Infrastructure as a Service (IaaS)

Roch Glitho, PhD

Full Professor

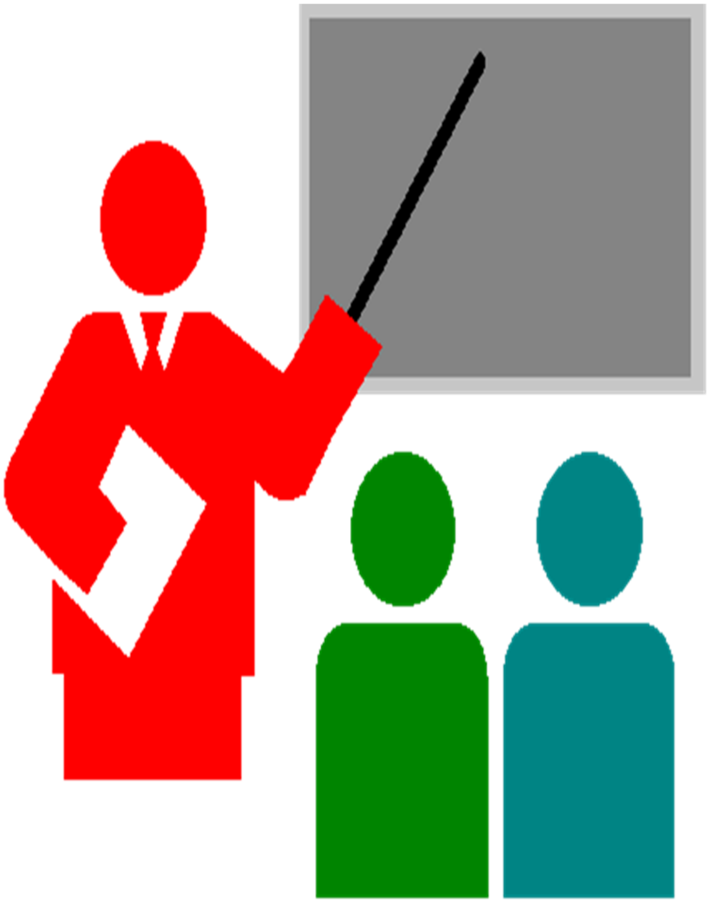
Ericsson / ENCQOR-5G Senior Industrial Research Chair

Cloud and Edge Computing for 5G and Beyond

My URL - http://users.encs.concordia.ca/~glitho

Concordia University
Engineering and Computer Science
Concordia Institute for Information Systems Engineering

IaaS

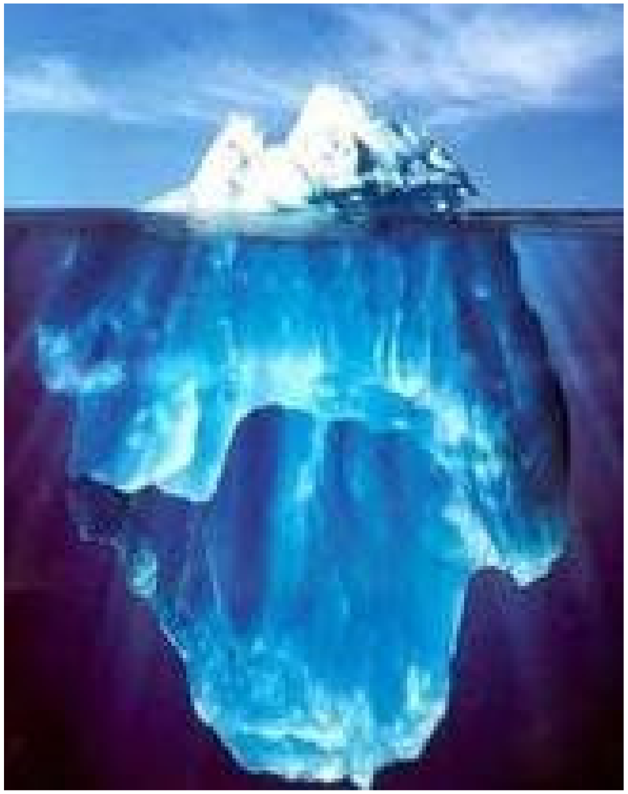


- Introduction
- Definitions and layers
- Challenges
- Resource Management
- Case Studies (XEN and Openstack)
- Serverless computing

Introduction

Introduction

Infrastructure as a Service (IaaS): immersed part II: End-user perspective)



Introduction

Infrastructure as a Service (IaaS): immersed part II: End-user perspective)

- Virtualized resources (CPU, memory, storage and eventually service substrates) used (on a pay per use basis) by applications
  - Examples
  - IBM Blue Cloud
  - Amazon EC2

Definitions and layered view

Definition

"An IaaS cloud enables on-demand provisioning of computational resources in the forms of virtual machines (VMs) deployed in a cloud provider data centres (such as Amazon's minimizing or even eliminating associated capital cost for cloud consumers .."

