needs of specific customers.
7. **DatacenterCharacteristics**: This class contains configuration information of data center resources.
8. **Host**: This class models a physical resource such as a compute or storage

server. It encapsulates important information such as the amount of memory and storage, a list and type of processing cores (to represent a multi-core machine), an allocation of policy for sharing the processing power among VMs, and policies for provisioning memory and bandwidth to the VMs.

9. **NetworkTopology**: This class contains the information for inducing network behavior (latencies) in the simulation. It stores the topology information, which is generated using the BRITE topology generator.
10. **RamProvisioner**: This is an abstract class that represents the provisioning policy for allocating primary memory (RAM) to the Virtual Machines. The execution and deployment of VM on a hostis feasible only if the RamProvisioner component approves that the host has the required amount of free memory. The RamProvisionerSimple does not enforce any limitation on the amount of memory that a VM may request. However, if the request is beyond the available memory capacity, then it is simply rejected.
11. **Vm**: This class models a Virtual Machine (VM), which is managed and hosted by a Cloud host component. Every VM component has access to a component that stores the following characteristics related to a VM (i.e.) accessible memory, processor, storage size, and the VM's internal provisioning policy that is extended from an abstract class called the CloudletScheduler.
12. **VmAllocationPolicy**: This abstract class represents a provisioning policy that a VM Monitor utilizes for allocating VMs to hosts. The main functionality of the VmAllocationPolicy is to select the available host in a data center that meets the memory, storage, and availability requirement for a VM deployment.
13. **VmScheduler**: This is an abstract class implemented by a Host component that models the policies (space-shared, time-shared) required for allocating processor cores to VMs. The functionalities of this class can easily be overridden to accommodate application-specific processor

sharing policies.

14. **CloudSim**: This is the main class, which is responsible for managing event queues and controlling step-by-step (sequential) execution of simulation events. Every event that is generated by the CloudSim entity at run-time is stored in the queue called future events. These events are sorted by their time parameter and inserted into the queue. Next, the events that are scheduled at each step of the simulation are removed from the future events queue and transferred to the deferred event queue. Following this, an event processing method is invoked for each entity, which chooses events from the deferred event queue and performs appropriate actions. Such an organization allows flexible management of simulation and provides the following powerful capabilities:
    a. Deactivation (holding/pausing) of entities.
    b. Context switching of entities between different states (e.g. waiting to active). Pausingand resuming the process of simulation.
    c. Creation of new entities at run-time.
    d. Aborting and restarting simulation at run-time.
15. **DeferredQueue**: This class implements the deferred event queue used by CloudSim.
16. **FutureQueue**: This class implements the future event queue accessed by CloudSim.
17. **CloudInformationService**: A CIS is an entity that provides resource registration, indexing, anddiscovering capabilities. CIS supports two basic primitives:
    a. **publish()**, which allows entities to register themselves with CIS and
    b. **search()**, which allows entities such as CloudCoordinator and Brokers in discovering status and endpoint contact address of other entities. This entity also notifies the other entities about the end of simulation.
18. **SimEntity**: This is an abstract class, which represents a simulation entity (such as Cloudlet, VM, Host etc) that is able to send messages to other

entities and process received messages as well as fire and handle events. SimEntity class provides the ability to schedule new events and send messages to other entities, where network delay is calculated according to the BRITE model. Once created, entities automatically register with CIS. All entities must extend this class and override its three core methods:

a. **startEntity**(), which define actions for entity initialization.
b. **processEvent**(), which define actions processing of events.
c. **shutdownEntity**(),which define actions entity destruction.

19. **CloudSimTags**. This class contains various static event/command tags that indicate the type ofaction that needs to be undertaken by CloudSim entities when they receive or send events.
20. **SimEvent**: This entity represents a simulation event that is passed between two or moreentities. SimEvent stores the following information about an event:
    a. type,
    b. init time,
    c. time at which the event should occur,
    d. finish time,
    e. time at which the event should be delivered to its destination entity,
    f. IDs of the source and destination entities,
    g. tag of the event, and
    h. Data that have to be passed to the destination entity.
21. **CloudSimShutdown**: This is an entity class that waits for the termination of all end-user andbroker entities, and then signals the end of simulation to CIS.

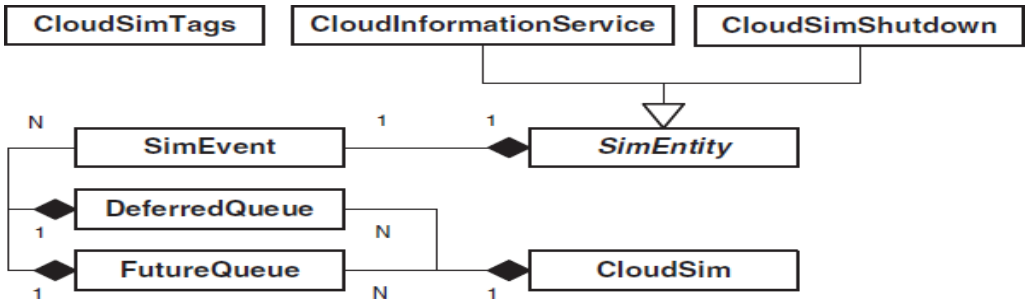


Figure 3: Interrelation of Core Simulation Classes

Following is the description of frequently used terms in this manual:

1. **Cloudlet**: set of tasks ready to submit for processing.
2. **VM (Virtual Machine)**: this term represents a logical image of machine defined with characteristics just like physical machine.
3. **Host**: This will represent a physical machine with specific characteristics. Each host will consist of minimum one or more processing element.
4. **Datacenter**: This will represent a physical setup, which will have more than one interconnected host setup using a specific topology (to form a compute cluster).
5. **Broker**: This will represent a user or machine though which infrastructure (datacenter resources) will be accessed by virtual machines and cloudlets will be allocated to virtual machines for execution.

4. Installing CloudSim with Eclipse

1. Before installing CloudSim following resources must be downloaded on the local system
   a. **Eclipse IDE for java developers**:
   i. Windows 32-bit:
   http://www.eclipse.org/downloads/download.php?file=/technology/epp/downl oads/release/juno/SR1/eclipse-java-juno-SR1-win32.zip

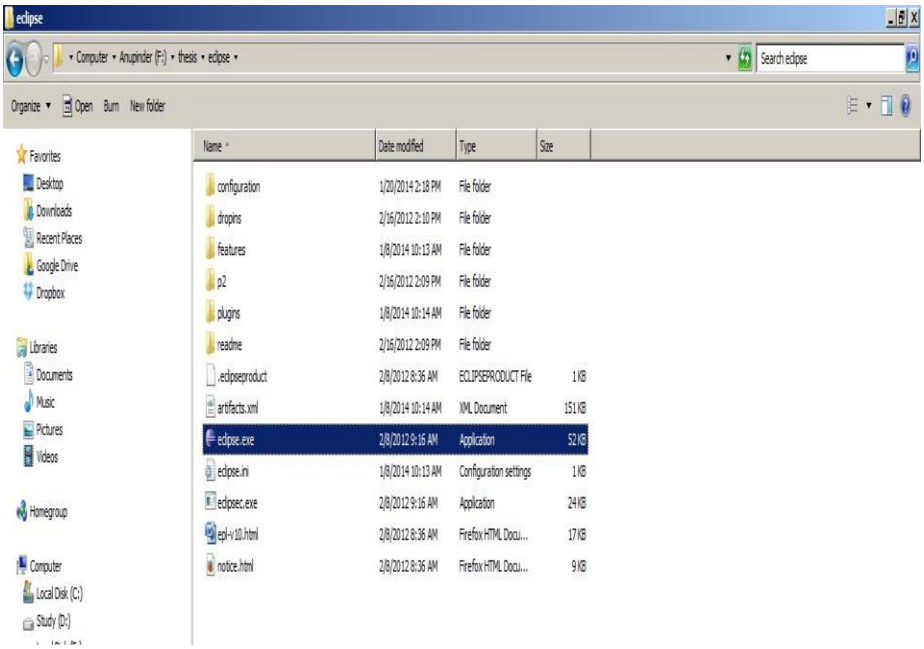
ii. Windows 64-bit:
http://www.eclipse.org/downloads/download.php?file=/technology/epp/downl oads/release/juno/SR1/eclipse-java-juno-SR1-win32-x86_64.zip

b. **CloudSim Project code**: CloudSim code can be downloaded from following project page https://code.google.com/p/cloudsim/, always download latest version to avoid any bug issues of previous packages.

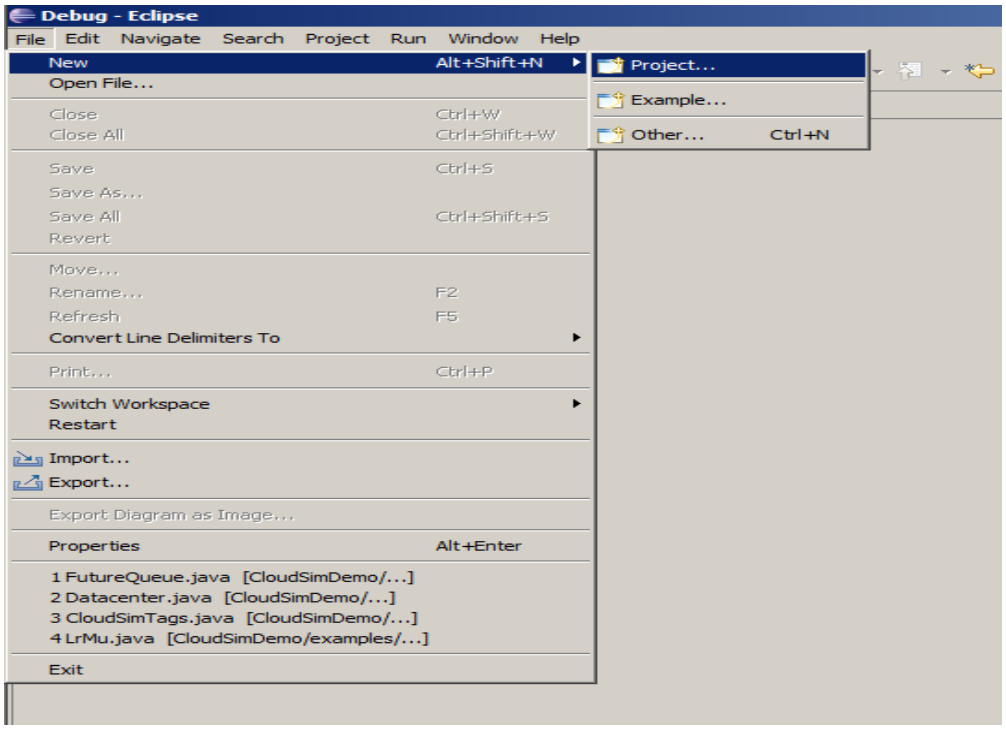
c. **One external requirement of cloudsim** i.e. common jar package of math related functions is to be downloaded from
http://www.trieuvan.com/apache//commons/math/binaries/commons-math3-3.2- bin.zip

2. Unzip **eclipse** and **cloudsim** to some common folder.

3. Open **eclipse.exe** from eclipse folder:

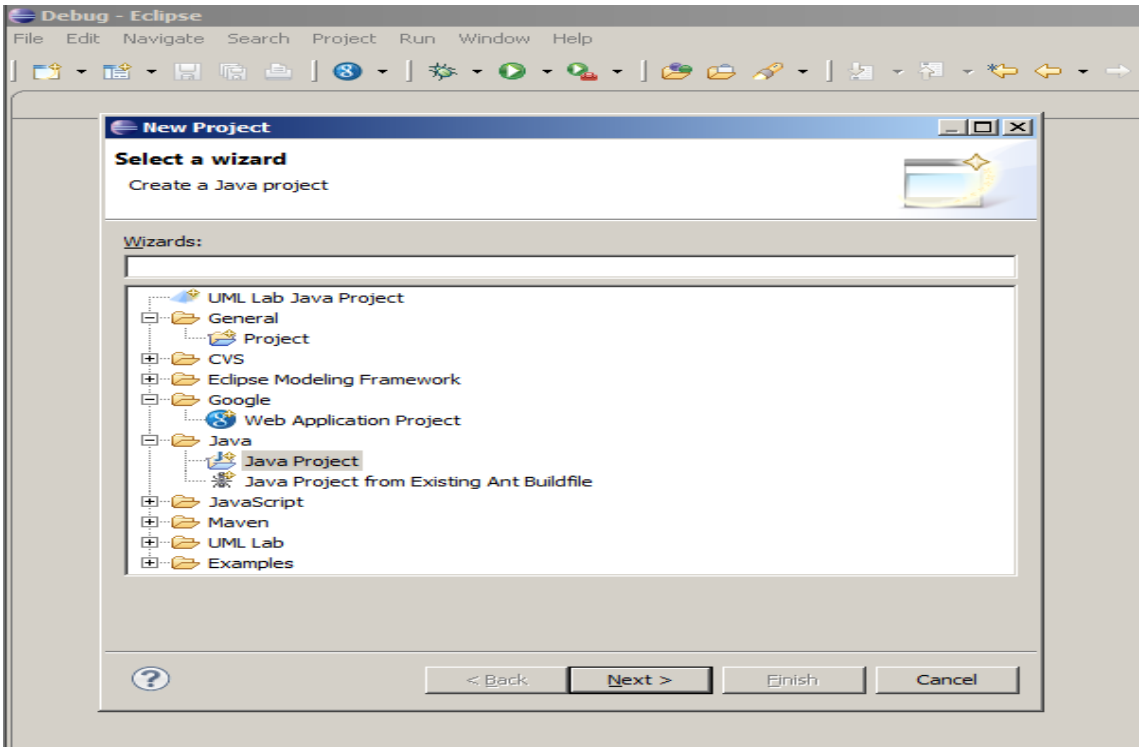


4. Now in Eclipse go to menu File-> New -> Project.

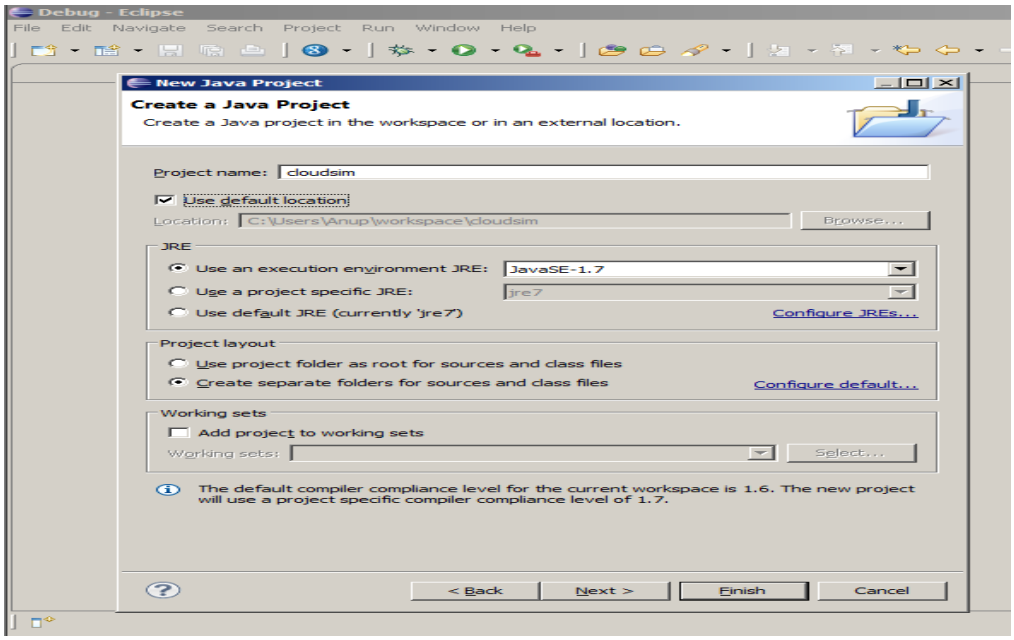


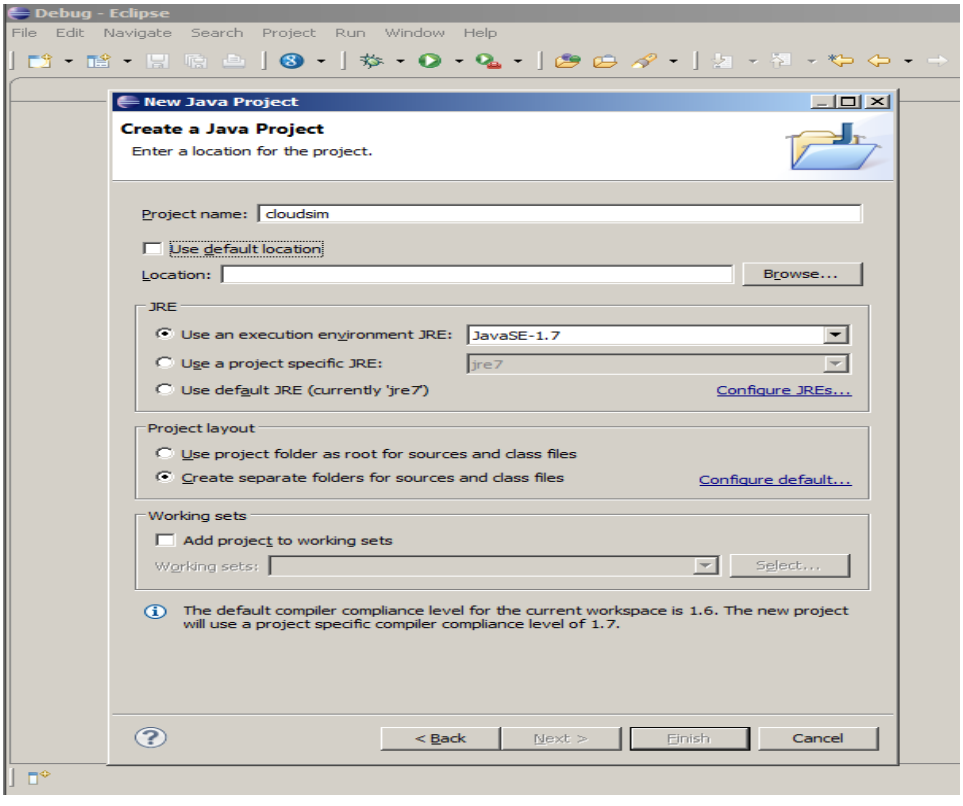
5. Now a project wizard will be opened. From this wizard choose ‘Java Project’

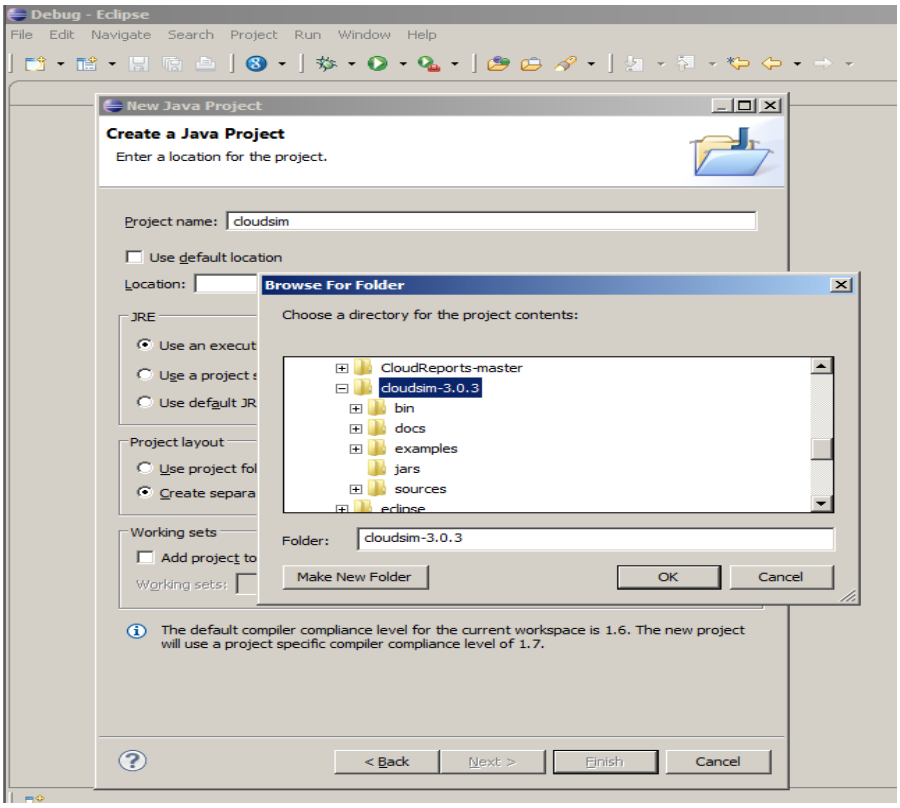
option and clicknext.

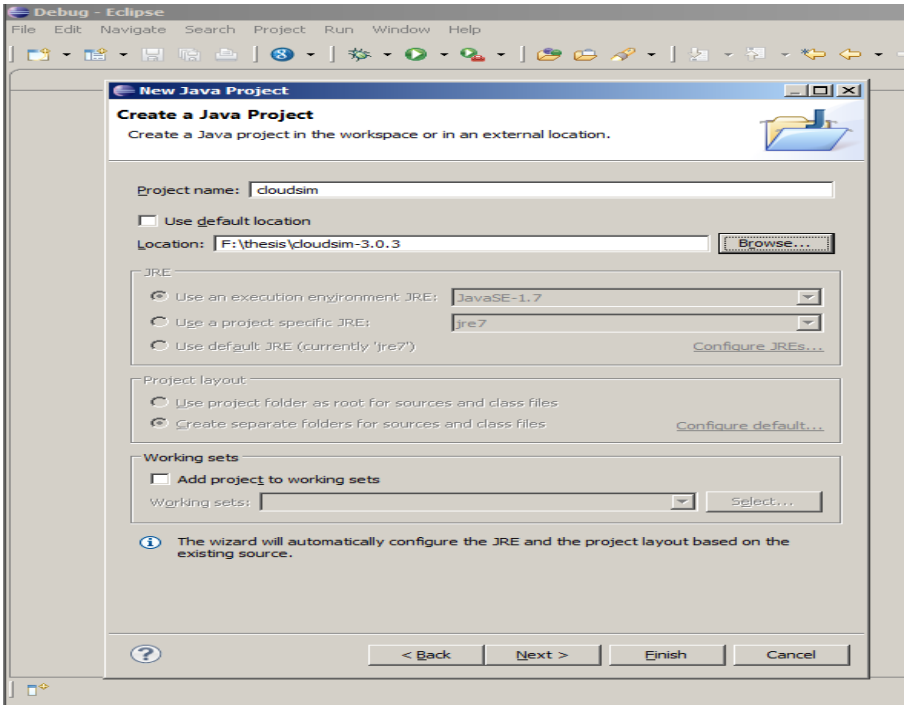


6. Now a detailed new project window will be opened, here you provide project name and thepath where you have unzipped the CloudSim project source code or you can put followingdetails as specified:
   a. **Project Name**: CloudSim.
   b. Unselect the 'Use default location' option and then click on browse to open the path where you have unzipped the cloudsim project and finally click Next to set project settings


Debug - Eclipse
File Edit Navigate Search Project Run Window Help
New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: cloudsim
Use default location
Location: C:\Users\Anup\workspace\cloudsim
Browse...
JRE
Use an execution environment JRE: JavaSE-1.7
Use a project specific JRE: jre7
Use default JRE (currently 'jre7')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...
The default compiler compliance level for the current workspace is 1.6. The new project will use a project specific compiler compliance level of 1.7.
< Back
Next >
Finish
Cancel

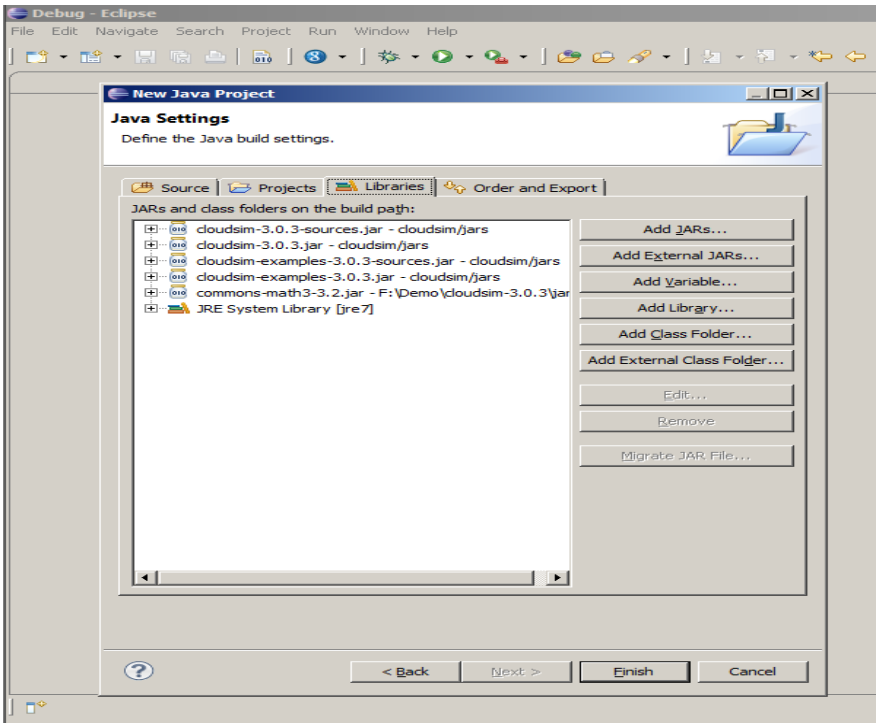

Debug - Eclipse
File Edit Navigate Search Project Run Window Help
New Java Project
Create a Java Project
Enter a location for the project.
Project name: cloudsim
Use default location
Location:
Browse...
JRE
Use an execution environment JRE: JavaSE-1.7
Use a project specific JRE: jre7
Use default JRE (currently 'jre7')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
Working sets:
Select...
The default compiler compliance level for the current workspace is 1.6. The new project will use a project specific compiler compliance level of 1.7.
< Back
Next >
Finish
Cancel



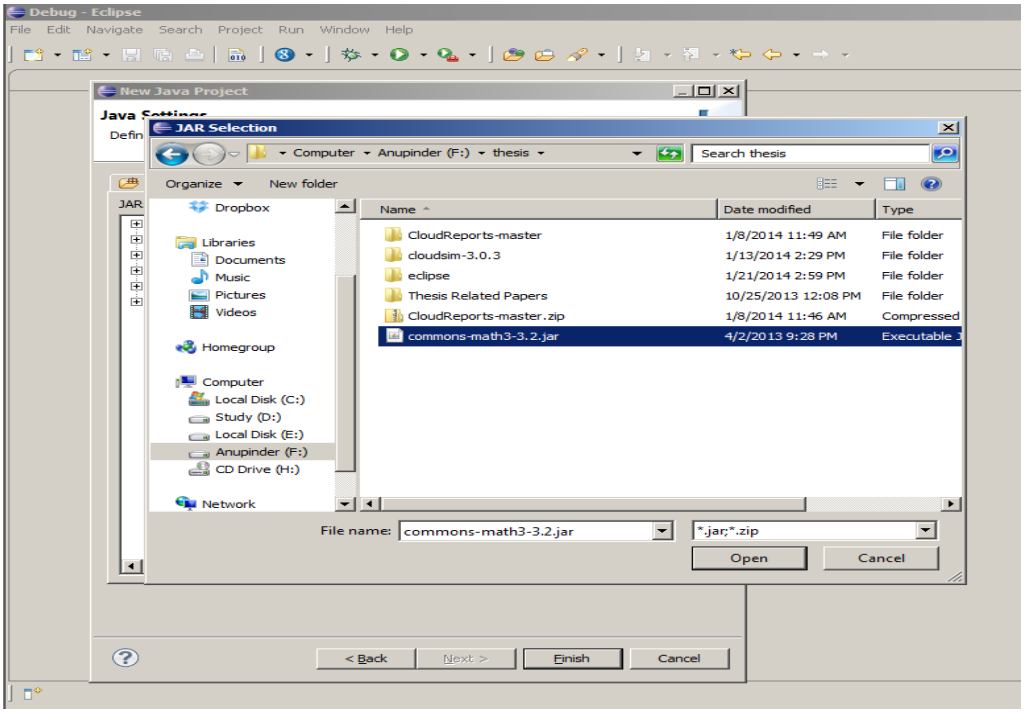


c. Now open ‘Libraries’ tab and click on add external jar (Math

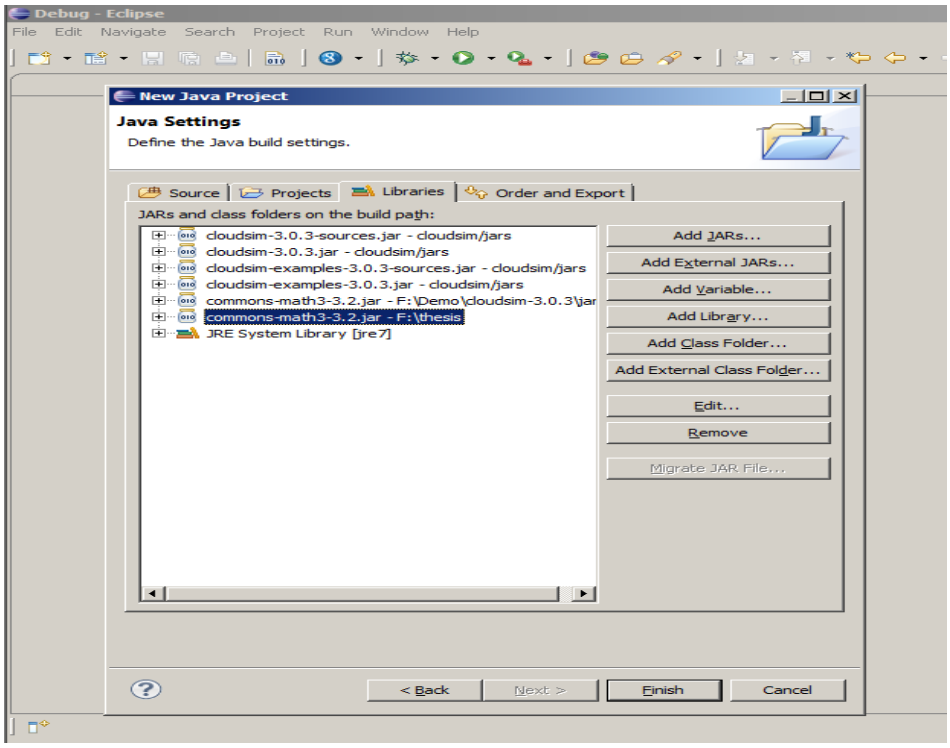
common jar will be included inproject from this step)



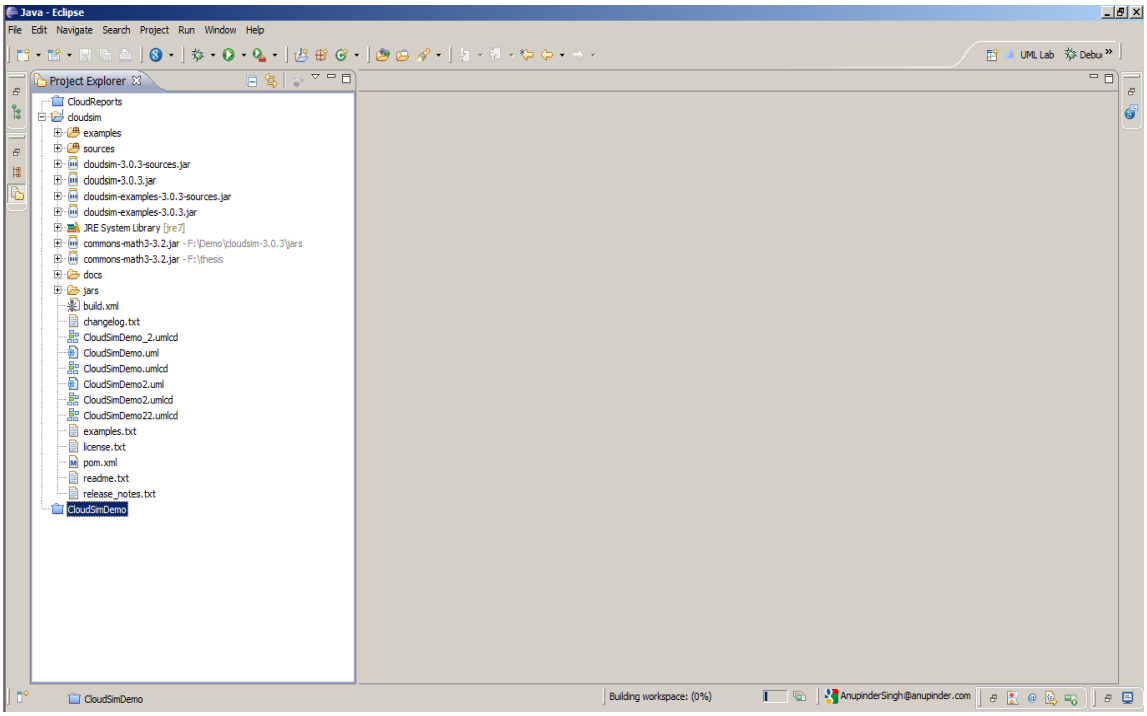
d. Open the path where you have unzipped the math common binaries and select ‘Common-math 3.3.2.jar’ and click on open.



