Cloud Computing: Overview & challenges

Aminata A. Garba

Outline

I. **Introduction**

II. **Virtualization**

III. **Resources Optimization**

VI. **Challenges**

A Historical Note

- 1960, the idea of organizing computation as a public utility like electricity was issued (John McCarthy)
- 1967, IBM introduced its first system with virtual memory (360/67)
- 1972, IBM released VM/370 system. VMs created for users and the VMM managed hardware resources multiplexing. Virtualization was motivated by the cost of hardware & the need to share hardware among users and applications
- 1980s & 1990s, the interest of virtualization dropped with a drop in hardware cost and the expansion of personal computers
- 2005, regain of interest in virtualization technology in research and commercial
- 2006, Amazon EC2 was initially released as a limited public cloud computing service
- 2008, Microsoft Windows Azure was announced & became commercially available in 2010
- 2011, iCloud was announced as a cloud storage and cloud computing services from Apple Inc. stores content such as music, photos allows access from Apple devices
- 2012, the Oracle Cloud was announced

Definitions

- Several definitions in the literature
- Cloud Computing [1] (NIST)
- 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

Cloud Features

- **On-demand self-service**
  - A consumer can unilaterally provision computing capabilities (server time, storage) as needed
  - No required human interaction with service provider

- **Broad network access**
  - Capabilities are available over the network
  - Accessed through standard mechanisms
  - Promote use by heterogeneous client platforms (e.g.,mobile phones, laptops)

- **Resource pooling**
  - The provider's computing resources are pooled to serve multiple consumers
  - Different physical and virtual resources dynamically assigned and reassigned according to consumer demand
  - Customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter)
  - Examples of resources include storage, processing, memory, network bandwidth, and virtual machines

Cloud Features (cont.)

- **Rapid elasticity**
  - Capabilities can be rapidly and elastically provisioned, to quickly scale out and rapidly released, to quickly scale in
  - Capabilities available for provisioning often appear to the consumer to be unlimited and can be purchased in any quantity at any time

- **Measured service**
  - Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g.,storage, processing, bandwidth)
  - Resource usage can be monitored, controlled & reported, providing transparency for both the provider and consumer of the utilized service

Deployment Models

- **Private cloud**
  - Infrastructure operated solely for an organization
  - Managed by the organization or a third party
  - May exist on premise or off premise

- **Community cloud**
  - Infrastructure shared by several organizations
  - Supports a specific community with shared concerns (e.g., mission, security requirements, policy, compliance considerations)
  - Managed by the organizations or a third party
  - May exist on premise or off premise

- **Public cloud**
  - Infrastructure is made available to the general public or a large industry group
  - Owned by an organization selling cloud services

- **Hybrid cloud**
  - A composition of two or more clouds (private, community or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability

Service Models

- Classified according to
  - Abstraction levels
  - Applications
- Models
  - **Software as a Service (SaaS)**
  - **Platform as a Service (PaaS)**
  - **Infrastructure as a Service (IaaS)**

Software as a Service (SaaS)

- Applications running on the cloud infrastructure belong to the provider
- Consumer can use the provider's applications running on the cloud infrastructure
- The consumer does not manage or control cloud infrastructure (network, servers, storage) or applications
- The applications are accessible from various client devices through a client interface such as a web browser (e.g., web-based email)
- Services are deployed and configured for user
- Possibly a limited user-specific configuration settings
  - Microsoft Office Suites
  - IBM iNotes
  - Google Docs
  - Photo Galleries (Yahoo Flickr, Google Picassa)

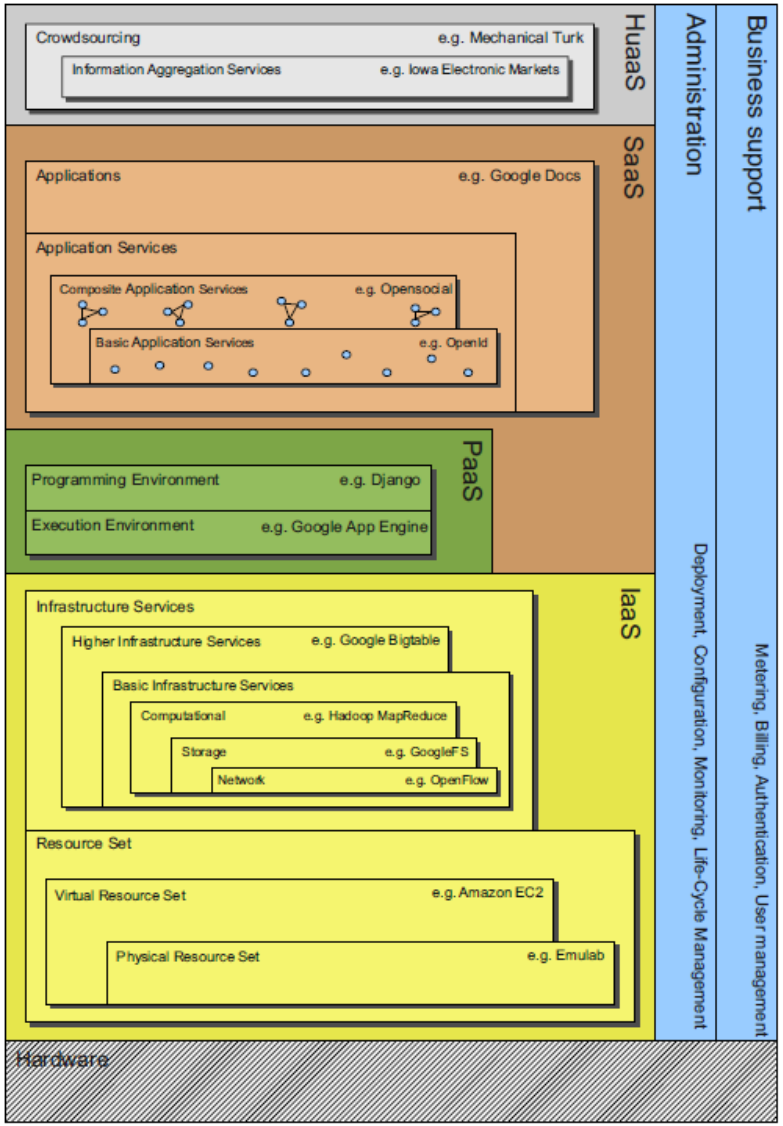
Platform as a Service (PaaS)