Reference 1

R. Moreno et al., Key Challenges in Cloud Computing: Enabling the Future Internet of Services, IEEE Internet Computing, July/August 2013

Note: A bit outdated since it does not include containers

Layering

(a) - Cloud consumers

Iaas consumers (e.g. PaaS, other clouds)

(b) - Cloud management layer

- Overall IaaS management
- Interface with cloud consumers

Layering

(c) - Virtual infrastructure management layer - Functionality includes:

- Provides uniform / homogeneous view of virtualized resources
  - Virtualization platform independent
- Handles VM life cycle
- Handle addition / failure of physical resources
- Server consolidation, high availability

(d) - Virtual machine management layer (i.e. hypervisors)

Layers

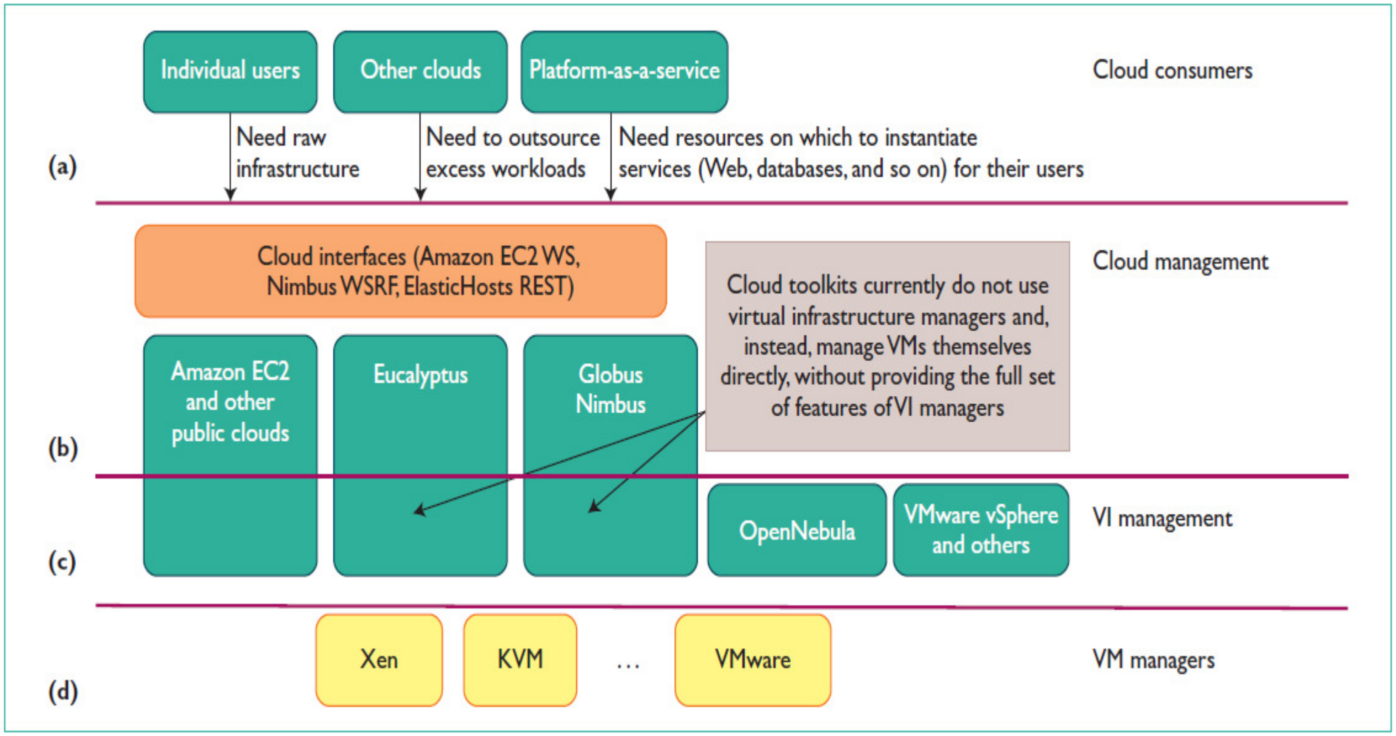


