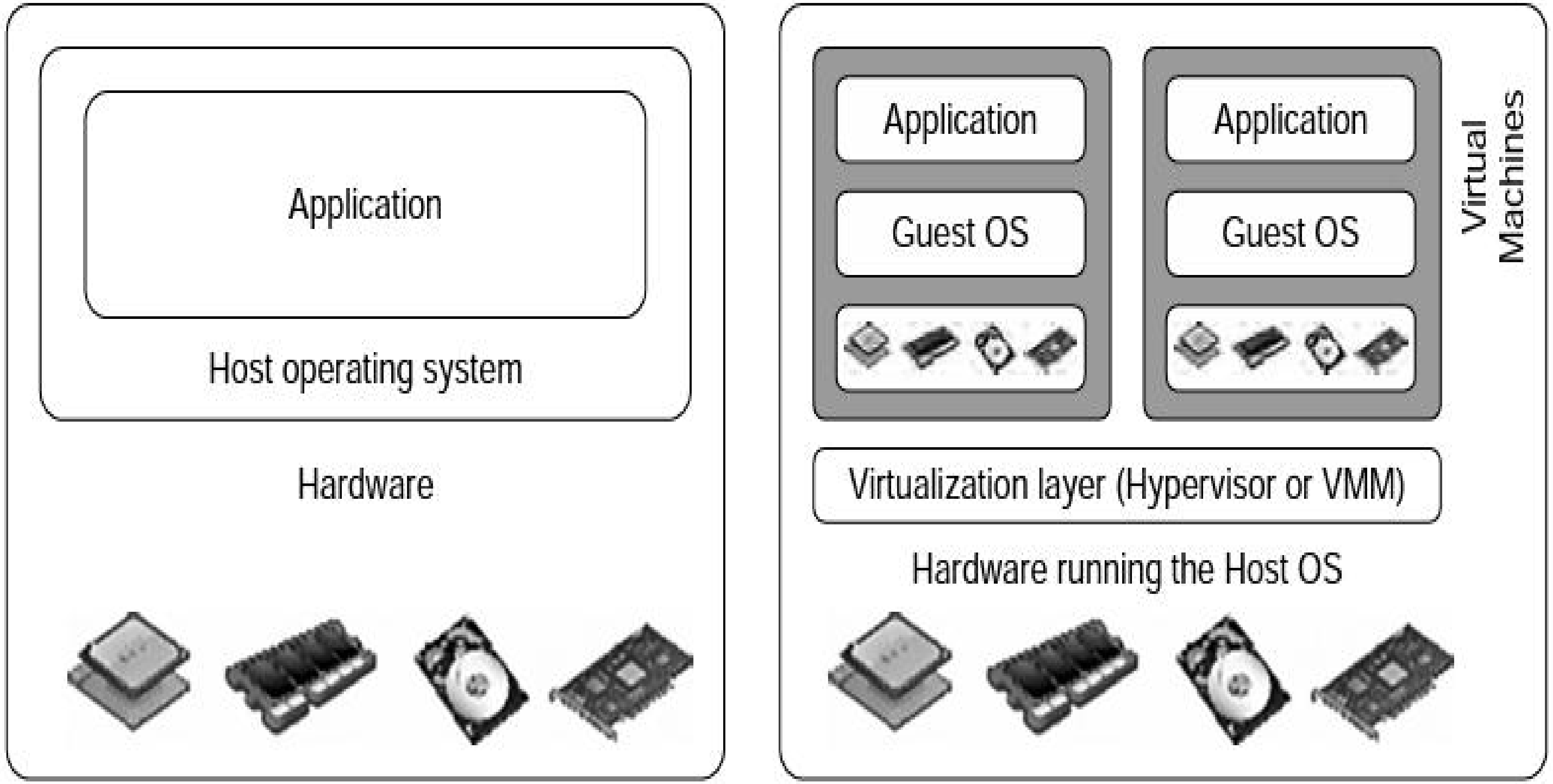
Virtualization

- Virtualization technology enable to create virtual representations of servers, storage, networks, and other physical machines. Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.
- Virtualization = Hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing.
- Virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

- Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine
- The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.
- Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system and software libraries) can be virtualized in various functional layers.

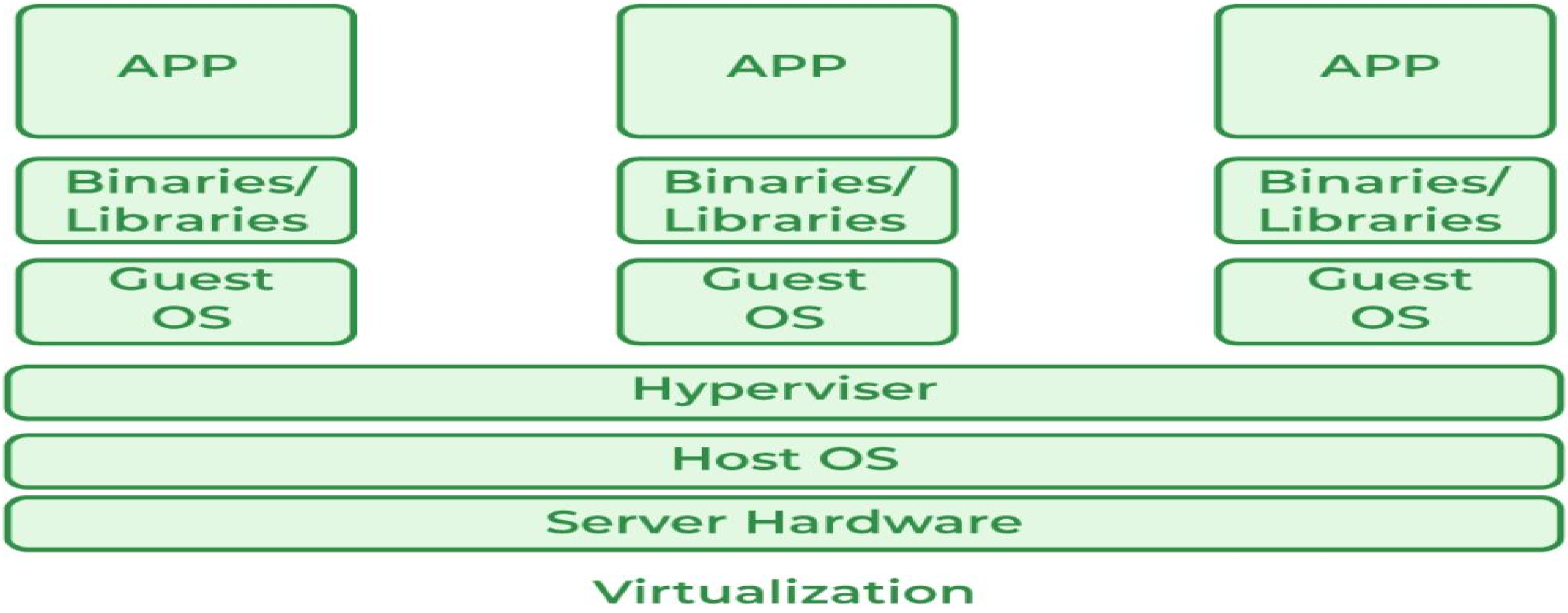
- According to a 2009 Gartner Report, virtualization was the top strategic technology poised to change the computer industry. With sufficient storage, any computer platform can be installed in another host computer, even if they use processors with different instruction sets and run with distinct operating systems on the same hardware



(a) Traditional computer

(b) After virtualization

Typical Virtualization scenario



Advantage of Virtualization

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.
- Pay peruse of the IT infrastructure on demand.
- Enables running multiple operating systems.

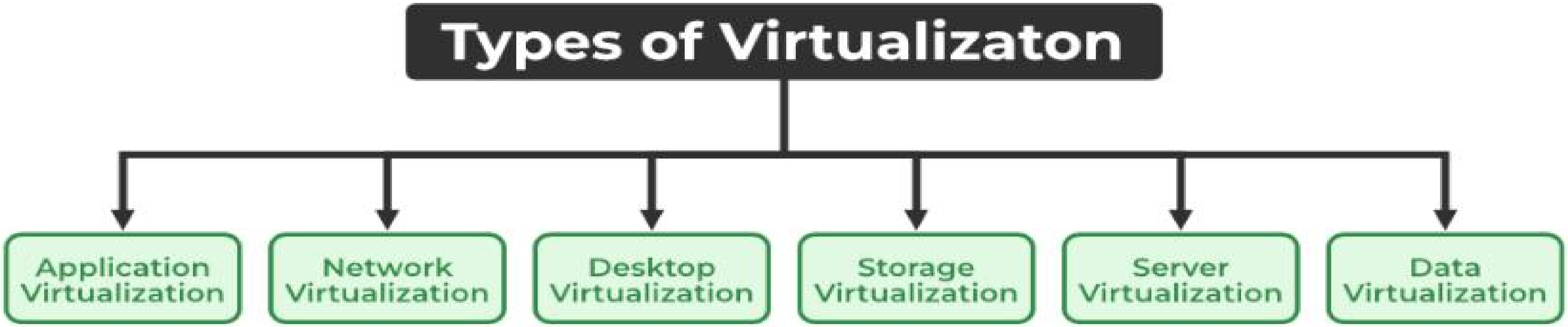
Host Machine

- The machine on which virtual machine is going to create is known as Host Machine.

Guest Machine

- The virtual machines which are created on Host Machine is called Guest Machine.

Types of Virtualization



Types of Virtualization

Why Virtualization in Cloud Computing

- Virtualization is very important concept in cloud computing. In cloud computing, a cloud vendor who will provide cloud services have all physical resources like server, storage device, network device etc. and these physical services are rented by cloud vendors so that user's will not worry about these physical services.

Pros and cons of Virtualization

Pros...

- Utilization of Hardware Efficiently
- Availability increases with Virtualization
- Disaster Recovery is efficient and easy

Cons...

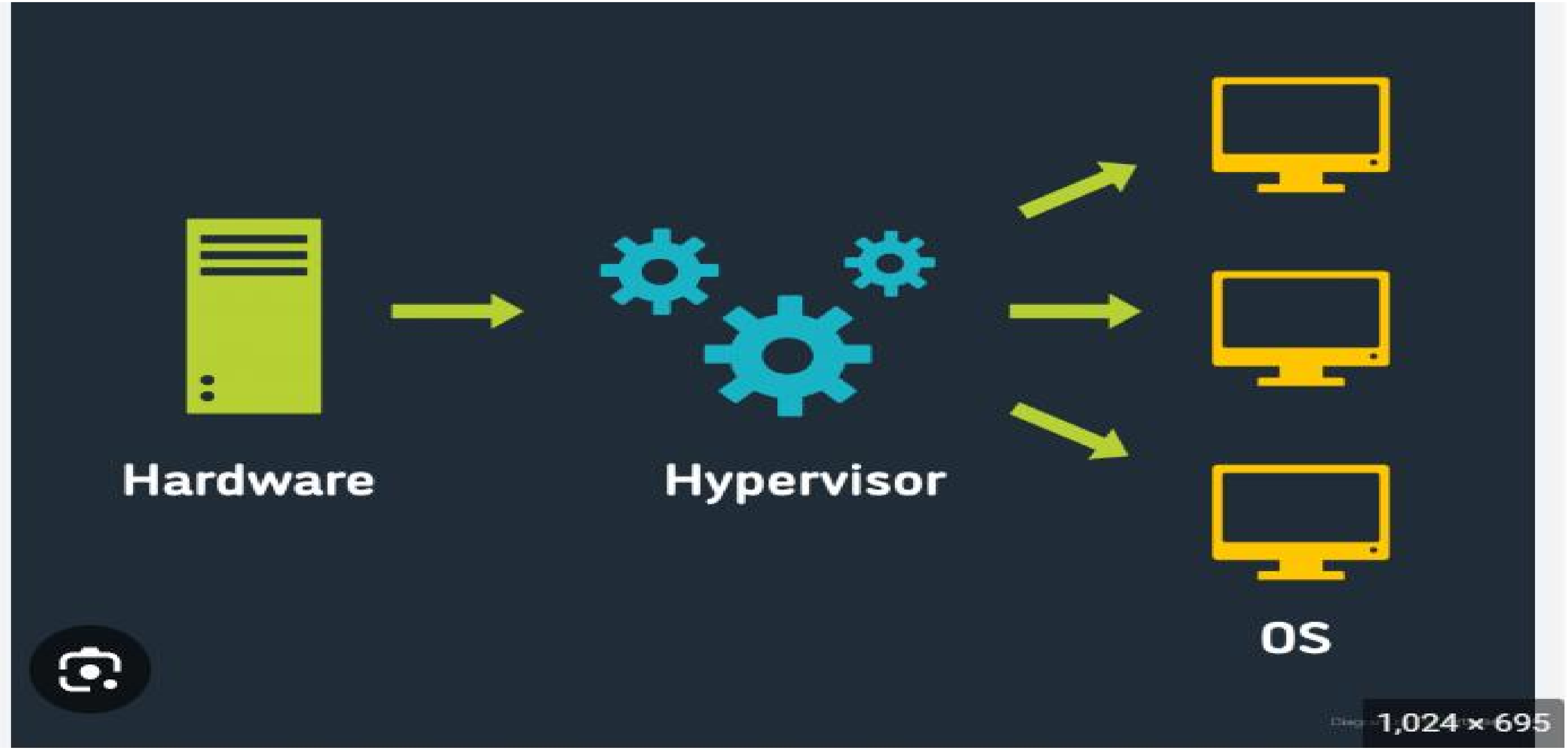
- Data can be at Risk
- Learning New Infrastructure
- High Initial Investment

What is Hypervisor?