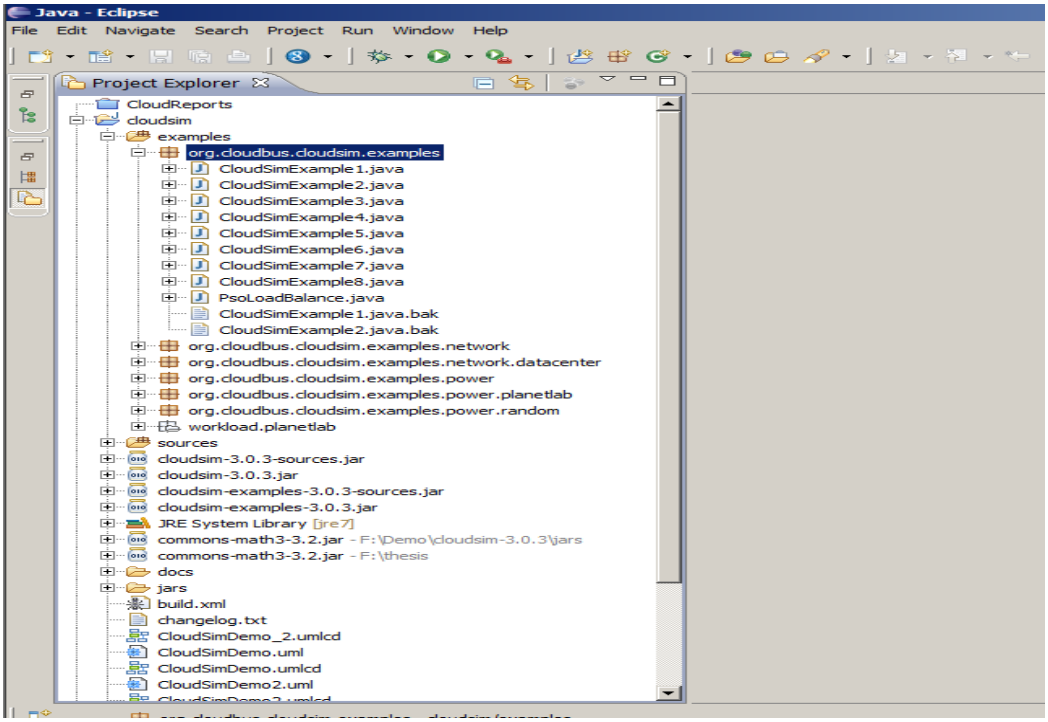
e. Ensure external jar that you opened in previous step is displayed in list and then click on‘Finish’ (your system may take 2-3 minutes to configure the project)



7. Once the project is configured you can open the project explorer and start exploring the cloudsim project. Also for the first time eclipse automatically start building the workspace for newly configured cloudsim project, which may take some time depending on the configuration of computer system. Following is the final screen which you will see after cloudsim is configured.



8. In this manual we will be discussing all the examples that are provided under package ‘org.cloudbus.cloudsim.examples’ (as shown in figure below).


Java - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
CloudReports
cloudsim
examples
org.cloudbus.cloudsim.examples
CloudSimExample1.java
CloudSimExample2.java
CloudSimExample3.java
CloudSimExample4.java
CloudSimExample5.java
CloudSimExample6.java
CloudSimExample7.java
CloudSimExample8.java
PsoLoadBalance.java
CloudSimExample1.java.bak
CloudSimExample2.java.bak
org.cloudbus.cloudsim.examples.network
org.cloudbus.cloudsim.examples.network.datacenter
org.cloudbus.cloudsim.examples.power
org.cloudbus.cloudsim.examples.power.planetlab
org.cloudbus.cloudsim.examples.power.random
workload.planetlab
sources
cloudsim-3.0.3-sources.jar
cloudsim-3.0.3.jar
cloudsim-examples-3.0.3-sources.jar
cloudsim-examples-3.0.3.jar
JRE System Library [jre7]
commons-math3-3.2.jar - F:\Demo\cloudsim-3.0.3\jars
commons-math3-3.2.jar - F:\thesis
docs
jars
build.xml
changelog.txt
CloudSimDemo_2.umlcd
CloudSimDemo.uml
CloudSimDemo.umlcd
CloudSimDemo2.uml

5. How CloudSim Works?

When any example in package ‘org.cloudbus.cloudsim.examples’ is executed, the sequence ofexecution will be as follows:

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 2

Aim: Install a Cloud Analyst and Integrate with Eclipse/Netbeans. Monitor the performance of an Existing Algorithms.

Theory:

CloudAnalyst

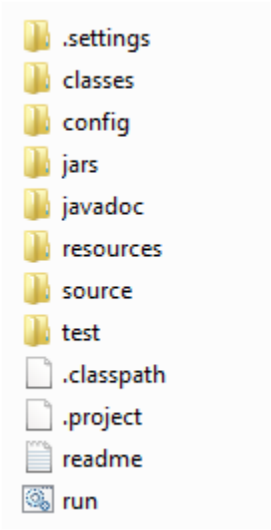
Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support evaluation of social networks tools according to geographic distribution of users and data centers. In this tool, communities of users and data centers supporting the social networks are characterized and, based on their location; parameters such as user experience while using the social network application and load on the data center are obtained/logged

RUN cloud analyst Simulator

1.Download CloudAnalyst

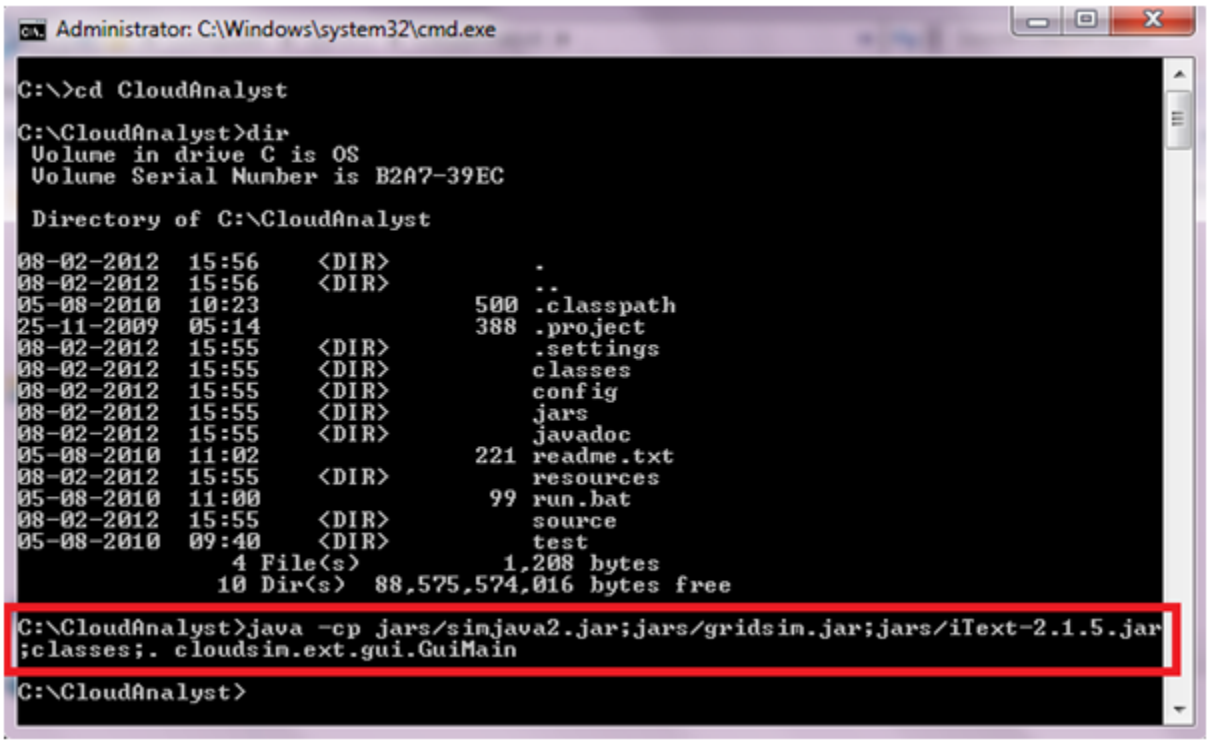
http://www.cloudbus.org/cloudsim/CloudAnalyst.zi

2. Extract Files from the Zip file which will give following folder structure



CLOUDANALYST FOLDER STRUCTURE

If you want to Run from Command line then



RUN CLOUDANALYST FROM COMMAND LINE

Or click on run.bat file

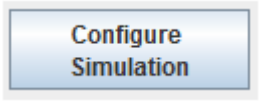
Done!!!

RUN CLOUDANALYST FROM COMMAND LINE: DASHBOARD

Click on Show Region Boundaries:

RUN CLOUDANALYST FROM COMMAND LINE: DASHBOARD – REGIONS

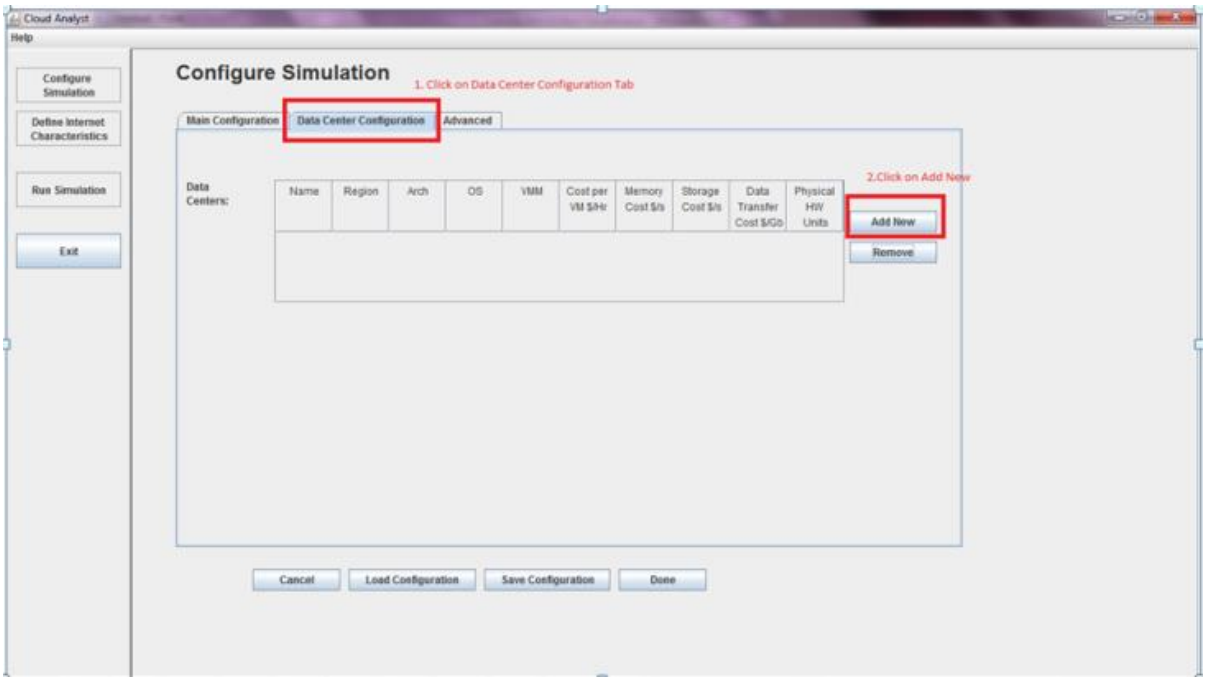
Click on CONFIGURE Simulation



Here you can configure:

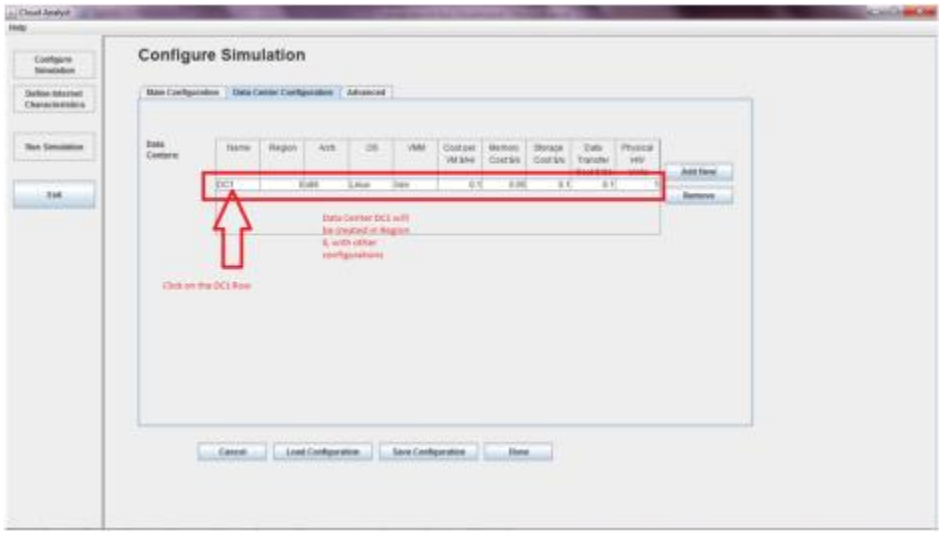
· Data Center Configuration

o Physical Hardware Details of Data center

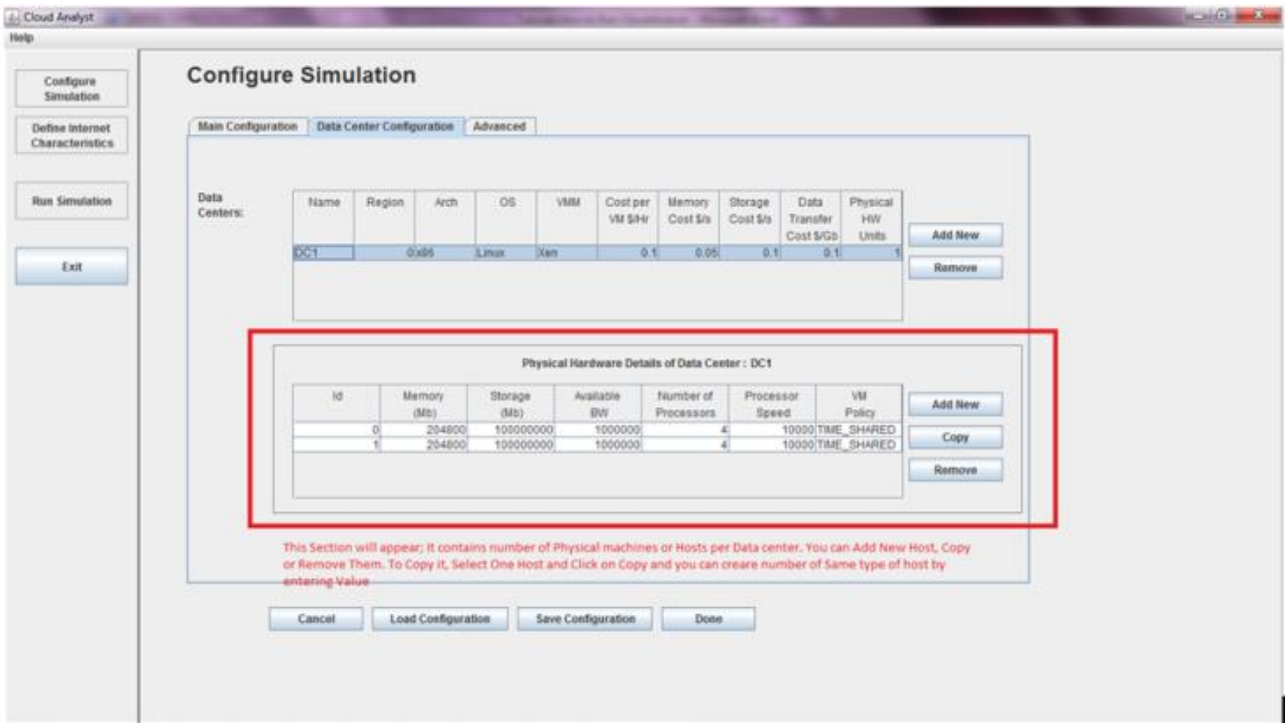


CLOUDANALYST – PHYSICAL HARDWARE DETAILS OF DATA CENTER

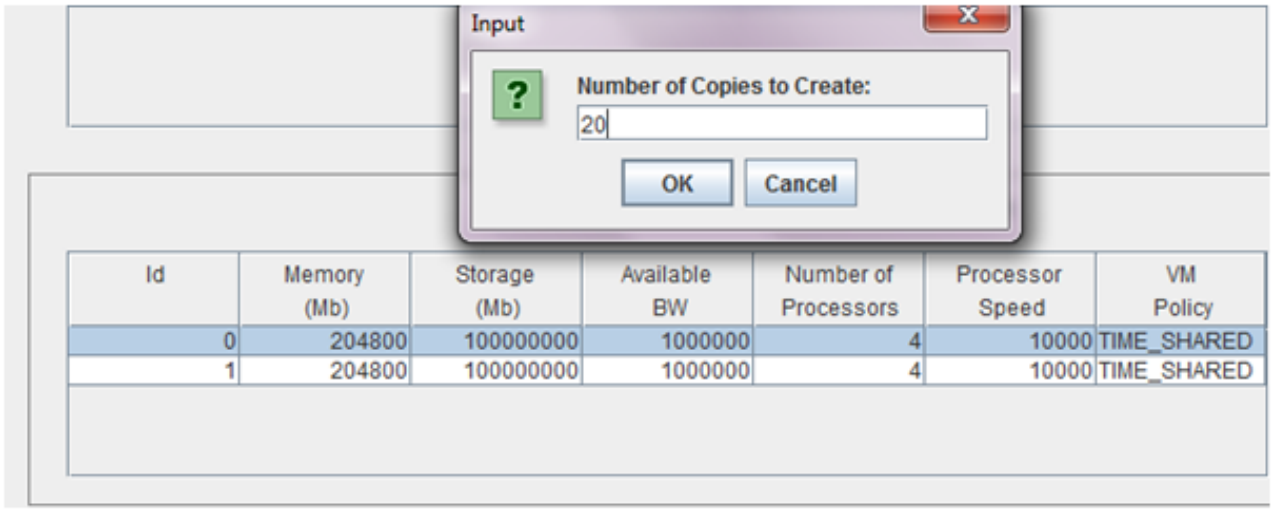
You will get:



CLOUDANALYST – DATA CENTER CONFIGURATIONS



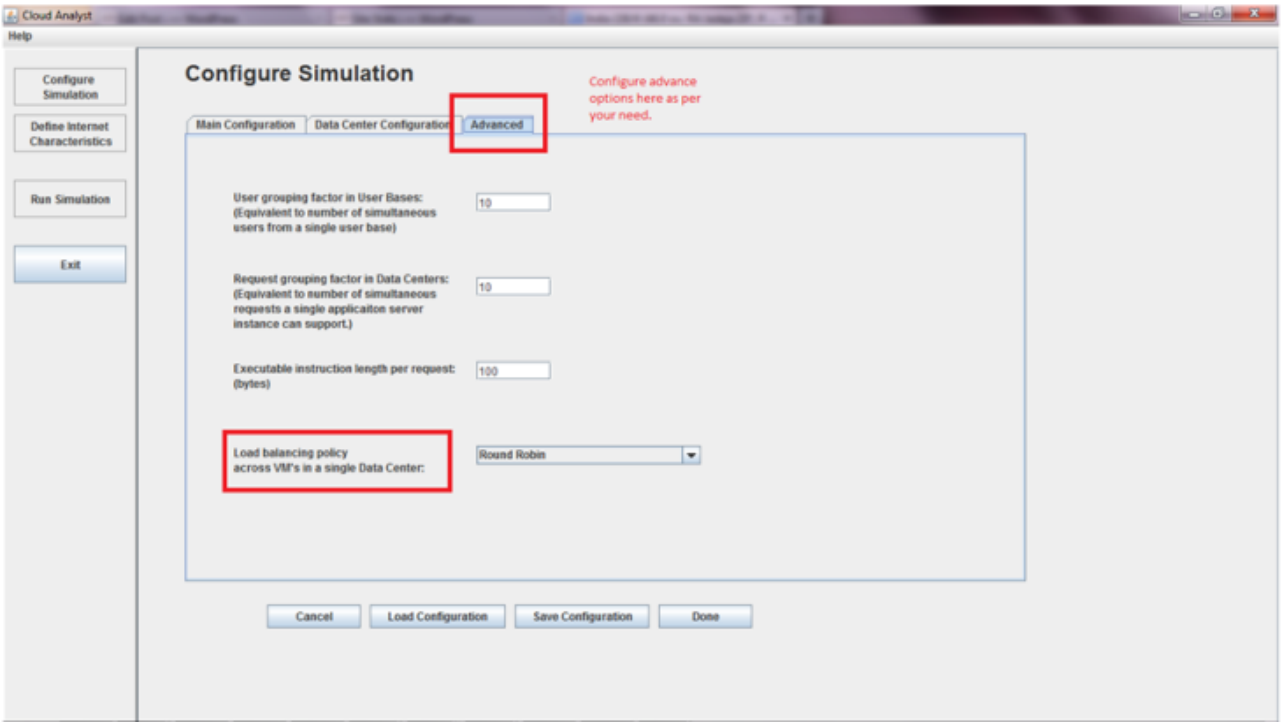
CLOUDANALYST – DATA CENTER CONFIGURATIONS DETAILS



CLOUDANALYST – DATA CENTER CONFIGURATIONS – COPY

.

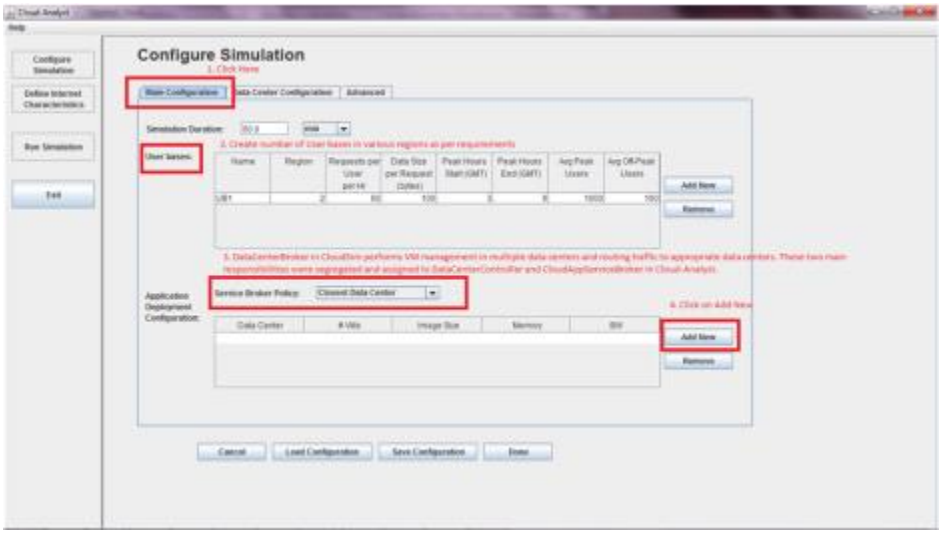
Advanced



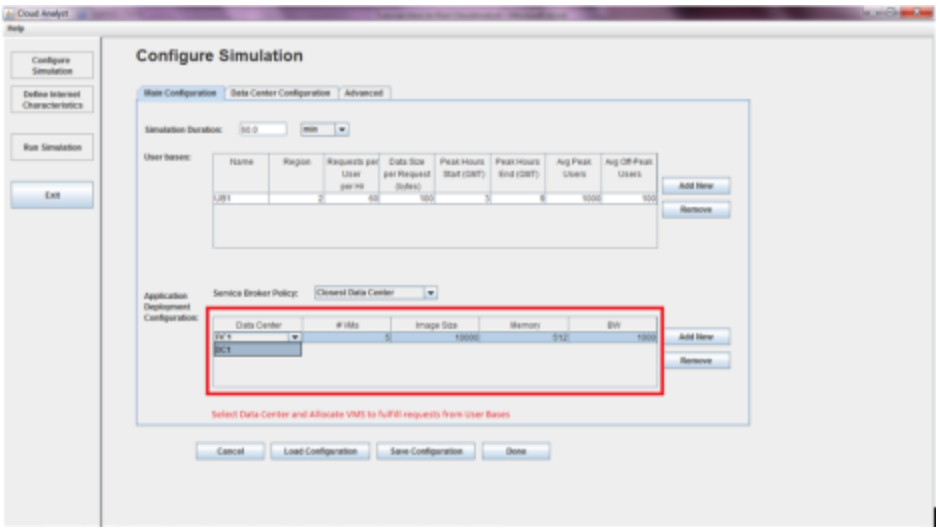
CLOUDANALYST – ADVANCED CONFIGURATIONS

- User Base: models a group of users and generates traffic representing the users.
- Application Deployment Configurations (Cloudlets)

o Service Broker Policy

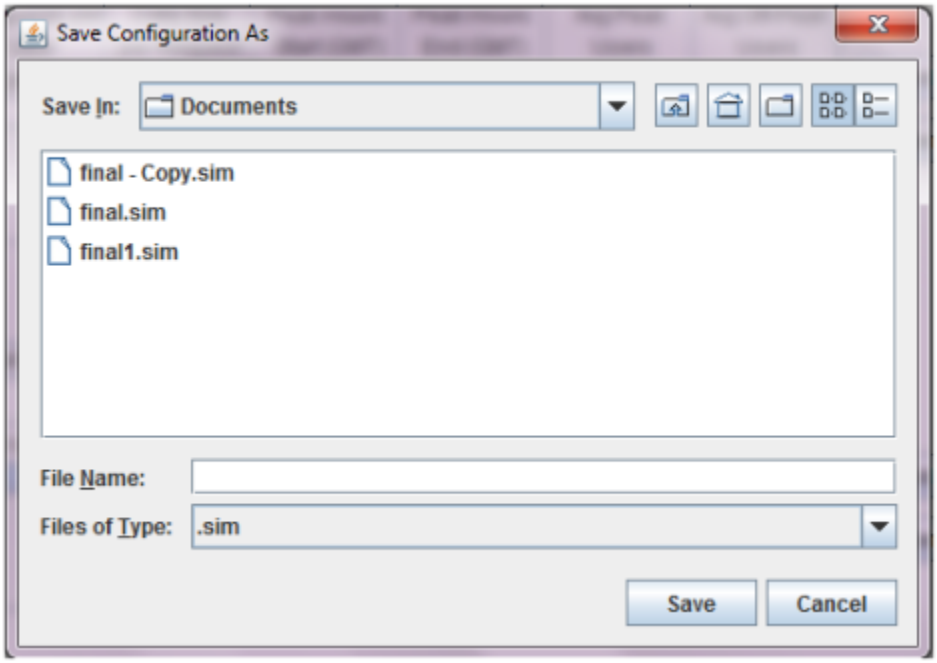


CLOUDANALYST – SERVICE BROKER CONFIGURATIONS



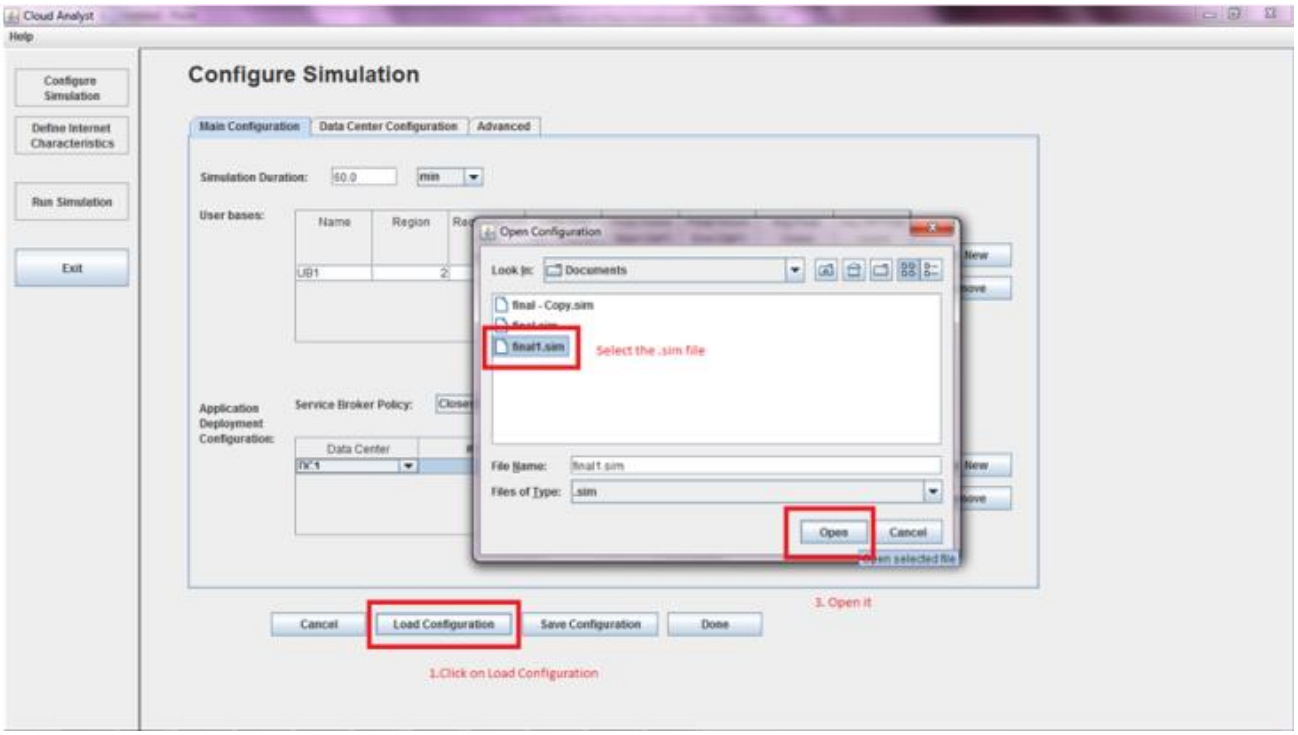
CLOUDANALYST – MAIN CONFIGURATIONS

You can Save this Configuration as well in case you want to use it later. It is stored as .sim file. XML data is generated and saved as Sim file.