Figure 1. The cloud ecosystem for building private clouds. (a) Cloud consumers need flexible infrastructure on demand. (b) Cloud management provides remote and secure interfaces for creating, controlling, and monitoring virtualized resources on an infrastructure-as-a-service cloud. (c) Virtual infrastructure (VI) management provides primitives to schedule and manage VMs across multiple physical hosts. (d) VM managers provide simple primitives (start, stop, suspend) to manage VMs on a single host.

B. Sotomayor et al., Virtual Infrastructure Management in Private and Hybrid Clouds, IEEE Internet Computing, September/October 2009

Challenges

IaaS Challenges

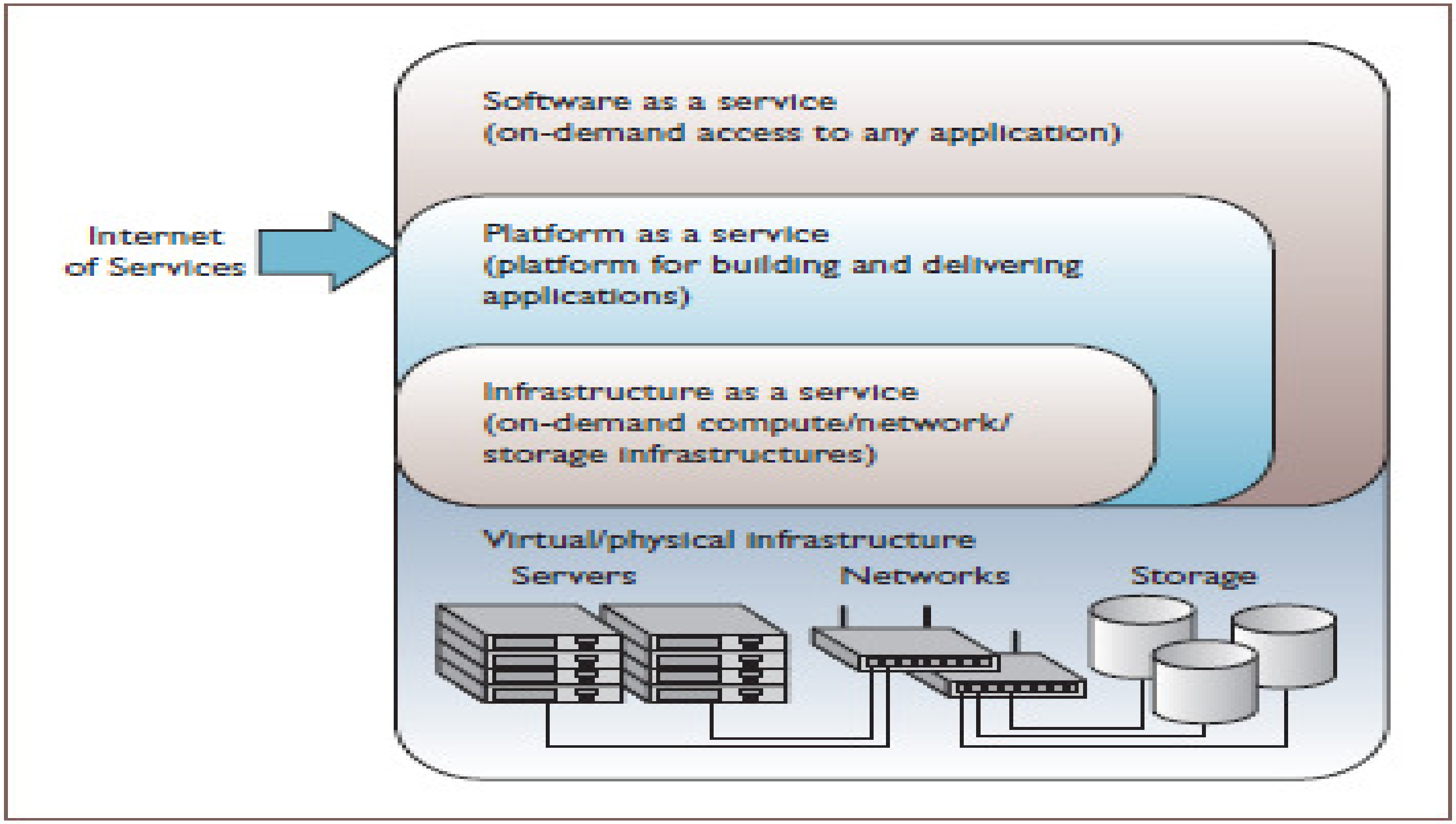


Figure 2. Cloud computing model for the future Internet of Services (IoS). Cloud technology enables the future IoS, and the infrastructure-as-a-service clouds represent this model's foundation.