- User can deploy applications using the provider cloud infrastructure
- Application are created using programming languages and tools supported by the provider's platform
- Consumer does not manage or control the cloud infrastructure (network, servers, storage) or software (OS and other software)
- Consumer has control over the deployed applications and possibly application hosting environment configurations
- User are limited to programming languages defined by the cloud provider
- Google App Engine
  - Developing and hosting Web applications
  - Limited to a number of applications: e.g., Python and Java API for the implementation of web applications
- Microsoft's Windows Azure
  - Data-intensive applications
- IBM Blue Cloud

Infrastructure as a Service (IaaS)

- Users can provision computing, storage and networking resources
- Users can deploy and run arbitrary software, which can include operating systems and applications
- Consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications
- Possibly limited control of select networking components (e.g., host firewalls)
- **Amazon EC2**
  - Users can create and use virtual machines (instances)
  - Instances are classified according to memory, CPU, storage, etc.
- **Amazon Simple Storage Service (S3)**
  - Public storage web service
  - Data is stored on multiple devices
  - Store for objects of variable sizes
  - Clients can read and update objects remotely using a web services interface

Architecture of Cloud Computing (Cloud Stack) [2]



Benefits of Clouds

- **Users**
  - Pay as you go model: users pay for needed resources (bandwidth, storage, processing)
  - Less operational & less/no infrastructure deployment and maintenance costs for users
  - Companies can focus on their main activities
  - High available throughput
  - Easier processing of large amount of data
- **Providers**
  - Economy of scale
  - Efficiency
  - Big revenues
- **Nation wide**
  - Accelerate nation economy
  - Create jobs and new revenues

Reality Check

- Capacity is limited
- Cloud providers promise more to customers than their capacity
- Reliability/Availability: several cloud services failover [6]
  - Microsoft Azure: 22h outage 13-14 March 2008
  - Google search outage (Programming error) 40 min, 31 Jan 2009
  - Gmail and Google APP Engine: 2h hours February 24 2009
  - S3 outage 5h (17 June 2008) and 6-8h (July 2008)
- Slow/down network
  - Cloud applications can not run

Service Level Agreement

- **Agreement between the provider and the consumer**
  - Duration of the agreement
  - Availability of the resources (probabilistic)
  - Amount of resources
    - Limits to the resource requirements
    - According to workload fluctuations resources may be added
  - Service Performance
  - Penalties
    - Provider: if resources are unavailable/credit for user
    - User: for using additional resources (may not be supported in case of high resources demand)

Requirements

- **Infrastructure**
  - Servers/clusters
  - Storage
  - Network (inside the cloud and WAN connecting data centers & users)
  - Power & cooling systems

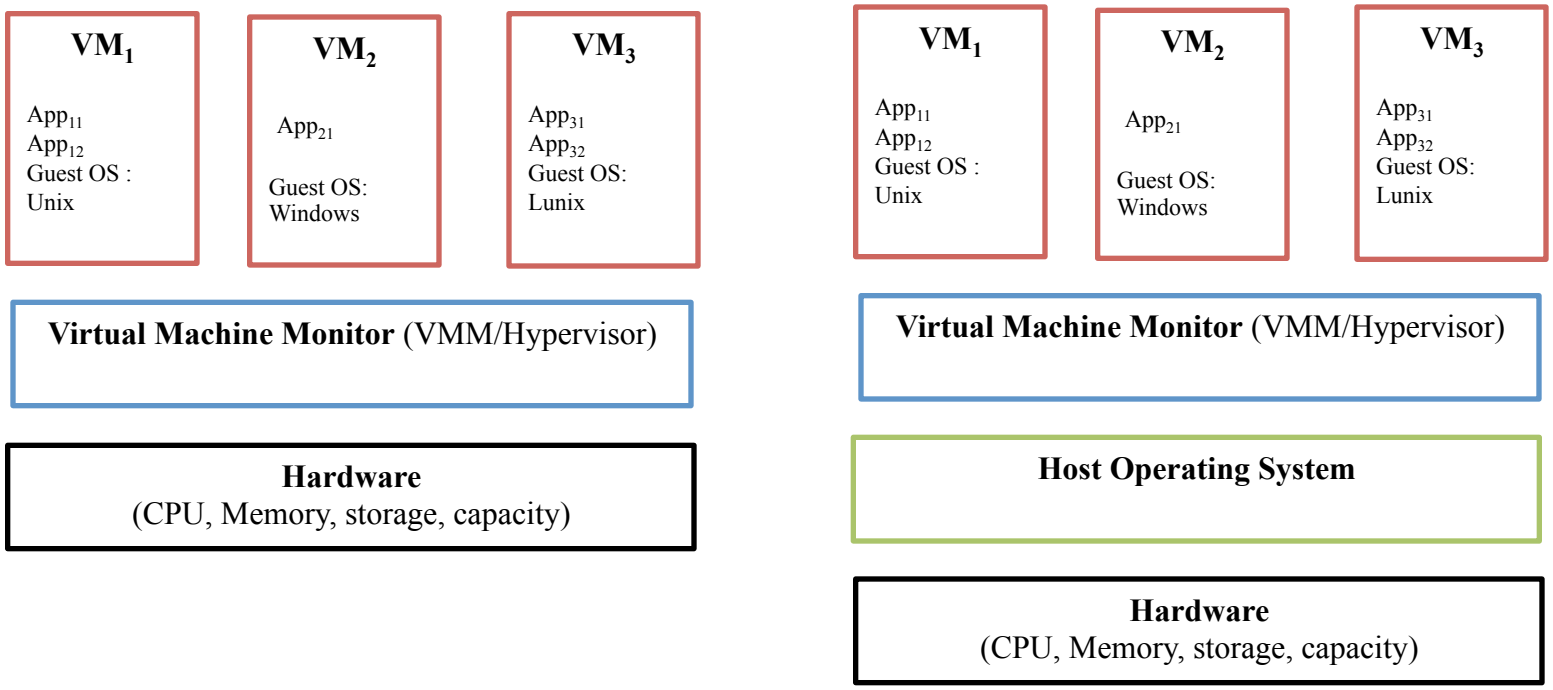
- **Tools**
  - Virtualization
    - Virtual Machines (VMs)
    - Virtual Machines Monitors (VMM)
  - Resource Management Tools