- A hypervisor is software that creates and runs virtual machines (VMs). A hypervisor, sometimes called a virtual machine monitor (VMM), isolates the hypervisor operating system and resources from the virtual machines and enables the creation and management of those VMs.
- The physical hardware, when used as a hypervisor, is called the host, while the many VMs that use its resources are guests.
- The hypervisor treats resources—like CPU, memory, and storage—as a pool that can be easily reallocated between existing guests or to new virtual machines.

Hypervisor...

- A hypervisor, also known as a virtual machine manager, is a program that permits multiple operating systems to share one hardware host.
- Hypervisor is compute virtualization software which facilitates manifold operating systems to operate on physical machines simultaneously.
- The hypervisor is the main constituent of the data center consolidation.
- Hypervisor has two main constituents—virtual machine monitor (VMM) and kernel.


Hardware
Hypervisor
OS
1,024 × 695

Types of Hypervisors

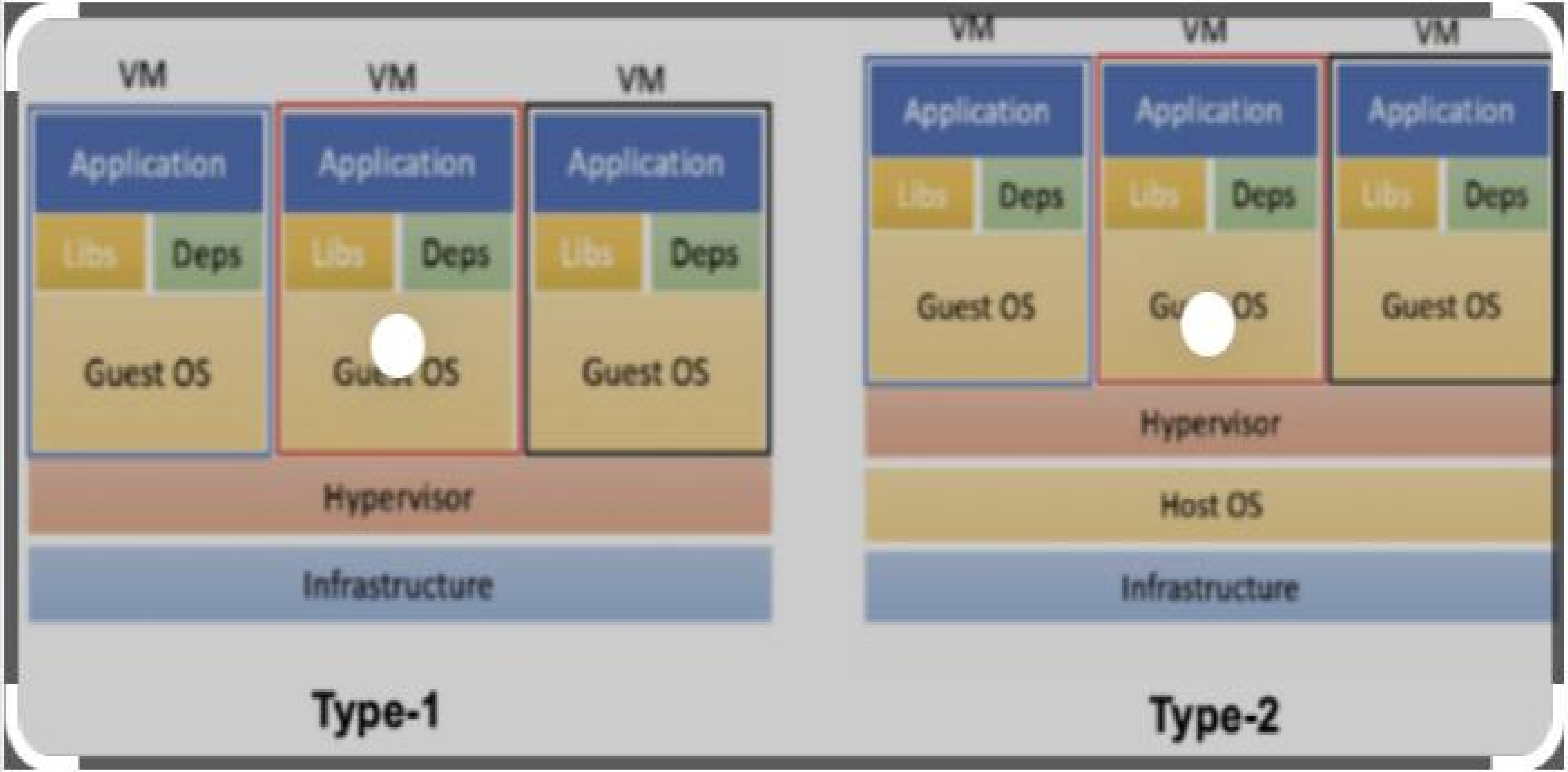
Type 1

- A type 1 hypervisor, also referred to as a native or bare metal hypervisor, runs directly on the host’s hardware to manage guest operating systems. It takes the place of a host operating system and VM resources are scheduled directly to the hardware by the hypervisor.
- This type of hypervisor is most common in an enterprise data center or other server-based environments.

Types…

Type 2

- A type 2 hypervisor is also known as a hosted hypervisor, and is run on a conventional operating system as a software layer or application.
- It works by abstracting guest operating systems from the host operating system. VM resources are scheduled against a host operating system, which is then executed against the hardware.
- A type 2 hypervisor is better for individual users who want to run multiple operating systems on a personal computer.
- VMware Workstation and Oracle VirtualBox are examples of a type 2 hypervisor.



VM
VM
VM
Application
Application
Application
Libs
Deps
Libs
Deps
Libs
Deps
Guest OS
Guest OS
Guest OS
Hypervisor
Infrastructure
Type-1
VM
VM
VM
Application
Application
Application
Libs
Deps
Libs
Deps
Libs
Deps
Guest OS
Guest OS
Guest OS
Hypervisor
Host OS
Infrastructure
Type-2

What is Virtual Machine VM???

- A virtual machine (VM) is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system (located off- or on-premises). Software called a hypervisor separates the machine’s resources from the hardware and provisions them appropriately so they can be used by the VM.
- VMs allow multiple different operating systems to run simultaneously on a single computer—like a Linux distro on a MacOS laptop. Each operating system runs in the same way an operating system or application normally would on the host hardware, so the end user experience emulated within the VM is nearly identical to a real-time operating system experience running on a physical machine.

How does VMs work?

- Virtualization technology allows you to share a system with many virtual environments. The hypervisor manages the hardware and separates the physical resources from the virtual environments. Resources are partitioned as needed from the physical environment to the VMs.
- When the VM is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor schedules the request to the physical system’s resources so that the virtual machine’s operating system and applications can access the shared pool of physical resources.

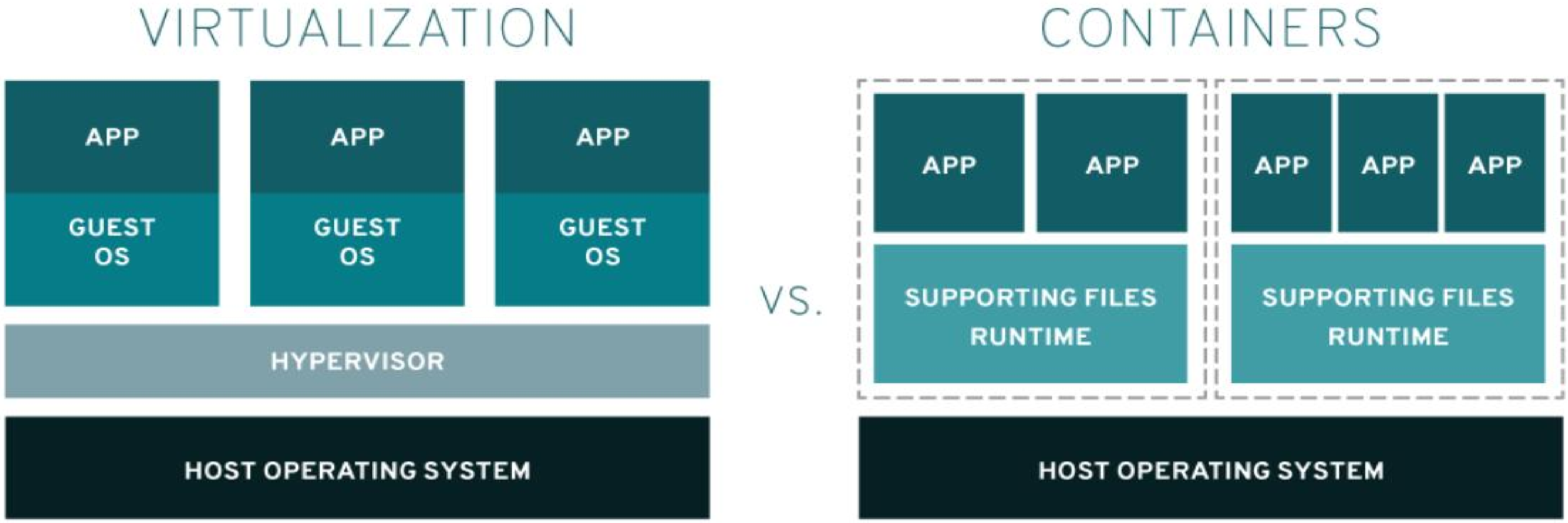
Application of VMs

- House traditional, legacy, and monolithic workloads
- Isolate risky development cycles
- Provision infrastructural resources (such as networks, servers, and data)
- Run a different OS inside another OS (such as running Unix on Linux)

Container

- Containers are typically measured by the megabyte. They don’t package anything bigger than an app and all the files necessary to run, and are often used to package single functions that perform specific tasks (known as a microservice). The lightweight nature of containers—and their shared operating system (OS)—makes them very easy to move across multiple environments. Best suited for ...
- Build cloud-native apps
- Package microservices
- Instill DevOps or CI/CD practices
- Move scalable IT projects across a diverse IT footprint that shares the same OS

VM versus Container Deployment



What is Data Center Virtualization (DCV)?

- Transfer of physical data centers into digital data centers (i.e., virtual) using a cloud software platform, enabling companies to remotely access information and applications.

Data Center Virtualization (DCV)

- Modern data centers are fully virtualized, software defined and highly automated, providing consistent infrastructure and application delivery across a hybrid cloud environment. You can begin your virtual data center journey with server virtualization. The next steps involve adding storage and network virtualization, moving toward a fully virtualized software-defined data center architecture. This means virtualized compute, storage, networking, security and management all on one consistent foundation.

Why has data center virtualization emerged?

- Traditional Data centers have been kept and managed in a physical state from one central location; however, developments in cloud computing and an increased demand for flexible IT solutions have led to emergent technologies that literally transform the traditional data center.

Data Center Virtualization

- It is process of creating a virtual server—sometimes called a software defined data center (SDCC)—from traditional, physical servers.
- The process abstracts physical hardware by mimicking its processors, operating system, and other resources with help from a hypervisor.
- Virtualized data center platforms can be managed from a central physical location (private cloud) or a remote third-party location (public cloud), or any combination of both (hybrid cloud).
- On-site virtualized servers are deployed, managed, and protected by private or in-house teams. Alternatively, third-party servers are operated in remote data centers by another service provider who offers cloud infrastructure solutions to many different organizations and companies based on set fees or a consumption-based (i.e., pay for what you use) model.