CLOUDANALYST – SAVE CONFIGURATIONS

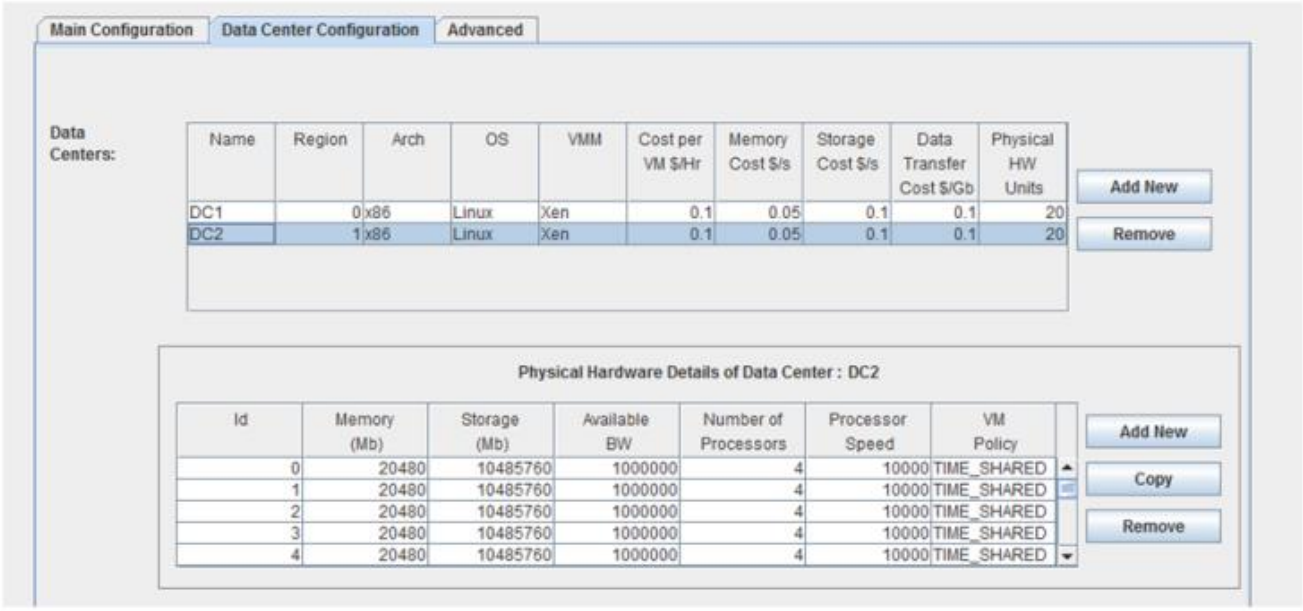
Saved configuration can be loaded anytime easily into CloudAnalyst



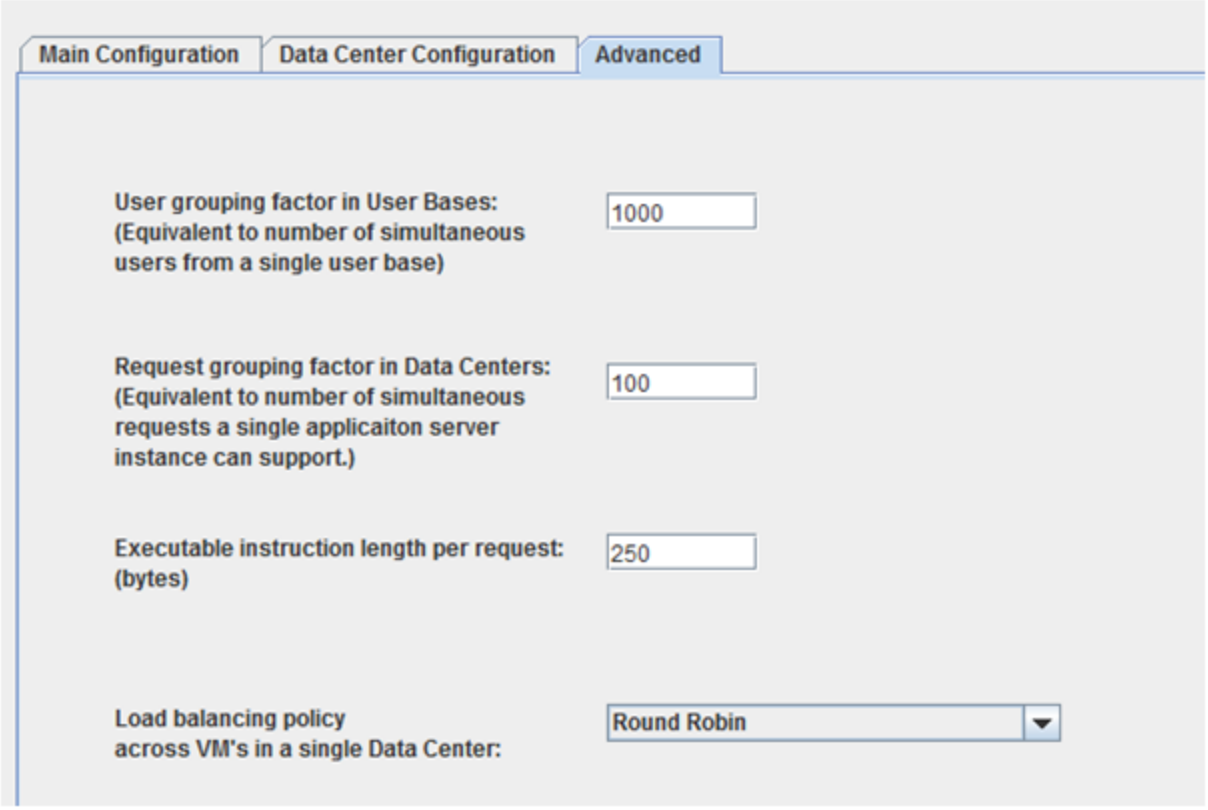
CLOUDANALYST – LOAD CONFIGURATIONS

So you need to enter data each time you want to run simulation.

I have created 2 DC in different region, each with 20 physical host

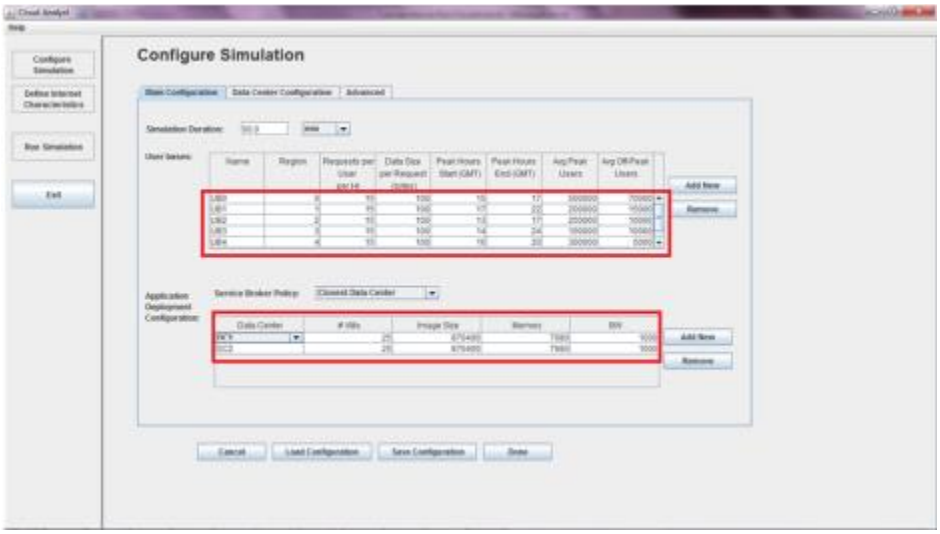


CLOUDANALYST – CONFIGURATIONS – DATA CENTERS IN DIFFERENT REGIONS



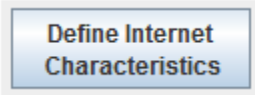
CLOUDANALYST – CONFIGURATIONS – ROUND ROBIN LOAD BALANCING POLICY

6 User bases and each in different region; 25 VMs are allocated to fulfil requests from all user bases from both the Data Center



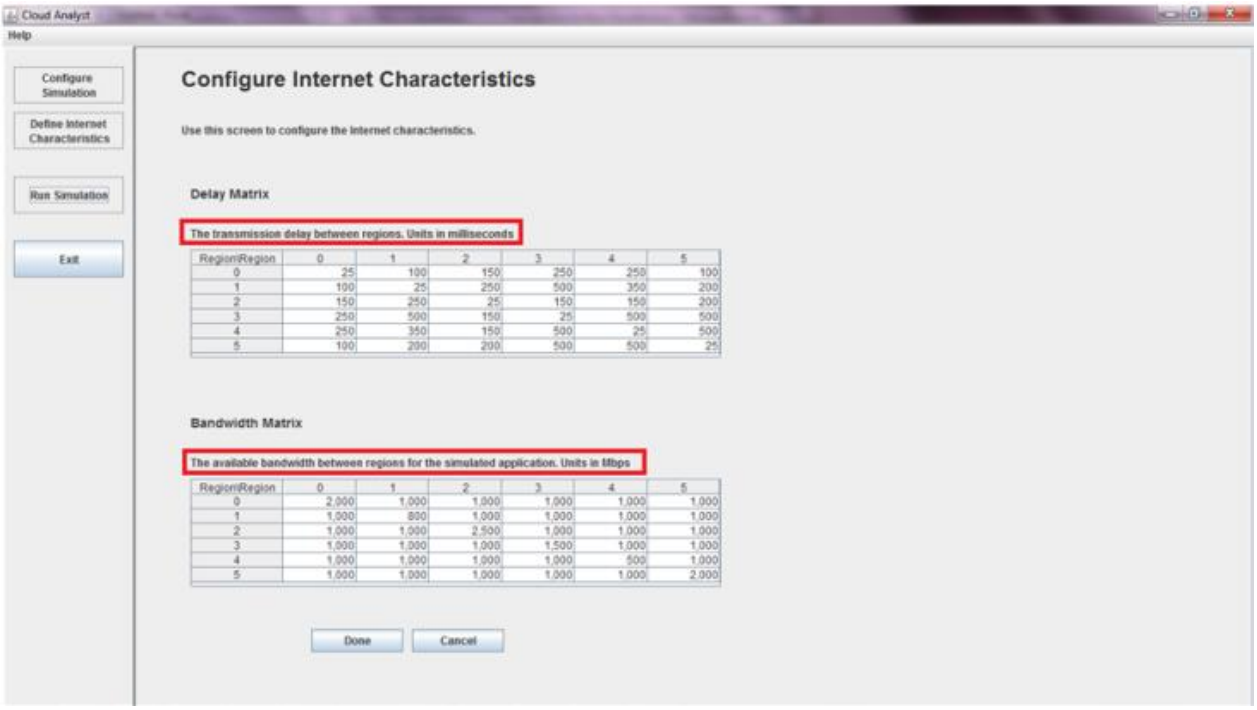
CLOUDANALYST – CONFIGURATIONS – USER BASE AND REGIONS

Once your are done with Configuration; click on Done!!!



CLOUDANALYST – CONFIGURATIONS – INTERNET CHARACTERISTICS

Click on

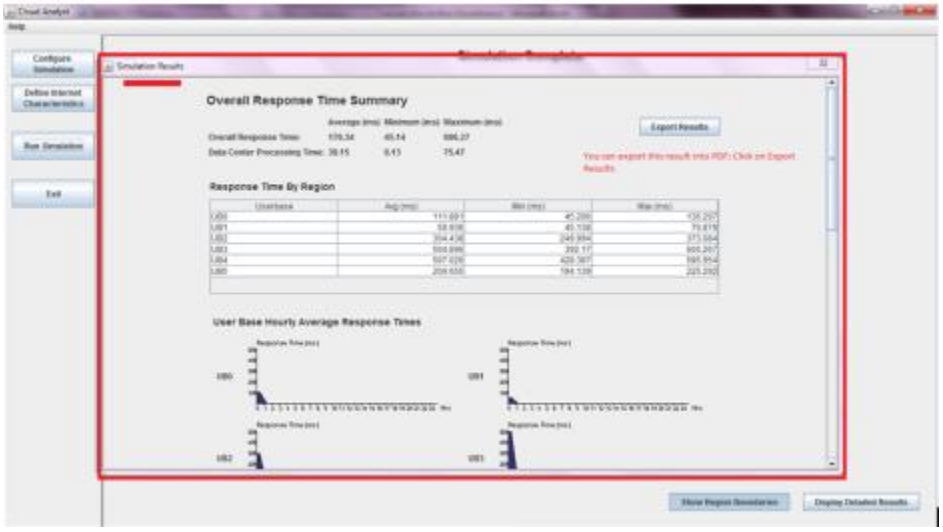


CLOUDANALYST – CONFIGURE INTERNET CHARACTERISTICS

Run Simulation

CLOUDANALYST – RUN SIMULATION

Simulation Results Window will Open



CLOUDANALYST – SIMULATION RESULTS

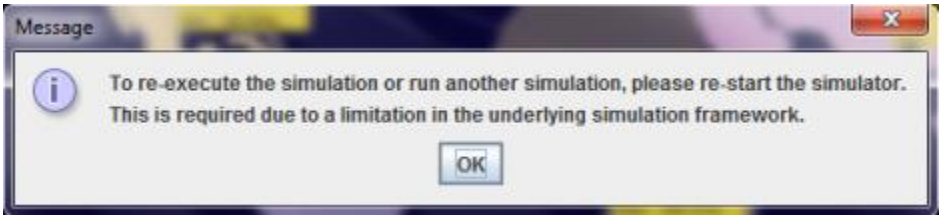
Close it.

Main Window will give all statistics

CLOUDANALYST – SIMULATION STATISTICS

If you will try to run Simulation again then it will give Error

CLOUDANALYST – SIMULATION ERROR

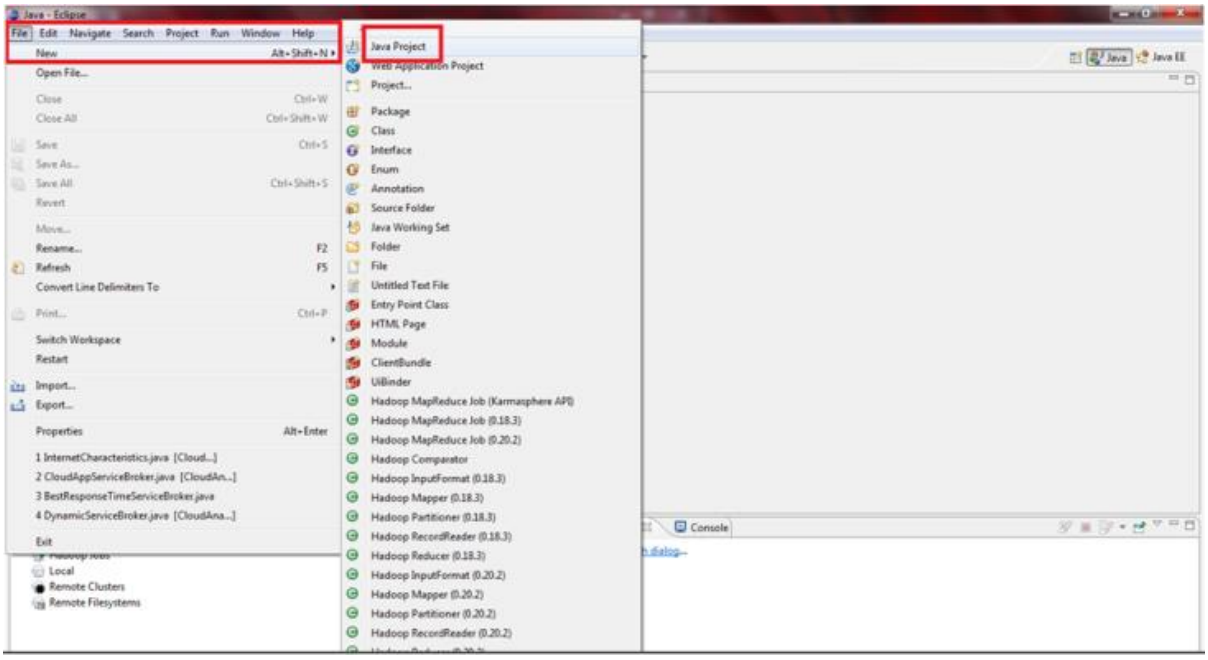


CLOUDANALYST – RESTART SIMULATION ERROR

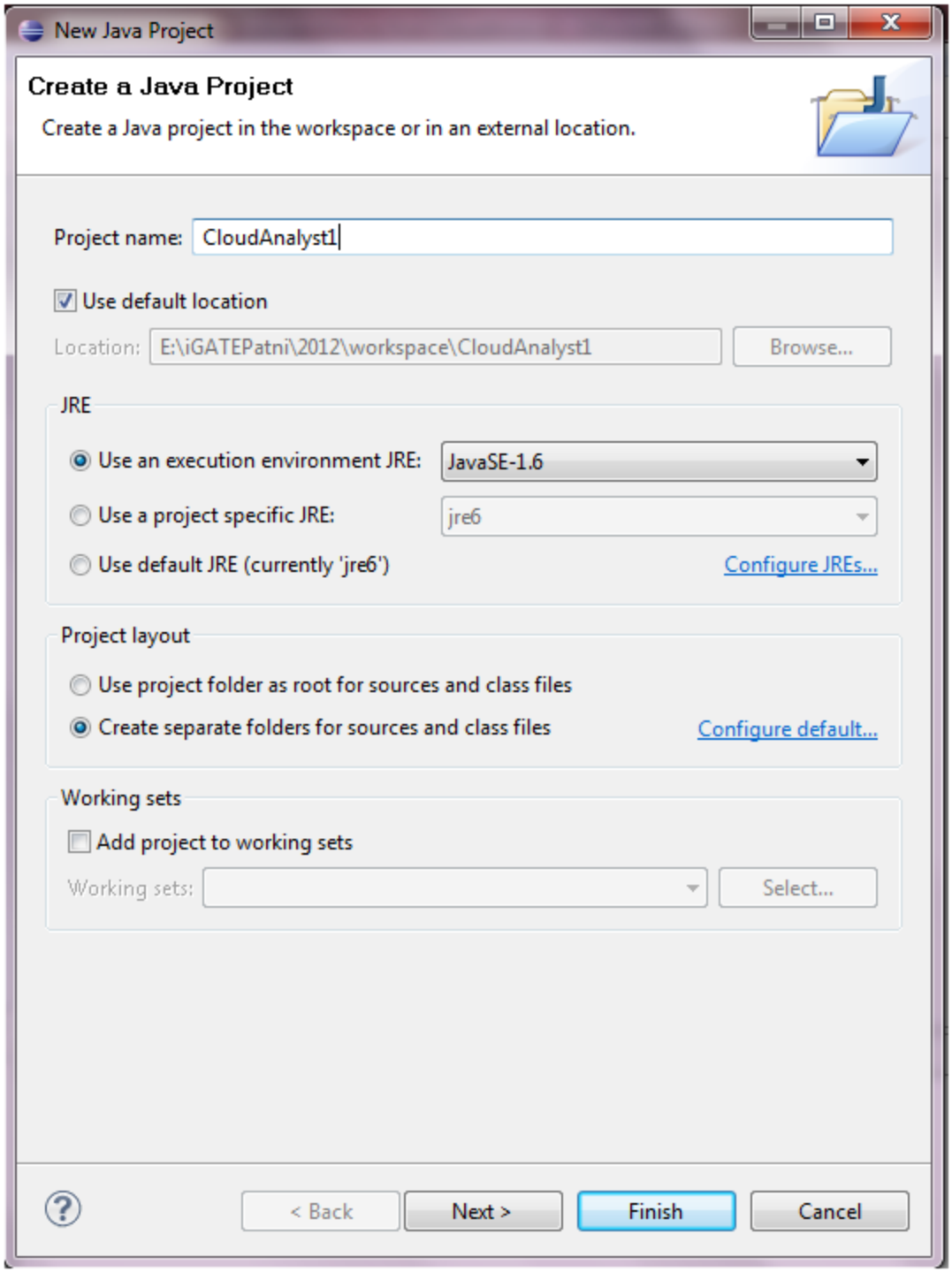
Explore and Edit Cloud Analyst

How to Configure CloudAnalyst in Eclipse

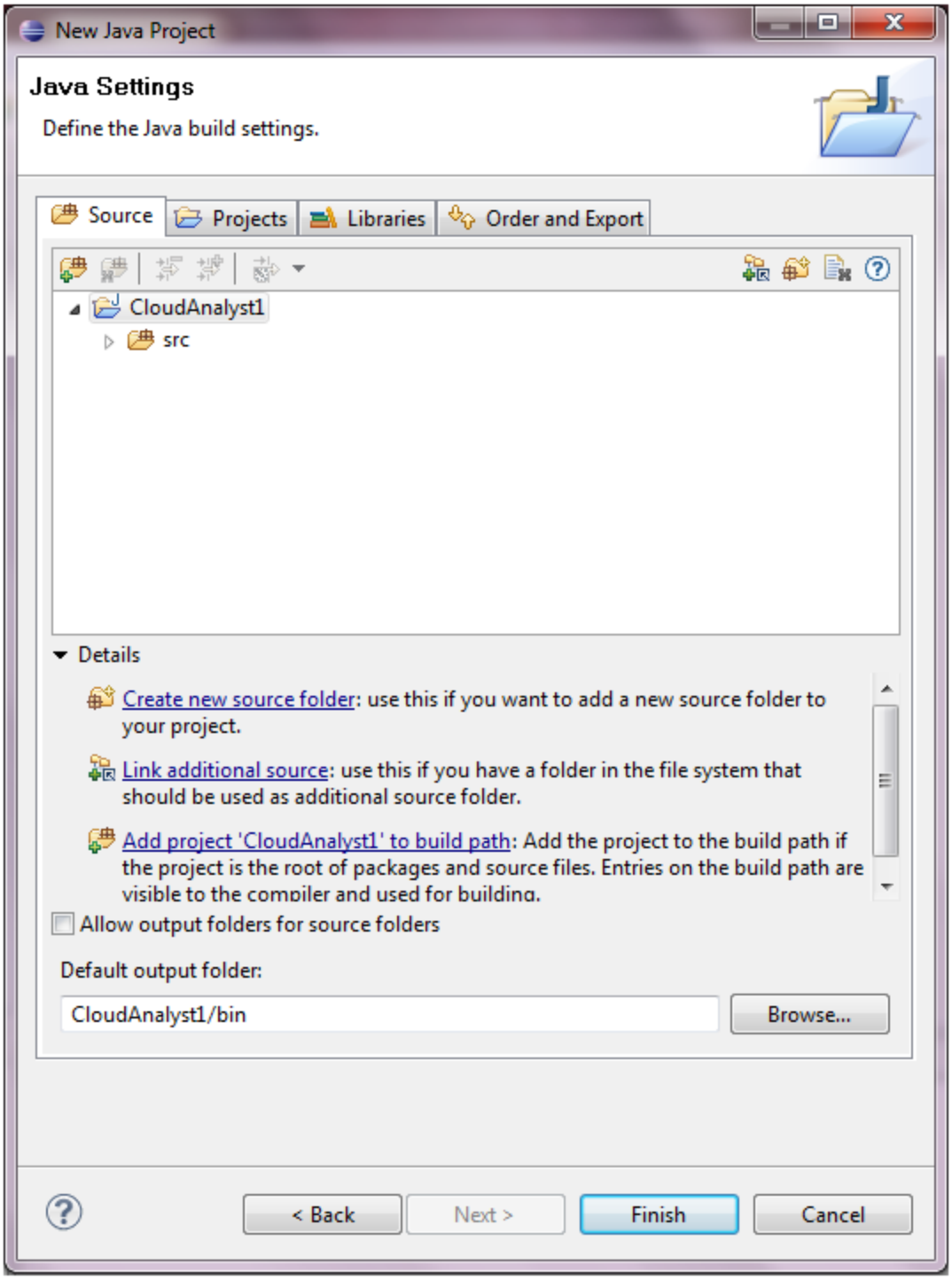
Create New Java Project



NEW JAVA PROJECT IN ECLIPSE



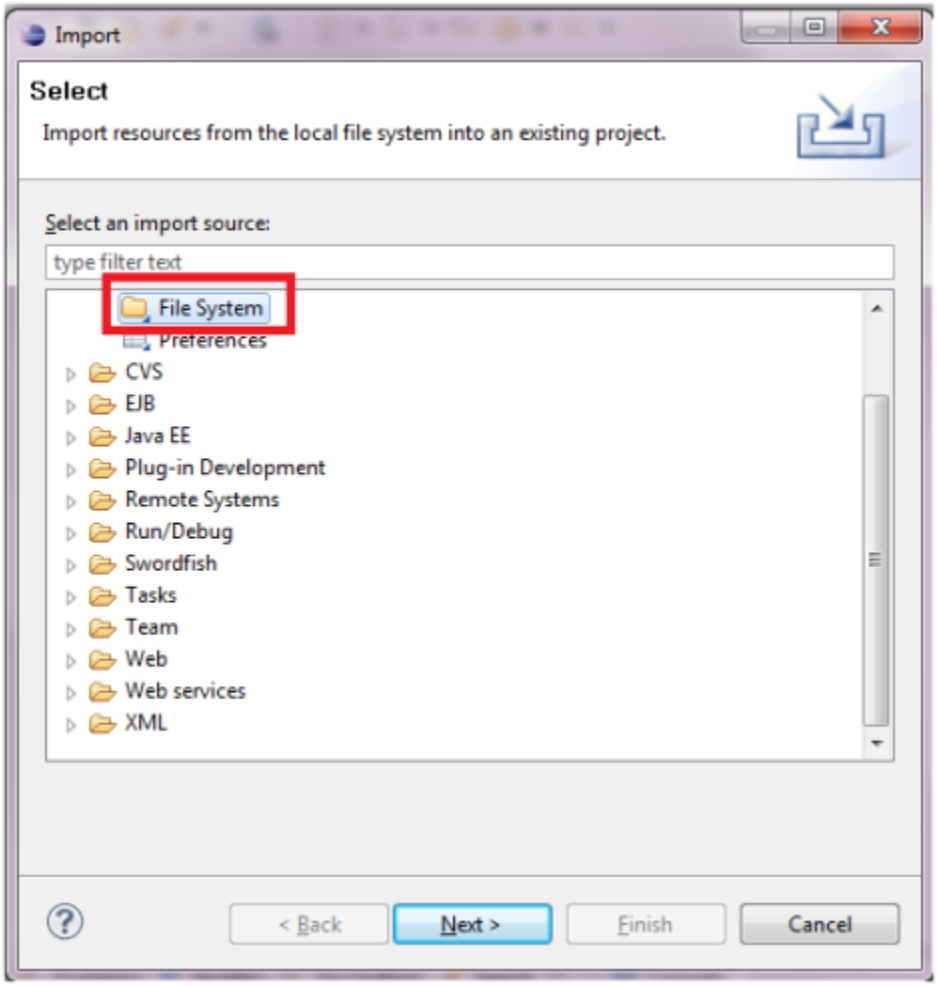
CREATE JAVA PROJECT IN ECLIPSE



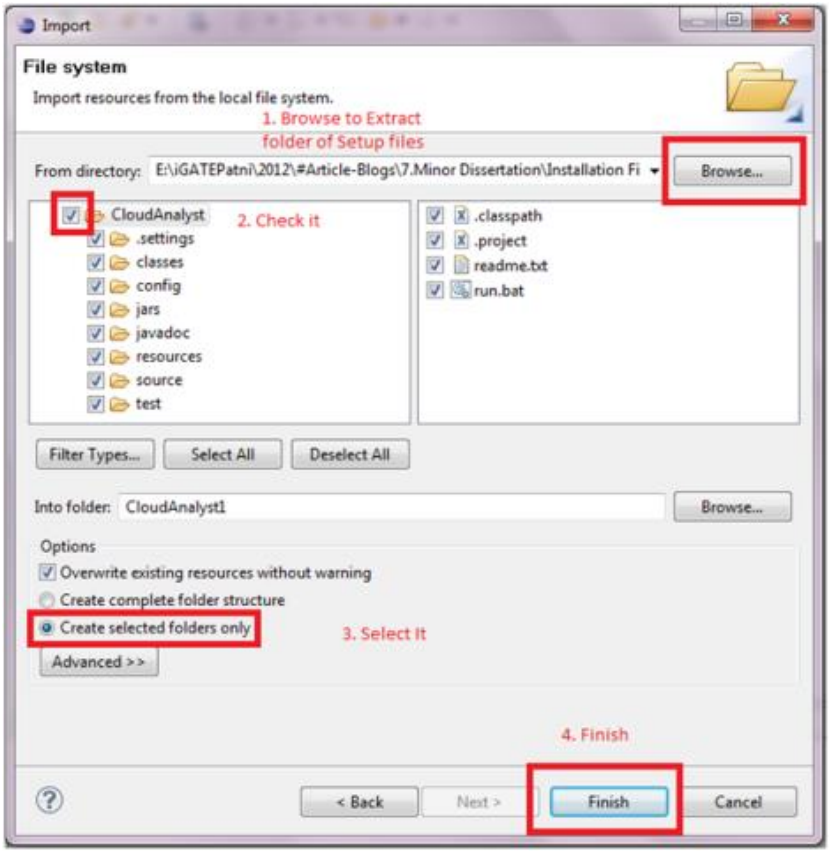
NEW JAVA PROJECT IN ECLIPSE: JAVA SETTINGS

Go to File-

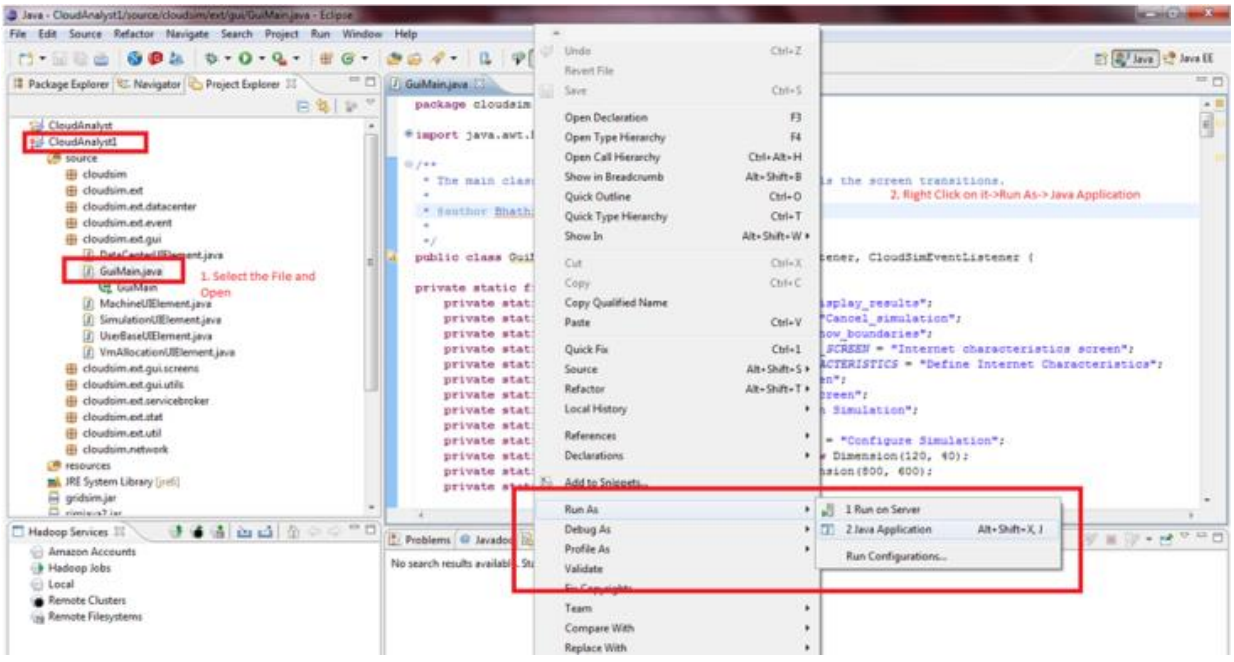
>Import



NEW JAVA PROJECT IN ECLIPSE: IMPORT SOURCE CODE FROM EXISTING PROJECT



NEW JAVA PROJECT IN ECLIPSE: IMPORT RESOURCES FROM LOCAL FILE SYSTEM



RUN CLOUDANALYST IN ECLIPSE

Done!!!