- **Consumers**
  - Platform for communication with users
  - Service Level Agreement
  - Billing system (for public clouds)

II. Virtualization

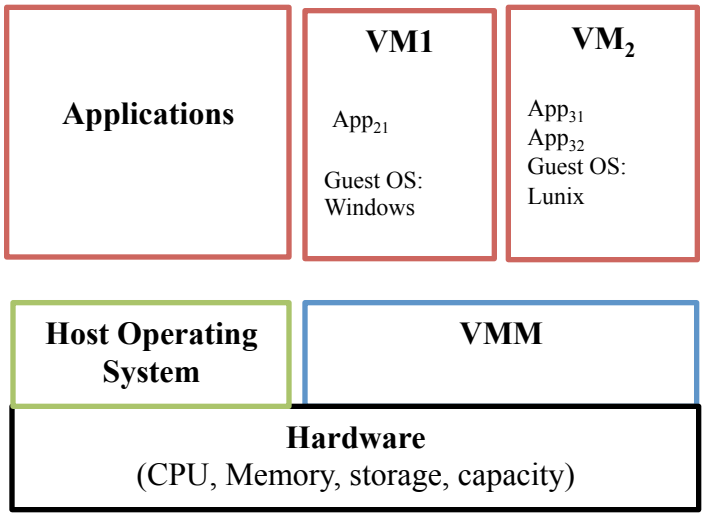
- **System virtualization**
  - Abstraction obtained using a software layer, Virtual Machine Monitor (VMM) or hypervisor, to partition the physical resources into virtual machines (VM)
  - VM (possibly) hosting different operating systems (guest OS) can run on the same physical machine
  - Hardware seen as a pool of resources to be shared by the VMs (Processor, Memory, Hard disk, Network)
- **Types of system virtualization**
  - **Full virtualization**
    - VMM on top of hardware
    - Guest OS unmodified
  - **Hosted virtualization**
    - Hypervisor is installed as an application on top of existing OS
    - Simplicity of installation
    - Increased overhead
  - **Hybrid**
    - VMM and the host operating system share the hardware

Examples



- A physical machine hosting 3 VMs with distinct OS, software & apps (full and hosted virtualization)

Examples (cont.)



- Hybrid system

Characteristics of Virtualization

- **Benefits**
  - **Isolation**
    - Hardware: distinct applications (different users) run simultaneously on the physical machine
    - Isolation from other VM
      - Reduce impact of failure to VM
      - Increase security
  - **Flexibility**
    - Add or reduce hardware resources
  - **Mobility**
    - Move virtual machine between different physical hosts

- **Disadvantages**
  - Increased overhead
  - Increased Processing
  - Reduced Performance

Virtual Machines Monitor (VMM)

- Also referred to as Hypervisor
- Software layer that partitions a server into virtual machines
  - VM can run existing software on the physical machine
  - Provides isolation between VM and from hardware
  - Multiplex virtual machines on a single hardware platform
  - Manage VM access to the physical resources
  - Migrate virtual machines across machines
    - Load balancing
    - Hardware failures
    - System scaling
    - Consolidate VM with low resources usage onto a single computer

Examples of VMM

- **Kernel-based virtual machine (KVM)**
  - Full virtualization
  - Supported by several platforms
  - Support unmodified guest operating systems (versions of Windows, Linux, UNIX)
  - Open source
- **VMWare**
  - Startup from Stanford University
  - Full, hybrid virtualization
  - Commercial
- **Xen hypervisor**
  - Started at Cambridge University
  - Paravirtualization
  - Open-source product used by other virtualization products

VMM Challenges

- VMM must be able to export a hardware interface to the software in a VM
  - Maintains control of the machine and retain the ability to interpose on hardware access
- VMM has to be compatible with the system (to run legacy software)
- Performance
  - Virtualization overhead
  - Run the VM at the same performance (speed) as the software would run on the real machine
- VMM failure will affect all the virtual machines running on the computer
- A security failure occurrence in the VMM will impact all the VMs

III. Resource Optimization

- **Optimize resource utilization**
  - CPU
  - Memory usage
  - Capacity/bandwidth
  - Storage
  - Energy consumption

- **Constraints**
  - Available resources
  - QoS: flexibility, latency, security, reliability, backup /replication, elasticity
  - Service Agreement Level (capacity, performance)
    - Priorities
    - Weights
  - Cost

$$\min f\downarrow \vdash \blacksquare Total\ available\ resource\ QoSusers\ SLACost \boxed{A} \quad (Memo$$