Benefits of DCV

- **Scalability:** Unlike physical servers, which need extensive and, at times, expensive sourcing and time management, virtual data centers are relatively simple, quick, and inexpensive to set up. They can be added in response to rapid rises in demand for processing and other resources, or downsized when they are no longer necessary—something that is not possible with metal servers.
- **Enhanced functionality:** Virtualized resources can be accessed from anywhere with a strong enough Internet connection. For example, employees can access data and other applications from remote locations, making productivity possible outside the office.

Continue...

- **Cost savings**: Data center virtualization eliminates the higher management and maintenance associated with physical servers, which are typically outsourced to third-party providers.

Cloud Computing Models

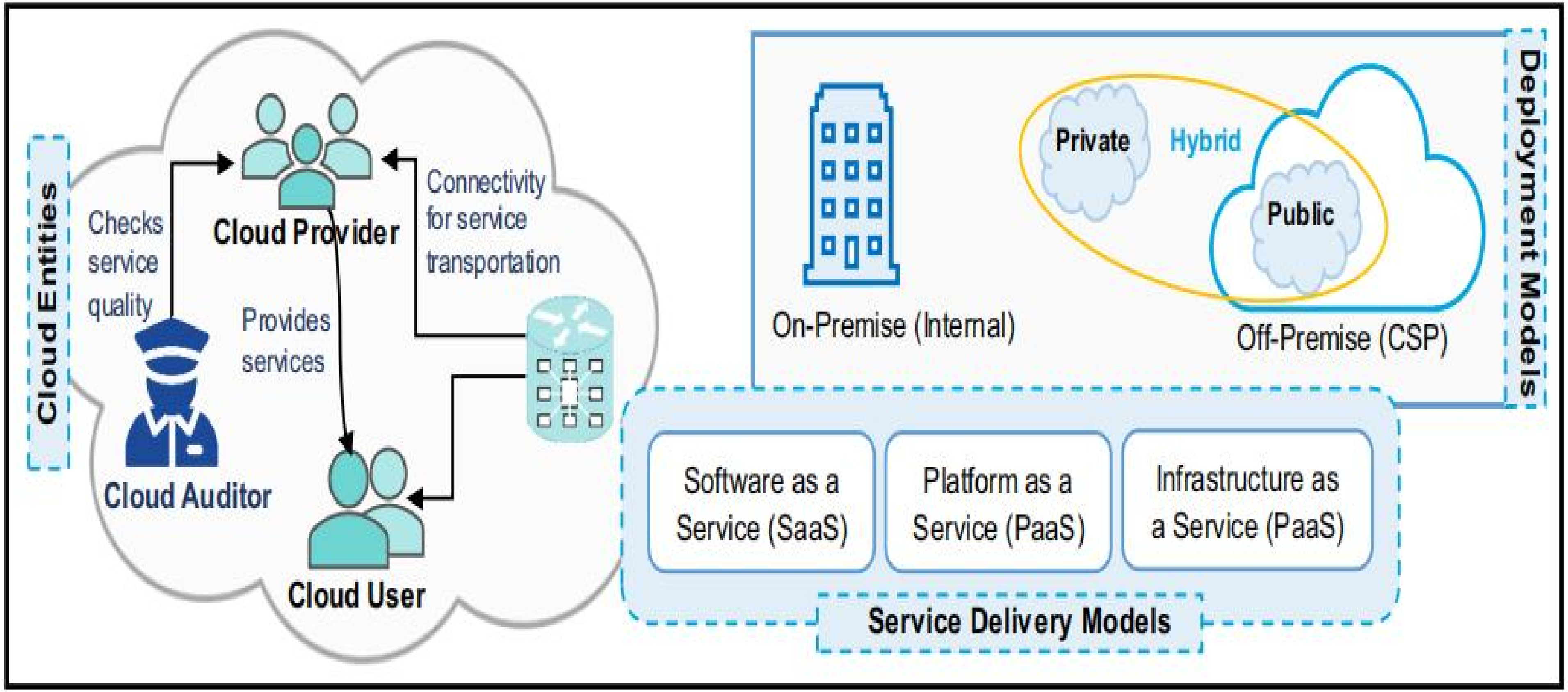


Fig. 1. Overview of Cloud Computing.

Virtualized Cloud Data Center

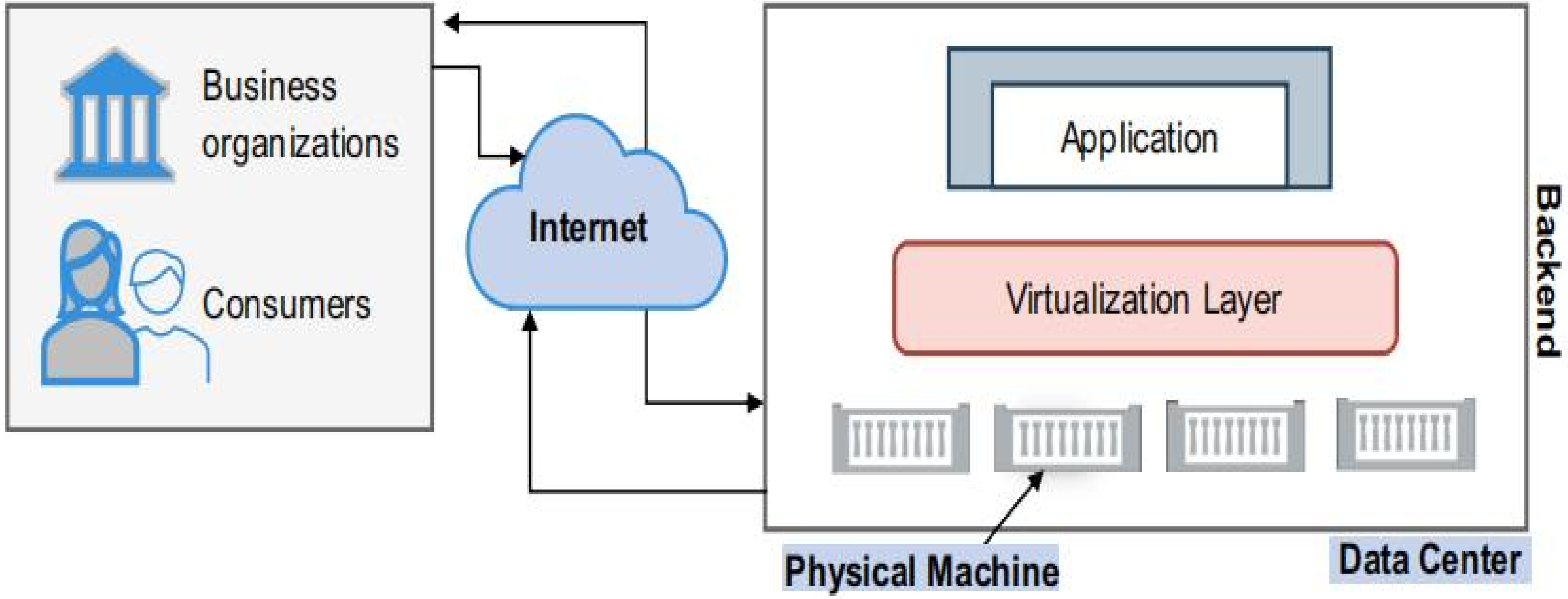
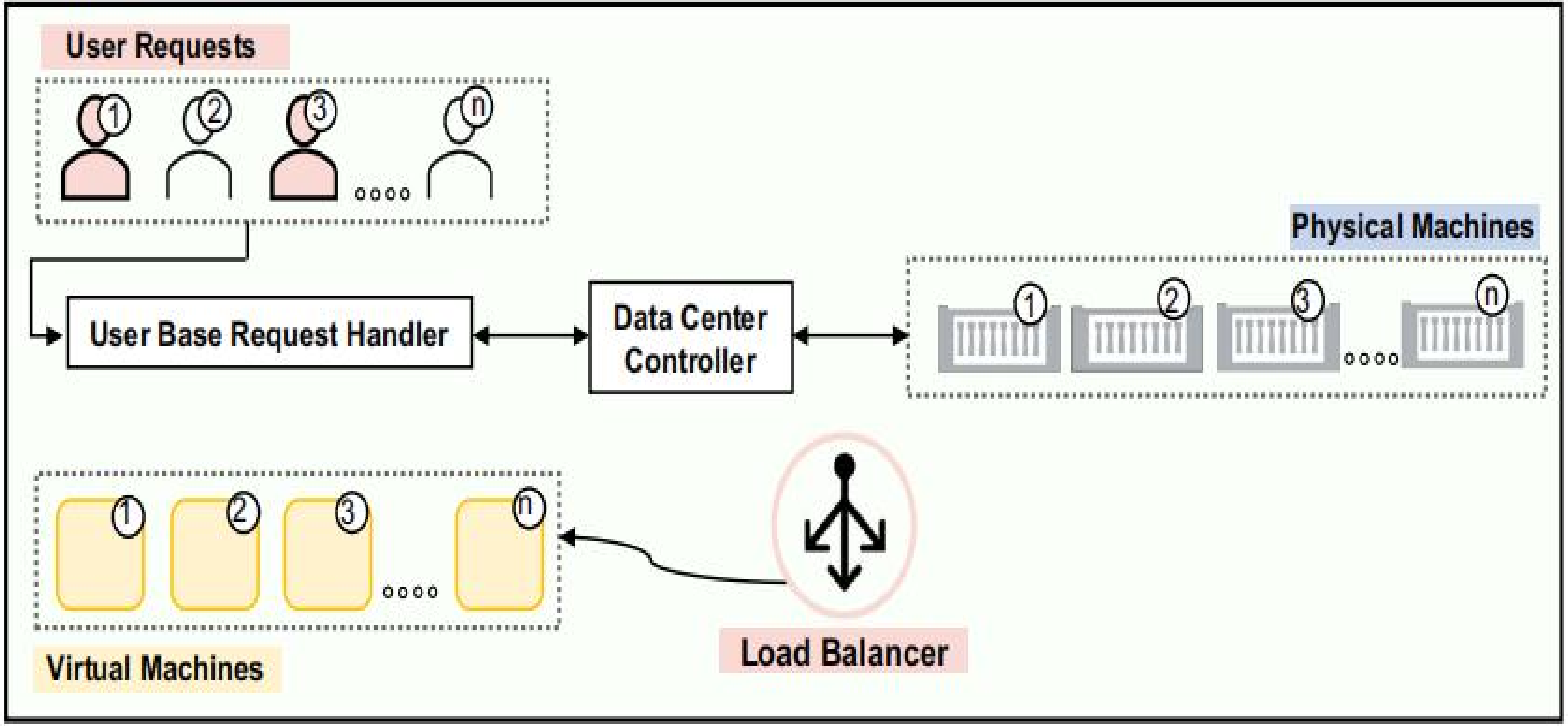


Fig. 2. Cloud Computing Architecture.

Load Balancer in VDC



Primary Goal of Load balancer

- Resource Allocation and Task Scheduling.