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 3

Aim: Modify or propose a new load balancing algorithm compatible with Cloud Analyst

Theory

Cloud computing is an attracting technology in the field of computer science. In Gartner's report [1], it says that the cloud will bring changes to the IT industry. The cloud is changing our life by providing users with new types of services. Users get service from a cloud without paying attention to the details [2]. NIST gave a definition of cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[3]. Cloud computing is an evolving paradigm with changing definitions, it is defined as a virtual infrastructure which provides shared information and communication technology services, via an internet i.e. cloud. Cloud computing provides a computer user access to Information Technology (IT) services (i.e., applications, servers, data storage) without requiring an understanding of the technology or even ownership of the infrastructure. Cloud Computing is getting advanced day by day. Cloud works on the principle of virtualization of resources with on-demand and pay-as–you go model policy. The five important characteristics of cloud computing defined by NIST includes On-demand self-service, Global network access, Distributed resource pooling, Scalable , Measured service[4]. Based on the domain or environment in which clouds are used, clouds can be divided into 3 categories Public Clouds (which are available to clients from a third party service provider via the

Internet. e.g., Amazon ,Google and IBM),Private Clouds(a business essentially turns its IT environment into a cloud and uses it to deliver services to the users),Hybrid Clouds(Hybrid cloud means either two separate clouds joined together (public, private, internal or external or a combination of virtualized cloud server instances used together with real physical hardware)[5]. In the whole, cloud computing provides us the attracting conventional services like: Software as a Service (SaaS) ,where the user uses different software applications from different servers through the Internet which can be consumed using web browsers without purchasing or maintaining overhead. Some examples are Gmail, Salesforce.com, etc. Platform as a Service (PaaS), where Application development framework offered as a service to developers for quick deployment of their code. Some examples for PaaS include Google App Engine, Heroku, Cloud Foundry, etc. Last but not the least Infrastructure as a Service (IaaS) where Shared infrastructure such as servers, storage and network are delivered as a service over the internet. Some examples include Amazon Web Services, Rackspace Cloud, etc[6]. Cloud computing architectures are inherently parallel, distributed and serve the needs of multiple clients in different scenarios. This distributed architecture deploys resources distributive to deliver services efficiently to users in different geographical channels [7].Clients in a distributed environment generate request randomly in any processor. So the major drawback of this randomness is associated with task assignment. The unequal task assignment to the processor creates imbalance i.e., some of the processors are overloaded and some of them are under loaded [8]. The objective of load balancing is to achieve a high user satisfaction and resource utilization ratio, and to avoid the situation where nodes are either heavily loaded or under loaded in the network, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption [9]. There are mainly two types of load balancing algorithms .In Static algorithm the traffic is divided evenly among the servers. This algorithm requires a prior knowledge of system resources, so that the decision of shifting of the load does not depend on the current state of system. Static algorithm is proper in the system which has low variation in load. In Dynamic algorithm the lightest server in the whole network or system is searched and preferred for balancing a load. For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load [10]. Cloud computing implements virtualization technique in which a single system can be virtualized into number of virtual systems [11]. Load balancing decides which client will use the

virtual machine and which requesting machines will be put on hold. Load balancing of the entire system can be handled dynamically by using virtualization technology where it becomes possible to remap Virtual Machines (VMs) and physical resources according to the change in load. Due to these advantages, virtualization technology is being comprehensively implemented in Cloud computing. A Virtual Machine (VM) is a software implementation of a computing environment in which an operating system (OS) or program can be installed and run. The Virtual Machine typically changes a physical computing environment and requests for CPU, memory, hard disk, network and other hardware resources are managed by a virtualization layer which translates these requests to the underlying physical hardware. VMs are created within a virtualization layer, such as a hypervisor or a virtualization platform that runs on top of a client or server operating system. This operating system is known as the host OS. The virtualization layer can be used to create many individual, isolated VM environments, where multiple requests or tasks can execute in multiple machines [12].

Dynamic Load Balancing Algorithms

A. Round Robin Algorithm

The name of this algorithm suggests that it works in round robin manner [13]. When the Data Center Controller gets a request from a client it notifies the round robin load balancer to allocate a new virtual machine (VM) for processing. Round robin load balancer (RRLB) picks a VM randomly from the group and returns the VM id to Data Center Controller for processing. In this way the subsequent requests are processed in a circular order. However there is a better allocation policy called weighted round robin balancer [14] in which we can assign a weight to each VM so that if one VM is capable of handling twice as much load as other then the former gets the weight of 2 whereas the later gets the weight of 1.

B. **Active Monitoring Load Balancing Algorithm**

This algorithm is also called as equally spread current execution (ESCE) load balancing. It uses active monitoring load balancer for equally spreading the execution of loads on different virtual machines. The steps of this algorithm are described as follows referring to Fig 1.

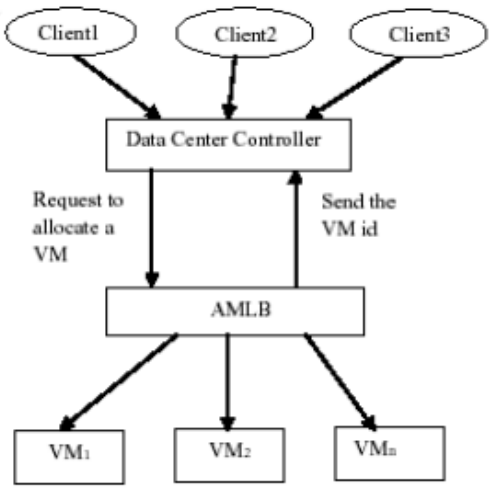


Figure 1.Active monitoring load balancing.

Active monitoring load balancer (AMLB) maintains an index table of virtual machines and the number of allocations assigned to each virtual machine. Data Center Controller receives a new request from a client. When a request for allocation of new VM from Data Center Controller arrives at AMLB, it parses the index table from top until the least loaded VM is found. When it finds, it returns the VM id to the Data Center Controller. If there is more than one found, AMLB uses first come first serve (FCFS) basis to choose the least loaded. Simultaneously, it also returns the VM id to the Data Center Controller. Then the Data Center communicates the VM identified by that id. The Data Center Controller notifies the AMLB about the new allocation. After that AMLB updates the allocation table by increasing the allocation count by 1 for that VM. When a VM suitably finishes processing the assigned request, it forwards a response to the Data Center Controller. On receiving the response it notifies the AMLB about the VM de-allocation The AMLB updates the allocation table by decreasing the allocation count for that VM by 1. Figure 1.Active monitoring load balancing.

C. Throttled Algorithm

This algorithm implements a throttled load balancer (TLB) to monitor the loads on each VM. Here each VM is assigned to only one task at a time and can be assigned another task only when the current task has completed successfully. The algorithm steps can be described as follows: The job of TLB is to maintain an index table of all VMs as well as their current states (Available or Busy). The client first makes a request to Data Centre

Controller for the allocation of appropriate VM and to perform the recommended job. The Data Centre Controller queries the TLB for allocation of the VM. The TLB scans the index table from top to bottom until the first available VM is found. If it finds, then TLB returns the VM id to the Data Center Controller. The Data Centre communicates the request to the VM identified by the id. Further, the Data Centre acknowledges TLB about the new allocation and revises the index table by increasing the allocation for that VM by 1. On the other hand, if the TLB doesn't find any VM in the available state it simply returns null. In this case Data Center Controller queues the request until the availability of any VM. When a VM suitably finishes processing the request, it forwards a response to the Data Center Controller. On receiving it, the Data Center Controller acknowledges the TLB regarding VM de-allocation. The TLB updates the allocation table by decreasing the allocation count for the VM by 1[15].

Response Time Calculation

The purpose of these algorithms is to calculate the expected response time. We use the following formula for calculation:

$$\text{Response Time} = \text{Fint} - \text{Arrt} + \text{Tdelay}$$

where, Arrt is the arrival time of user request and Fint is the finish time of user request and Tdelay is the transmission delay.

However, Tdelay can be calculated as:

$$\text{Tdelay} = \text{Tlatency} + \text{Ttransfer Here,}$$

Tlatency is the network latency and Ttransfer is the time taken to transfer the size of data of a single request (D) from source location to destination.Tlatency is taken from the latency matrix (after applying Poisson distribution on it for distributing it)held in the internet characteristics.

$$\text{Ttransfer} = D / \text{Bwperuser}$$

where

$$\text{Bwperuser} = \text{Bwtotal} / N;$$

Bwtotal is the total available bandwidth (held in the internet characteristics) and N is the number of user requests currently in transmission. The internet characteristics also keep track of the number of user requests in-flight between two regions for the value of N

Simulation Step

In order to analyze the above discussed algorithms we have used the tool called Cloud Analyst [16]. Basically cloud analyst is a cloudsim [17] based GUI tool used for modeling and analysis of large scale cloud computing environment. Moreover, it enables the modeler to execute the simulation repeatedly with the modifications to the parameters quickly and easily. The following diagram shows the GUI interface of cloud analyst tool. It comes with three important menus: configure simulation, define internet characteristics and run simulation [17] [18].These menus are used for setting of the entire simulation process.

Figure2. GUI Interface of Cloud Analyst.

The tool provides us the feature of switching algorithms according to our requirement. Simulation setup and analysis of results are carried out for a period of 60 minutes by taking different numbers of users, 3 data centers i.e. DC1, DC2, and DC3 having 75, 50 and 25 numbers of VMs respectively. The other parameters are fixed according to Table 1 as shown.

TABLE 1: Setting of Parameters

Parameter	Value passed
VM-image size	10000
VM-memory	1024MB
VM-Bandwidth	1000
Service Broker Policy	Opimise response time
Data center architecture	x86
Data center-OS	Linux
Data center-VMM	Xen
Data center-No of VMs	DC1-75 DC2-25 DC3-50
Data center-memory per machine	2GB
Data center-storage per machine	1GB
Data center-available bandwidth per machine	1000000
Data center-processor speed	1000
Data center-VM policy	Time shared
User grouping factor	1000
Request grouping factor	250
Executable instruction length	250

After performing the simulation, the results computed by cloud analyst is as shown in the tables given below. We have used the above defined configuration for each load balancing policy one by one and depending on that the result calculated for the metrics like response time, request processing time in fulfilling the request has been shown. Overall response time calculated by the cloud analyst for each loading policy has been shown in the table 2, 3 and 4 respectively. As can be seen from the figure the overall response time of Round Robin policy and ESCEL policy is almost same while that of the Throttled Policy is somewhat low as compared to other two policies.

TABLE 2: Response time using Round Robin Policy

	Overall Response time	Datacenter processing time
Avg(ms)	187.78	29.13
Min(ms)	55.85	12.51
Max(ms)	384.61	61.69

TABLE 3: Response time using Active Monitoring Load Balancing Policy

	Overall Response time	Datacenter processing time
Avg(ms)	187.72	29.13
Min(ms)	55.85	12.51
Max(ms)	384.61	61.69

TABLE 4: Response time using Throttled Policy

	Overall Response time	Datacenter processing time
Avg(ms)	187.68	29.13
Min(ms)	55.85	12.51
Max(ms)	381.96	61.69

From the table it is clear that average processing time at datacenters for all three policies is same.

Conclusion

The response time and data transfer cost is a challenge of every engineer to develop the products that can increase the business performance in the cloud based sector. The several strategies lack efficient scheduling and load balancing resource allocation techniques leading to increased operational cost and give less customer satisfaction. A greater challenge in minimization of

response time is widely seen for each and every engineer of IT sector to develop the products which can increase the efficiency of business performance, customer satisfaction in cloud based environment. Keeping these things in mind we have simulated three different load balancing algorithms: Round robin, Active monitoring and Throttled for executing the user requests in cloud environment. Each algorithm is observed and their scheduling criteria like average response time and data center processing time are found. We have found that the parameters: response time, data processing time are almost similar in case of Round Robin and active monitoring load balancing policies. However, these parameters are slightly improved in case of Throttled load balancing. Hence we conclude that Throttled load balancing is an effective and efficient one than the other two that discussed. The performance of the given algorithms can also be increased by varying different parameters or developing an innovative algorithm for provisioning of cloud computing resources. Future work can be based on this algorithm modified and implemented for real time system. Better response time can be expected if we apply some evolutionary algorithms such as PSO, ACO, and ABC instead of classical algorithms.

References

1. R. Hunter, The why of cloud, http://www.gartner.com /DisplayDocument?doc cd=226469&ref= g noreg, 2012.
2. M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis,and A. Vakali, Cloud computing: Distributed internet computing for IT and scientific research, Internet Computing, vol.13, no.5, pp.10-13, Sept.-Oct. 2009.
3. P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf,2012. techtarget.com /definition/public-cloud, 2012.
4. Mell,P, Grance, T "The NIST definition of Cloud Computing" version 15. National Institute of Standards and Technology (NIST),Information Technology Laboratory (October 7, 2009)
5. AlexaHuth and JamesCebula"The Basics of Cloud Computing",www.us-cert.gov/reading.../USCERT CloudComputing HuthCebula."
6. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for

Delivering Computing as the 5th Utility, Future Generation Computer Systems", Volume 25, Number 6, Pages: 599-616, ISSN: 0167-739X, Elsevier Science, Amsterdam, The Netherlands, June 2009.

7. M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, September/October 2009, pages 10-13.
8. A. Khiyaita, H. EI Bakkli, M. Zbakh ,Dafir EI Kettani," Load Balancing Cloud Computing: State Of Art", 2010,IEEE.
9. Ram Prassd Pandhy (107CS046), P Goutam Prasad rao (107CS039). "Load balancing in cloud computing system" Department of computer science and engineering National Institute of Technology Rourkela, Rourkela-769008, Orissa, India May-2011.
10. Load Balancing in Cloud computing, http://community.citrix.com /display/ cdn/ Load+Balancing.
11. J. Sahoo, S. Mohapatra and R. lath "Virtualization: A survey on concepts, taxonomy and associated security issues" computer and network technology (ICCNT), IEEE, pp. 222-226. April 2010.
12. Bhaskar. R, Deepu.S. R and Dr.B. S. Shylaja "Dynamic Allocation Method For Efficient Load Balancing In Virtual Machines For Cloud Computing Environment" September 2012.
13. Saroj Hiranwal and Dr. K.C. Roy, "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice" International Journal Of Computer Science And Communication JulyDecember 2011 ,Vol. 2, No. 2 , Pp. 319-323.
14. Jasmin James and Dr. Bhupendra Verma, "Efficient VM load balancing algorithm for a cloud computing environment", International Journal on Computer Science and Engineering (IJCSE), 09 Sep 2012. 44 International Journal of Engineering Research & Technology (IJERT) IJERT www.ijert.org NCRTS`14 Conference Proceedings ISSN: 2278-0181
15. Soumya Ranjan Jena and Zulfikhar Ahmad, " Response Time Minimization of Different Load Balancing Algorithms in Cloud Computing Environment", International Journal of Computer Applications, Volume 69,No.17,Pages:22-23 May 2013.
16. Bhathiya Wickremasinghe, "CloudAnalyst: A CloudSim based Tool for Modelling and

Analysis of Large Scale Cloud Computing Environments" MEDC project report, 433-659 Distributed Computing project, CSSE department., University of Melbourne, 2009.

17. R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling And Simulation Of Scalable Cloud Computing Environments And The Cloudsim Toolkit: Challenges And Opportunities," Proc. Of The 7th High Performance Computing And Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.
18. Judith Hurwitz, Robin Bloor, and Marcia Kaufman, "Cloud computing for dummies" Wiley Publication (2010).

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 4

Aim: Integrating GoogleApp Engine API's in Eclipse and develop an application in Java/Python on the top of Google Cloud.

How to use **Eclipse** to create a **Google App Engine** (GAE) **Java** project (hello world example), run it locally, and deploy it to Google App Engine account.

Requirement: Tools used

1. JDK 1.6
2. Eclipse 3.7 + Google Plugin for Eclipse
3. Google App Engine Java SDK 1.6.3.1

Theory:

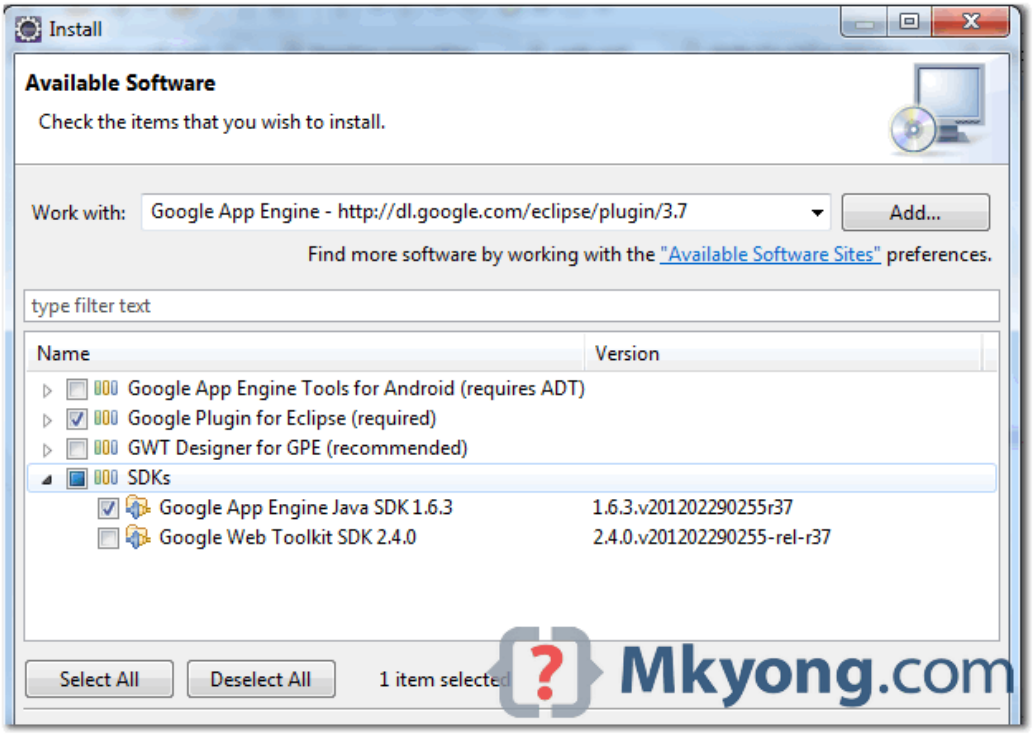
1. Install Google Plugin for Eclipse

how to install Google Plugin for Eclipse.

In Eclipse 3.7, click "**Help**" –> "**Install New Software…**", copy and paste following URL :

http://dl.google.com/eclipse/plugin/3.7

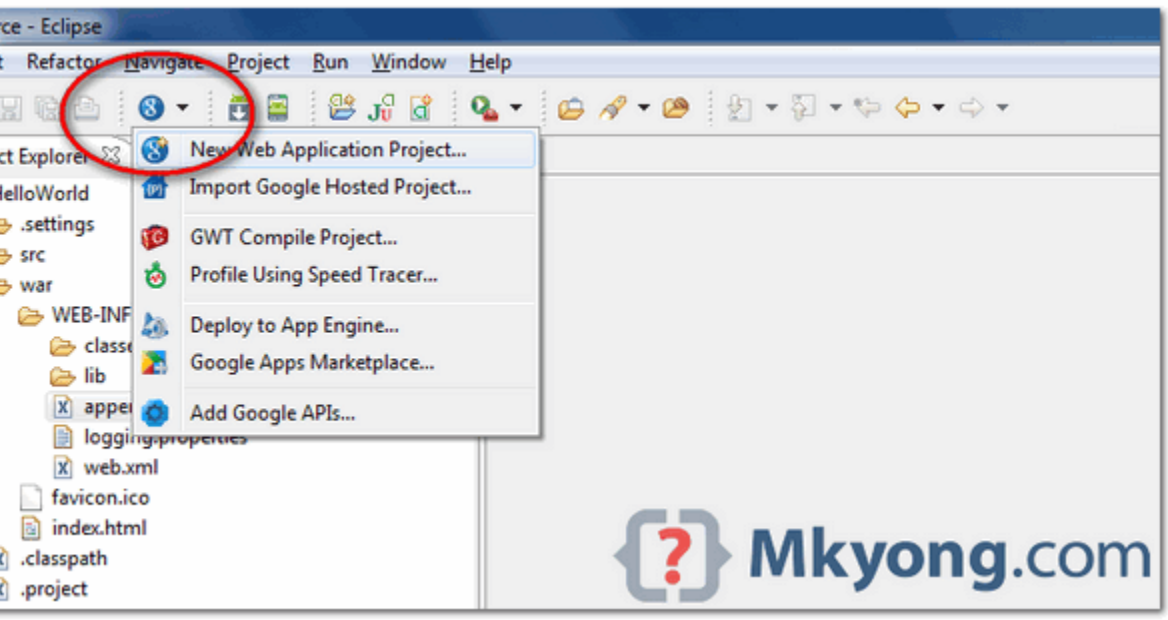
Figure : Select "Google Plugin for Eclipse (required), and Google App Engine SDK for Java.



Verification

Wait few minutes for the installation progress, when done, Eclipse prompts you to restart, click yes and the Google Plugin for Eclipse is installed.

Figure – A small Google icon is available in the Eclipse toolbar.



If you install the Google App Engine Java SDK together with "**Google Plugin for Eclipse**", then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.

2. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select "**New Web Application Project…**"

Figure – New Web Application Project

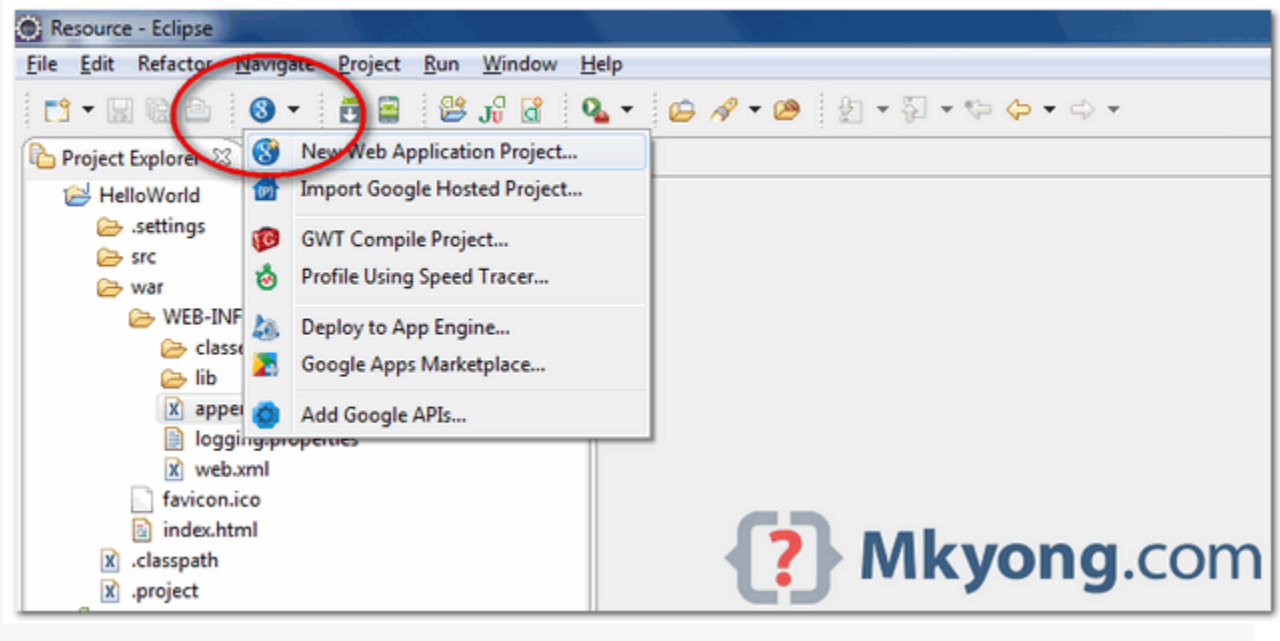
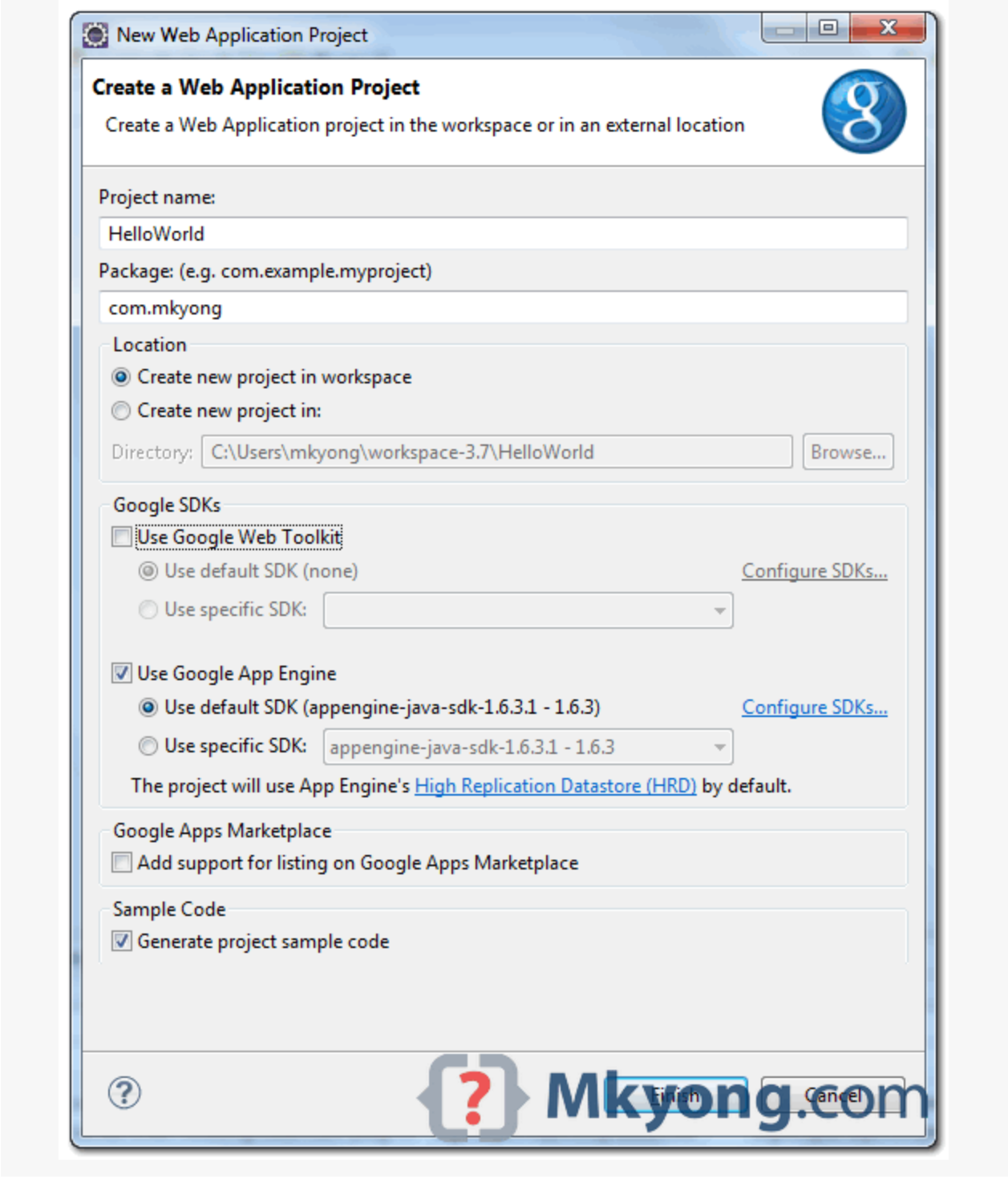


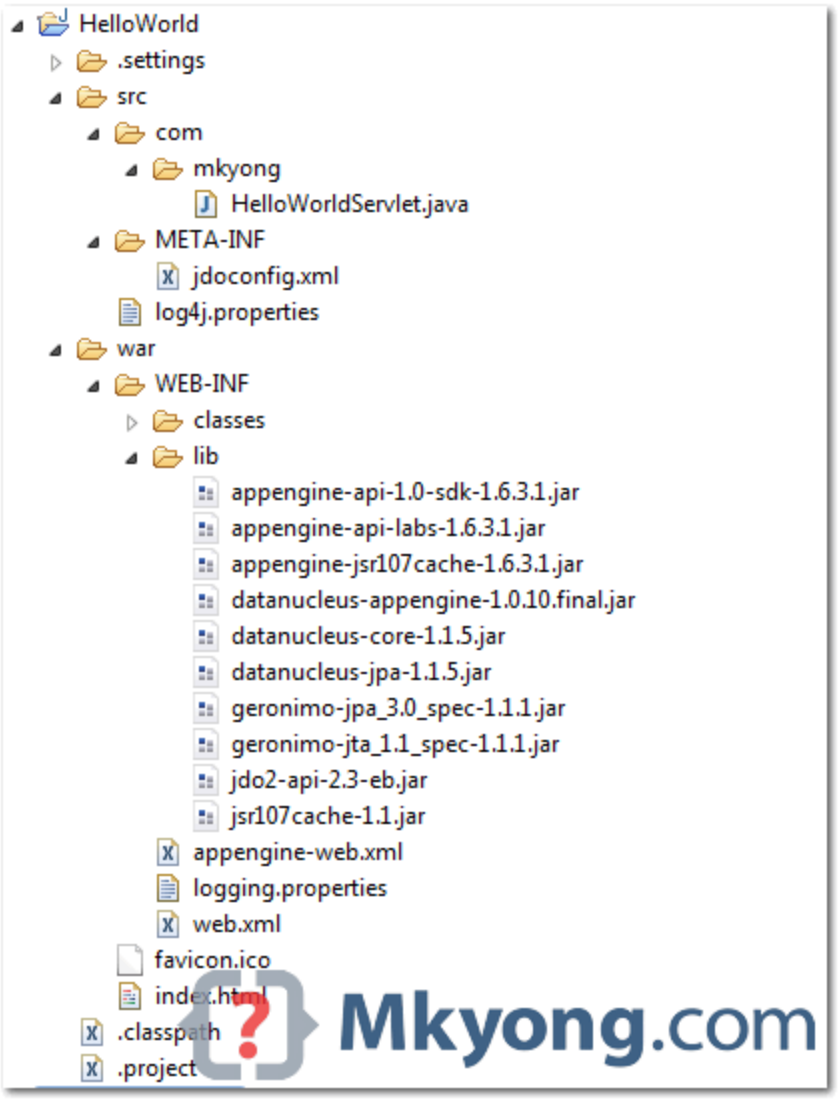
*Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link.*



Click finished, Google Plugin for Eclipse will generate a sample project automatically.

3. Hello World

Review the generated project directory.