Challenges

- **Modeling issue**
  - Modeling Memory, CPU, capacity, storage, cost, SLA, QoS is difficult
- **Complexity/Difficulty**
  - Complex numerical (computations) problem
  - Heavy processing
  - Function of allocated resources is non linear with multiple unknowns
  - Hard analytical analysis and optimization
  - NP hard problem
- Tradeoffs
  - between complexity and optimized solution
  - It is usually difficult to consider multiple resource simultaneously
  - Identify a single resource with the highest impact on the resource allocation
  - Scheduling and capacity allocation algorithms usually consider one argument or heuristic models

Resource Management

- Admission control
  - Resource availability
  - SLA
- Resource allocation
  - Resource provisioning
  - Scheduling and Routing
- Resource Management (QoS & SLA Compliance)
  - VM migration
  - Dynamic scaling
  - Replication

Admission Control

- **Admission control**
  - Accommodate new users
  - Provide required capacity with a given (usually high) probability
  - Select requests to be admitted when the server is overloaded
  - Police the incoming requests
- **Request-based admission control**
  - Reject new requests if the servers are running to their capacity
  - For a session with multiple requests, some requests may be admitted and other rejected
- **Session-based admission control**
  - Reject new sessions
  - Once a session is admitted, all future requests belonging to that session are admitted
- **SLA strategies**
  - Users are admitted according to their SLA
  - High-priority vs low priority user
- **QoS-aware admission**
  - Control Depends on the QoS
  - High capacity user may not be admitted during overload situations
- **Policy-based admission control**
  - Based on a given policy

Resource Allocation: Provisioning

- Allocate hardware resources to applications

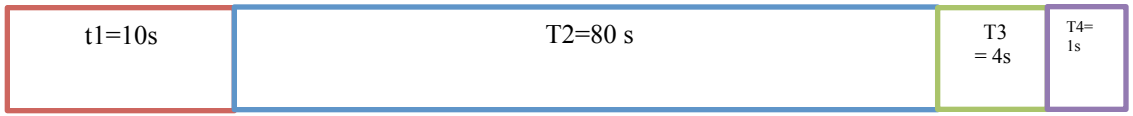
- **Provisioning Goals**
  - Elasticity
  - Scalability

- **Challenges**
  - Under-provisioning: SLA is not satisfied
  - Over-provisioning: resource utilization is not optimized

Scheduling

- **Problem formulation**
  - Matching tasks to machines (VM)
    - Throughput: number of tasks completed
    - Turnaround time: time to complete a task
    - Response time: time to access resources
- **Methods**
  - Optimal solution is a hard problem
  - Heuristic methods
  - Several algorithms are proposed in the literature based on some criteria
- **Dynamic vs Static**
  - Static: tasks are known prior to execution
  - Dynamic: scheduling is performed when the task arrives
- **Preemptive and non-preemptive**
  - A new task can interrupt execution of an old task
- **Time Constrained Scheduling**
  - A deadline is set
  - Priority can apply