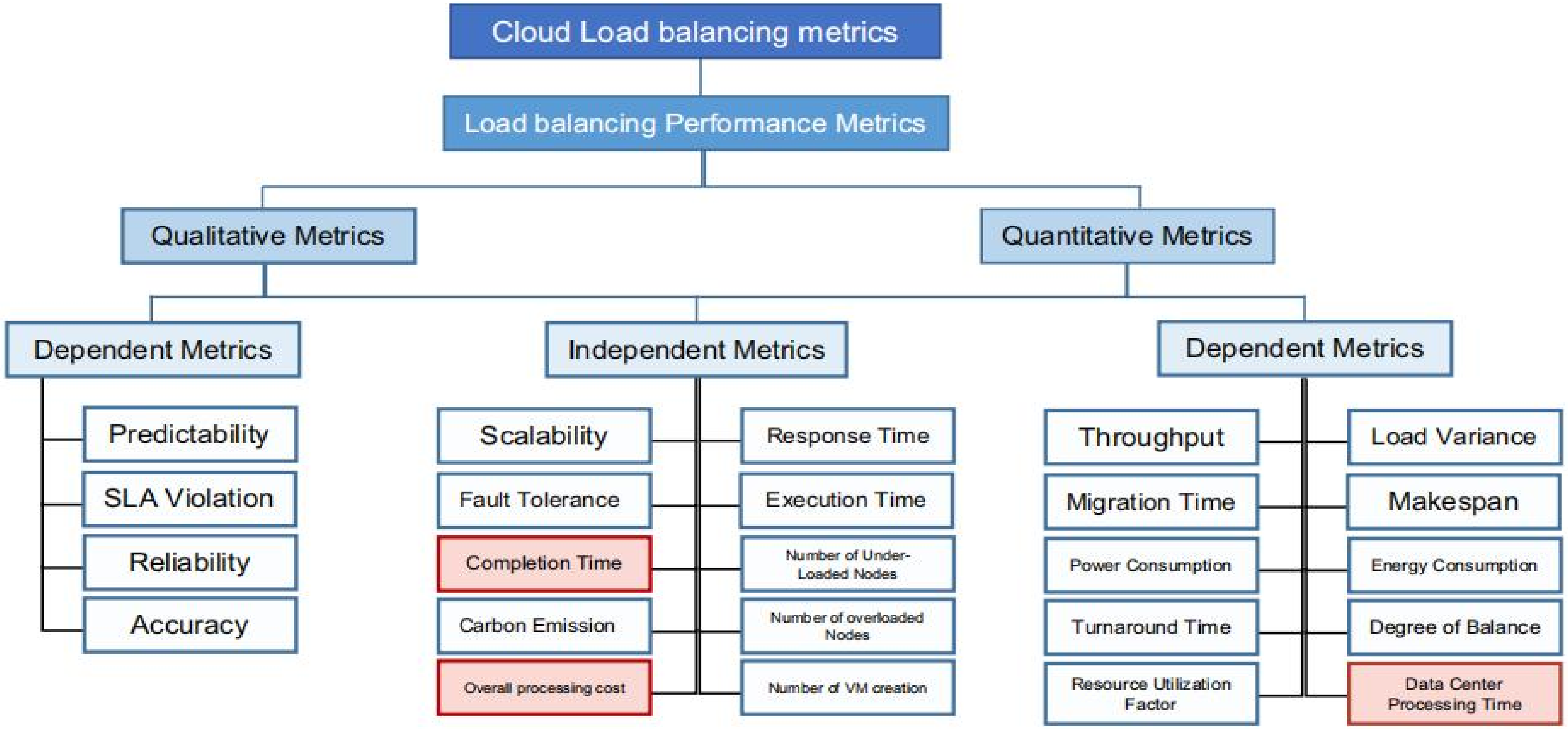


Fig. 6. Classification of Load Balancing Metrics.

Metrics

- **Resource Utilization (RU)**: The extent of utilizing the resources(e.g. memory, CPU, etc.) in the system. It is to measure the degree of RU in the cloud Data Center. As long as there's an increase in demand for services, RU is essential. A maximum RU is required for a good performance of the load balancing algorithm.

- **Scalability (S)**
- Similar to a system, an algorithm should perform well under any unexpected circumstances. Meaning regardless of the increase in the number of tasks and load, the algorithm should remain scalable. Highly scalable for a good performanceof the load balancing algorithm

- ***Resource Utilization (RU):*** The extent of utilizing the resources (e.g. memory, CPU, etc.) in the system. It is to measure the degree of RU in the cloud Data Center. As long as there's an increase in demand for services, RU is essential. A maximum RU is required for a good performance of the load balancing algorithm.
- ***Scalability (S)***: Similar to a system, an algorithm should perform well under any unexpected circumstances. Meaning regardless of the increase in the number of tasks and load, the algorithm should remain scalable. Highly scalable for a good performance of the load balancing algorithm.
- ***Throughput (TP):*** Represents the measure of the number of job requests that have been executed and processed successfully per unit time in the VM. It is the amount of data transferring from one place to another. High TP for a good performance of the load balancing algorithm.
- ***Response Time (RT):*** Amount of time taken by the algorithm to respond to a task. It takes into account the waiting time, transmission time, and service time. It is how much time is needed to respond to a user inquiry. Minimum RT is required in a good load balancing algorithm.

- ***Associated Overhead (AO):*** Amount of overhead formed while executing the load balancing algorithm. It could occur due to a high number of task migration and inter-process communication. A balanced load of system results in minimum overhead by the load balancing algorithm.
- ***Fault Tolerance (FT):*** A load balancing should be able to perform well despite the failure of some system elements. Meaning if one VM is overloaded, another available VM should be able to execute tasks. High FT for a good performance of the load balancing algorithm.
- ***Migration Time (MT):*** It is the total amount of time needed to migrate a task from one VM to another. The migration process should occur without affecting the system's availability. It highly depends on the virtualization concept in the cloud. Low MT for a good performance of load balancing algorithm.
- ***SLA Violation:*** Denotes the number of reductions of SLA violation factors in terms of deadline constraint, priority etc. Violations in SLA occurs due to situations where resources (VMs) are unavailable because they're overloaded. Minimum SLA for a higher level of user satisfaction.

- **Static Load Balancing (SLB) Algorithms**: in a static environment, the load balancing algorithms' process depends on prior knowledge (Alam and Ahmad Khan, 2017) of the system state along with its properties and capabilities. Examples of prior information could include memory, storage capacity, and processing power. Such information represents the load of the system. Static-based algorithms do not take into account dynamic changes to the load during runtime (Mj et al., 2014). Thus, the major drawback of these algorithms is low fault tolerance due to sudden changes in load.
- **Dynamic Load Balancing (DLB) Algorithms:** these types of algorithms are known to be better and adaptable for Load Balancing. Unlike SLB algorithms, in a dynamic environment, the load balancing algorithms take into account the previous state of the system (Alam and Ahmad Khan, 2017). Thus, the major benefit of these algorithms is flexibility although they

- **Nature-inspired Load Balancing (NLB) Algorithms:** such algorithms represent biological processes or activities based on human nature (Thakur and Goraya, 2017) such as the genetic process of the searching method of bees to find honey. These processes are modeled mathematically to adopt the natural processes to perform load balancing in CC. The development of these intelligent algorithms perform better for complex and dynamic systems.

Dynamic algorithms are better than static as no prior knowledge is needed and it takes the current state of the system which makes it more efficient for distributed cloud systems (Fatima et al., 2019). Also, dynamic algorithms eliminates the overhead for storing the previous state of the system and these algorithms have higher runtime complexity compared to static (Kamboj and Ghumman, 2016; Haryani and Jagli, 2014). Nature-inspired algorithms are known to be more intelligent as they belong to the metaheuristic class of load balancing algorithms and can resemble natural phenomena explained by natural sciences (Siddique and Adeli, 2015).

Resource Management

- The term resource management refers to the operations used to control how capabilities provided by Cloud resources and services are made available to other entities, whether users, applications, or services.
- Types of Resources
- Physical Resource: Computer, disk, database, network, etc.

- Logical Resource: Execution, monitoring, and application to communicate

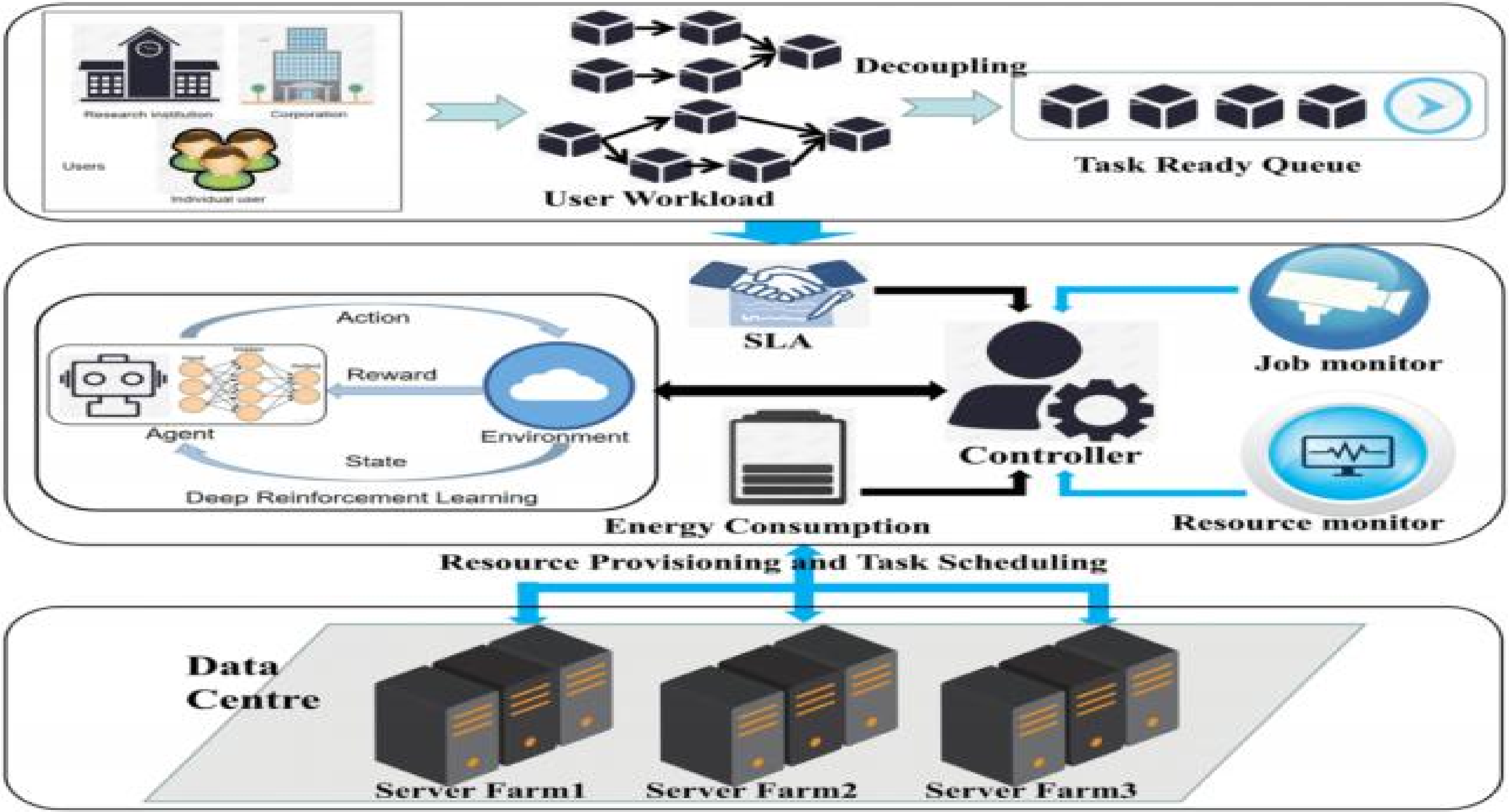
Resource Management Models

- Resource in the cloud is shared by all users at the same time. It allows the user to reserve the VM's memory to ensure that the memory size requested by the VM is always available to operate locally on clouds with a good enough level of QoS (Quality of Service) being delivered to the end user.

Introduction

- Cloud computing is becoming the standard of IT infrastructure and replacing the traditional on-promise deployment paradigm.
- Three levels of service in cloud computing are abstracted as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS)
- Resource management is the primary issue as the demand grows for provisioning resources and computation in cloud systems.