R. Moreno et al., Key Challenges in Cloud Computing: Enabling the Future Internet of Services, IEEE Internet Computing, July/August 2013

Examples of IaaS services

- Computing
- Storage
- Networking
- (Virtual) sensor/robot that can carry out given task(s)

Single IaaS providers

- Dynamic service provisioning
  - Dynamic mapping of services onto resources via environment (e.g. virtualization platform, software library) independent interfaces
    - Require advanced SOA interfaces

Single IaaS providers

- QoS and SLA negotiation
  - Require expressive mechanisms for QoS and SLA negotiation
    - Note: There are several negotiation mechanisms (e.g. offer / response. Offer/counter offer)

Single IaaS providers

- Service scalability
- Service elasticity
- Service monitoring, billing and payment
- Context / situation awareness
  - Geographic restrictions
    - Service deployed close to a group of users
    - Legal restrictions

Aggregating IaaS providers

- Service deployment across different providers
  - Hybrid clouds
  - Cloud brokering
    - Allow consumer to select most appropriate IaaS providers
      - Publication / discovery

Aggregating IaaS providers

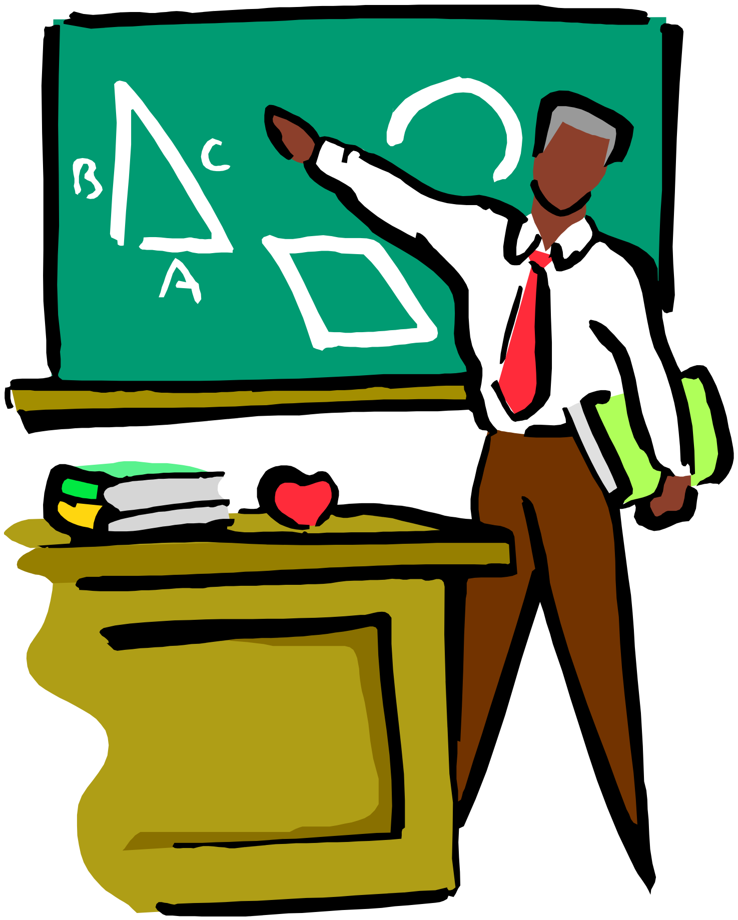
- Interoperability
- Portability

Non functional challenges

- Security
- Availability, reliability and resilience
- Energy efficiency

Resource Management

Resource Management



- 1 - The problem
- 2 – VM migration (One of the techniques for tackling the problem)
- 3 - On VM Migration based algorithms for IaaS