Examples

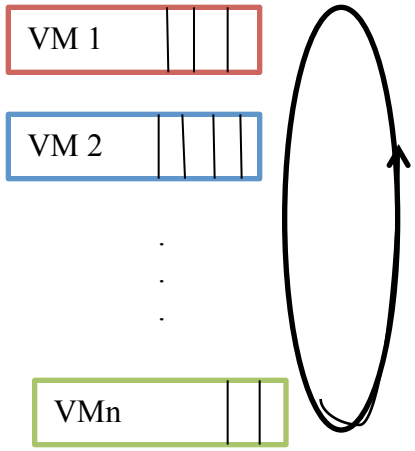


FIFO : Average waiting time (turnaround time) : (10+90+94+95)/3=96s

Min-Min : Average turnaround time : (1+ 5 + 15+95)/3=38s

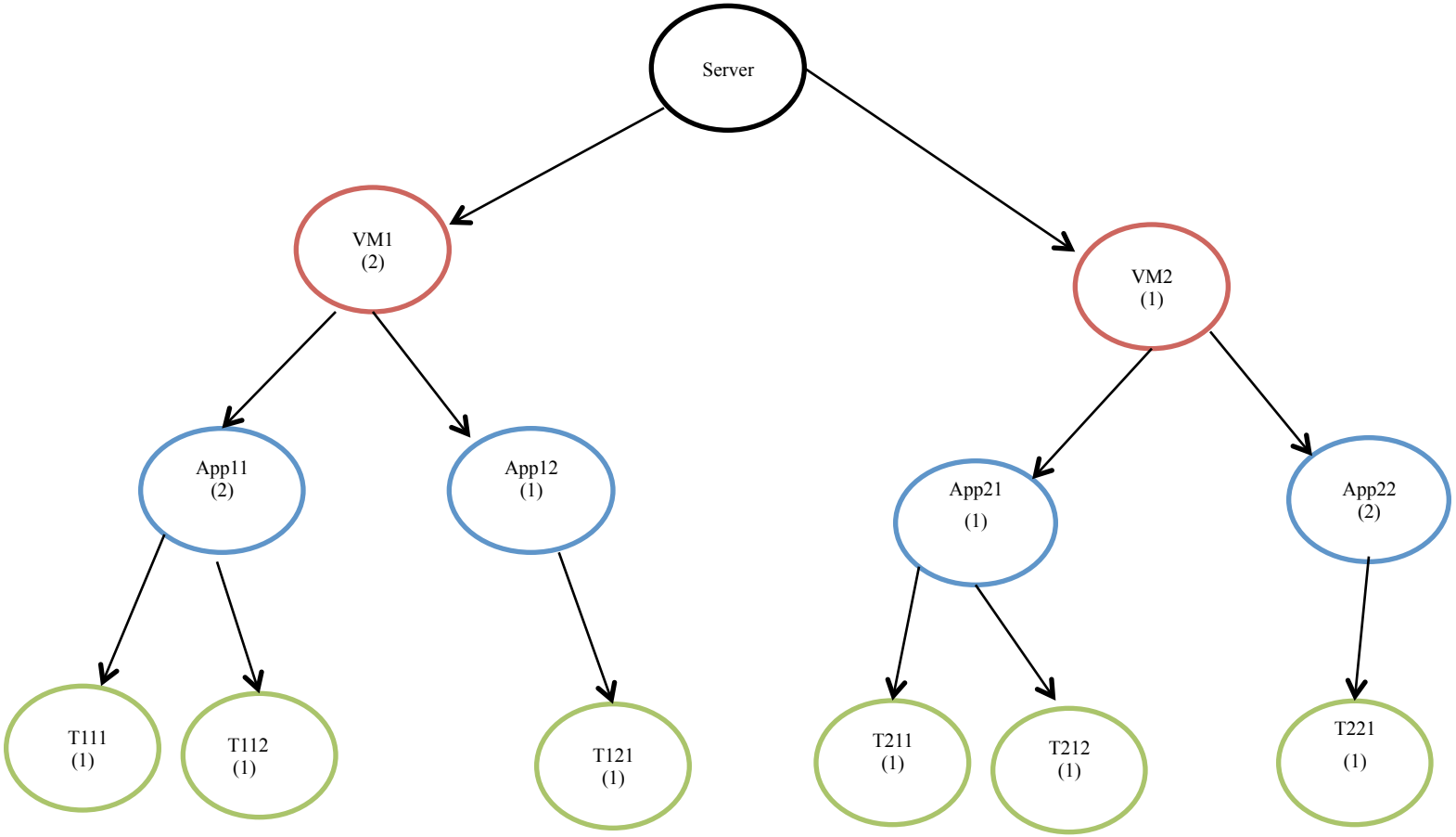
Max-Min : Average turnaround time : (80+ 90 + 94+95)/3=119s