Cloud Resource Management



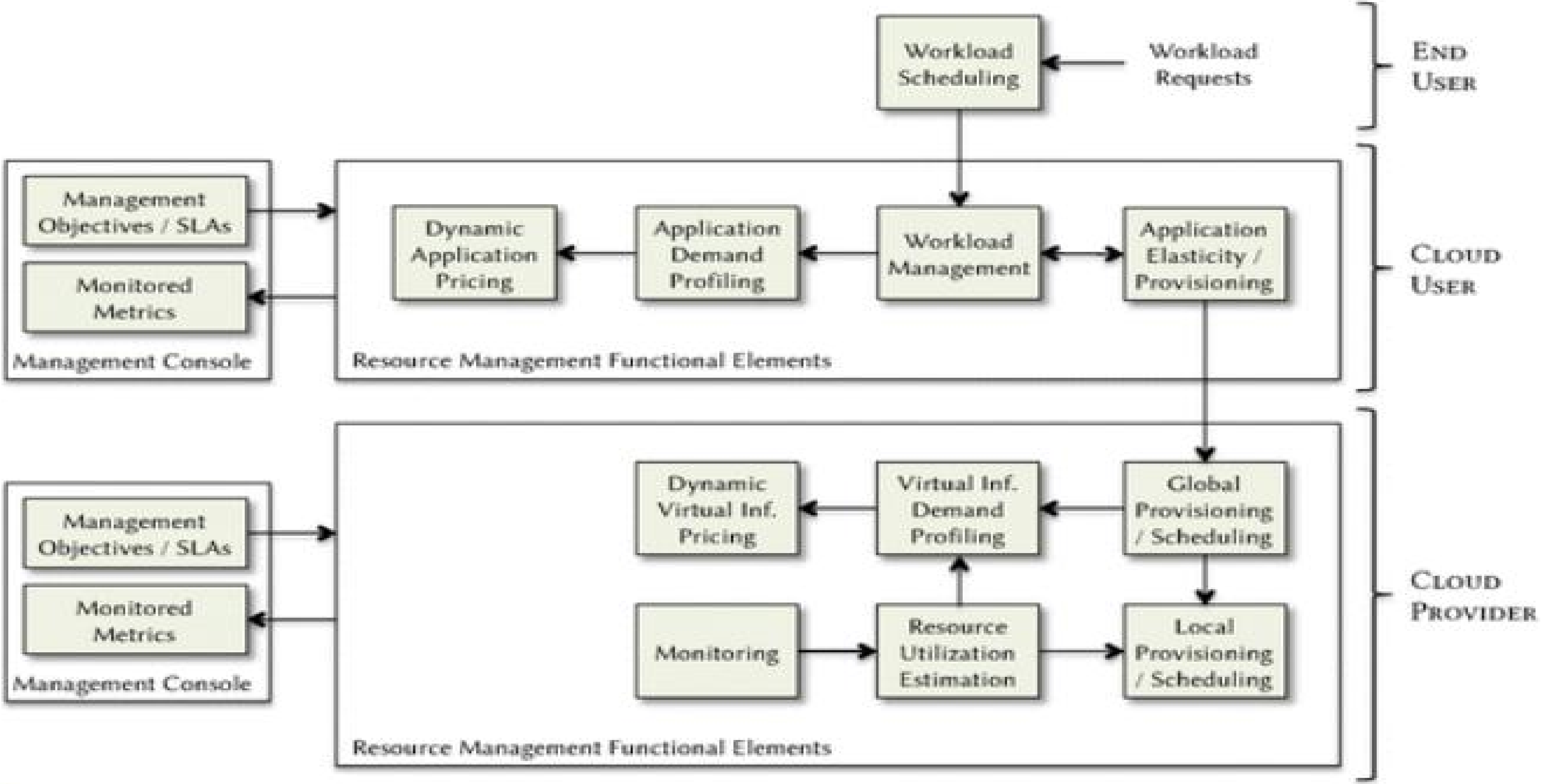
Introduction

- Cloud is a complex system with a very large number of shared resources subject to unpredictable requests and affected by external events it cannot control.
- Cloud resource management requires complex policies and decisions for multi-objective optimization.
- Cloud resource management is extremely challenging because of the complexity of the system which makes it impossible to have accurate global state information and because of the unpredictable interactions with the environment.

Motivation

- **Cloud resource management** .
  - Requires complex policies and decisions for multi-objective optimization.
  - It is challenging - the complexity of the system makes it impossible to have accurate global state information.
  - Affected by unpredictable interactions with the environment, e.g., system failures, attacks.
  - Cloud service providers are faced with large fluctuating loads which challenge the claim of cloud elasticity.
- The strategies for resource management for IaaS, PaaS, and SaaS are different.

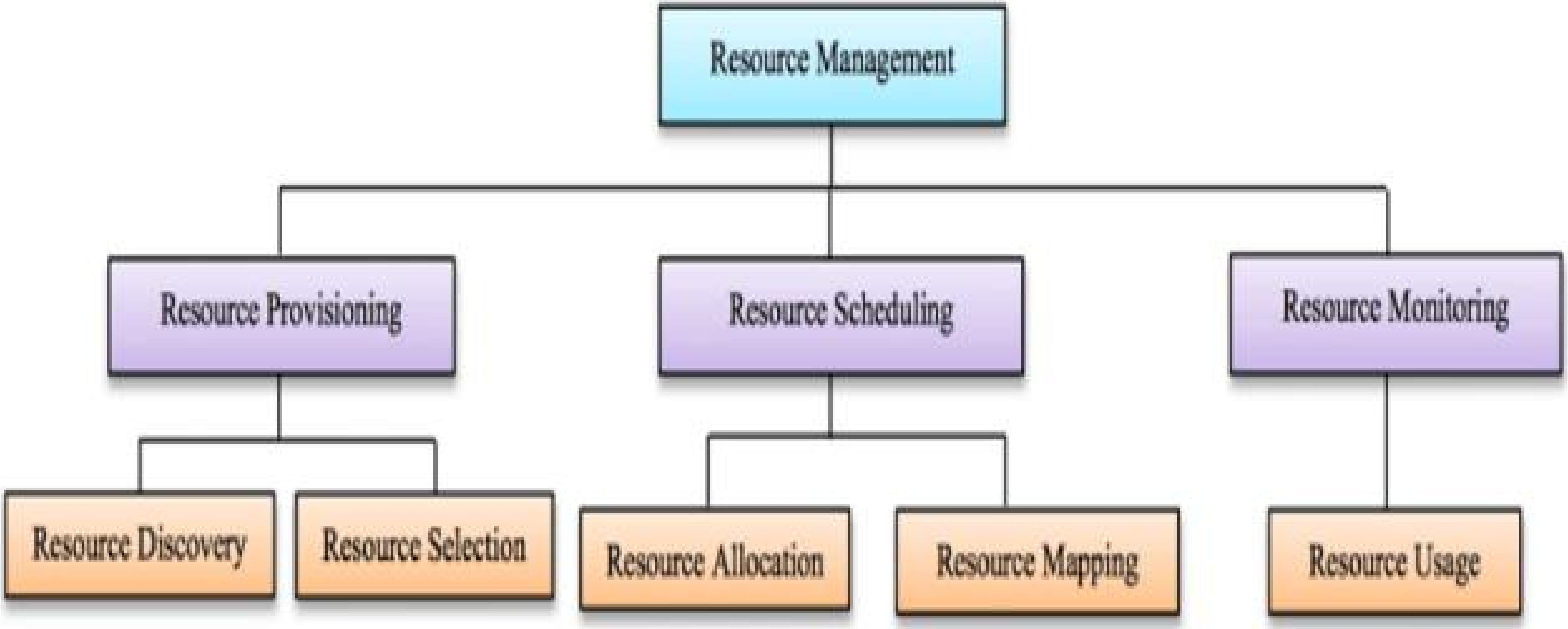
Resource management in Cloud



Resource management and scheduling

- Critical function of any man-made system.
- It affects the three basic criteria for the evaluation of a system:
    - Functionality.
    - Performance.
    - Cost.
- **Scheduling in a computing system** → deciding how to allocate resources of a system, such as CPU cycles, memory, secondary storage space, I/O and network bandwidth, between users and tasks.
- Policies and mechanisms for resource allocation.
    - Policy → principles guiding decisions.
    - Mechanisms → the means to implement policies

Cloud Resource Management



Resource management comprises nine components:

1. **Provisioning:** Assignment of resources to a **workload**.
2. **Allocation:** Distribution of resources among competing workloads.
3. **Adaptation:** Ability to dynamically adjust resources to fulfill workload requirements.
4. **Mapping:** Correspondence between resources required by the workload and resources provided by the cloud infrastructure.
5. **Modeling:** Framework that helps to predict the resource requirements of a workload by representing the most important attributes of resource management, such as states, transitions, inputs, and outputs within a given environment.

Resource management comprises nine components:

6. **Estimation:** Guess of the actual resources required for executing a workload.
7. **Discovery:** Identification of a list of resources that are available for workload execution.
8. **Brokering:** Negotiation of resources through an agent to ensure their availability at the right time to execute the workload.
9. **Scheduling:** A timetable of events and resources, determining when a workload should start or end depending on its duration, predecessor activities, predecessor relationships, and resources allocated

Cloud Workload

- The work function (application or service) processed by a remote server or instance at any given time; it generally has users or applications interacting with it through the Internet. **Cloud workloads** can range from a web server to a database to a container.
- The term *workload* as commonly defined in computer science and the IT domain has primarily been characterized from a **system resource usage** or capacity perspective and, therefore, is very restrictive.
- On UNIX platforms, a workload was referred to in the context of the **number of processes** that were running on the UNIX host.
- The UNIX operating system went as far as providing an uptime command. This command provided a convenient way for a system administrator to examine the workload that was currently running on the system and a means to assess whether the *load* on the system was increasing or decreasing.

Workload model

- **Non-interactive workloads** are the workloads that can be processed on only one specific server and do not need interactions with other servers (such as scientific computing and image processing).

- A distinction should be made between interactive and non-interactive workloads; the management techniques for **interactive workloads**, e.g., web services, involve flow control and dynamic application placement, while those for **non-interactive** workloads are focused on scheduling.
- A fair amount of work reported in the literature is devoted to resource management of interactive workloads, some to non-interactive, and only a few, e.g., to heterogeneous workloads, a combination of the two.

The processes of interactive and non-interactive workloads.

- The Nuomi is a group buying application, Waimai is a take-out service, and Alipay is an online payment platform.
- When a user clicks an item on Nuomi, the latency is quite short because this query does not require many interactions among services.

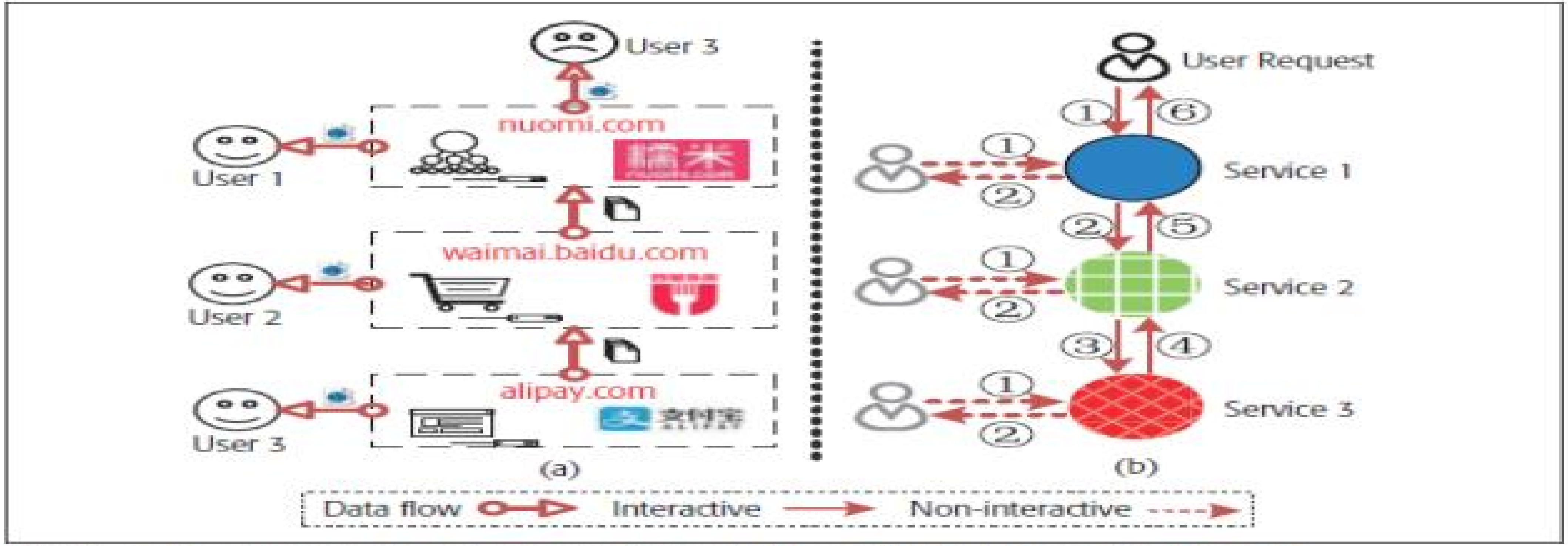


FIGURE The processes of interactive and non-interactive workloads.

The processes of interactive and non-interactive workloads.