Resource management problem

- **On virtual machine life cycle**
  - Determine expected resources needed at deployment
  - Configure VM
  - Decide where to place it
  - Start VM
  - Track resource usage
    - Hot spot: Inadequate resource to meet performance
    - Cold spot: Overprovisioned resources

    Note: Resource usage pattern is highly dependent on applications types

Resource management problem

- **Overall problem**

  "Minimized resource utilization while meeting IaaS users performance requirements"

Resource management problem

- **How could it be solved?**
  - Server consolidation
    - Avoid cold spots on physical machines
  - Load balancing
    - Avoid discrepancy in resource usage on the different physical machines
  - Hot spot mitigation
    - Avoid conditions where a VM does not have enough resource to meet the performance requirements

Resource management problem

- **How could it be solved (Reference 4)?**

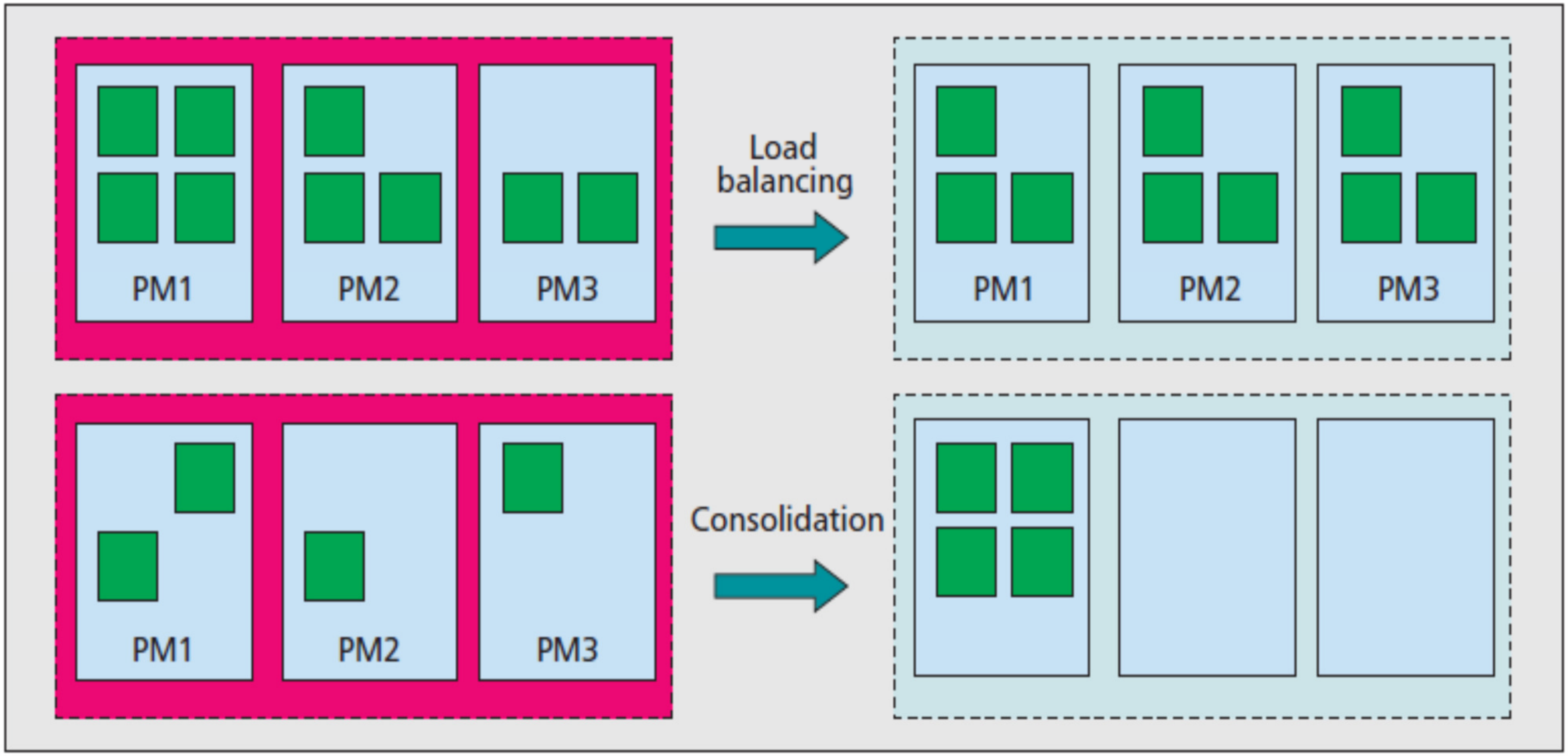


Figure 3. *Load balancing and consolidation scenarios.*

Resource management problem

- **How could it be solved?**
  - Complementary / non mutually exclusive techniques
    - VM reconfiguration
      - Add / reduce resources
    - New VM instantiations
    - VM migration