Examples

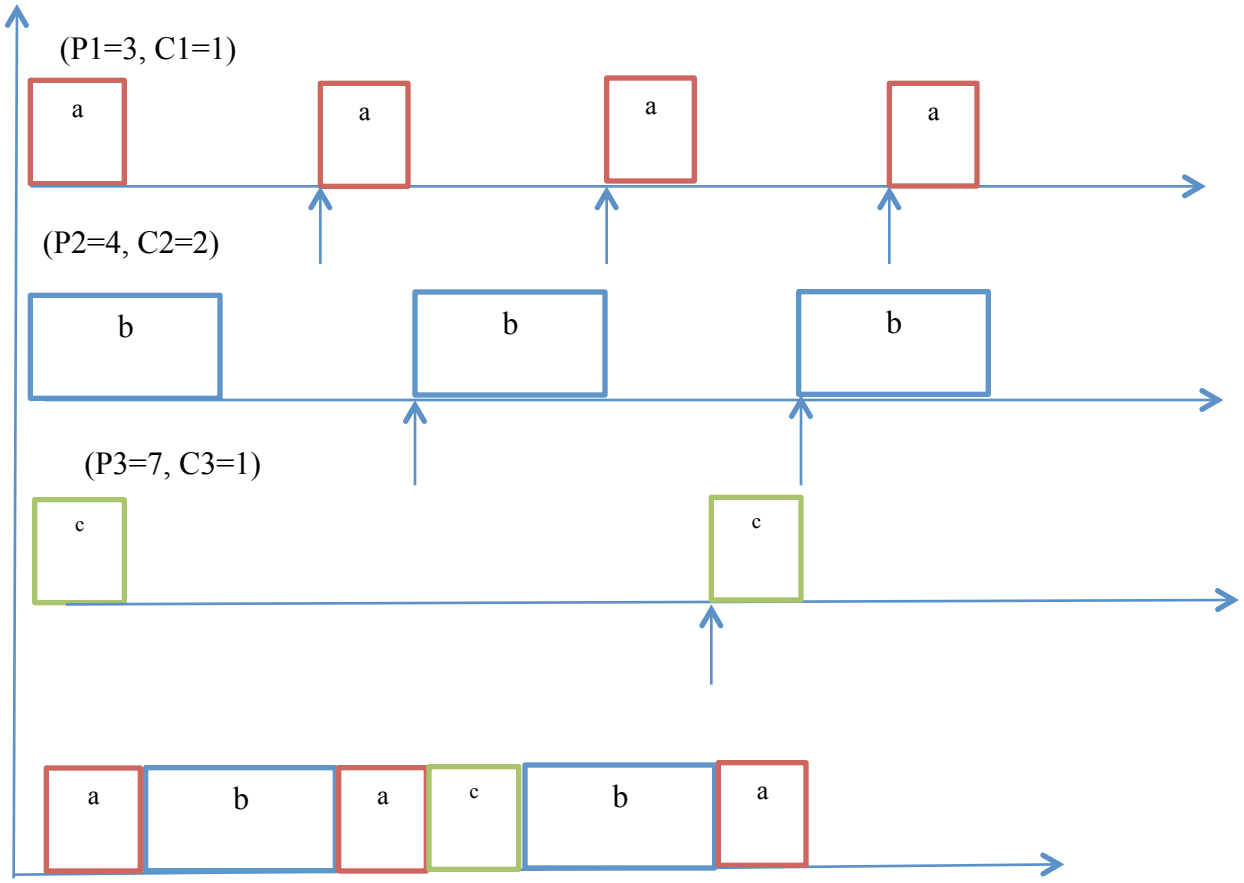


Fair Queuing

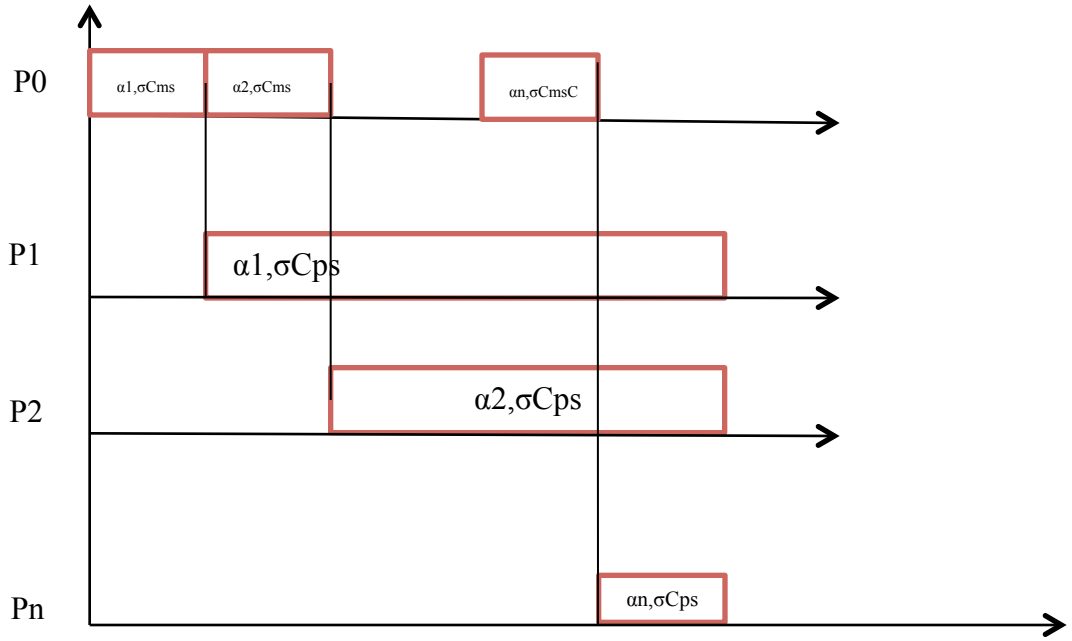
Start-Time Fair Queuing



Earliest Deadline First (EDF)



Optimal Partitioning Rule



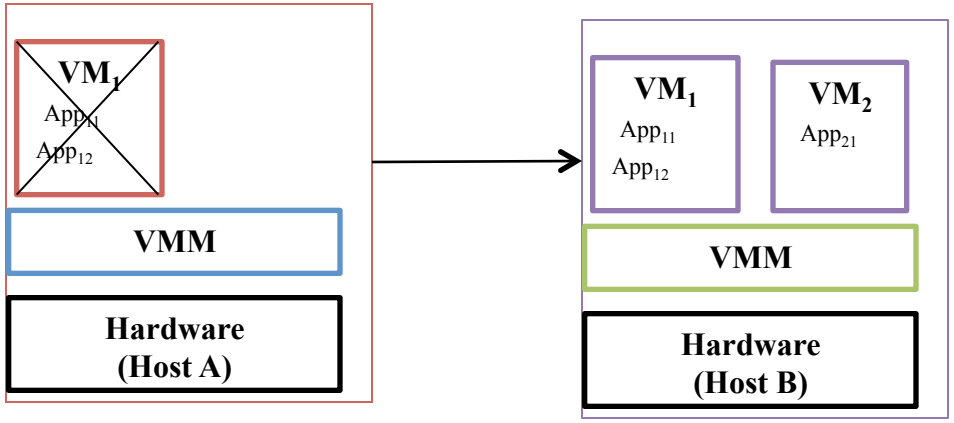
- αi: data fraction allocated to the ith node
- αiσ: amount of data to be transmitted to the ith node for processing, $0 < \alpha i \leq 1$, sum (αi)=1
- C_{ms}: Cost of transmitting a unit workload
- C_{ps}: Cost of processing a unit workload

- [(Lin et. al)]
- Workload is partitioned to ensure the earliest possible completion time
- Tasks are required to complete at the same time
- Based on divisible load theory (DLT), which states that the optimal execution time is obtained when all nodes allocated to the task complete their computation at the same time
- The head node distributes the data to worker nodes

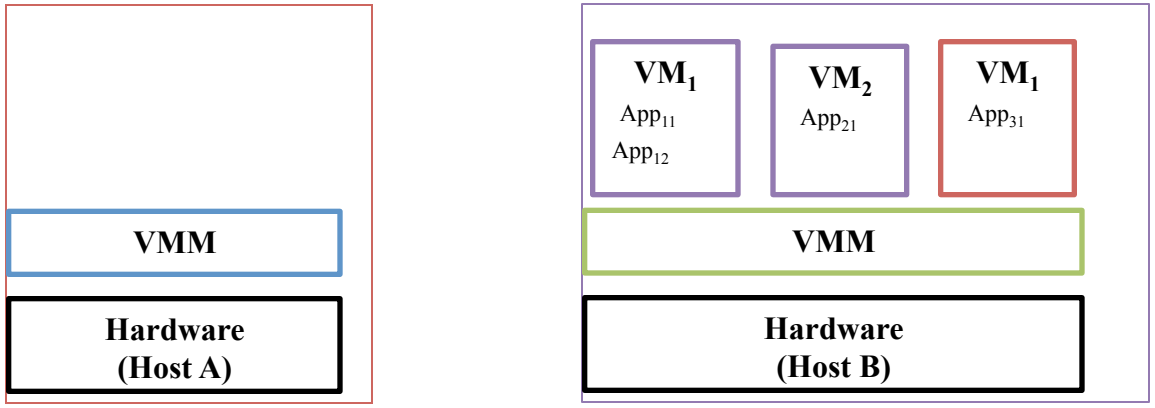
VM Migration

- Moving a VM from one physical host to another
  - VM running (live migration)
  - No impact on other running VMs
- Objective
  - Reduce impact of hardware failure
  - Load balancing
  - Power management (server consolidation)
  - Failure or Network maintenance
- Features
  - Minimize the migration time
  - Minimize the time during which services are not available (downtime)