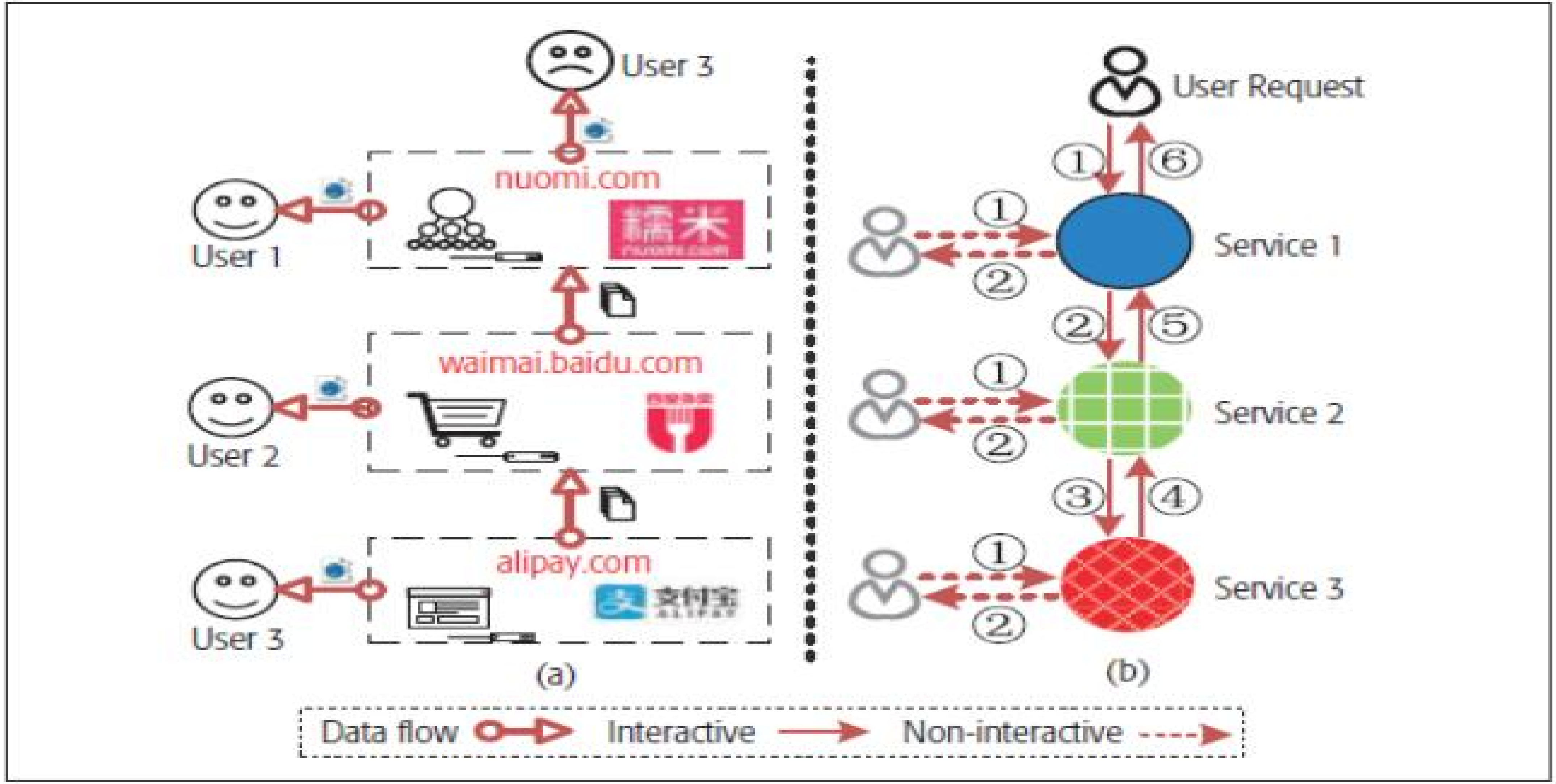


FIGURE The processes of interactive and non-interactive workloads.

The processes of interactive workloads

- When the user orders a take-out and purchases the item, the request goes through Nuomi, Waimai, and then Alipay.
- This interactive workload consists of several highly-dependent operations that have to be processed on different servers separately.
- As shown in Fig. b, there are six procedures for interactive workloads and only two procedures for non-interactive workloads.
- The interactive workloads in chained applications will introduce extra latency to users because these requests will be handled by different services for multiple times.

The processes of interactive workloads

- When service 1 receives an interactive request (TCP:0) that cannot be served in service 1 (i.e., this request requires data from services 2 and 3), then service 1 will make a new TCP connection (TCP:1) to service 2, and the original connection (TCP:0) would be hung up.

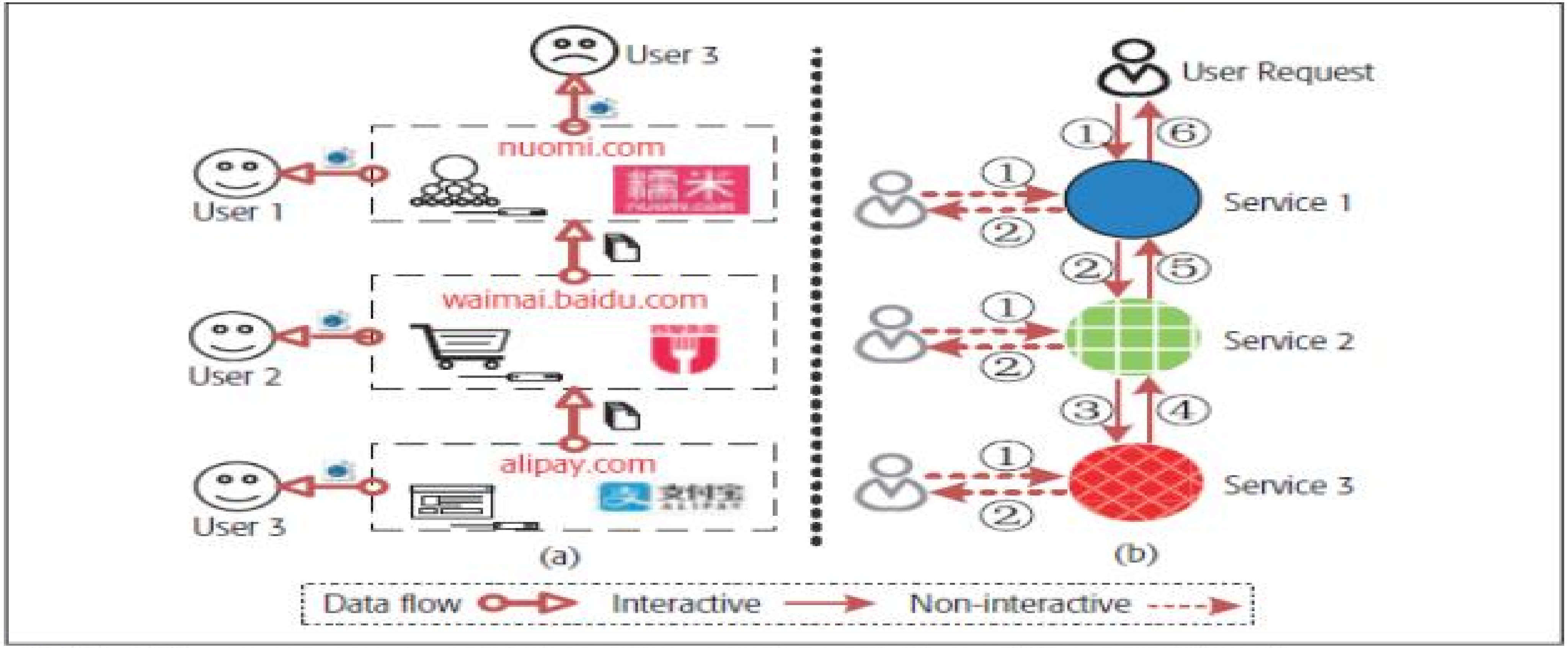


FIGURE The processes of interactive and non-interactive workloads.

Simulator for Cloud Infrastructure

Simulation is nothing but an experiment with the help of computers, without performing actual experiment. It saves lot of money which is required, if we actually perform experiments with the real system.

This simulation toolkit allows the researchers as well as cloud developers to test the performance of the potential cloud application for performance testing in a controlled and easy to setup environment.

1.3.1 CloudSim [2]

CloudSim is a framework developed by the GRIDS laboratory of University of Melbourne which enables seamless modelling, simulation and experimenting on designing Cloud computing infrastructures. CloudSim is a self-contained platform which can be used to model data centers, service brokers, scheduling and allocation policies of a large scaled Cloud platform. It provides a virtualization engine with extensive features for modelling the creation and life cycle management of virtual engines in a data center. CloudSim framework is built on top of GridSim framework also developed by the GRIDS laboratory.

The CloudAnalyst is built directly on top of CloudSim framework leveraging the features of the original framework and extending some of the capabilities of CloudSim.

1.3.2 GridSim [3]

GridSim toolkit was developed by Buyya et al to address the problem of near impossibility of performance evaluation of real large scaled distributed environments (typically Grid systems but also P2P networks) in a repeatable and controlled manner. The GridSim toolkit is a Java based simulation toolkit that supports modelling and simulation of heterogeneous Grid resources and users spread across multiple organizations with their own policies. It supports multiple application models and provides primitives for creation of application tasks, mapping of tasks to resources and managing such tasks and resources.

1.3.3 simjava [4]

SimJava is the underlying event based simulation toolkit used in both CloudSim and GridSim.

1.4 Terminology and Abbreviations

Data Transmission Latency	This document uses "Transmission Latency" to mean the network delay (based on geographical distance, operation of network equipment etc.) between two points. This can be considered equivalent to half of the ping round-trip time.
Data Transfer Time	Data Transfer Time is the time taken by a given amount of data to be transported from one point to another. This is taken to be equivalent to the available bandwidth divided by the size of the unit of data.
Response Time	The time taken by an Internet application defined as the time interval between sending the request and receiving a response
VM	Virtual Machine
VMM	Virtual Machine Monitor

Features of Simulator

2.2.1 Ease of use

Ease of setting up and executing a simulation experiment is the main point of having a simulation tool. The simulator needs to provide an easy to use graphical user interface which is intuitive yet comprehensive.

2.2.2 Ability to define a simulation with a high degree of configurability and flexibility

Perhaps the most important feature is the level of configurability the tool can provide. A simulation, especially of the nature of modelling something as complex as an Internet Application depends on many parameters and most of the time the values for those parameters need to be assumed. Therefore it is important to be able to enter and change those parameters quickly and easily and repeat simulations.

2.2.3 Graphical output

A picture is said to be worth a thousand words. Graphical output in the form of tables and charts is highly desirable to summarise the potentially large amount of statistics that is collected during the simulation. Such effective presentation helps in identifying the important patterns of the output parameters and helps in comparisons between related parameters.

Features of Simulator

2.2.4 Repeatability

Repeatability of experiments is a very important requirement of a simulator. The same experiment with the same parameters should produce similar results each time the simulation is executed. Otherwise the simulation becomes just a random sequence of events rather than a controlled experiment.

It is also helpful to be able to save an experiment (the set of input parameters) as a file and also be able to save the results of an experiment as a file.

2.3 Simulation Output / What is being Measured

Following are the statistical measures produced as output of the simulation in the initial version of the simulator.

- Response time of the simulated application
  - Overall average, minimum and maximum response time of all user requests simulated
  - The response time broken down by user groups, located within geographical regions
  - The response time further broken down by the time showing the pattern of change over the duration of a day
- The usage patterns of the application
  - How many users use the application at what time from different regions of the world, and the overall effect of that usage on the data centers hosting the application
- The time taken by data centers to service a user request
  - The overall request processing time for the entire simulation
  - The average, minimum and maximum request processing time by each data center
  - The response time variation pattern during the day as the load changes
- The cost of operation

Cloud Analyst Simulator

3.2.1 Region

In the CloudAnalyst the world is divided in to 6 'Regions' that coincide with the 6 main continents in the World. The other main entities such as User Bases and Data Centers belong to one of these regions. This geographical grouping is used to maintain a level of realistic simplicity for the large scaled simulation being attempted in the CloudAnalyst.

3.2.2 Internet

The CloudAnalyst Internet is an abstraction for the real world Internet, implementing only the features that are important to the simulation. It models the Internet traffic routing around the globe by introducing suitable transmission latency and data transfer delays. The transmission latency and the available bandwidth between the 6 regions are configurable.

4.1 Setting up a Simulation

To set up a simulation you need to carry out the following steps. (Please note the screens mentioned here are explained in detail in the next section.)

1. Define user bases – Using User Base entities define the users of the application, their geographic distribution, and other properties such as the number of users, the frequency of usage and the pattern of usage such as peak hours. This is done in the Main tab of the Configure Simulation screen.
2. Define data centers – Using the Data Centers tab of the Configuration screen define the data centers you wish to use in the simulation. Define all the hardware and accounting aspects of the data centers here.
3. Allocate Virtual Machines for the application in Data Centers – Once the data centers have been created, you need to allocate virtual machines in them for the simulated application using the Main tab of the Configurations screen. A data center defined in step 2 above does not get included in the simulation unless it is allocated in this step. You can allocate multiple types of virtual machines in the same data center during this step.
4. Review and adjust the advanced parameters in the Advanced tab of the Configuration Screen.
5. Review and adjust the network latency and bandwidth matrices on the Internet Characteristics screen.