Nothing special, a standard Java web project structure.

```
HelloWorld/
 src/
  ...Java source code...
  META-INF/
   ...other configuration...
 war/
  ...JSPs, images, data files...
  WEB-INF/
   ...app configuration...
   lib/
```

```
      ...JARs for libraries...
    classes/
      ...compiled classes...
```

The extra is this file "appengine-web.xml", Google App Engine need this to run and deploy the application.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
 <application></application>
 <version>1</version>

 <!-- Configure java.util.logging -->
 <system-properties>
   <property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
 </system-properties>

</appengine-web-app>
```

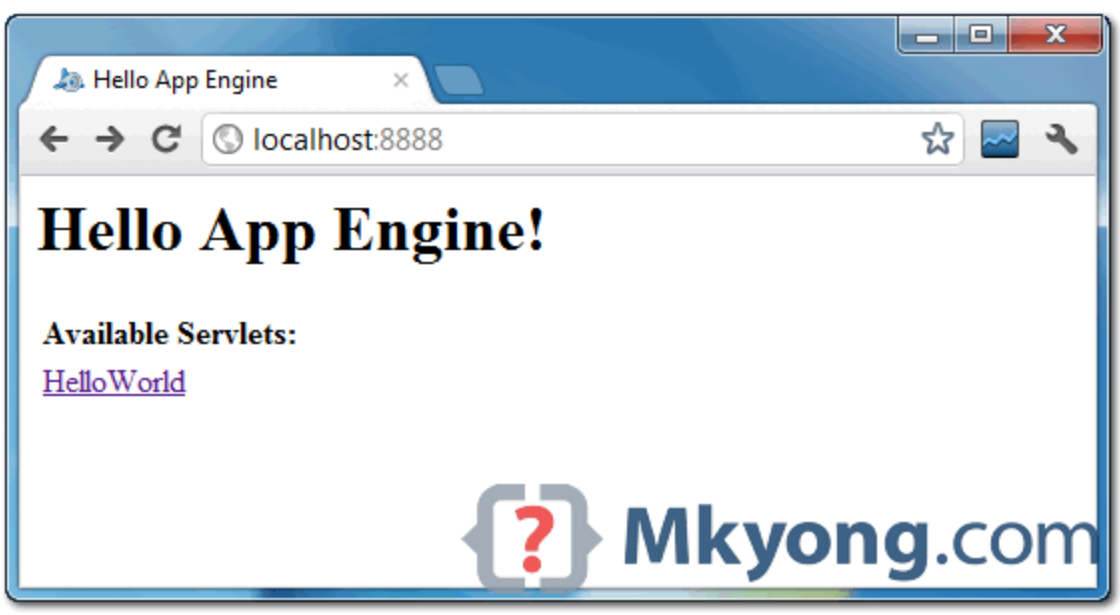
4. Run it local

Right click on the project and run as "**Web Application**".

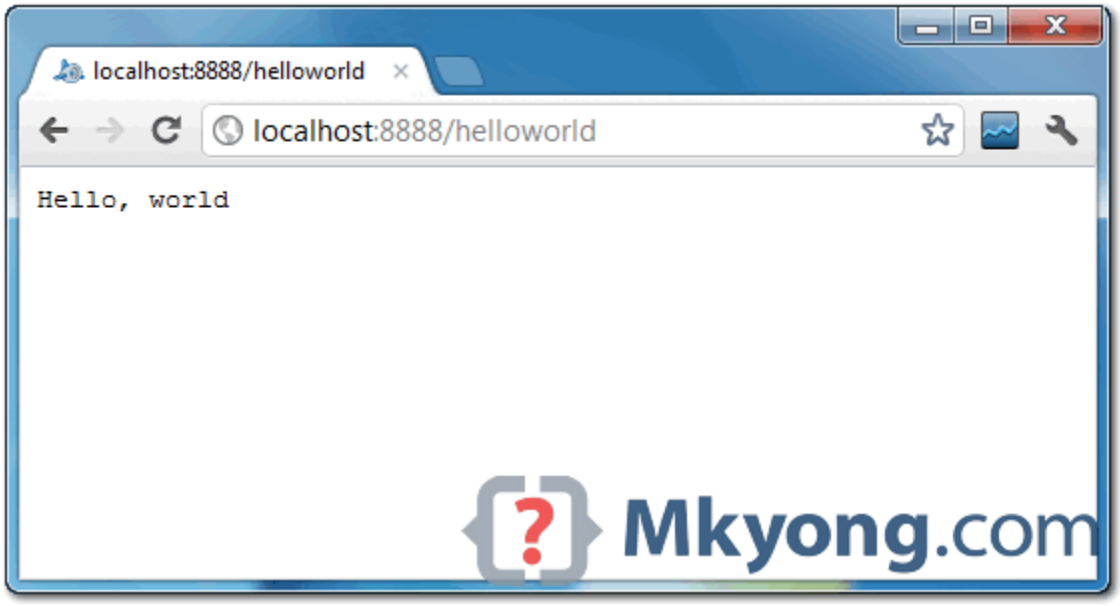
Eclipse console :

```
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start
INFO: The admin console is running at http://localhost:8888/_ah/admin
```

Access URL http://localhost:8888/, see output



and also the hello world servlet – http://localhost:8888/helloworld



5. Deploy to Google App Engine

Register an account on https://appengine.google.com/, and create an application ID for your web application.

In this demonstration, I created an application ID, named "mkyong123", and put it in appengine-web.xml.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>mkyong123</application>
  <version>1</version>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
  </system-properties>

</appengine-web-app>
```

To deploy, see following steps:

Figure 1.1 – Click on GAE deploy button on the toolbar.

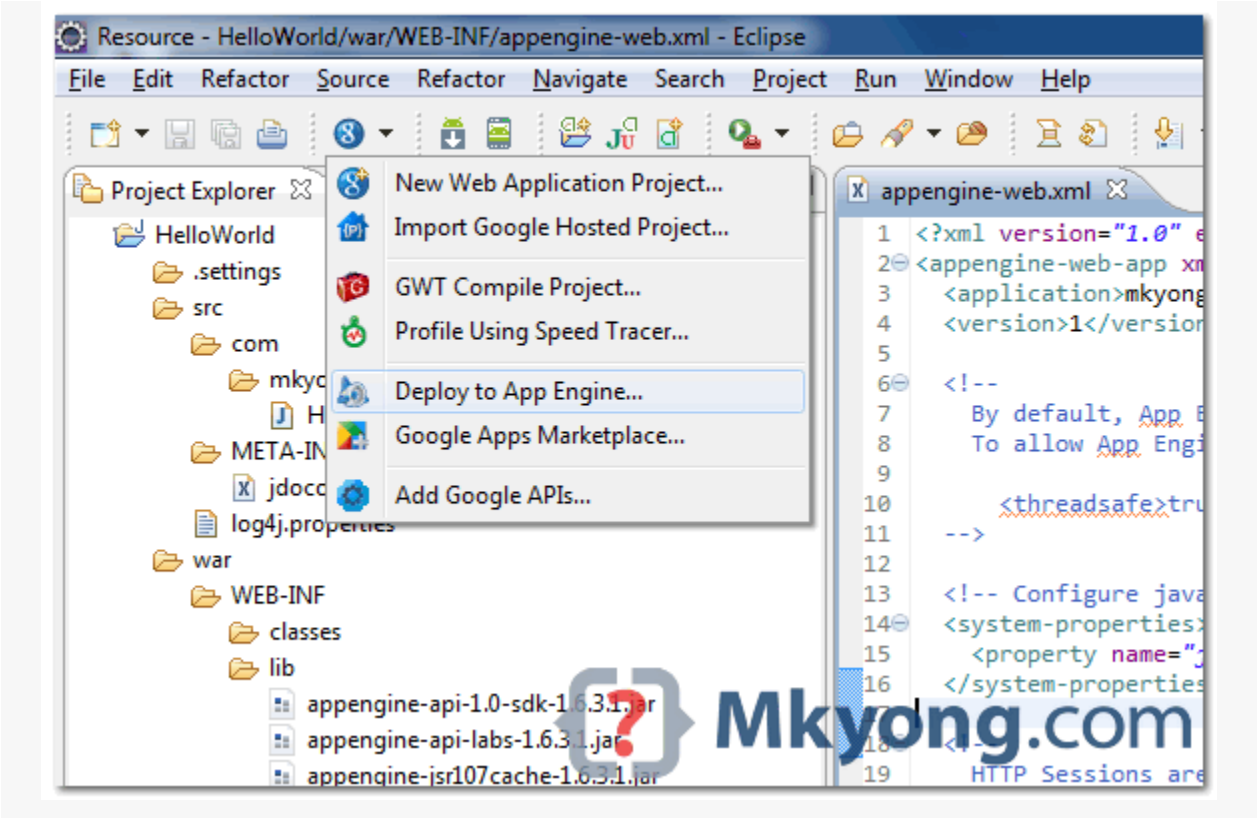


Figure 1.2 – Sign in with your Google account and click on the Deploy button.

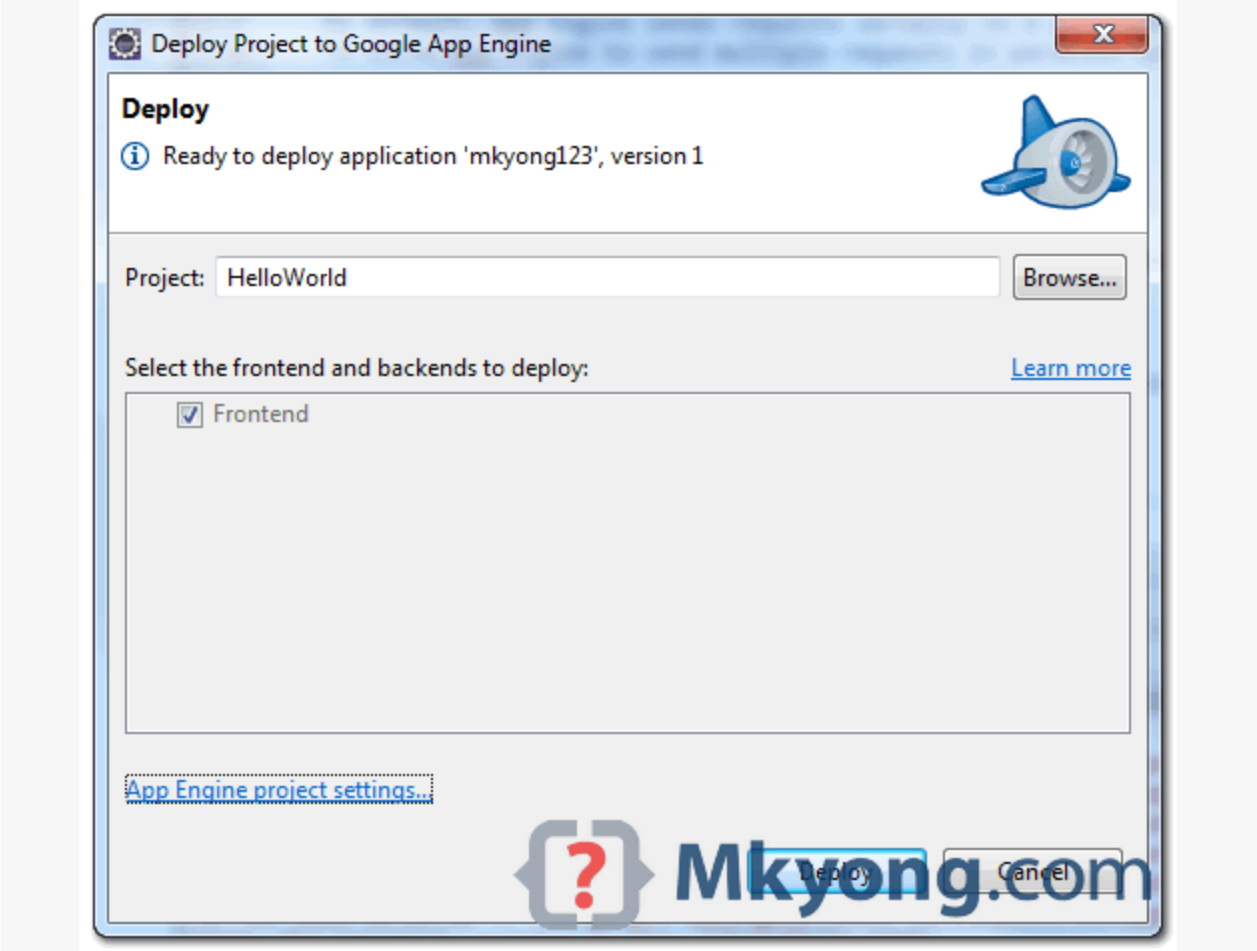


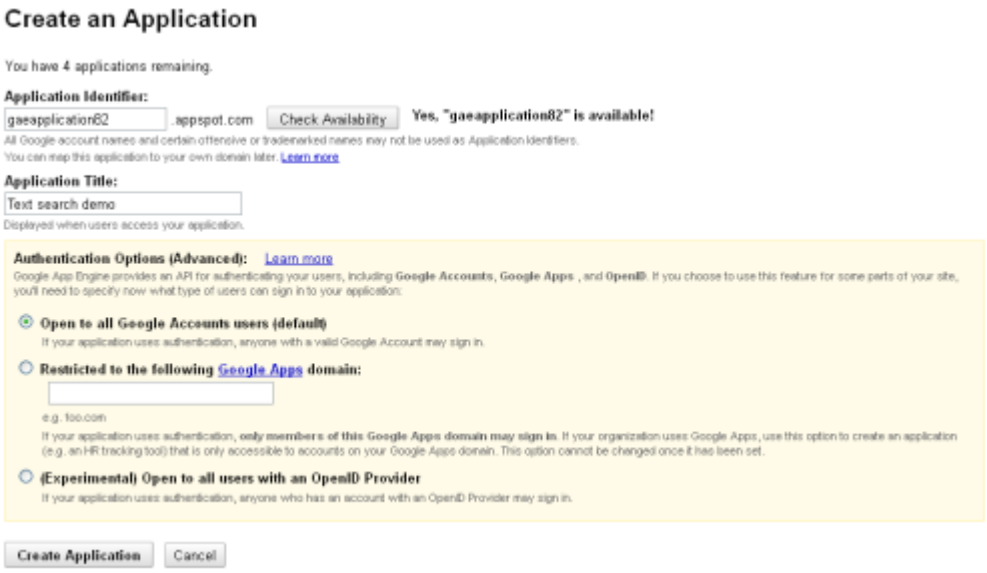
Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/

Done.

Deploying a New Application GAE

With Codenvy you can deploy Java, Python and PHP apps to Google App Engine.

- create a new project from scratch and choose either Java web application or Python and Google App Engine as PaaS (if you have already created a project, then open it and go to **PaaS > Google App Engine > Create Application**)
- enter project name and choose a **Template**
- check **Use existing GAE ID** if you want to overwrite an existing app
- click **Create** button
- if you deploy your first app to GAE from Codenvy you need to authenticate
- allow access to proceed
- enter required information at the GAE webpage *(Application Title is optional)*



Click to see a full sized image

- once you click **Create Application**, the browser's tab will be automatically closed in a few seconds
- when you are back to your Codenvy workspace, click **Deploy** to push the app to GAE

The process may take several minutes, and you will see a confirmation message in the Output panel with the application url - *yourappname.appspot.com*

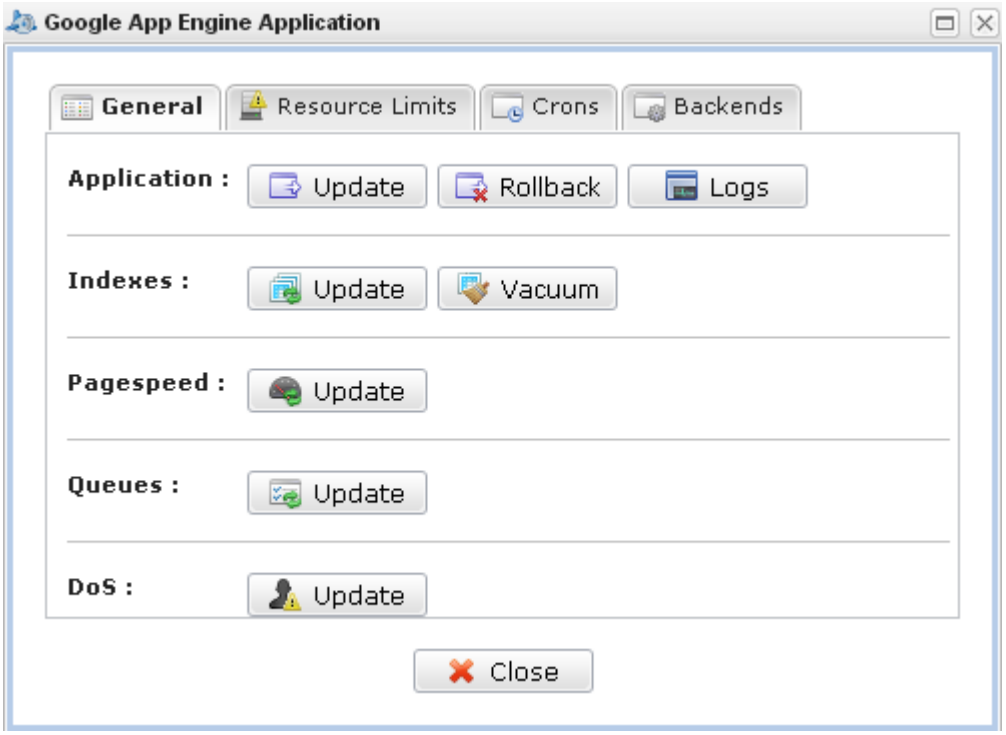
Make sure you use the same Google ID to log in to Codenvy and Google App Engine. Using different accounts may cause 401 or 404 errors. 401 error can be fixed by logging out, and then logging in Google App Engine at **PaaS > Google App Engine > Logout/Login**

Watch the entire process of deploying an app to GAE in the below video

Updating an Existing Application GAE

Having deployed your application to GAE, you can make some changes to the code and easily update it.

- The application is updated at **Project > PaaS > Google App Engine**
- The project is re-built and re-deployed once you press **Application Update** button. An alternative way to update your GAE app is to go to **PaaS > Google App Engine > Update Application**.

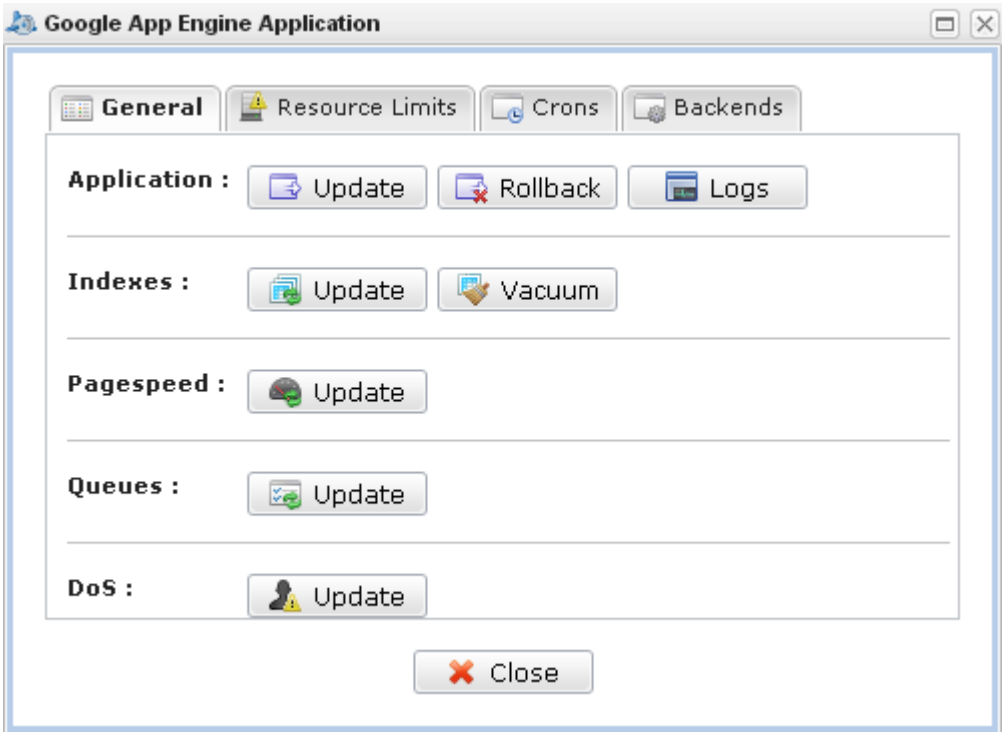


Three messages (project build, start and end of deployment) appear in the Output panel one by on. Once done, changes are implemented in the application hosted on appspot.com.

Managing GAE Applications

You can manage your GAE application at **Project >PaaS > Google App Engine.**

It’s possible to modify and vacuum Indexes, PageSpeed, Queues, DoS as well as have a look at recourse limits, cron jobs and backends.



Import an Existing GAE Application

If you have a GAE application which you need to import to Codenvy, here’s a workaround (this is not a direct import of source code, so it will take a few minutes or so):

- download source code of your app (of course, this step can be omitted if you have in on GitHub and sync it regularly). You can download source code of your Java and Python apps using SDK command line (check out GAE documentation).
- push this code to GitHub or whatever remote repository you use
- clone your GitHub project to Codenvy
- open *appengine-web.xml* file and edit application ID, if necessary, for example *<application>javagae112</application>* (enter the app ID you need to update on GAE)
- if you want to create a new version of the same app, you can change it as well, for example *<version>2</version>*

```
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>java112</application>
```

<version>1</version>

- update application at **Project > PaaS > Google App Engine**.

Once the app is updated, you can change and update it anytime directly from Codenvy.

Since Codenvy uses Maven as a build manager, the projects you clone should also be built with Maven, i.e. contain pom.xml file in the root project folder.

You may have a look at this short video demonstrating the procedure of importing an existing GAE app to Codenvy using GitHub.

Device Support

Codenvy currently supports all desktop and laptop devices. We currently provide touch device support through the use of the Puffin Web Browser which virtualizes double clicks and right clicks. We have not yet created a native touch UI design. Vote for this feature at our Uservoice page.

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 5

Aim: Make the registration groupwise on Google and register your application by using google application-ID

Theory:

Google Sign-In manages the OAuth 2.0 flow and token lifecycle, simplifying your integration with Google APIs. A user always has the option to revoke access to an application at any time.
This document describes how to complete a basic Google Sign-In integration.

Create authorization credentials

Any application that uses OAuth 2.0 to access Google APIs must have authorization credentials that identify the application to Google's OAuth 2.0 server. The following steps explain how to create credentials for your project. Your applications can then use the credentials to access APIs that you have enabled for that project.

1. Go to the Credentials page.
2. Click **Create credentials > OAuth client ID**.
3. Select the **Web application** application type.
4. Name your OAuth 2.0 client and click **Create**

After configuration is complete, take note of the client ID that was created. You will need the client ID to complete the next steps. (A client secret is also created, but you need it only for server-side operations.)

Load the Google Platform Library

You must include the Google Platform Library on your web pages that integrate Google Sign-In.
<script src="https://apis.google.com/js/platform.js" async defer></script>

Specify your app's client ID

Specify the client ID you created for your app in the Google Developers Console with the google-signin-client_id meta element.

<meta name="google-signin-client_id"
content="YOUR_CLIENT_ID.apps.googleusercontent.com">

Note: You can also specify your app's client ID with the **client_id** parameter of the **gapi.auth2.init()** method.

Add a Google Sign-In button

The easiest way to add a Google Sign-In button to your site is to use an automatically rendered sign-in button. With only a few lines of code, you can add a button that automatically configures itself to have the appropriate text, logo, and colors for the sign-in state of the user and the scopes you request.

To create a Google Sign-In button that uses the default settings, add a div element with the class g-signin2 to your sign-in page:

Get profile information

After you have signed in a user with Google using the default scopes, you can access the user's Google ID, name, profile URL, and email address.

To retrieve profile information for a user, use the getBasicProfile() method.

```
function onSignIn(googleUser) {
  var profile = googleUser.getBasicProfile();
  console.log('ID: ' + profile.getId()); // Do not send to your backend! Use an ID token instead.
  console.log('Name: ' + profile.getName());
  console.log('Image URL: ' + profile.getImageUrl());
  console.log('Email: ' + profile.getEmail()); // This is null if the 'email' scope is not present.
}
```

Important: Do not use the Google IDs returned by **getId()** or the user's profile information to communicate the currently signed in user to your backend server. Instead, send ID tokens, which can be securely validated on the server.

Sign out a user

You can enable users to sign out of your app without signing out of Google by adding a sign-out button or link to your site. To create a sign-out link, attach a function that calls the GoogleAuth.signOut() method to the link's onclick event.

```
<a href="#" onclick="signOut();">Sign out</a>
<script>
  function signOut() {
    var auth2 = gapi.auth2.getAuthInstance();
    auth2.signOut().then(function () {
      console.log('User signed out.');
    });
  }
</script>
```

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 6

Aim: Creating a Warehouse Application in SalesForce.com's Force.com.

Theory:

Steps to create an application in Force.com by declarative model

Step 1: Click on Setup → Create → Objects → New custom object →

Label: MySale

Pular Label: MySales

Object Name: MySale

Record Name: MySale

Description Data Type: Text

→ Click on Save.

Step 2: Under MySale Go to Custom Field and Relationships →Click on New Custom Field

Creating 1st Field:--

select Data type as Auto Number →next→

Enter the details → Field Label: PROD_ID→ Display Format: MYS-{0000}

→Starting Number: 1001 →Field Name: PRODID→Next→Save & New

Creating 2nd Field:--

select Data type as Date → next →

Enter the details → Field Label: Date of Sale → Field Name: Date_of_Sale

→Default Value: Today()-1 →Next → Save & New

Creating 3rd Field:--

select Data type as Number →next

→Enter the details → Field Label: Quantity Sold→ Length:3 Decimal places:0 →Default Value: Show Formulae Editor:1 →Next →Save & New

Creating 4th Field:--

select Data type as Currency→next

→Enter the details→Field Label: Rate →Field Name: Rate→ Length:4 Decimal places:2 →Default Value: 10 →Next→ Save & New

Creating 5th Field:--

select Data type as Currency → next→

MySale field→Quantity__Sold__c*Rate__c→next →save.

Now create an App

Setup→ Create→ App→ new→ MyShop→ Next→

Select an Image → Next → Add Object → MySales.

Now create an Tab

Setup →Create→ Tab→ New Custom Tab →Choose MySales object → select tab style → save.

On the top in the tab bar you can see the tab which has been created by you click on the tab you can see your object is opened just click on new button and provide the details mentioned.

Conclusion: In this we have created a MyShop Application on Force.com using declarative model.

****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 7

Aim: Creating an Application in SalesForce.com using Apex programming Language.

Theory:

Step1:

Log into your Sandbox or Developers Organization.

Click on setup →create→ objects →new custom objects.

Enter Book for label.

Enter Books for plural label.

Click Save.

Step 2:

Now let‘s create a custom field.

In the custom field & relationship section of the Book

Object click new. Select Number for the datatype &

next.

Enter Price for the field Label.

Enter 16 in the length text box.

Enter 2 in the decimal places & Next….next…. save.

Step 3:

Clilck setup →Develop →Apex Classes & click new

In the class Editor enter this

class public class

MyHelloWorld{

public static void applyDiscount(Book c[] books)

```
 {
    for(Book c b:books)
    {b.Price  c*=0.9;}
}
}
```

Step 4:

Add a trigger

A trigger is a piece of code that can execute objects before or after specific data manipulation language events occurred.

Click on setup→ create→ objects→click the object you have created ex:

Book Scroll down you can see Trigger Click on New

In the trigger Editor enter this class

```
trigger HelloWorldTrigger on Book c(before insert)
{
Book c[] books=Trigger.new;

MyHelloWorld.applyDiscount(books);
}
```

Step 5:

Click on setup →create →tabs →new custom tab → choose Book

→ next&.next&..save.

Click on tab Books → new→ insert a name for Book→ insert price for that book→ click on save.

Conclusion:

Thus we have studied how to create and run an application in salesforce developers site by using APEX programming language.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 8

Aim: To study & Implement Web services in SOAP for JAVA Applications.

Theory:

Overview of Web Services

Web services are application components that are designed to support interoperable machine- to-machine interaction over a network. This interoperability is gained through a set of XML- based open standards, such as the Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and Integration (UDDI). These standards provide a common and interoperable approach for defining, publishing, and using web services.

Choosing a Container:

You can either deploy your web service in a web container or in an EJB container. This depends on your choice of implementation. If you are creating a Java EE application, use a web container in any case, because you can put EJBs directly in a web application. For example, if you plan to deploy to the Tomcat Web Server, which only has a web container, create a web application, not an EJB module.

- Choose File > New Project. Select Web Application from the Java Web category .Name the project `Calculator WS Application`. Select a location for the project. Click Next.
- Select your server and Java EE version and click Finish.

Creating a Web Service from a Java Class

- Right-click the `Calculator WS Application` node and choose New > Web Service.

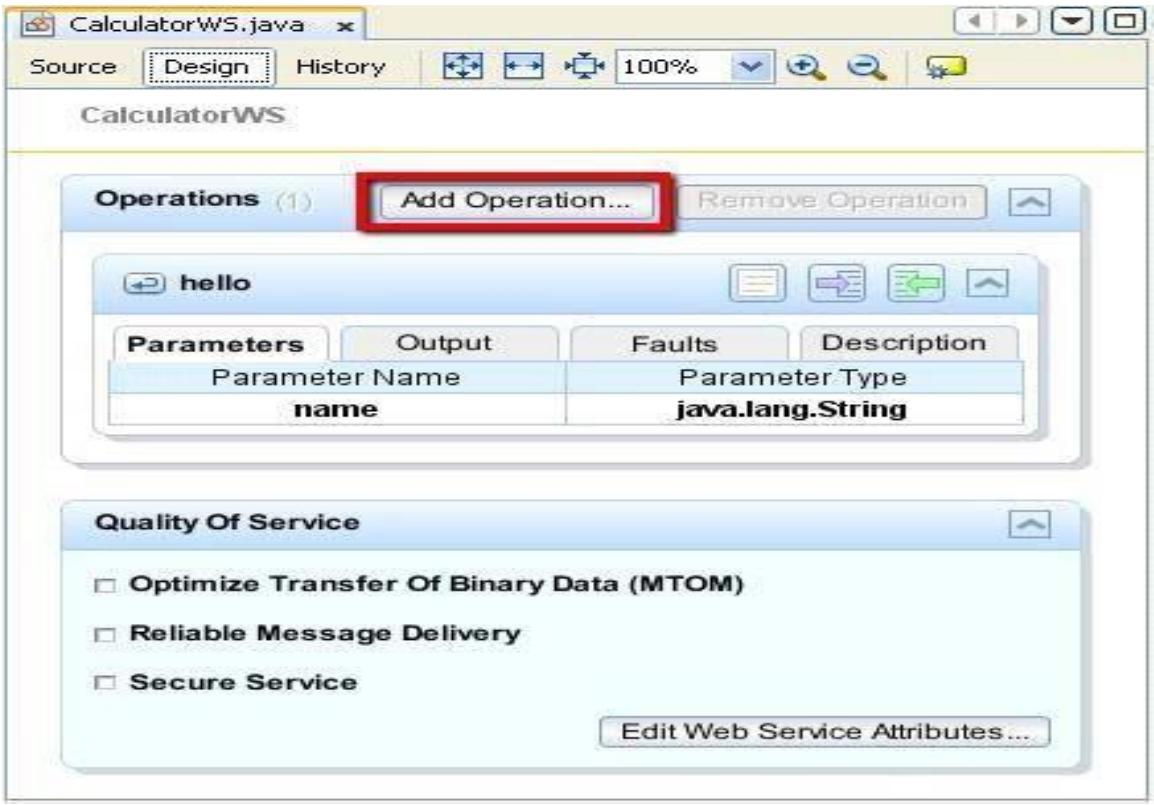
- Name the web service `Calculator WS` and type `org.me.calculator` in Package. Leave Create Web Service from Scratch selected.
- If you are creating a Java EE project on GlassFish or WebLogic, select Implement Web Service as a Stateless Session Bean.
- Click Finish. The Projects window displays the structure of the new web service and the source code is shown in the editor area.

Adding an Operation to the Web Service

The goal of this exercise is to add to the web service an operation that adds two numbers received from a client. The NetBeans IDE provides a dialog for adding an operation to a web service. You can open this dialog either in the web service visual designer or in the web service context menu.

To add an operation to the web service:

- Change to the Design view in the editor.