Example

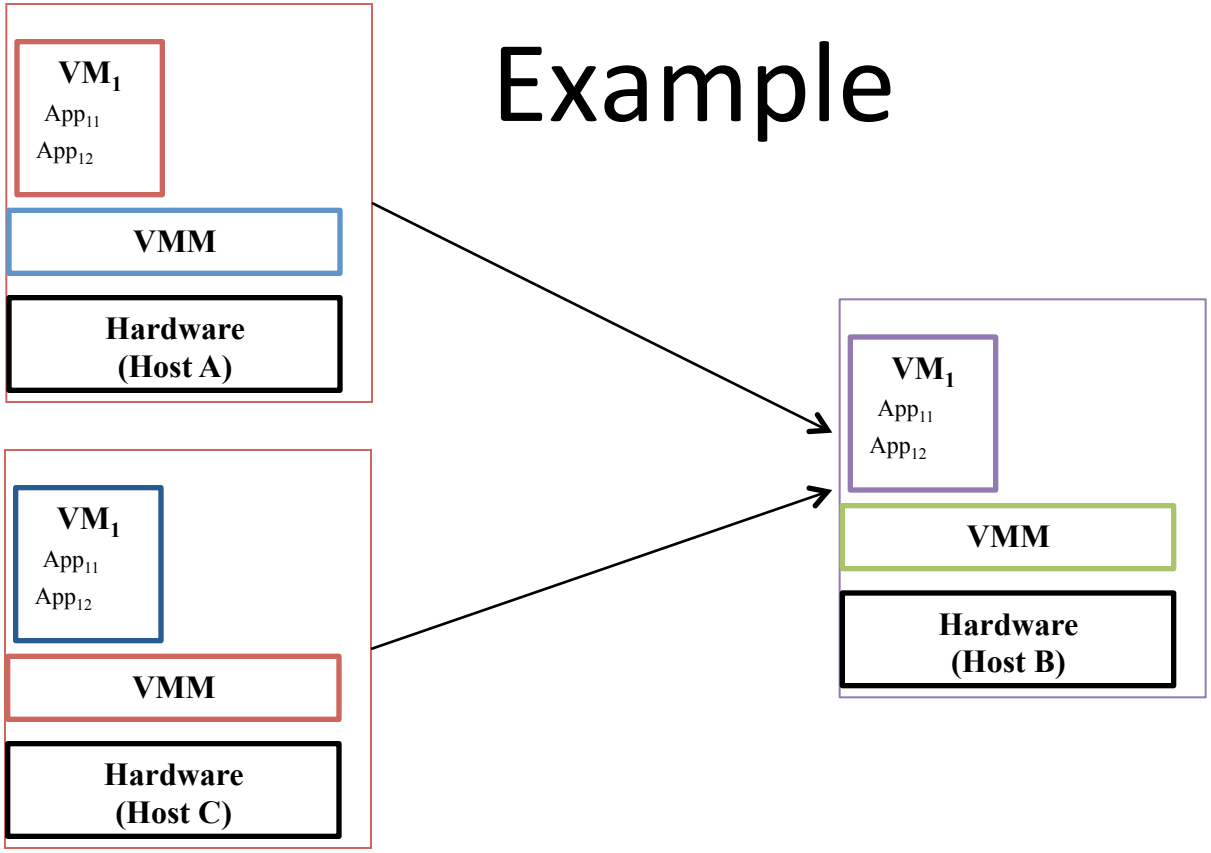


Before Migration

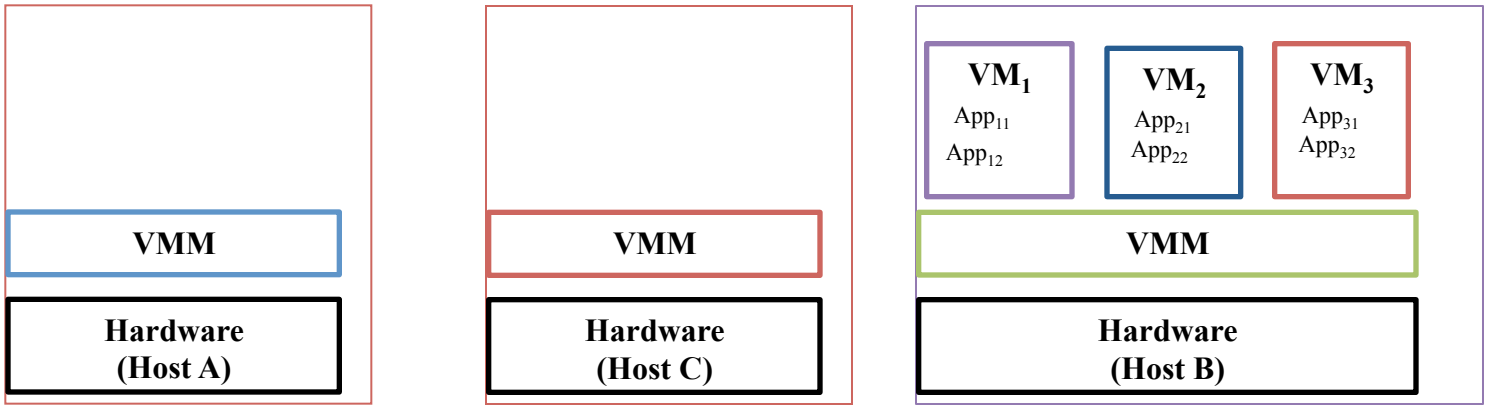


After Migration

Example

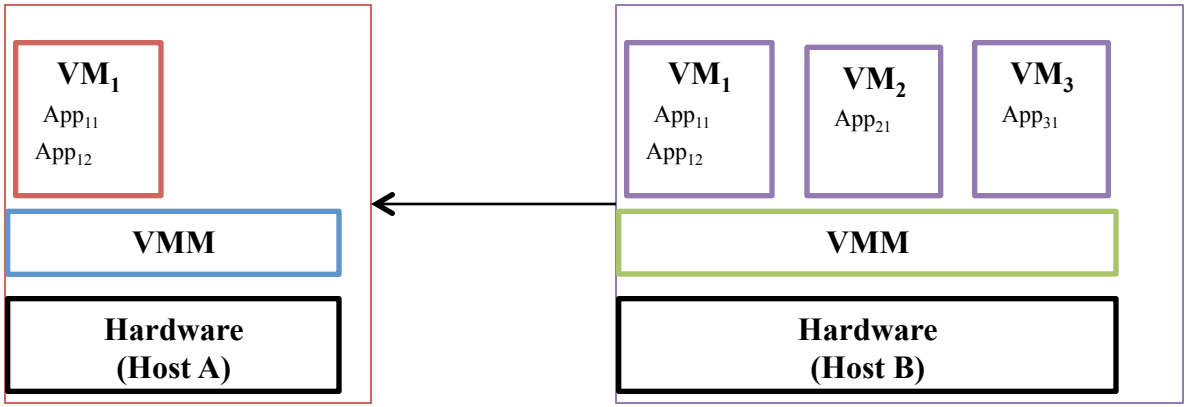


Before Migration

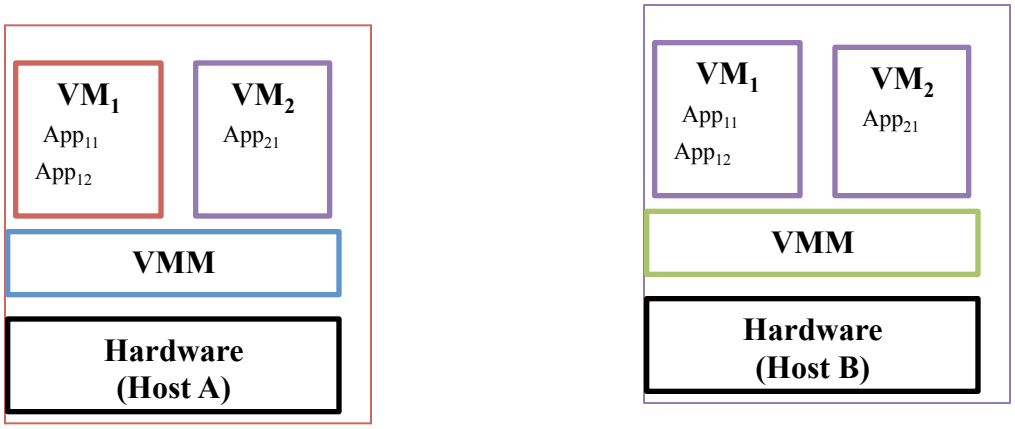


After Migration

Example



Before Migration



After Migration

IV. Cloud Computing Challenges

- Scalability/Elasticity
- Performance & Fault Tolerance
- Security/ Confidentiality
  - Data location
  - Data security
  - Data loss
  - Data management by a third party
  - Data access through Internet
- Resource allocation
- Cost optimization
  - Energy cost
  - Networking/bandwidth
- Performance, QoS, Service Level Agreement (SLA)
- Portability

Security Issues

- Management delegation
  - Responsibilities (agreed on the SLA)
  - Data location
  - Identity and consumer interface management
  - Compliance of the cloud provider
  - Data portability

- Data security
  - Instance security
  - Malicious attack
  - Failure to isolate from attacked VM
  - Hypervisor attack

- Network security
  - Encryption of data packets
  - VPN

Specific Challenges in Africa

- Energy (power and cooling)
- Networking
- Data protection
- How can cloud computing be viable in Africa?
- What are the alternatives for cloud computing development in Africa?

Summary

- Introduction to Cloud Computing
- Virtualization
- Resources Optimization
- Resources Management Strategies
- Resource Management Tools
  - Virtual infrastructure management
  - Data management tools