Main Screen with Simulation Panel

Figure 7. CloudAnalyst Main Screen

The configuration options on the main tab are:

1. Simulation time – the duration of the simulation which can be given in minutes, hours or days
2. User Bases Table – This is a table listing out all the user bases in the simulation. Each user base has following configurable fields, represented by a single row in the table.
    a. Name
    b. Region
    c. Requests per user per hour
    d. Data size per request
    e. Peak hours
    f. Average users during peak hours
    g. Average users during off-peak hours

    The Add and Remove buttons next to the table can be used to add or remove user bases from the configuration.

3. Application Deployment Configuration – This table lists how many virtual machines are allocated for the application in each data center from the Data Centers tab, along with the details of a virtual machine. The fields are:
    a. Data Center – This is a drop down listing the names of data centers created in the Data Center tab.
    b. Number of VMs – How many VMs to be allocated to the application from the selected data center
    c. Image Size – a single VM image size in bytes
    d. Memory – amount of memory available to a single VM
    e. BW – amount of bandwidth available to a single VM

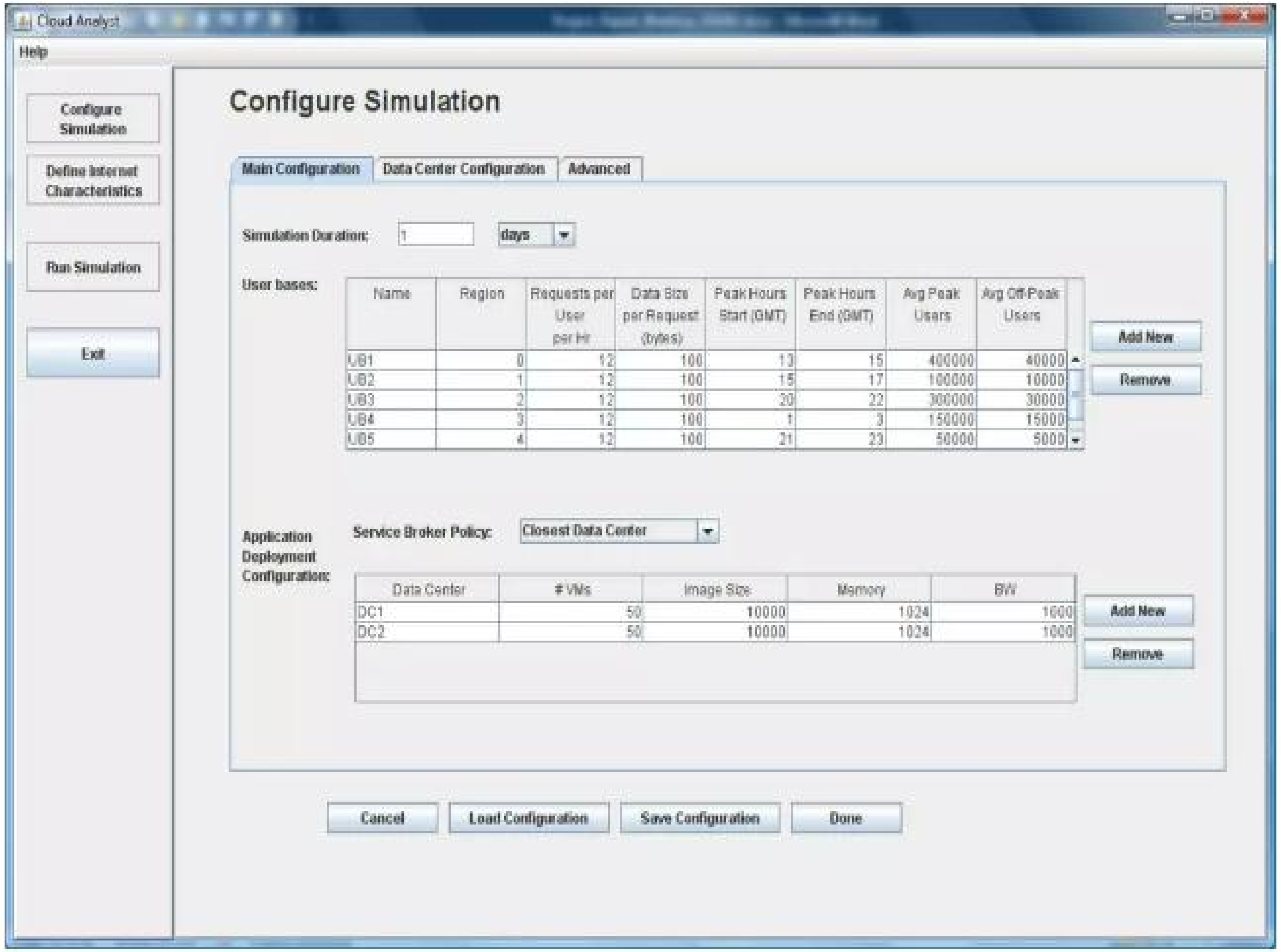


Figure 8. Configure Simulation Screen - Main Tab

4. Service Broker Policy – This drop down allows you to select the brokerage policy between data centers that decide which data center should receive traffic from which user base. The available policies are:
   a. Closest data center – The data center with the least network latency (disregarding network bandwidth) from a particular user base is sent all the requests from that user base.
   b. Optimize response time – This policy attempts to balance the load between data centers when one data center gets over loaded. Please see 3.5.2.2 for the algorithm used.

The "Save Configuration" button allows you to save the configuration created as a file. Simulation files are saved with a .sim extension. Similarly using the "Load Configuration" button you can load a previously saved simulation configuration.

4.2.2.2 Data Center Tab

The data center tab allows you to define the configuration of a data center (see Figure 8 below). The table at the top lists the data centers and using the Add/Remove buttons you can add or remove data centers to the configuration. The parameter fields are:

1. Name
2. Region
3. Architecture – Architecture of the servers used in the data center. e.g. X86
4. Operating System – e.g. Linux
5. Virtual Machine Monitor (VMM)
6. Cost per VM Hour
7. Cost per 1Mb Memory Hour
8. Storage cost per Gb
9. Data Transfer cost per Gb (both in and out)
10. Number of servers

When you select a data center from this table a second table will appear below it with the details of the server machines in the data center. The parameters for each machine can be given according to the available fields.

1. Machine Id
2. Memory

3.2.3 Cloud Application Service Broker

The traffic routing between User Bases and Data Centers is controlled by a Service Broker that decides which Data Center should service the requests from each user base. Current version of CloudAnalyst implements three types of service brokers each implementing a different routing policy.

1. Service Proximity based routing.

   In this case the proximity is the quickest path to the data center from a user base based on network latency. The service broker will route user traffic to the closest data center in terms of transmission latency.

2. Performance Optimized routing

   Here the Service Broker actively monitors the performance of all data centers and directs traffic to the data center it estimates to give the best response time to the end user at the time it is queried.

3. Dynamically reconfiguring router

   This is an extension to Proximity based routing, where the routing logic is very similar, but the service broker is entrusted with the additional responsibility of scaling the application deployment based on the load it is facing. This is done by increasing or decreasing the number of VMs allocated in the data center, according to the current processing times as compared against best processing time ever achieved.

When you select a data center from this table a second table will appear below it with the details of the server machines in the data center. The parameters for each machine can be given according to the available fields.

1. Machine Id
2. Memory

3. Storage
4. Available network bandwidth
5. Number of processors
6. Processor speed (MIPS)
7. VM allocation policy (time shared/space shared)

4. Load balancing policy – the load balancing policy used by all data centers in allocating requests to virtual machines. Available policies are:
    a. Round-robin
    b. Equally Spread Current Execution Load – The load balancer keeps track of how many Cloudlets are currently being processed by each VM and tries to even out the active load.
    c. Throttled – The load balancer throttles the number of requests assigned to a single VM. See section 3.5.1.1 for the throttling algorithm.

The Internet Characteristics screen can be used to set the Internet latency and bandwidth parameters. It presents two matrices for these two categories.

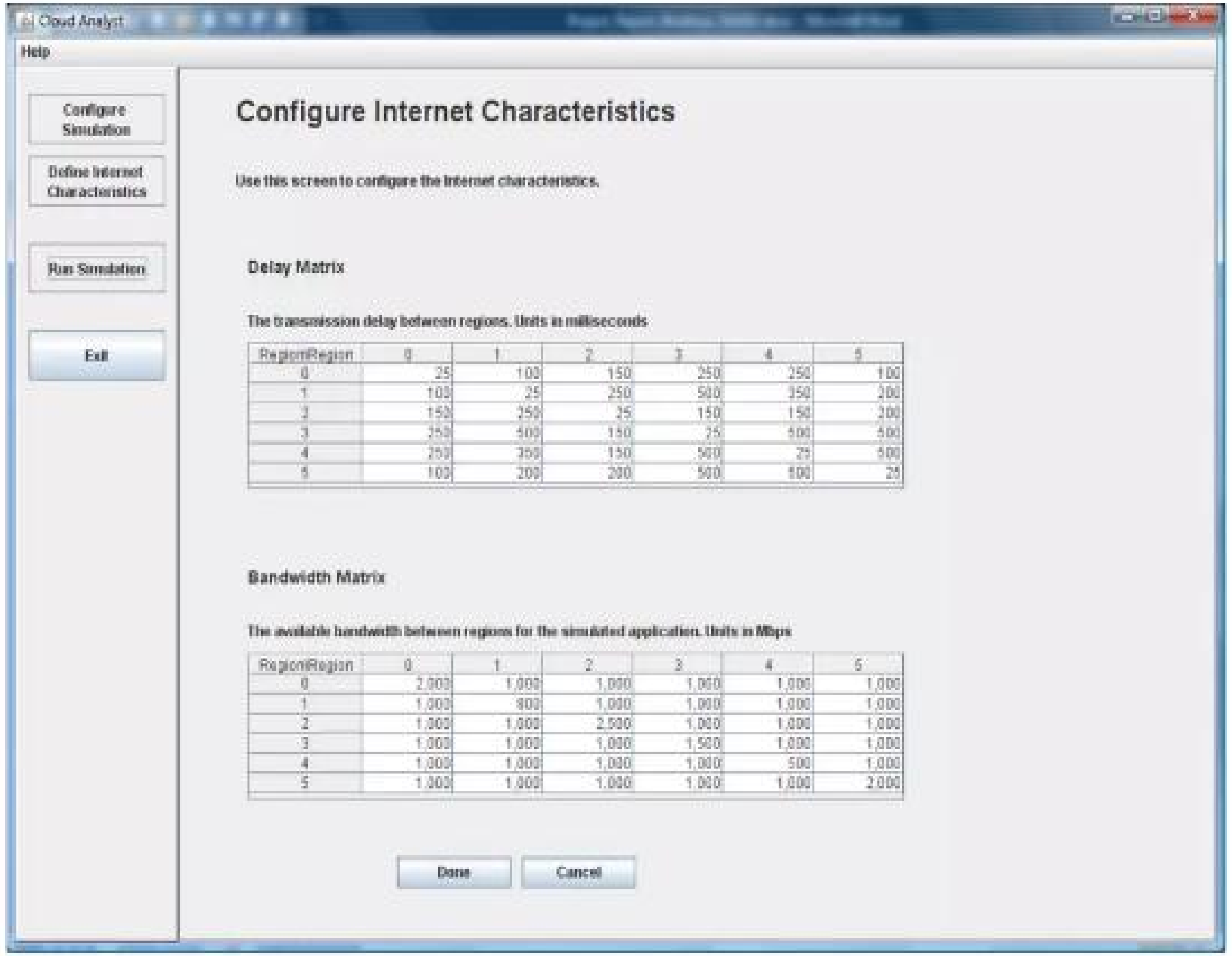


Figure 11. Internet Characteristics Screen

Running the Simulator

Once the above screens have been used to successfully create a simulation configuration, the user has to go back to the main screen and execute the simulation by selecting the "Run Simulation" from the control panel. This will start the simulation and the progress bar at the top of the simulation panel shows the percentage completion of the simulation. The simulation screen will display a simple animation showing which user bases are sending messages to which data centers.

A simulation can be cancelled before the completion of the run, using the cancel button at the bottom right hand corner. It may take a while after clicking the cancel button for the simulation to halt as it will continue to gather the simulation data of the requests that had been generated before cancelling but had not been completed.

Figure 12. Simulation Panel During a Simulation

Once the simulation is completed the main response times will be displayed on the simulation panel next to each user base. The detailed results can be viewed by clicking the "View Detailed Results" button that appears at the right hand bottom corner of the screen after the simulation has completed.