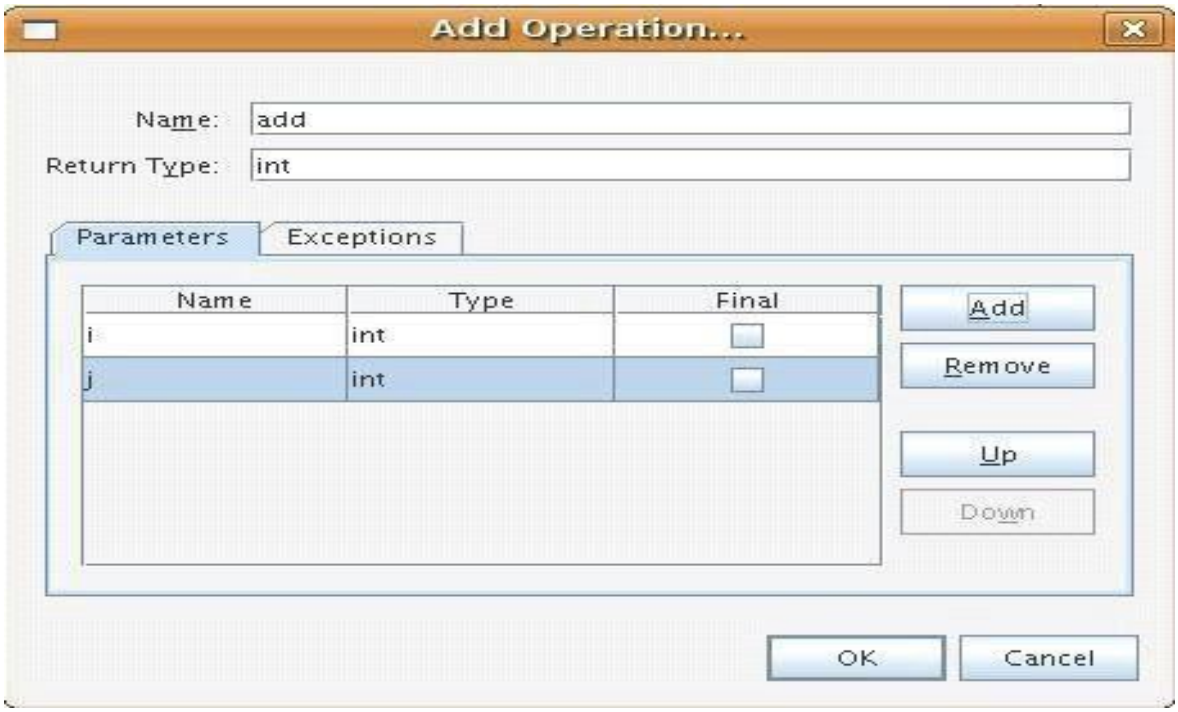


- Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.

- In the upper part of the Add Operation dialog box, type `add` in Name and type `int` in the Return Type drop-down list.

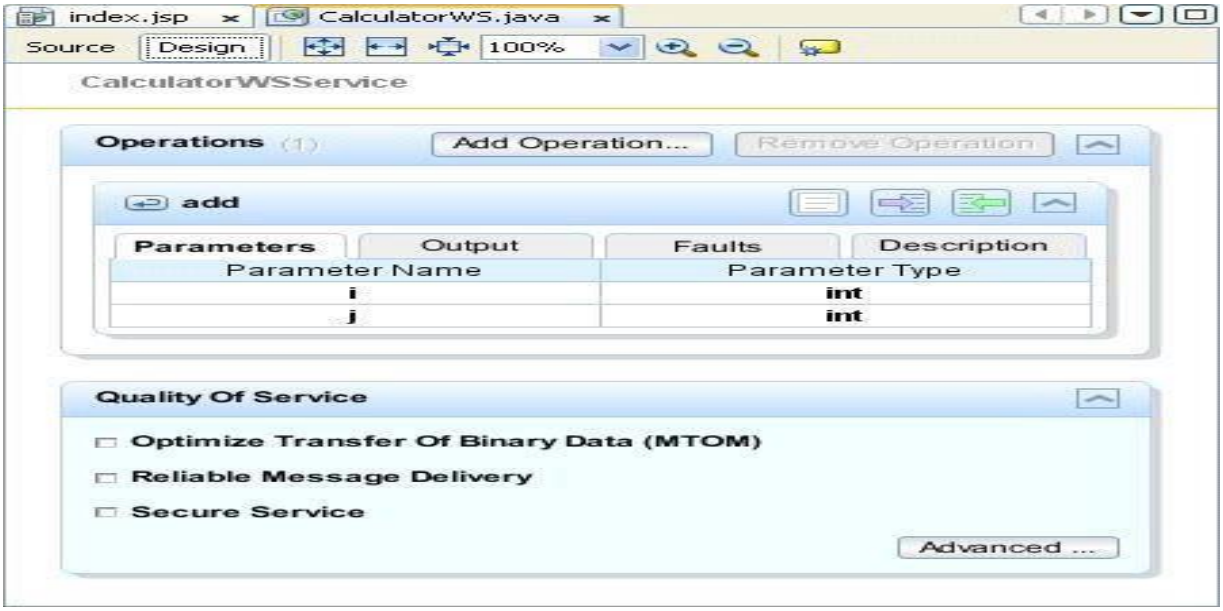
- In the lower part of the Add Operation dialog box, click Add and create a parameter of type `int` named `i`.

Click Add again and create a parameter of type `int` called `j`.



- Click OK at the bottom of the Add Operation dialog box. You return to the editor.
- Remove the default `hello` operation, either by deleting the `hello()` method in the source code or by selecting the `hello` operation in the visual designer and clicking Remove Operation.

The visual designer now displays the following:



- Click Source and view the code that you generated in the previous steps. It differs whether you created the service as a Java EE stateless bean or not. Can you see the difference in the screenshots below? (A Java EE 6 or Java EE 7 service that is not implemented as a stateless bean resembles a Java EE 5 service.)

Note. In NetBeans IDE 7.3 and 7.4 you will notice that in the generated `@WebService` annotation the service name is specified explicitly:`@WebService(serviceName = "CalculatorWS")`.

9. In the editor, extend the skeleton `add` operation to the following (changes are in bold): @WebMethod

```
public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {
    int k = i + j;
    return k;
    }
```

As you can see from the preceding code, the web service simply receives two numbers and then returns their sum. In the next section, you use the IDE to test the web service.

Deploying and Testing the Web Service

After you deploy a web service to a server, you can use the IDE to open the server's test client, if the server has a test client. The GlassFish and WebLogic servers provide test clients.

If you are using the Tomcat Web Server, there is no test client. You can only run the project and see if the Tomcat Web Services page opens. In this case, before you run the project, you need to make the web service the entry point to your application. To make the web service the entry point to your application, right-click the CalculatorWSApplication project node and choose Properties. Open the Run properties and type `/CalculatorWS` in the Relative URL field. Click OK. To run the project, right-click the project node again and select Run.

To test successful deployment to a GlassFish or WebLogic server:

- Right-click the project and choose Deploy. The IDE starts the application server, builds the application, and deploys the application to the server. You can follow the progress of these operations in the CalculatorWSApplication (run-deploy) and the GlassFish server or Tomcat tabs in the Output view.
- In the IDE's Projects tab, expand the Web Services node of the CalculatorWSApplication project. Right-click the CalculatorWS node, and choose Test Web Service.
- The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish server. For the Tomcat Web Server and deployment of EJB modules, the situation is different:
- If you deployed to the GlassFish server, type two numbers in the tester page, as shown below:

Consuming the Web Service

Now that you have deployed the web service, you need to create a client to make use of the web service's `add` method. Here, you create three clients— a Java class in a Java SE application, a servlet, and a JSP page in a web application.

Note: A more advanced tutorial focusing on clients is Developing JAX-WS Web Service Clients.

Client 1: Java Class in Java SE Application

In this section, you create a standard Java application. The wizard that you use to create the application also creates a Java class. You then use the IDE's tools to create a client and consume the web service that you created at the start of this tutorial.

.

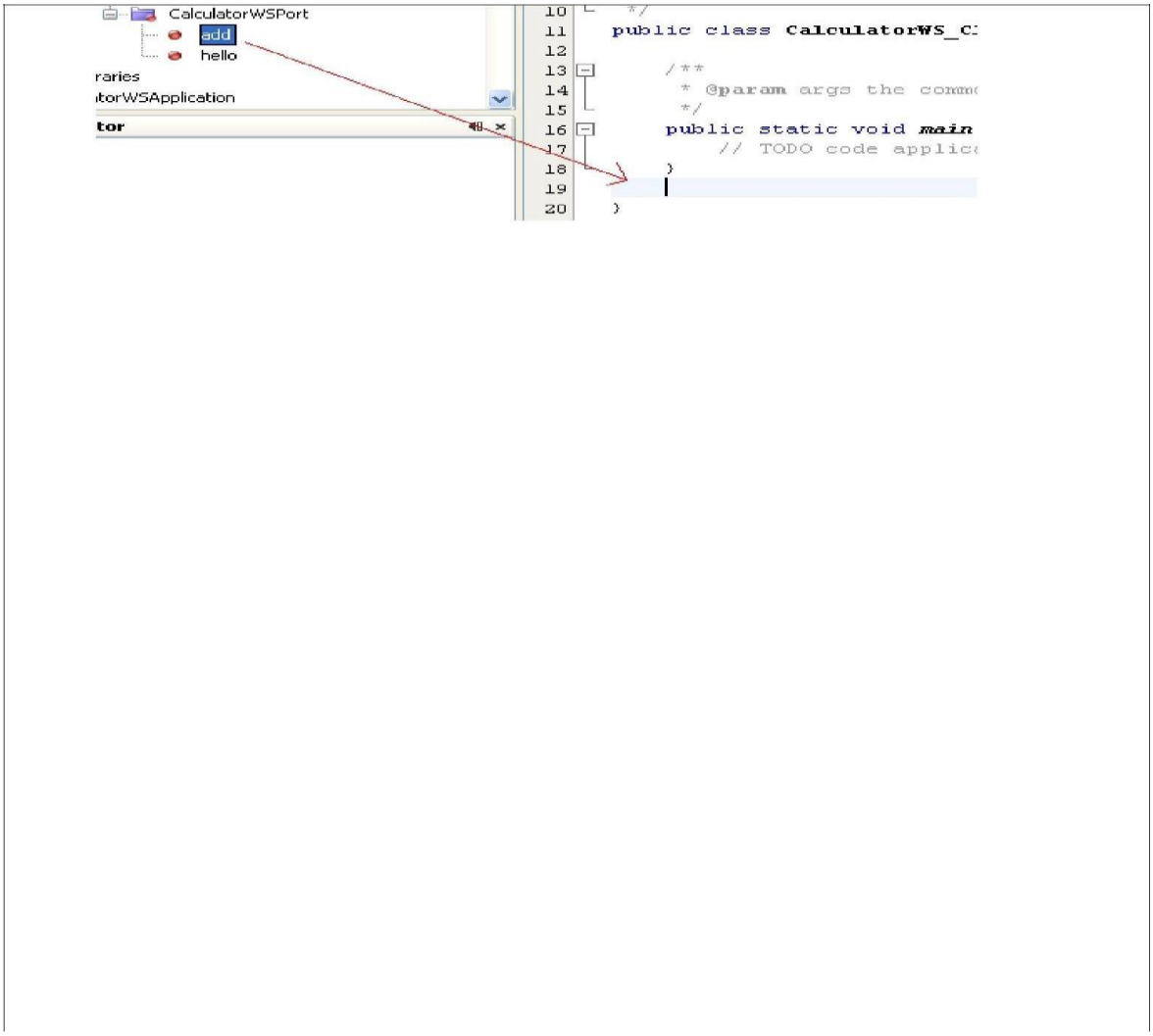
Choose File > New Project (Ctrl-Shift-N on Linux and Windows, ⌘-Shift-N on MacOS).

- Select Java Application from the Java category. Name the project`CalculatorWS_Client_Application`. Leave Create Main Class selected and accept all other default settings. Click Finish.
- Right-click the `CalculatorWS_Client_Application` node and choose New > Web Service Client. The New Web Service Client wizard opens.
- Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.
- Do not select a package name. Leave this field empty.
- Leave the other settings at default and click Finish.

The Projects window displays the new web service client, with a node for the `add` method that you created:

Double-click your main class so that it opens in the Source Editor.

Drag the `add` node below the `main()` method.



Note: Alternatively, instead of dragging the `add` node, you can right-click in the editor and then choose Insert Code > Call Web Service Operation.

- In the `main()` method body, replace the TODO comment with code that initializes values for `i` and `j`, calls `add()`, and prints the result.
- public static void main(String[] args) { int i = 3;

```
    int j = 4;
    int result = add(i, j);
    System.out.println("Result = " + result);
}
```

- Right-click the project node and choose Run.

The Output window now shows the sum:

```
compile:
  Run result =7
  BUILD SUCCESSFUL (total time: 1 second)
```

Conclusion:

Thus we have studied use of webservices using SOAP for a java application.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 9

Aim: Implementation of Para-Virtualization using VM Ware‘s Workstation/ Oracle‘s Virtual Box and Guest O.S.

Aim: Implementation of Virtual Box for Virtualization of any OS.

Theory:

Virtual Box is a cross-platform virtualization application. What does that mean? For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. Secondly, it extends the capabilities of your existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time. So, for example, you can run Windows and Linux on your Mac, run Windows Server 2008 on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like the only practical limits are disk space and memory. Virtual Box is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

The techniques and features that Virtual Box provides are useful for several scenarios:

- **Running multiple operating systems simultaneously.** Virtual Box allows you to run more than one operating system at a time. This way, you can run software written for one operating system on another (for example, Windows software on Linux or a Mac) without having to reboot to use it. Since you can configure what kinds of "virtual" hardware should be presented to each such operating system, you can install an old operating system such as DOS or OS/2 even if your real computer's hardware is no longer supported by that operating system.
- **Easier software installations.** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Virtual Box, such a complex setup (then

often called an "appliance") can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Virtual Box.

- **Testing and disaster recovery.** Once installed, a virtual machine and its virtual hard disks can be considered a "container" that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.
- **Infrastructure consolidation.** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

Some Terminologies used:

When dealing with virtualization (and also for understanding the following chapters of this documentation), it helps to acquaint oneself with a bit of crucial terminology, especially the following terms:

Host operating system (host OS): This is the operating system of the physical computer on which Virtual Box was installed. There are versions of Virtual Box for Windows, Mac OS X, Linux and Solaris hosts.

Guest operating system (guest OS): This is the operating system that is running inside the virtual machine. Theoretically, Virtual Box can run any x86 operating system (DOS, Windows, OS/2, FreeBSD, Open BSD), but to achieve near-native performance of the guest code on your machine, we had to go through a lot of optimizations that are specific to certain operating systems. So while your favorite operating system *may* run as a guest, we officially support and optimize for a select few (which, however, include the most common ones).

Virtual machine (VM): This is the special environment that Virtual Box creates for your guest operating system while it is running. In other words, you run your guest operating system "in" a VM. Normally, a VM will be shown as a window on your computers desktop, but depending on which of the various frontends of VirtualBox you use, it can be displayed in full screen mode or remotely on another computer. In a more abstract way, internally, VirtualBox thinks of a VM as a set of parameters that determine its behavior. They include

hardware settings (how much memory the VM should have, what hard disks VirtualBox should virtualize through which container files, what CDs are mounted etc.) as well as state information (whether the VM is currently running, saved, its snapshots etc.). These settings are mirrored in the VirtualBox Manager window as well as the VBoxManage command line program;

Guest Additions: This refers to special software packages which are shipped with VirtualBox but designed to be installed *inside* a VM to improve performance of the guest OS and to add extra features.

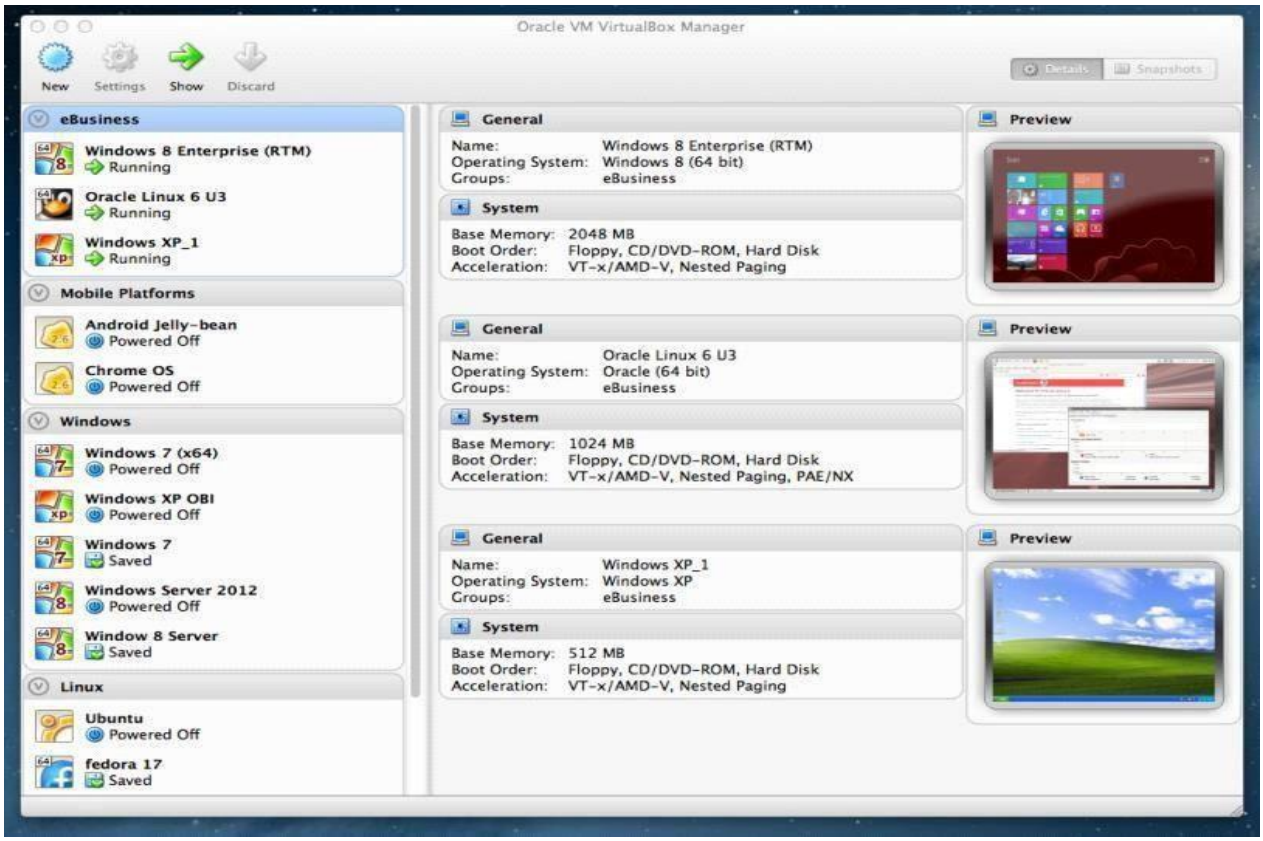
Starting Virtual Box:

After installation, you can start VirtualBox as follows:

- On a Windows host, in the standard "Programs" menu, click on the item in the "VirtualBox" group. On Vista or Windows 7, you can also type "VirtualBox" in the search box of the "Start" menu.
- On a Mac OS X host, in the Finder, double-click on the "VirtualBox" item in the "Applications" folder. (You may want to drag this item onto your Dock.)
- On a Linux or Solaris host, depending on your desktop environment, a "VirtualBox" item may have been placed in either the "System" or "System Tools" group of your "Applications" menu. Alternatively, you can type VirtualBox in a terminal.

When you start VirtualBox for the first time, a window like the following should come up:

This window is called the **"VirtualBox Manager".** On the left, you can see a pane that will later list all your virtual machines. Since you have not created any, the list is empty. A row of buttons above it allows you to create new VMs and work on existing VMs, once you have some. The pane on the right displays the properties of the virtual machine currently selected, if any. Again, since you don't have any machines yet, the pane displays a welcome message. To give you an idea what VirtualBox might look like later, after you have created many machines, here's another example:



Creating your first virtual machine:

Click on the "New" button at the top of the VirtualBox Manager window. A wizard will pop up to guide you through setting up a new virtual machine (VM)

On the following pages, the wizard will ask you for the bare minimum of information that is needed to create a VM, in particular:

- The **VM name** will later be shown in the VM list of the VirtualBox Manager window, and it will be used for the VM's files on disk. Even though any name could be used, keep in mind that once you have created a few VMs, you will appreciate if you have given your VMs rather informative names; "My VM" would thus be less useful than "Windows XP SP2 with OpenOffice".
- For **"Operating System Type",** select the operating system that you want to install later. The supported operating systems are grouped; if you want to install something very unusual that is not listed, select "Other". Depending on your selection, Virtual Box will enable or disable certain VM settings that your guest operating system may require. This is particularly important for 64-bit guests (see Section 3.1.2,64-bit guests). It is therefore recommended to always set it to the correct value.

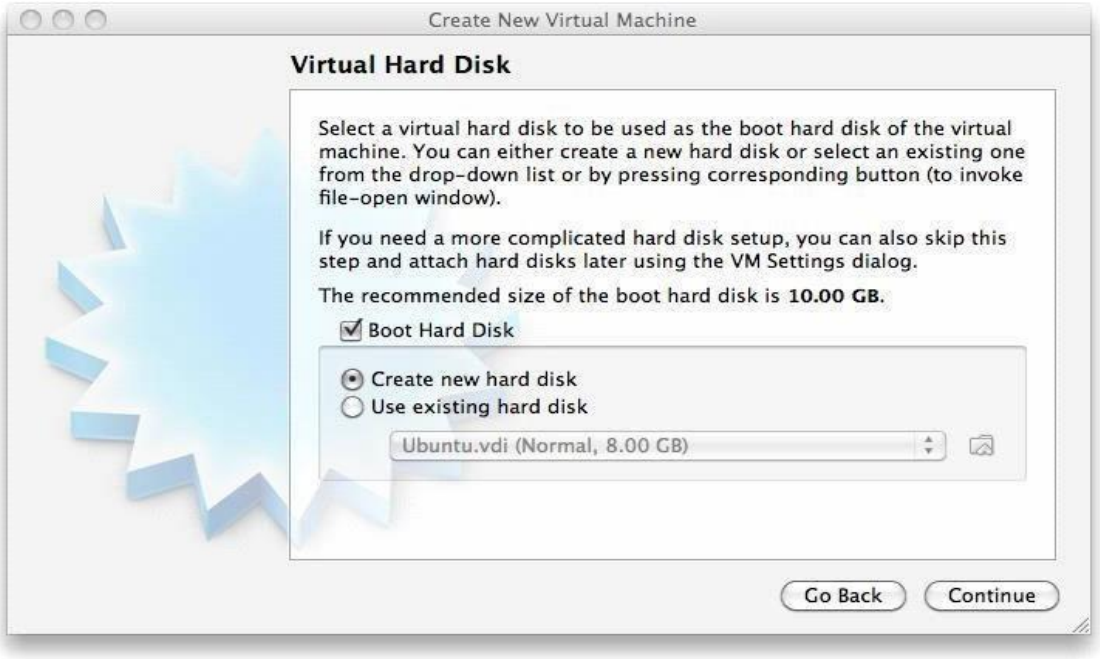
- On the next page, select the **memory (RAM)** that Virtual Box should allocate every time the virtual machine is started. The amount of memory given here will be taken away from your host machine and presented to the guest operating system, which will report this size as the (virtual) computer's installed RAM.

  A Windows XP guest will require at least a few hundred MB RAM to run properly, and Windows Vista will even refuse to install with less than 512 MB. Of course, if you want to run graphics-intensive applications in your VM, you may require even more RAM.

  So, as a rule of thumb, if you have 1 GB of RAM or more in your host computer, it is usually safe to allocate 512 MB to each VM. But, in any case, make sure you always have at least 256 to 512 MB of RAM left on your host operating system. Otherwise you may cause your host OS to excessively swap out memory to your hard disk, effectively bringing your host system to a standstill. As with the other settings, you can change this setting later, after you have created the VM.

4. Next, you must specify a **virtual hard disk** for your VM. There are many and potentially complicated ways in which VirtualBox can provide hard disk space to a VM, but the most common way is to use a large image file on your "real" hard disk, whose contents VirtualBox presents to your VM as if it were a complete hard disk. This file represents an entire hard disk then, so you can even copy it to another host and use it with another VirtualBox installation.

The wizard shows you the following window:



Here you have the following options:

- To create a new, empty virtual hard disk, press the **"New"** button.
- You can pick an **existing** disk image file. The **drop-down list** presented in the window contains all disk images which are currently remembered by VirtualBox, probably because they are currently attached to a virtual machine (or have been in the past). Alternatively, you can click on the small **folder button** next to the drop-down list to bring up a standard file dialog, which allows you to pick any disk image file on your host disk

Most probably, if you are using VirtualBox for the first time, you will want to create a new disk image. Hence, press the "New" button. This brings up another window, the **"Create New Virtual Disk Wizard",** which helps you create a new disk image file in the new virtual machine's folder.

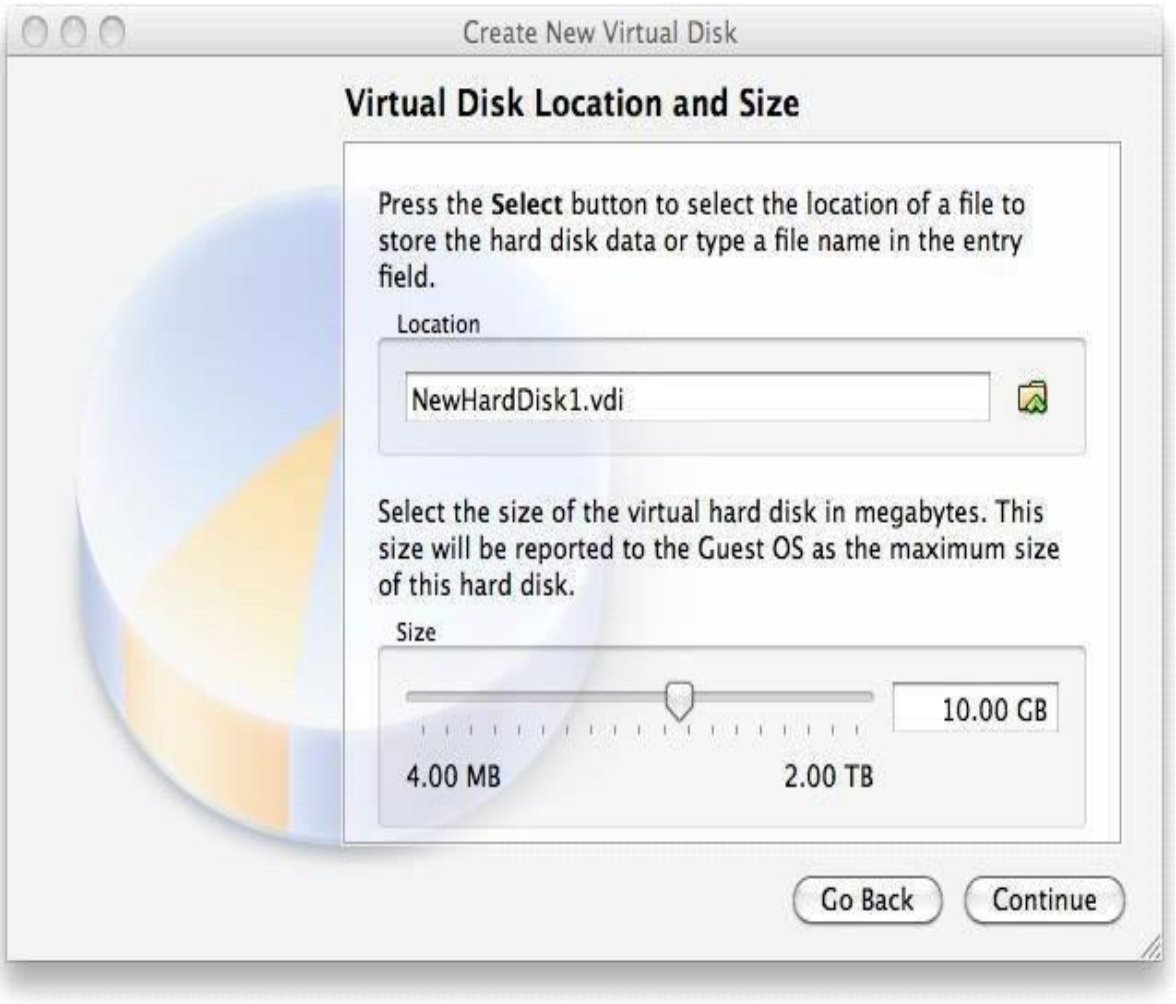
VirtualBox supports two types of image files:

- A **dynamically allocated file** will only grow in size when the guest actually stores data on its virtual hard disk. It will therefore initially be small on the host hard drive and only later grow to the size specified as it is filled with data.
- A **fixed-size file** will immediately occupy the file specified, even if only a fraction

of the virtual hard disk space is actually in use. While occupying much more space, a fixed-size file incurs less overhead and is therefore slightly faster than a dynamically allocated file.

After having selected or created your image file, again press **"Next"** to go to the next page.

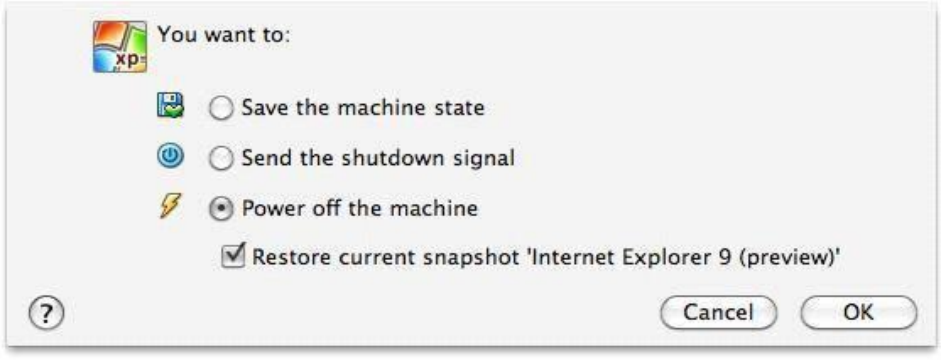
- After clicking on **"Finish"**, your new virtual machine will be created. You will then see it in the list on the left side of the Manager window, with the name you entered initially.



Running your virtual machine: To start a virtual machine, you have several options:

- Double-click on its entry in the list within the Manager window or
- select its entry in the list in the Manager window it and press the "Start" button at the top or
- for virtual machines created with VirtualBox 4.0 or later, navigate to the "VirtualBox VMs" folder in your system user's home directory, find the subdirectory of the machine you want to start and double-click on the machine settings file (with a .vbox file extension). This opens up a new window, and the virtual machine which you selected will boot up. Everything which would normally be seen on the virtual system's monitor is shown in the window. In general, you can use the virtual machine much like you would use a real computer. There are couple of points worth mentioning however.

Saving the state of the machine: When you click on the "Close" button of your virtual machine window (at the top right of the window, just like you would close any other window on your system), VirtualBox asks you whether you want to "save" or "power off" the VM. (As a shortcut, you can also press the Host key together with "Q".)