On Virtual Machine Migration

- VM Migration
  - Process of transferring a VM with its state from one physical machine to another
    - Program transfer from one machine to another machine is not new, e.g.
      - Mobile agents (Late 90s, early 200s): Program than can start execution in a physical machine, suspend execution, then move to another machine to resume execution, then move again if necessay

On Virtual Machine Migration

- VM Migration
  - Techniques
    - Suspend and copy
      - Suspend a VM execution, copy state (memory + processor), then resume execution on target machine
        - Downtime proportional to VM size and network bandwidth available for transfer

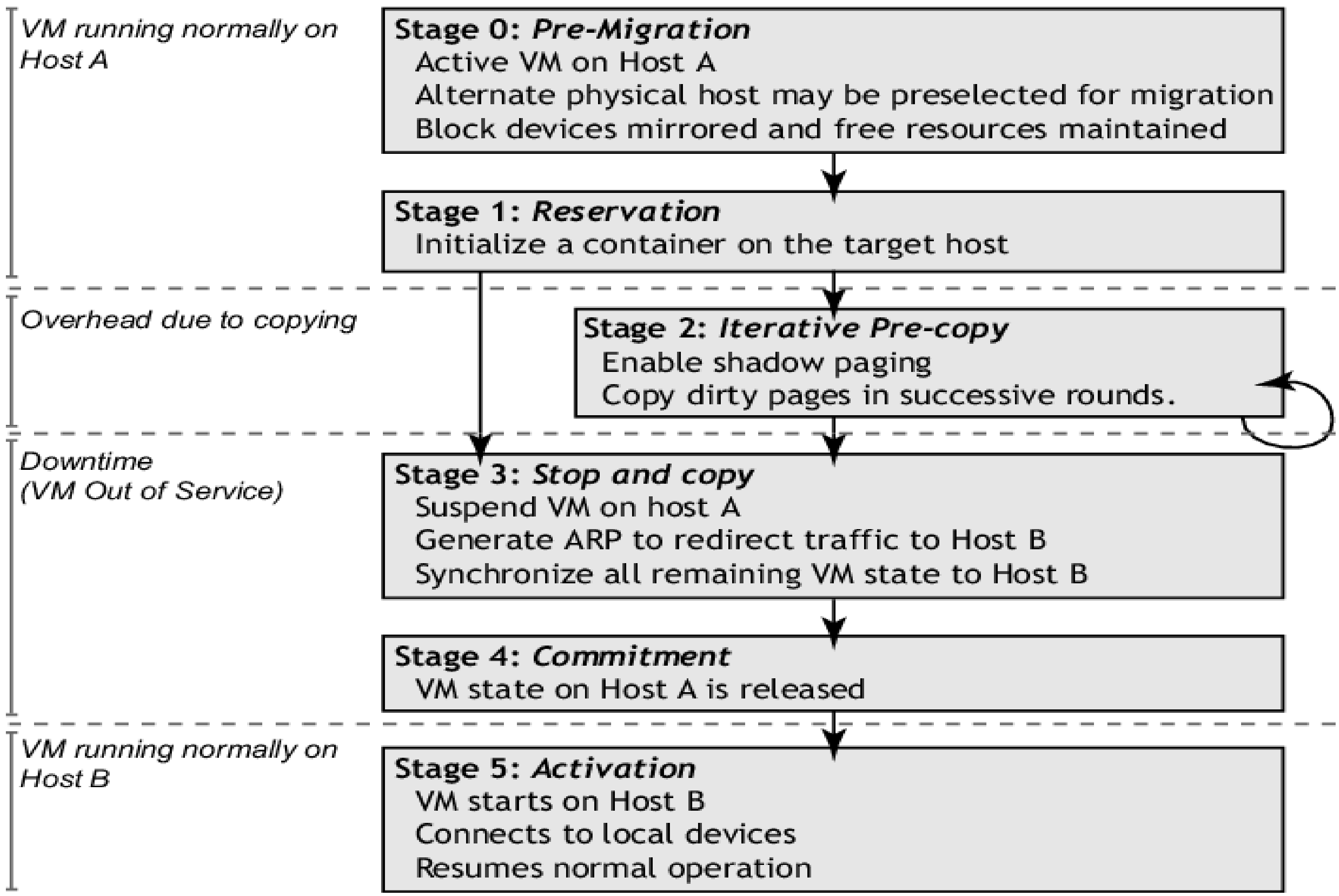
On Virtual Machine Migration

- VM Migration
  - Techniques
    - Live migration (Pre-copy and post – copy)
      - Minimize down time by either pre-copying or post copying state