The results screen will list out the data collected from the simulation. This includes:

1. Overall response time summary (for all the user bases)
2. Response time by user base in tabular format
3. Response time by user base in graphical format broken down into the 24 hours of the day
4. Request servicing time by each data center in tabular format
5. Request servicing time by data center in graphical format broken down into 24 hours of the day
6. Data center loading (number of requests serviced) in graphical format broken down in to 24 hours of the day
7. Cost details

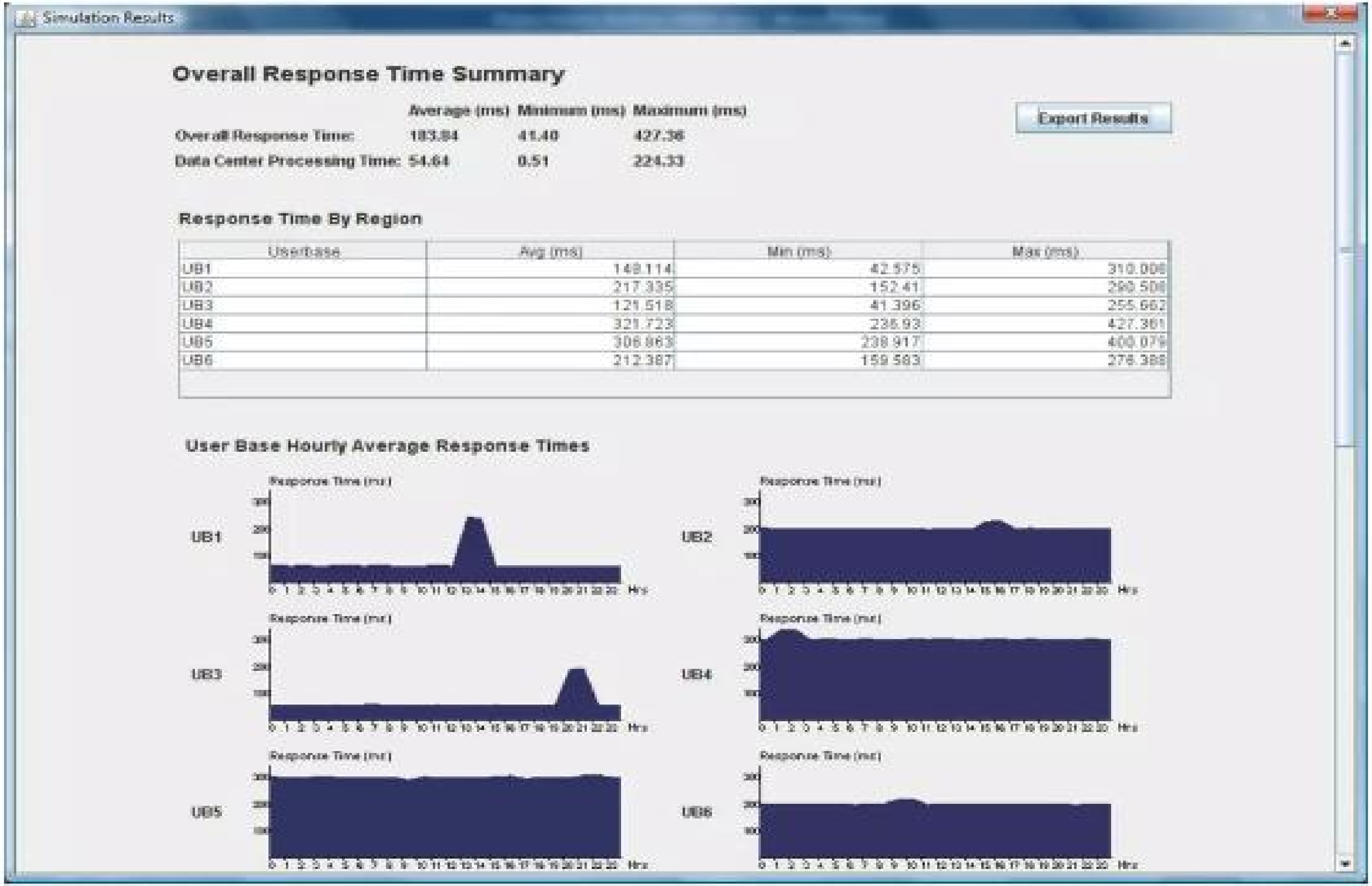


Figure 13. Results Screen

3.2.4 User Base

A User Base models a group of users that is considered as a single unit in the simulation and its main responsibility is to generate traffic for the simulation.

A single User Base may represent thousands of users but is configured as a single unit and the traffic generated in simultaneous bursts representative of the size of the user base. The modeller may choose to use a User Base to represent a single user, but ideally a User Base should be used to represent a larger number of users for the efficiency of simulation.

3.2.5 InternetCloudlet

An InternetCloudlet is a grouping of user requests. The number of requests bundled into a single InternetCloudlet is configurable in CloudAnalyst. The InternetCloudlet carries information such as the size of a request execution command, size of input and output files, the originator and target application id used for routing by the Internet and the number of requests.

3.2.6 Data Center Controller

The Data Center Controller is probably the most important entity in the CloudAnalyst. A single Data Center Controller is mapped to a single cloudsim.DataCenter object and manages the data center management activities such as VM creation and destruction and does the routing of user requests received from User Bases via the Internet to the VMs. It can also be viewed as the façade used by CloudAnalyst to access the heart of CloudSim toolkit functionality.

3.2.7 VmLoadBalancer

The Data Center Controller uses a VmLoadBalancer to determine which VM should be assigned the next Cloudlet for processing. Currently there are three VmLoadBalancers implementing three load balancing policies which can be selected as required by the modeller.

1. Round-robin Load Balancer – uses a simple round-robin algorithm to allocate VMs
2. Active Monitoring Load Balancer – this version load balances the tasks among available VM's in a way to even out the number of active tasks on each VM at any given time.
3. Throttled Load Balancer – this ensures only a pre-defined number of Internet Cloudlets are allocated to a single VM at any given time. If more request groups are present than

3.4.2 Calculating the Data Transmission Delay

The data transmission delay is calculated using the following formula:

$$T_{total} = T_{latency} + T_{transfer}$$

Where $T_{latency}$ is the network latency and $T_{transfer}$ is the time taken to transfer the size of data of a single request (D) from source location to destination. $T_{latency}$ is taken from the latency matrix (after applying Poisson distribution on it for distributing it) held in the InternetCharacteristics.

$$T_{transfer} = D \,/\, Bw_{peruser}$$

Where $Bw_{peruser} = Bw_{total} \,/\, N_r$

Bw_{total} is the total available bandwidth (held in the InternetCharacteristics) and N_r is the number of user requests currently in transmission. The InternetCharacteristics also keeps track of the number of user requests in-flight between two regions for the value of N_r.

3.5.1 VM Load Balancing Algorithms

As mentioned in section 3.2.7 The data centers use the VMLoadBalancer to load balance requests between the available virtual machines, and in the current version there are 3 variants. The first – round-robin is straight forward and is not explained further here. The algorithms used in the other two are explained below.

3.5.1.1 Throttled Load Balancer

Following is the algorithm used by the ThrottledVmLoadBalancer.

1. ThrottledVmLoadBalancer maintains an index table of VMs and the state of the VM (BUSY/AVAILABLE). At the start all VM's are available.
2. DataCenterController receives a new request.
3. DataCenterController queries the ThrottledVmLoadBalancer for the next allocation.
4. ThrottledVmLoadBalancer parses the allocation table from top until the first available VM is found or the table is parsed completely.

   If found:

   a. The ThrottledVmLoadBalancer returns the VM id to the DataCenterController
   b. The DataCenterController sends the request to the VM identified by that id.
   c. DataCenterController notifies the ThrottledVmLoadBalancer of the new allocation
   d. ThrottledVmLoadBalancer updates the allocation table accordingly

   If not found:

   e. The ThrottledVmLoadBalancer returns -1.
   f. The DataCenterController queues the request
5. When the VM finishes processing the request, and the DataCenerController receives the response cloudlet, it notifies the ThrottledVmLoadBalancer of the VM de-allocation.
6. The DataCenerController checks if there are any waiting requests in the queue. If there are, it continues from step 3.
7. Continue from step 2.

Task Schedulling

- The scheduling of tasks in cloud means choose the best suitable resource available for execution of tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible.

- In scheduling algorithms list of tasks is created by giving priority to each and every tasks where setting of priority to different different tasks can be based on various parameters. Tasks are then chooses according to their priorities and assigned to available processors and computer machines which satisfy a predefined objective function.

Types of Scheduling

- **Static scheduling** schedule tasks in known environment i.e. it already has the information about complete structure of tasks and mapping of resources before execution, estimates of task execution/running time.

- **Dynamic scheduling** must depend on not only the submitted tasks to cloud environment but also the current states of system and computer machines to make scheduling decision.

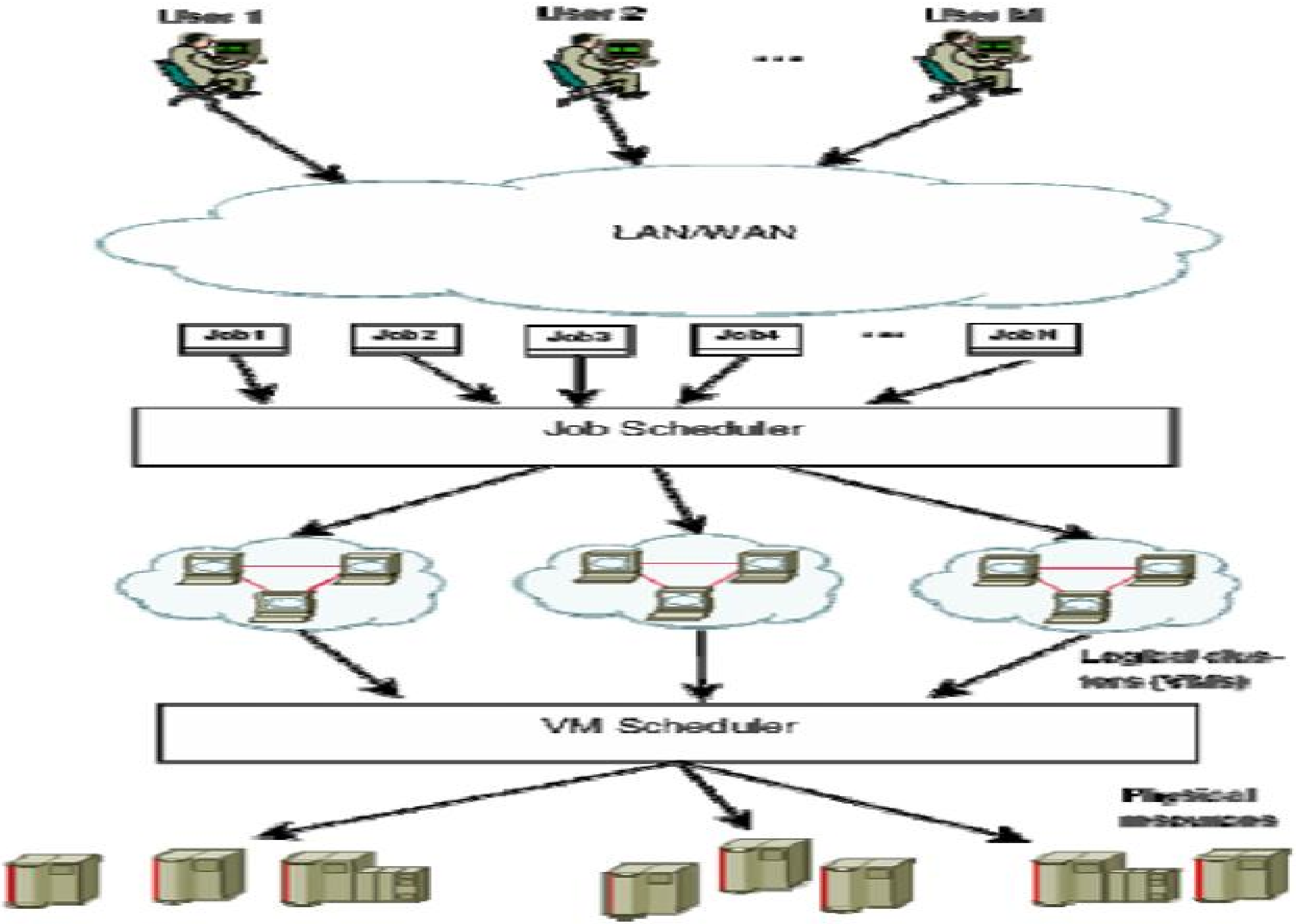


Fig.3. General View of Task Scheduling