The difference between these three options is crucial. They mean:

- **Save the machine state:** With this option, VirtualBox "freezes" the virtual machine by completely saving its state to your local disk. When you start the VM again later, you will find that the VM continues exactly where it was left off. All your programs will still be open, and your computer resumes operation. Saving the state of a virtual machine is thus in some ways similar to suspending a laptop computer (e.g. by closing its lid).
- **Send the shutdown signal.** This will send an ACPI shutdown signal to the virtual machine, which has the same effect as if you had pressed the power button on a real computer. So long as the VM is running a fairly modern operating system, this should trigger a proper shutdown mechanism from within the VM.
- **Power off the machine:** With this option, VirtualBox also stops running the virtual machine, but *without* saving its state. As an exception, if your virtual machine has any snapshots (see the next chapter), you can use this option to quickly **restore the current snapshot** of the virtual machine. In that case, powering off the machine will not disrupt its state, but any changes made since that snapshot was taken will be lost. The **"Discard"** button in the VirtualBox Manager window discards a virtual machine's saved state. This has the same effect as powering it off, and the same warnings apply.

Importing and exporting virtual machines

VirtualBox can import and export virtual machines in the industry-standard Open Virtualization Format (OVF). OVF is a cross-platform standard supported by many virtualization products which allows for creating ready-made virtual machines that can then be imported into a virtualizer such as VirtualBox. VirtualBox makes OVF import and export easy to access and supports it from the Manager window as well as its command-line interface. This allows for packaging so-called **virtual appliances**: disk images together with configuration settings that can be distributed easily. This way one can offer complete ready-to-use software packages (operating systems with applications) that need no configuration or installation except for importing into VirtualBox.

Appliances in OVF format can appear in two variants:

- They can come in several files, as one or several disk images, typically in the widely- used VMDK format (see Section 5.2,Disk image files (VDI, VMDK, VHD, HDD)❙❙) and a textual description file in an XML dialect with an .ovf extension. These files must then reside in the same directory for Virtual Box to be able to import them.
- Alternatively, the above files can be packed together into a single archive file, typically with an .ova extension. (Such archive files use a variant of the TAR archive format and can therefore be unpacked outside of Virtual Box with any utility that can unpack standard TAR files.)

Select "File" -> "Export appliance". A different dialog window shows up that allows you to combine several virtual machines into an OVF appliance. Then, select the target location where the target files should be stored, and the conversion process begins. This can again take a while.

Conclusion:

Thus we have studied use of Multiple OS using Virtual Box by virtualizing.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 10

Aim: Installation and Configuration of Hadoop.

Theory:

Hadoop-1.2.1 Installation Steps for Single-Node Cluster (On Ubuntu 12.04)

- Download and install VMware Player depending on your Host OS (32 bit or 64 bit https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_play er/6_0
- Download the .iso image file of Ubuntu 12.04 LTS (32-bit or 64-bit depending on your requirements) http://www.ubuntu.com/download/desktop
- Install Ubuntu from image in VMware. (For efficient use, configure the Virtual Machine to have at least 2GB (4GB preferred) of RAM and at least 2 cores of processor

JAVA INSTALLATION

- sudo mkdir -p /usr/local/java
- cd ~/Downloads
- sudo cp -r jdk-8-linux-i586.tar.gz /usr/local/java
- sudo cp -r jre-8-linux-i586.tar.gz /usr/local/java
- cd /usr/local/java
- sudo tar xvzf jdk-8-linux-i586.tar.gz
- sudo tar xvzf jre-8-linux-i586.tar.gz

- ls a jdk1.8.0 jre1.8.0 jdk-8-linux-i586.tar.gz jre-8-linux-i586.tar.gz

- sudo gedit /etc/profile

- JAVA_HOME=/usr/local/java/jdk1.7.0_4 PATH=$PATH:$HOME/bin:$JAVA_HOME /binJRE_HOME=/usr/local/java/jdk1.7.0_45/jre PATH=$PATH:$HOME/bin:$JRE_HOME/binHADOOP_HOME=/home/hadoop/adoop-1.2.1 PATH=$PATH:$HADOOP_HOME/binexpor t JAVA_HOME export JRE_HOME export PATH
- sudo update-alternatives --install "/usr/bin/java" "java" /usr/local/java/jdk1.8.0/jre/bin/java" 1
- sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/local/java/jdk1.8.0/bin/javac" 1 13.sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/local/java/jdk1.8.0/bin/javaws" 1
- sudo update-alternatives --set java /usr/local/java/jdk1.8.0/jre/bin/java
- sudo update-alternatives --set javac /usr/local/java/jdk1.8.0/bin/javac
- sudo update-alternatives --set javaws /usr/local/java/jdk1.8.0/bin/javaws
- . /etc/profile
- java -version

java version "1.8.0"
Java(TM) SE Runtime Environment (build 1.8.0-b132)
Java HotSpot(TM) Client VM (build 25.0-b70, mixed mode)

HADOOP INSTALLATION

- open Home
- create a floder hadoop
- copy from downloads hadoop-1.2.1.tar.gz to hadoop

- right click on hadoop-1.2.1.tar.gz and Extract Here
- cd hadoop/
- ls -a

. .. hadoop-1.2.1 hadoop-1.2.1.tar.gz 25. edit the file
conf/hadoop-env.sh

The java implementation to use. Required. export JAVA_HOME=/usr/local/java/jdk1.8.0

26. cd hadoop-1.2.1

----------STANDALONE OPERATION----------

- mkdir input
- cp conf/*.xml input
- bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'
- cat output/*

----------PSEUDO DISTRIBUTED OPERATION ---------- //WORDCOUNT

- conf/core-site.xml:

<configuration> <property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>

- conf/hdfs-site.xml:

<configuration> <property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

- conf/mapred-site.xml:

<configuration> <property>

<name>mapred.job.tracker</name>

<value>localhost:9001</value>

</property>

</configuration>

- ssh localhost
- ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
- cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
- bin/hadoop namenode -format
- bin/start-all.sh

Run the following command to verify that hadoop services are running $ jps If everything was successful, you should see following services running

2583 DataNode

2970 ResourceManager

3461 Jps

3177 NodeManager

2361 NameNode

2840 SecondaryNameNode

Conclusion:

Thus we have studied how to install and configure hadoop on Ubuntu operating system.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 11

Aim: Create an application (Ex: Word Count) using Hadoop Map/Reduce.

Theory:

THE MAPREDUCE MODEL

Traditional parallel computing algorithms were developed for systems with a small number of processors, dozens rather than thousands. So it was safe to assume that processors would not fail during a computation. At significantly larger scales this assumption breaks down, as was experienced at Google in the course of having to carry out many large-scale computations similar to the one in our word counting example. The MapReduce parallel programming abstraction was developed in response to these needs, so that it could be used by many different parallel applications while leveraging a common underlying fault-tolerant implementation that was transparent to application developers. Figure 11.1 illustrates MapReduce using the word counting example where we needed to count the occurrences of each word in a collection of documents.

MapReduce proceeds in two phases, a distributed _map‘ operation followed by a distributed _reduce‘ operation; at each phase a configurable number of M _mapper‘ processors and R _reducer‘ processors are assigned to work on the problem (we have used$M$ = 3 and R = 2 in the illustration). The computation is coordinated by a single master process (not shown in the figure).

A MapReduce implementation of the word counting task proceeds as follows: In the map phase each mapper reads approximately 1/M th of the input (in this case documents), from the global file system, using locations given to it by the master. Each mapped then performs a_map‘ operation to compute word frequencies for its subset of documents. These frequencies are *sorted* by the words they represent and written to the *local* file system of the mapper. At the next phase reducers are each assigned a subset of words; in our illustration the first reducer is assigned w1 and w2 while the second one handles w3 and w4. In fact during the map phase itself each mapper writes one file per reducer, based on the words assigned to each reducer, and keeps the master informed of these

file locations. The master in turn informs the reducers where the partial counts for their words have been stored on the local files of respective mappers; the reducers then make remote procedure call requests to the mappers to fetch these. Each reducer performs a reduce‘ operation that sums up the frequencies for each word, which are finally written back to the GFS file system

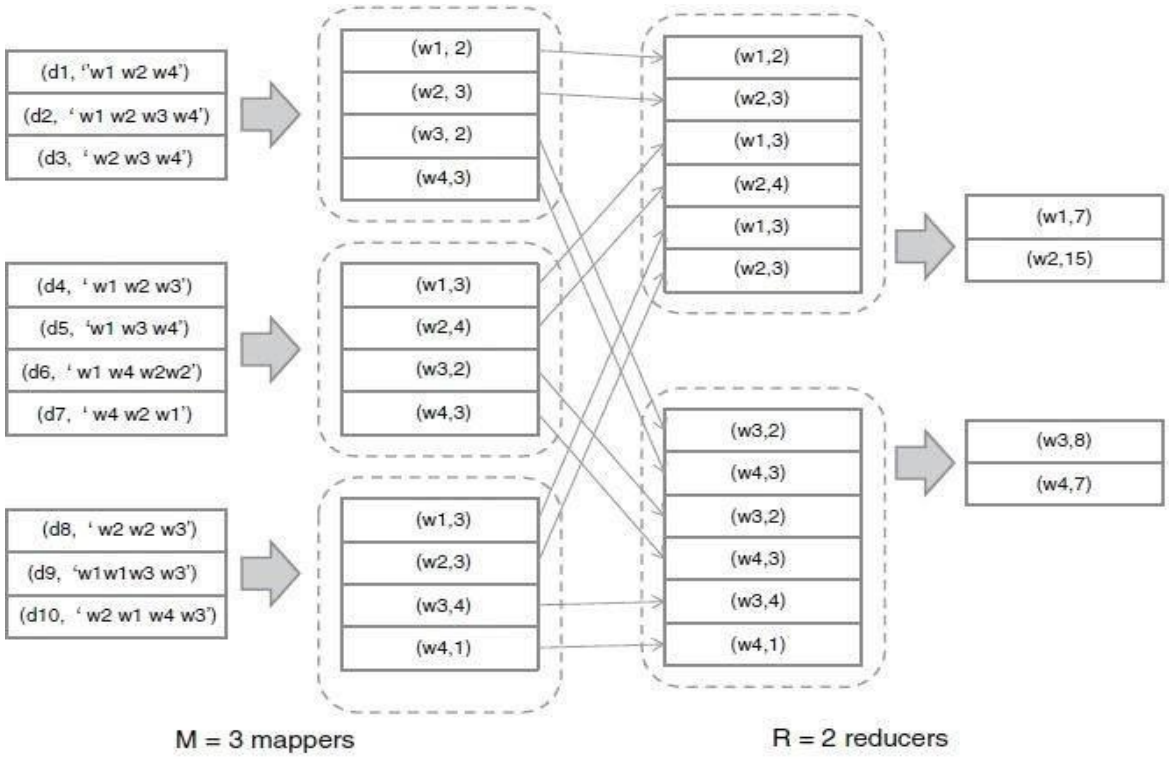


FIGURE 11.1. MapReduce model

The MapReduce programming model generalizes the computational structure of the above example. Each map operation consists of transforming one set of key-value pairs to another:

Map: $(k1, v1) \rightarrow [(k2, v2)]$……

In our example each map operation takes a document indexed by its id and emits a list if word-count pairs indexed by word-id: $(dk, [w1 \ldots\ldots\ldots\ldots wn]) \rightarrow [(wi, ci)]$. The reduce operation groups the results of the map step using the same key $k2$ and performs a function f on the list of values that correspond to each

Reduce: *(k2, [v2]) → (k2, f ([v2]))*…….

In our example each reduce operation sums the frequency counts for each word:

$$(w_i, [c_i]) \rightarrow \left(w_i, \sum_i c_i\right).$$

The implementation also generalizes. Each mapper is assigned an input-key range (set of values for $k1$) on which map operations need to be performed. The mapper writes results of its map operations to its local disk in R partitions, each corresponding to the output-key range (values of $k2$) assigned to a particular reducer, and informs the master of these locations. Next each reducer fetches these pairs from the respective mappers and performs reduce operations for each key $k2$ assigned to it. If a processor fails during the execution, the master detects this through regular heartbeat communications it maintains with each worker, wherein updates are also exchanged regarding the status of tasks assigned to workers.

If a mapper fails, then the master reassigns the key-range designated to it to another working node for re-execution. Note that re-execution is required even if the mapper had completed some of its map operations, because the results were written to local disk rather than the GFS. On the other hand if a reducer fails only its remaining tasks (values $k2$) are reassigned to another node, since the completed tasks would already have been written to the GFS.

Finally, heartbeat failure detection can be fooled by a wounded task that has a heartbeat but is making no progress: Therefore, the master also tracks the overall progress of the computation and if results from the last few processors in either phase are excessively delayed, these tasks are duplicated and assigned to processors who have already completed their work. The master declares the task completed when any one of the duplicate workers complete.

Such a fault-tolerant implementation of the MapReduce model has been implemented and is widely used within Google; more importantly from an enterprise perspective, it is also available as an open source implementation through the Hadoop project along with the HDFS distributed file system.

The MapReduce model is widely applicable to a number of parallel computations, including database-oriented tasks which we cover later. Finally we describe one more example, that of indexing a large collection of documents, or, for that matter any data including database records: The map task consists of emitting a word-document/record id pair for each word: $(dk, [w1 \ldots wn]) \rightarrow [(wi, dk)]$. The reduce step groups the pairs by word and creates an index entry for each word: $[(wi, dk)] \rightarrow (wi, [di1 \ldots dim])$

Indexing large collections is not only important in web search, but also a critical aspect of handling structured data; so it is important to know that it can be executed efficiently in parallel using

MapReduce. Traditional parallel databases focus on rapid query execution against data warehouses that are updated infrequently; as a result these systems often do not parallelize index creation sufficiently well.

Open in any Browser

- Open in any Browser NameNode - http://localhost:50070/
- Open in any Browser JobTracker - http://localhost:50030/
- open hadoop/hadoop-1.2.1 create a document type something in that

document and save it as test.txt

- bin/hadoop fs -ls /

Found 1 items

drwxr-xr-x - vishal supergroup 0 2014-04-15 01:13 /tmp

- bin/hadoop fs -mkdir example
- bin/hadoop fs -ls /user/vishal/

Found 1 items

drwxr-xr-x - vishal supergroup /user/vishal/example

- bin/hadoop fs -copyFromLocal test.txt /user/vishal/example
- bin/hadoop jar hadoop-examples-1.2.1.jar wordcount /user/vishal/example/test.txt /hello

(OR)

- In Eclipse New →Java Project→ Provide Project Name→Next→ Select Libraries→ Add Externals JARs→ Go to Hadoop→ hadoop-1.2.1→ select all jar files →again click on Add→External JARs →go to Hadoop →hadoop-1.2.1 →lib →select all JAR files click on Finish.
- Right Click on Src Folder →Select Class→ Provide a Class name: WCE→ Package name: com.WordCount.Example →Click on Finish.

```
package com.WordCount.Example;


import java.io.IOException;
 import java.util.*;
import
org.apache.hadoop.fs.Path;
import
org.apache.hadoop.conf.*;
import
org.apache.hadoop.io.*;
import
org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WCE
{
public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text,
IntWritable>
{
private final static IntWritable one = new
 IntWritable(1); private Text word = new Text();
public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException
{
 String line = value.toString();
StringTokenizer tokenizer = new
StringTokenizer(line); While
(tokenizer.hasMoreTokens())
{
```

```
word.set(tokenizer.nextToken()); output.collect(word, one);
}
}
}
public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text,
IntWritable> {
public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable>
output, Reporter reporter) throws IOException {
int sum = 0;
while (values.hasNext())
{
sum += values.next().get();
}
output.collect(key, new IntWritable(sum));
}}

public static void main(String[] args) throws Exception
{ JobConf conf = new
JobConf(WCE.class);
conf.setJobName("wordcount");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
 conf.setMapperClass(Map.class);
 conf.setCombinerClass(Reduce.class);
conf.setReducerClass(Reduce.class);
FileInputFormat.addInputPath(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
JobClient.runJob(conf);
```

}}

- Right Click on Project Name→ New→ File→sample→ type something in the sample filr
- Right Click on Project Name→Click on Java→ JAR File→ Provide a JAR File Name →Select The Location where to save the JAR file.
- Right Click on Project Name → Run as Run Configuration → Java Application new In Main → WordCount →Click on Search and click on the JAR File which you have →created → Click on Arguments→ Provide under Program arguments → sample output →Click on Run.
- Right Click on Project Name→ Refresh
  An output file is created in your project.

Conclusion: Hence we have implemented Map Reduce example such as Word Count program on an file which will count the no.of times a word repeats in the given file.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 12

Aim: : Case Study: PAAS (Face book, Google App Engine)

Theory:

Platform-as-a-Service (PaaS):

Cloud computing has evolved to include platforms for building and running custom web-based applications, a concept known as Platform-as-a- Service. PaaS is an outgrowth of the SaaS application delivery model. The PaaS model makes all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet, all with no software downloads or installation for developers, IT managers, or end users. Unlike the IaaS model, where developers may create a specific operating system instance with homegrown applications running, PaaS developers are concerned only with webbased development and generally do not care what operating system is used. PaaS services allow users to focus on innovation rather than complex infrastructure. Organizations can redirect a significant portion of their budgets to creating applications that provide real business value instead of worrying about all the infrastructure issues in a roll-your-own delivery model. The PaaS model is thus driving a new era of mass innovation. Now, developers around the world can access unlimited computing power. Anyone with an Internet connection can build powerful applications and easily deploy them to users globally.

Google App Engine:

Architecture :

The Google App Engine (GAE) is Google`s answer to the ongoing trend of Cloud

Computing offerings within the industry. In the traditional sense, GAE is a web application hosting service, allowing for development and deployment of web-based applications within a pre- defined runtime environment. Unlike other cloud-based hosting offerings such as Amazon Web Services that operate on an IaaS level, the GAE already provides an application infrastructure on the PaaS level. This means that the GAE abstracts from the underlying hardware and operating system layers by providing the hosted application with a set of application-oriented services. While this approach is very convenient for developers of such applications, the rationale behind the GAE is its focus on scalability and usage-based infrastructure as well as payment.

Costs :

Developing and deploying applications for the GAE is generally free of charge but restricted to a certain amount of traffic generated by the deployed application. Once this limit is reached within a certain time period, the application stops working. However, this limit can be waived when switching to a billable quota where the developer can enter a maximum budget that can be spent on an application per day. Depending on the traffic, once the free quota is reached the application will continue to work until the maximum budget for this day is reached. Table 1 summarizes some of the in our opinion most important quotas and corresponding amount per unit that is charged when free resources are depleted and additional, billable quota is desired.

Features :

With a Runtime Environment, the Data store and the App Engine services, the GAE can be divided into three parts.

Runtime Environment

The GAE runtime environment presents itself as the place where the actual application is executed. However, the application is only invoked once an HTTP request is processed to the GAE via a web browser or some other interface, meaning that the application is not constantly running if no invocation or processing has been done. In case of such an HTTP request, the request handler forwards the request and the GAE selects one out of many possible Google servers where the application is then instantly deployed and executed for a certain amount of

time (8). The application may then do some computing and return the result back to the GAE request handler which forwards an HTTP response to the client. It is important to understand that the application runs completely embedded in this described sandbox environment but only as long as requests are still coming in or some processing is done within the application. The reason for this is simple: Applications should only run when they are actually computing, otherwise they would allocate precious computing power and memory without need. This paradigm shows already the GAE's potential in terms of scalability. Being able to run multiple instances of one application independently on different servers guarantees for a decent level of scalability. However, this highly flexible and stateless application execution paradigm has its limitations.

Requests are processed no longer than 30 seconds after which the response has to be returned to the client and the application is removed from the runtime environment again (8). Obviously this method accepts that for deploying and starting an application each time a request is processed, an additional lead time is needed until the application is finally up and running. The GAE tries to encounter this problem by caching the application in the server memory as long as possible, optimizing for several subsequent requests to the same application. The type of runtime environment on the Google servers is dependent on the programming language used. For Java or other languages that have support for Java-based compilers (such as JRuby, Rhino and Groovy) a Java-based Java Virtual Machine (JVM) is provided. Also, GAE fully supports the Google Web Toolkit (GWT), a framework for rich web applications. For Python and related frameworks a Python-based environment is used.

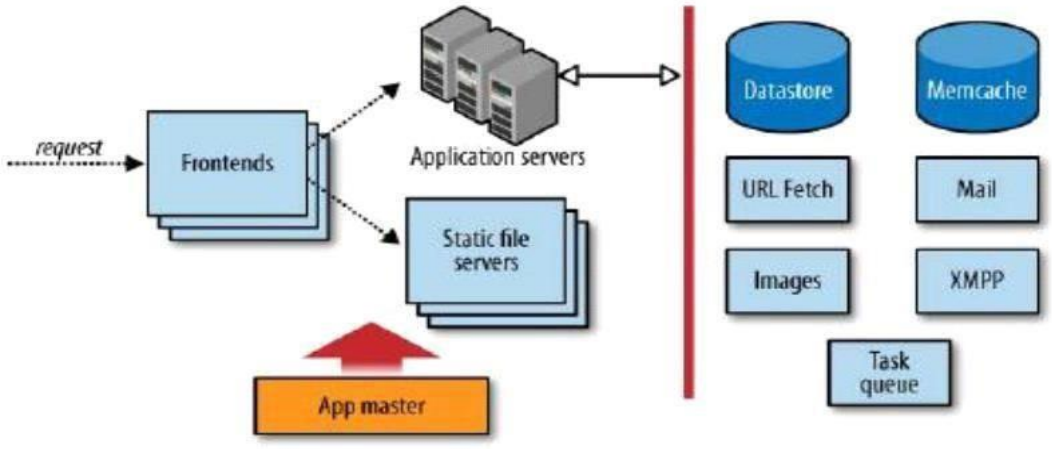


FIGURE 4: STRUCTURE OF GOOGLE APP ENGINE (13)

Persistence and the datastore

As previously discussed, the stateless execution of applications creates the need for a datastore that provides a proper way for persistence. Traditionally, the most popular way of persisting data in web applications has been the use of relational databases. However, setting the focus on high flexibility and scalability, the GAE uses a different approach for data persistence, called *Bigtable* (14). Instead of rows found in a relational database, in Google‘s *Bigtable* data is stored in *entities*. Entities are always associated with a certain *kind.* These entities have *properties*, resembling columns in relational database schemes. But in contrast to relational databases, entities are actually schemaless, as two entities of the same kind not necessarily have to have the same properties or even the same type of value for a certain property.

The most important difference to relational databases is however the querying of entities withina Bigtable datastore. In relational databases queries are processed and executed against a database at application runtime. GAE uses a different approach here. Instead of processing a query at application runtime, queries are pre-processed during compilation time when a corresponding index is created. This index is later used at application runtime when the actual query is executed. Thanks to the index, each query is only a simple table scan where only the exact filter value is searched. This method makes queries very fast compared to relational databases while updating entities is a lot more expensive.

Transactions are similar to those in relational databases. Each transaction is atomic, meaning that it either fully succeeds or fails. As described above, one of the advantages of the GAE is its scalability through concurrent instances of the same application. But what happens when two instances try to start transactions trying to alter the same entity? The answer to this is quite simple: Only the first instance gets access to the entity and keeps it until the transaction is completed or eventually failed. In this case the second instance will receive a concurrency failure exception. The GAE uses a method of handling such parallel transactions called optimistic concurrency control. It simply denies more than one altering transaction on an entity and implicates that an application running within the GAE should have a mechanism trying to get write access to an entity multiple times before finally giving up.
Heavily relying on indexes and optimistic concurrency control, the GAE allows performing queries very fast even at higher scales while assuring data consistency.

Services

As mentioned earlier, the GAE serves as an abstraction of the underlying hardware and operating system layers. These abstractions are implemented as services that can be directly called from the actual application. In fact, the datastore itself is as well a service that is controlled by the runtime environment of the application.

MEM CACHE

The platform innate memory cache service serves as a short-term storage. As its name suggests, it stores data in a server‘s memory allowing for faster access compared to the datastore. Memcache is a non-persistent data store that should only be used to store temporary data within a series of computations. Probably the most common use case for Memcache is to store session specific data (15). Persisting session information in the datastore and executing queries on every page interaction is highly inefficient over the application lifetime, since session-owner instances are unique per session (16). Moreover, Memcache is well suited to speed up common datastore queries (8). To interact with the Memcache

GAE supports JCache, a proposed interface standard for memory caches (17).

URL FETCH

Because the GAE restrictions do not allow opening sockets (18), a URL Fetch service can be used to send HTTP or HTTPS requests to other servers on the Internet. This service works asynchronously, giving the remote server some time to respond while the request handler can do other things in the meantime. After the server has answered, the URL Fetch service returns response code as well as header and body. Using the Google Secure Data Connector an application can even access servers behind a company‘s firewall (8).

MAIL

The GAE also offers a mail service that allows sending and receiving email messages. Mails can be sent out directly from the application either on behalf of the application‘s administrator or on behalf of userswith Google Accounts. Moreover, an application can receive emails in the form of HTTP requests initiated by the App Engine and posted to the app at multiple addresses. In contrast to incoming emails, outgoing messages may also have an attachment up to 1 MB (8).

XMPP

In analogy to the mail service a similar service exists for instant messaging, allowing an application to send and receive instant messages when deployed to the GAE. The service allows communication to and from any instant messaging service compatible to XMPP (8), a set of open technologies for instant messaging and related tasks (19).

IMAGES

Google also integrated a dedicated image manipulation service into the App Engine. Using this service images can be resized, rotated, flipped or cropped (18). Additionally it is able to combine several images into a single one, convert between several image formats and enhance photographs. Of course the API also provides information about format, dimensions and a histogram of color values (8).

USERS

User authentication with GAE comes in two flavors. Developers can roll their own authentication service using custom classes, tables and Memcache or simply plug into Google‘s Accounts service. Since for most applications the time and effort of creating a sign-up page and store user passwords is not worth the trouble (18), the User service is a very convenient functionality which gives an easy method for authenticating users within applications. As byproduct thousands of Google Accounts are leveraged. The User service detects if a user has signed in and otherwise redirect the user to a sign-in page. Furthermore, it can detect whether the current user is an administrator, which facilitates implementing admin-only areas within the application (8).

OAUTH

The general idea behind OAuth is to allow a user to grant a third party limited permission to access protected data without sharing username and password with the third party. The OAuth specification separates between a consumer, which is the application that seeks permission on accessing protected data, and the service provider who is storing protected data on his users' behalf (20). Using Google Accounts and the GAE API, applications can be an OAuth service provider (8).

SCHEDULED TASKS AND TASK QUEUES

Because background processing is restricted on the GAE platform, Google introduced task queues as another built-in functionality (18). When a client requests an application to do certain steps, the application might not be able to process them right away. This is where the task queues come into play. Requests that cannot be executed right away are saved in a task queue that controls the correct sequence of execution. This way, the client gets a response to its request right away, possibly with the indication that the request will be executed later (13). Similar to the concept of task queues are corn jobs. Borrowed from the UNIX world, a GAE cron job is a scheduled job that can invoke a request handler at a pre-specified time (8).

BLOBSTORE

The general idea behind the blobstore is to allow applications to handle objects that are much larger than the size allowed for objects in the datastore service. Blob is short for binary large object and is designed to serve large files, such as video or high quality images. Although blobs can have up

to 2 GB they have to be processed in portions, one MB at a time. This restriction was introduced to smooth the curve of datastore traffic. To enable queries for blobs, each has a corresponding blob info record which is persisted in the datastore (8), e. g. for creating an image database.

ADMINISTRATION CONSOLE

The administration console acts as a management cockpit for GAE applications. It gives the developer real-time data and information about the current performance of the deployed application and is used to upload new versions of the source code. At this juncture it is possible to test new versions of the application and switch the versions presented to the user. Furthermore, access data and logfiles can be viewed. It also enables analysis of traffic so that quota can be adapted when needed. Also the status of scheduled tasks can be checked and the administrator is able to browse the applications datastore and manage indices (8).

App Engine for Business

While the GAE is more targeted towards independent developers in need for a hosting platform for their medium-sized applications, Google`s recently launched App Engine for Business tries to target the corporate market. Although technically mostly relying on the described GAE, Google added some enterprise features and a new pricing scheme to make their cloud computing platform more attractive for enterprise customers (21). Regarding the features, App Engine for Business includes a central development manager that allows a central administration of all applications deployed within one company including access control lists. In addition to that Google now offers a 99.9% service level agreement as well as premium developer support. Google also adjusted the pricing scheme for their corporate customers by offering a fixed price of $8 per user per application, up to a maximum of $1000, per month. Interestingly, unlike the pricing scheme for the GAE, this offer includes unlimited processing power for a fixed price of $8 per user, application and month. From a technical point of view, Google tries to accommodate for established industry standards, by now offering SQL database support in addition to the existing Bigtable datastore described above (8).

APPLICATION DEVELOPMENT USING GOOGLE APP ENGINE

General Idea

In order to evaluate the flexibility and scalability of the GAE we tried to come up with an application that relies heavily on scalability, i.e. collects large amounts of data from external sources. That way we hoped to be able to test both persistency and the gathering of data from external sources at large scale. Therefore our idea has been to develop an application that connects people`s delicious bookmarks with their respective Facebook accounts. People using our application should be able to see what their Facebook friends‘ delicious bookmarks are, provided their Facebook friends have such a delicious account. This way a user can get a visualization of his friends‘ latest topics by looking at a generated tag cloud giving him a clue about the most common and shared interests.

PLATFORM AS A SERVICE: GOOGLE APP ENGINE:--

The Google cloud, called Google App Engine, is a _platform as a service‘ (PaaS) offering. In contrast with the Amazon infrastructure as a service cloud, where users explicitly provision virtual machines and control them fully, including installing, compiling and running software on them, a PaaS offering hides the actual execution environment from users. Instead, a software platform is provided along with an SDK, using which users develop applications and deploy them on the cloud. The PaaS platform is responsible for executing the applications, including servicing external service requests, as well as running scheduled jobs included in the application. By making the actual execution servers transparent to the user, a PaaS platform is able to share *application* servers across users who need lower capacities, as well as automatically scale resources allocated to applications that experience heavy loads. Figure 5.2 depicts a user view of Google App Engine. Users upload code, in either Java or Python, along with related files, which are stored on the Google File System, a very large scale fault tolerant and redundant storage system. It is important to note that an application is immediately available on the internet as soon as it is successfully uploaded (no virtual servers need to be explicitly provisioned as in IaaS).

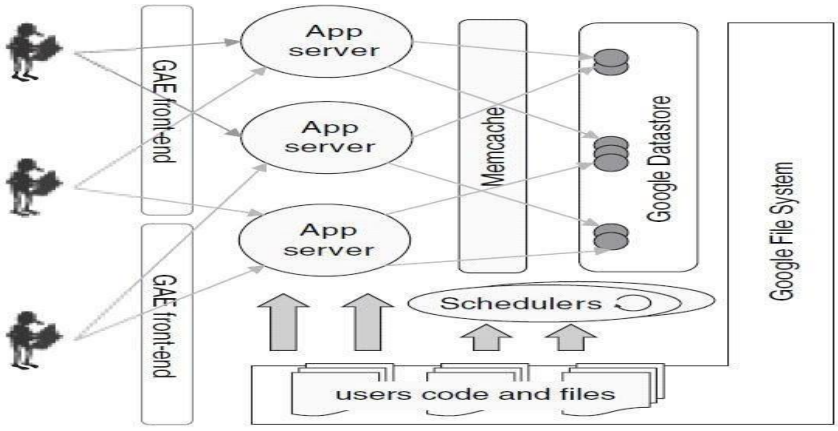


FIGURE 5.2. Google App Engine

Resource usage for an application is metered in terms of web requests served and CPU- hours actually spent executing requests or batch jobs. Note that this is very different from the IaaS model: A PaaS application can be deployed and made globally available 24×7, but charged only when *accessed* (or if batch jobs run); in contrast, in an IaaS model merely making an application continuously available incurs the full cost of keeping at least some of the servers running all the time. Further, deploying applications in Google App Engine is free, within usage limits; thus applications can be developed and tried out free and begin to incur cost only when actually accessed by a sufficient volume of requests. The PaaS model enables Google to provide such a free service because applications do not run in dedicated virtual machines; a deployed application that is not accessed merely consumes storage for its code and data and expends no CPU cycles.