  - Computational tools
- Resource Management Challenges

References

1. P. Mell and T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Information Technology Laboratory, Technical Report , 2011.
2. A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape," ICSE Workshop on Software Engineering Challenges of Cloud Computing, pp. 23-31, 2009.
3. Mendel Rosenblum, Tal Garfinkel: "Virtual Machine Monitors: Current Technology and Future Trends", IEEE Computer Society, 2005
4. Sherif Sakr, Anna Liu, Daniel M. Batista, and Mohammad Alomari, A Survey of Large Scale Data Management Approaches in Cloud Environments, IEEE Communications Surveys & Tutorials, 2011.
5. B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, Virtual infrastructure management in private and hybrid clouds, IEEE Internet Computing Conference, September/October, 2009.
6. B.P. Rimal, E. Choi, I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", IEEE, 5th International Conference on INC, IMS and IDC, 2009.
7. F. Chang, J. Ren, and R. Viswanathan. Optimal resource allocation in clouds. IEEE Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, pages 418 –425, july 2010
8. C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines", Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI '05), 2005.
9. K. J. Duda and D. R. Cheriton. Borrowed-virtual-time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler, Proceedings of the 17th ACM SOSP, 1999.
10. http://aws.amazon.com/ec2/
11. http://www.microsoft.com/windowsazure/