GAE applications are served by a large number of web servers in Google‘s data centers that execute requests from end-users across the globe. The web servers load code from the GFS into memory and serve these requests. Each request to a particular application is served by any one of GAE‘s web servers; there is no guarantee that the same server will serve requests to any two requests, even from the same HTTP session. Applications can also specify some functions to be executed as batch jobs which are run by a scheduler.

Google Datastore:--

Applications persist data in the Google Datastore, which is also (like Amazon SimpleDB) a non- relational database. The Datastore allows applications to define structured types (called _kinds‘) and store their instances (called _entities‘) in a distributed manner on the GFS file system. While one can view Datastore _kinds‘ as table structures and entities as records, there are important differences between a relational model and the Datastore, some of which are also illustrated in Figure 5.3.

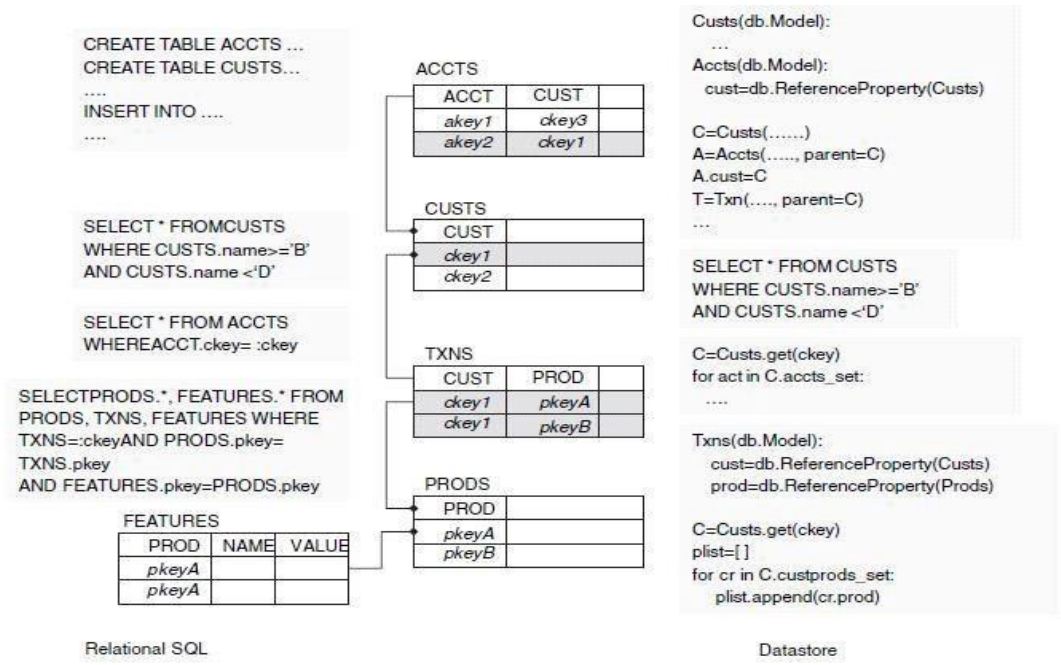


FIGURE 5.3. Google Datastore

Unlike a relational schema where all rows in a table have the same set of columns, all entities of a _kind‘ need not have the same properties. Instead, additional properties can be added to any entity. This feature is particularly useful in situations where one cannot foresee all the potential properties in a model, especially those that occur occasionally for only a small subset of records. For example, a model storing _products‘ of different types (shows, books, etc.) would need to allow each product to have a different set of features. In a relational model, this would probably be implemented using a separate FEATURES table, as shown on the bottom left of Figure 5.3. Using the Datastore, this table (_kind‘) is not required; instead, each product entity can be assigned a different set of properties at runtime. The Datastore allows simple queries with conditions, such as the first query shown in Figure 5.3 to retrieve all customers having names in some lexicographic range. The query syntax (called GQL) is essentially the same as SQL, but with some restrictions. For example, all inequality conditions in a querymust be on a single property; so a query that also filtered customers on, say, their _type‘, would be illegal in GQL but allowed in SQL.

Relationships between tables in a relational model are modeled using foreign keys. Thus, each account in the ACCTS table has a pointer *ckey* to the customer in the CUSTS table that it belongs to. Relationships are traversed via queries using foreign keys, such as retrieving all accounts for a particular customer, as shown. The Datastore provides a more object-oriented approach to relationships in persistent data. Model definitions can include references to other models; thus each entity of the Accts kind‘ includes a reference to its customer, which is an entity of the Custs _kind.‘ Further, relationships defined by such references can be traversed in *both* directions, so not only can one directly access the customer of an account, but also *all* accounts of a given customer, without executing any query operation, as shown in the figure.

GQL queries *cannot* execute joins between models. Joins are critical when using SQL to efficiently retrieve data from multiple tables. For example, the query shown in the figure retrieves details of all products bought by a particular customer, for which it needs to join data from the transactions (TXNS), products (PRODS) and product features (FEATURES) tables. Even though GQL does not allow joins, its ability to traverse associations between entities often enables joins to be avoided, as shown in the figure for the above example: By storing references to customers and products in the Txns model, it is possible to retrieve all transactions for a given customer through a reverse traversal of the customer reference. The product references in each transaction then yield all products and their features (as discussed earlier, a separate Features model is not required because of schema

flexibility). It is important to note that while object relationship traversal can be used as an alternative to joins, this is not always possible, and when required joins may need to be explicitly executed by application code.

The Google Datastore is a distributed object store where objects (entities) of all GAE applications are maintained using a large number of servers and the GFS distributed file system. From a user perspective, it is important to ensure that in spite of sharing a distributed storage scheme with many other users, application data is (a) retrieved efficiently and (b) atomically updated. The Datastore provides a mechanism to group entities from different _kinds‘ in a hierarchy that is used for both these purposes. Notice that in Figure 5.3entities of the Accts and Txns _kinds‘ are instantiated with a parameter _parent‘ that specifies a particular customer entity, thereby linking these three entities in an _entity group‘. The Datastore ensures that all entities belonging to a particular group are stored close together in the distributed file system (we shall see how in Chapter 10). The Datastore allows processing steps to be grouped into transactions wherein updates to data are guaranteed to be atomic; however this also requires that each transaction only manipulates entities belonging to the same entity group. While this transaction model suffices for most on line applications, complex batch updates that update many unrelated entities cannot execute atomically, unlike in a relational database where there are no such restrictions.

Amazon SimpleDB:--

Amazon SimpleDB is also a nonrelational database, in many ways similar to the Google Datastore.

SimpleDB_domains‘ correspond to _kinds‘, and _items‘ to entities; each item can have a number of attribute-value pairs, and different items in a domain can have different sets of attributes, similar to Datastore entities. Queries on SimpleDB domains can include conditions, including inequality conditions, on any number of attributes. Further, just as in the Google Datastore, joins are not permitted. However, SimpleDB does not support object relationships as in Google Datastore, nor does it support transactions. It is important to note that all data in SimpleDB is replicated for redundancy, just as in GFS. Because of replication, SimpleDB

features an _eventual consistency‘ model, wherein data is guaranteed to be propagated to at least one replica and will eventually reach all replicas, albeit with some delay. This can result in perceived inconsistency, since an immediate read following a write may not always yield the result written. In the case of Google Datastore on the other hand, writes succeed only when all replicas are updated; this avoids inconsistency but also makes writes slower.

PAAS CASE STUDY: FACEBOOK

Facebook provides some PaaS capabilities to application developers:--

- Web services remote APIs that allow access to social network properties, data,Like button, etc.
- Many third-parties run their apps off Amazon EC2, and interface to Facebook via its APIs PaaS
- IaaS
- Facebook itself makes heavy use of PaaS services for their own private cloud
- Key problems: how to analyze logs, make suggestions, determine which ads to place.

Facebook API: Overview:--

What you can do:

- Read data from profiles and pages
- Navigate the graph (e.g., via friends lists)
- Issue queries (for posts, people, pages, ...)

Facebook API: The Graph API :

```
{
"id": "1074724712",
"age_range": {
```

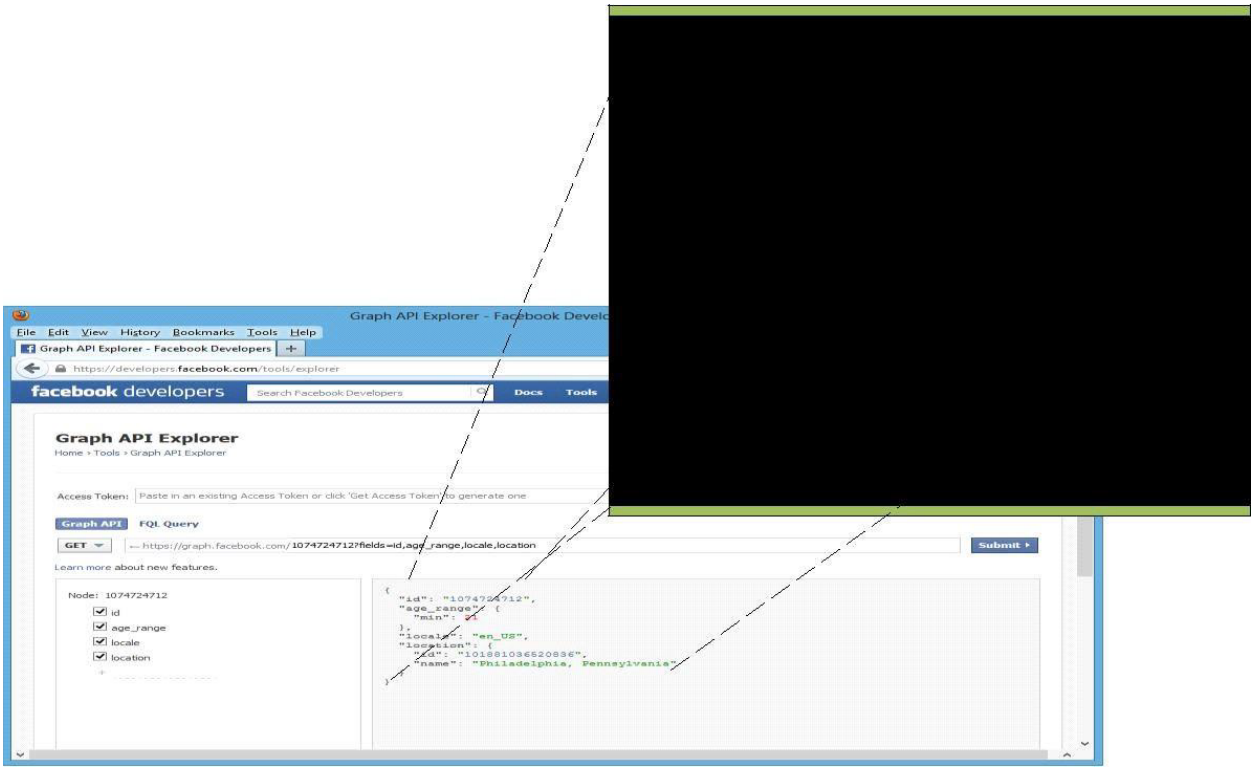
```
"min": 21

},

"locale":
"en_US",
"location": {

"id": "101881036520836",

"name": "Philadelphia, Pennsylvania"
}
 }
```

- Requests are mapped directly to HTTP:
    - https://graph.facebook.com/(identifier)?fields=(fieldList)
- Response is in JSON

Uses several HTTP methods:

- GET for reading
- POST for adding or modifying
- DELETE for removing
- IDs can be numeric or names
- /1074724712 or /andreas.haeberlen
- Pages also have IDs
- Authorization is via 'access tokens'
- Opaque string; encodes specific permissions (access user location, but not interests, etc.)
- Has an expiration date, so may need to be refreshed

Facebook Data Management / Warehousing Tasks

Main tasks for "cloud" infrastructure:

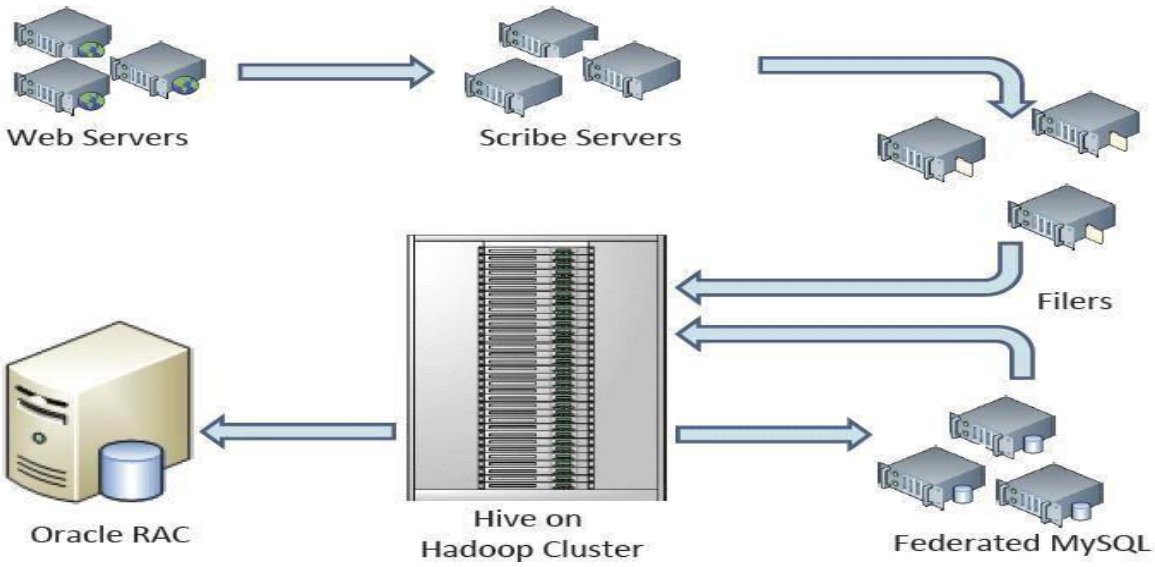
- Summarization (daily, hourly)
  - to help guide development on different components
  - to report on ad performance
  - recommendations

Ad hoc analysis:

- Answer questions on historical data – to help with managerial decisions
- Archival of logs
- Spam detection
- Ad optimization

- Initially used Oracle DBMS for this
  - But eventually hit scalability, cost, performance bottlenecks
    just like Salesforce does now

Data Warehousing at Facebook:



PAAS AT FACEBOOK:

- Scribe – open source logging, actually records the data that will be analyzed by Hadoop
- Hadoop (MapReduce – discussed next time) as batch processing engine for data analysis
    - As of 2009: 2nd largest Hadoop cluster in the world, 2400 cores, > 2PB data with
      > 10TB added every day
- Hive – SQL over Hadoop, used to write the data analysis queries
- Federated MySQL, Oracle – multi-machine DBMSs to store query results

Example Use Case 1: Ad Details

- Advertisers need to see how their ads are performing
    - Cost-per-click (CPC), cost-per-1000-impressions (CPM)
    - Social ads – include info from friends
    - Engagement ads – interactive with video
- Performance numbers given:
    - Number unique users, clicks, video views, …
- Main axes:

    - Account, campaign, ad
    - Time period
    - Type of interaction
    - Users
- Summaries are computed using Hadoop via Hive

Use Case 2: Ad Hoc analysis, feedback

- Engineers, product managers may need to understand what is going on
    - e.g., impact of a new change on some sub-population
- Again, Hive-based, i.e., queries are in SQL with database joins
    - Combine data from several tables, e.g., click-through rate = views combined with clicks
- Sometimes requires custom analysis code with sampling

CONCLUSION :

Cloud Computing remains the number one hype topic within the IT industry at present. Our evaluation of the Google App Engine and facebook has shown both functionality and limitations of the platform. Developing and deploying an application within the GAE is in fact quite easy and in a way shows the progress that software development and deployment has made. Within our application we were able to use the abstractions provided by the GAE without problems, although the concept of Bigtable requires a big change in mindset when developing. Our scalability testing showed the limitations of the GAE at this point in time. Although being an extremely helpful feature and a great USP for the GAE, the built-in scalability of the GAE suffers from both purposely-set as well as technical restrictions at the moment. Coming back to our motivation of evaluating the GAE in terms of its sufficiency for serious large-scale applications in a professional environment, we have to conclude that the GAE not (yet) fulfills business needs for enterprise applications at present.

*****

Class: MCA

Subject: Lab I- High Performance Computing Paradigms and Applications

Experiment No. 13

Aim: AWS Case Study: Amazon.com.

Theory: About AWS

- Launched in 2006, Amazon Web Services (AWS) began exposing key infrastructure services to businesses in the form of web services -- now widely known as cloud computing.
- The ultimate benefit of cloud computing, and AWS, is the ability to leverage a new business model and turn capital infrastructure expenses into variable costs
- Businesses no longer need to plan and procure servers and other IT resources weeks or months inadvance.
- Using AWS, businesses can take advantage of Amazon's expertise and economies of scale to access resources when their business needs them, delivering results faster and at a lower cost.
- Today, Amazon Web Services provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in 190 countries around the world.

  Amazon.com is the world‘s largest online retailer. In 2011, Amazon.com switched from tape backup to using Amazon Simple Storage Service (Amazon S3) for backing up the majority of its Oracle databases. This strategy reduces complexity and capital expenditures, provides faster backup and restore performance, eliminates tape capacity planning for backup and archive, and frees up administrative staff for higher value operations. The company was able to replace their backup tape infrastructure with cloud-based Amazon S3 storage, eliminate backup software, and experienced a 12X performance improvement, reducing restore time from around 15 hours to 2.5 hours in select scenarios

With data center locations in the U.S., Europe, Singapore, and Japan, customers across all industries

are taking advantage of the following benefits:

- **Low Cos**
- **Agility and Instant Elasticity**
- **Open and Flexible**
- **Secure**

The Challenge

As Amazon.com grows larger, the sizes of their Oracle databases continue to grow, and so does the sheer number of databases they maintain. This has caused growing pains related to backing up legacy Oracle databases to tape and led to the consideration of alternate strategies including the use of Cloud services of Amazon Web Services (AWS), a subsidiary of Amazon.com. Some of the business challenges Amazon.com faced included:

- Utilization and capacity planning is complex, and time and capital expense budget are at a premium. Significant capital expenditures were required over the years for tape hardware, data center space for this hardware, and enterprise licensing fees for tape software. During that time, managing tape infrastructure required highly skilled staff to spend time with setup, certification and engineering archive planning instead of on higher value projects. And at the end of every fiscal year, projecting future capacity requirements required time consuming audits, forecasting, and budgeting.
- The cost of backup software required to support multiple tape devices sneaks up on you. Tape robots provide basic read/write capability, but in order to fully utilize them, you must invest in proprietary tape backup software. For Amazon.com, the cost of the software had been high, and added significantly to overall backup costs. The cost of this software was an ongoing budgeting pain point, but one that was difficult to address as long as backups needed to be written to tape devices

Maintaining reliable backups and being fast and efficient when retrieving data requires a lot of time and effort with tape. When data needs to be durably stored on tape, multiple copies are required. When everything is working correctly, and there is minimal contention for tape resources, the tape robots and backup software can easily find the required data. However, if there is a hardware failure, human intervention is necessary to restore from tape. Contention for tape drives resulting from multiple users' tape requests slows down restore processes even more. This adds to the recovery time objective (RTO) and makes achieving it more challenging compared to backing up to Cloud storage.

Why Amazon Web Services?

Amazon.com initiated the evaluation of Amazon S3 for economic and performance improvements related to data backup. As part of that evaluation, they considered security, availability, and performance aspects of Amazon S3 backups. Amazon.com also executed a cost-benefit analysis to ensure that a migration to Amazon S3 would be financially worthwhile. That cost benefit analysis included the following elements:

- Performance advantage and cost competitiveness. It was important that the overall costs of the backups did not increase. At the same time, Amazon.com required faster backup and recovery performance. The time and effort required for backup and for recovery operations proved to be a significant improvement over tape, with restoring from Amazon S3 running from two to twelve times faster than a similar restore from tape. Amazon.com required any new backup medium to provide improved performance while maintaining or reducing overall costs. Backing up to on-premises disk based storage would have improved performance, but missed on cost competitiveness. Amazon S3 Cloud based storage met both criteria.
- Greater durability and availability. Amazon S3 is designed to provide 99.999999999% durability and 99.99% availability of objects over a given year. Amazon.com compared these figures with those observed from their tape infrastructure, and determined that Amazon S3 offered significant improvement.

- Less operational friction. Amazon.com DBAs had to evaluate whether Amazon S3 backups would be viable for their database backups. They determined that using Amazon S3 for backups was easy to implement because it worked seamlessly with Oracle RMAN.
- Strong data security. Amazon.com found that AWS met all of their requirements for physical security, security accreditations, and security processes, protecting data in flight, data at rest, and utilizing suitable encryption standards.

The Benefits

With the migration to Amazon S3 well along the way to completion, Amazon.com has realized several benefits, including:

- Elimination of complex and time-consuming tape capacity planning. Amazon.com is growing larger and more dynamic each year, both organically and as a result of acquisitions. AWS has enabled Amazon.com to keep pace with this rapid expansion, and to do so seamlessly. Historically, Amazon.com business groups have had to write annual backup plans, quantifying the amount of tape storage that they plan to use for the year and the frequency with which they will use the tape resources. These plans are then used to charge each organization for their tape usage, spreading the cost among many teams. With Amazon S3, teams simply pay for what they use, and are billed for their usage as they go. There are virtually no upper limits as to how much data can be stored in Amazon S3, and so there are no worries about running out of resources. For teams adopting Amazon S3 backups, the need for formal planning has been all but eliminated.
- Reduced capital expenditures. Amazon.com no longer needs to acquire tape robots, tape drives, tape inventory, data center space, networking gear, enterprise backup software, or predict future tape consumption. This eliminates the burden of budgeting for capital equipment well in advance as well as the capital expense.
- Immediate availability of data for restoring – no need to locate or retrieve physical tapes. Whenever a DBA needs to restore data from tape, they face delays. The tape backup software needs to read the tape catalog to find the correct files to restore, locate the correct tape, mount the tape, and read the data from it. In almost all cases the data is spread across multiple tapes, resulting in further delays. This, combined with contention for tape drives resulting from multiple users‘ tape requests, slows the process down even more. This is especially severe during critical events such as a data center outage, when many databases

must be restored simultaneously and as soon as possible. None of these problems occur with Amazon S3. Data restores can begin immediately, with no waiting or tape queuing – and that means the database can be recovered much faster.

- Backing up a database to Amazon S3 can be two to twelve times faster than with tape drives. As one example, in a benchmark test a DBA was able to restore 3.8 terabytes in 2.5 hours over gigabit Ethernet. This amounts to 25 gigabytes per minute, or 422MB per second. In addition, since Amazon.com uses RMAN data compression, the effective restore rate was 3.37 gigabytes per second. This 2.5 hours compares to, conservatively, 10-15 hours that would be required to restore from tape.
- Easy implementation of Oracle RMAN backups to Amazon S3. The DBAs found it easy to start backing up their databases to Amazon S3. Directing Oracle RMAN backups to Amazon S3 requires only a configuration of the Oracle Secure Backup Cloud (SBC) module. The effort required to configure the Oracle SBC module amounted to an hour or less per database. After this one- time setup, the database backups were transparently redirected to Amazon S3.
- Durable data storage provided by Amazon S3, which is designed for 11 nines durability. On occasion, Amazon.com has experienced hardware failures with tape infrastructure – tapes that break, tape drives that fail, and robotic components that fail. Sometimes this happens when a DBA is trying to restore a database, and dramatically increases the mean time to recover (MTTR). With the durability and availability of Amazon S3, these issues are no longer a concern.
- Freeing up valuable human resources. With tape infrastructure, Amazon.com had to seek out engineers who were experienced with very large tape backup installations – a specialized, vendor-specific skill set that is difficult to find. They also needed to hire data center technicians and dedicate them to problem-solving and troubleshooting hardware issues – replacing drives, shuffling tapes around, shipping and tracking tapes, and so on. Amazon S3 allowed them to free up these specialists from day-to-day operations so that they can work on more valuable, business-critical engineering tasks.
- Elimination of physical tape transport to off-site location. Any company that has been storing Oracle backup data offsite should take a hard look at the costs involved in transporting, securing and storing their tapes offsite – these costs can be reduced or possibly eliminated by storing the data in Amazon S3.

As the world‘s largest online retailer, Amazon.com continuously innovates in order to provide improved customer experience and offer products at the lowest possible prices. One such innovation has been to replace tape with Amazon S3 storage for database backups. This innovation is one that can be easily replicated by other organizations that back up their Oracle databases to tape.

Products & Services

- Compute
- Content Delivery
- Database
- Deployment & Management
- E-Commerce
- Messaging
- Monitoring
- Networking
- Payments & Billing
- Storage
- Support
- Web Traffic
- Workforce

Products & Services

Compute

- **Amazon Elastic Compute Cloud** (EC2)

- Amazon Elastic Compute Cloud delivers scalable, pay-as-you-go compute capacity in the cloud.

- **Amazon Elastic MapReduce**

Amazon Elastic MapReduce is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

- **Auto Scaling**

Auto Scaling allows to automatically scale our Amazon EC2capacity up or down according to conditions we define.

Content Delivery

- **Amazon CloudFront**

Amazon CloudFront is a web service that makes it easy to distribute content with low latency via a global network of edge locations.

Database

- **Amazon SimpleDB**

Amazon SimpleDB works in conjunction with Amazon S3 and AmazonEC2 to run queries on structured data in real time.

- **Amazon Relational Database Service (RDS)**

Amazon Relational Database Service is a web service that makes it easy to set up, operate, and scale a relational database in the cloud.

- **Amazon ElastiCache**

Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in- memory cache in the cloud.

E-Commerce

- ›**Amazon Fulfillment Web Service** (FWS)

Amazon Fulfillment Web Service allows merchants to deliver products using Amazon.com‘s worldwide fulfillment capabilities.

Deployment & Management

- ✓ **:AWS Elastic Beanstalk**

AWS Elastic Beanstalk is an even easier way to quickly deploy and manage applications in the AWS cloud. We simply upload our application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

- ›**AWS CloudFormation**

AWS CloudFormation is a service that gives developers and businesses an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion.

Monitoring

- ›**Amazon CloudWatch**

Amazon CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2

Messaging

- ›**Amazon Simple Queue Service** (SQS)

Amazon Simple Queue Service provides a hosted queue for storing messages as they travel between computers, making it easy to build automated workflow between Web services.

- ›**Amazon Simple Notification Service (SNS)**

Amazon Simple Notification Service is a web service that makes it easy to set up, operate, and send notifications from the cloud.

- ›**Amazon Simple Email Service (SES)**

Amazon Simple Email Service is a highly scalable and cost-effective bulk and transactional email-sending service for the cloud.

Workforce

- ›**Amazon Mechanical Turk**

Amazon Mechanical Turk enables companies to access thousands of global workers on demand and programmatically integrate their work into various business processes.

Networking

- ›**Amazon Route 53**

- Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service.

- ›**Amazon Virtual Private Cloud** (VPC)

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a private, isolated section of the Amazon Web Services (AWS) Cloud where we can launch AWS resources in a virtual network that you define. With Amazon VPC, we can define a virtual network topology that closely resembles a traditional network that you might operate in your own datacenter.

- ›**AWS Direct Connect**

AWS Direct Connect makes it easy to establish a dedicated network connection from your premise to AWS, which in many cases can reduce our network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

- **Elastic Load Balancing**

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances.

Payments & Billing

- ›**Amazon Flexible Payments Service** (FPS)

Amazon Flexible Payments Service facilitates the digital transfer of money between any two entities, humans or computers.

- ›**Amazon DevPay**

Amazon DevPay is a billing and account management service which enables developers to collect payment for their AWS applications.

- **Storage**

›Amazon Simple Storage Service (S3)

Amazon Simple Storage Service provides a fully redundant data storage infrastructure for storing and retrieving any amount of data, at any time, from anywhere on the Web.

›Amazon Elastic Block Store (EBS)

Amazon Elastic Block Store provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are off-instance storage that persists independently from the life of an instance.

›AWS Import/Export

AWS Import/Export accelerates moving large amounts of data into and out of AWS using portable storage devices for transport.

Support

›AWS Premium Support AWS Premium Support is a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.

Web Traffic

›Alexa Web Information Service

Alexa Web Information Service makes Alexa‘s huge repository of data about structure and traffic patterns on the Web available to developers.

›Alexa Top Sites

Alexa Top Sites exposes global website traffic data as it is continuously collected and updated by Alexa Traffic Rank.

Amazon CloudFront

- Amazon CloudFront is a web service for content delivery.
- It integrates with other Amazon Web Services to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no commitments.
- Amazon CloudFront delivers our static and streaming content using a global network of edge locations edge locations.
- Requests for our objects are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

Amazon CloudFront

- Amazon CloudFront is optimized to work with other Amazon Web Services, like Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2).
- Amazon CloudFront also works seamlessly with any origin server, which stores the original, definitive versions of our files.

- Like other Amazon Web Services, there are no contracts or monthly commitments for using Amazon CloudFront _ we pay only for as much or as little content as you actually deliver through the service.

Amazon Simple Queue Service (Amazon SQS)

- Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable, hosted queue for storing messages as they travel between computers.
- By using Amazon SQS, developers can simply move data between distributed components of their applications that perform different tasks, without losing messages or requiring each component to be always available.

  Amazon SQS makes it easy to build an automated workflow, working in close conjunction with the Amazon Elastic Compute Cloud (Amazon EC2) and the other AWS infrastructure web services.

Amazon Simple Queue Service (Amazon SQS)

- Amazon SQS works by exposing Amazon_s web-scale messaging infrastructure as a web service.
- Any computer on the Internet can add or read messages without any installed software or special firewall configurations.
- Components of applications using Amazon SQS can run independently, and do not need

  to be on the same network, developed with the same technologies, or running at the same time

BigTable

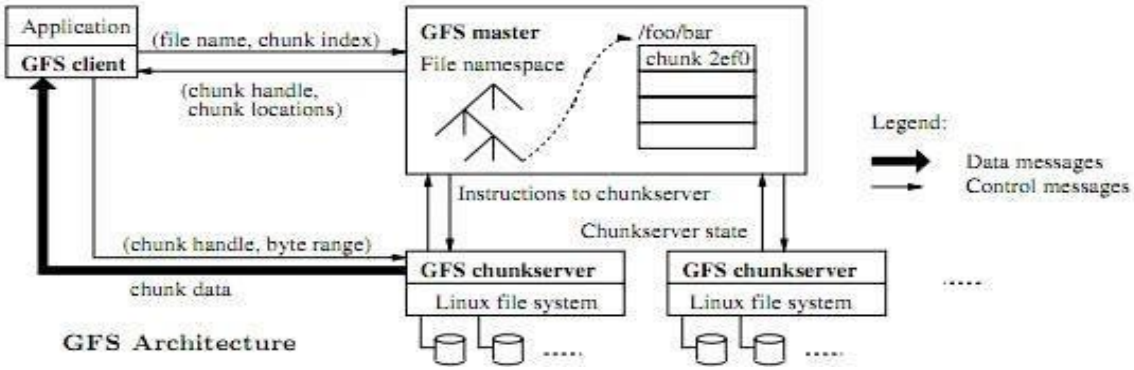
- Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size: petabytes of data across thousands of commodity servers.
- Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance.
- These applications place very different demands on Bigtable, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to real-time data serving).
- Despite these varied demands, Bigtable has successfully provided a fexible, high-performance solution for all of these Google products.

The Google File System(GFS)

- The Google File System (GFS) is designed to meet the rapidly growing demands of Google_s data processing needs.
- GFS shares many of the same goals as previous distributed file systems such as performance, scalability, reliability, and availability.
- It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.
- While sharing many of the same goals as previous distributed file systems, file system has successfully met our storage needs.
- It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets.

-

The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients.



Conclusion:

Thus we have studied a case study on amazon web services.

*****