On Virtual Machine Migration

- VM Migration
  - Techniques
    - Live migration (Pre-copy) –
      - Pre-copy iteratively memory state to a set threshold (or until a condition is met) while still executing on host machine
      - Execution is suspended, processor state and remaining memory state are copied, and VM is restarted on target machine

On Virtual Machine Migration (Ref.5)



Virtual Machine Migration Based Algorithms

- Goals
  - Consolidation
  - Load balancing
  - Hot spot mitigation

Virtual Machine Migration Based Algorithms

- Questions
  - When to migrate?
  - Which VM to migrate?
  - Where to migrate the VM?

Virtual Machine Migration Based Algorithms

- Examples of constraints
  - Migration process overhead
  - Impact on applications
  - Degree of improvements in intended goals

Virtual Machine Migration Based Algorithms

- When to migrate?
  - Periodically, e.g.
    - data centres in several time zones
  - Hot spot
  - Excessive spare capacity
    - under-utilized VM migrated to free servers
    - VM migrated from overloaded servers
  - Load imbalance
  - Addition / removal of physical server

Virtual Machine Migration Based Algorithms

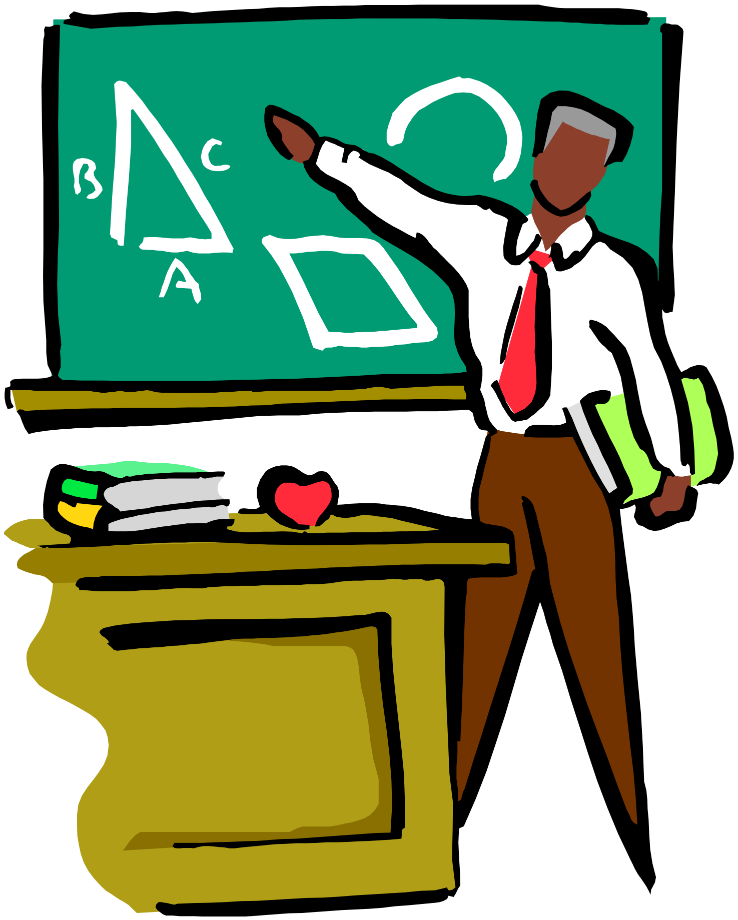
- Which VM to migrate?
  - overloaded VMs
  - Holistic approach
    - The overloaded VM is not always the best choice (Time for migrating it might be too high – A less overloaded VM might be considered)
  - Affinity based
    - If 2 VMs are communicating, it might not make sense to move one and let the other due to delays

Virtual Machine Migration Based Algorithms

- Where to migrate?
  - Available resource capacity
  - Affinity

Case Studies

Case studies



- **Case study 1: XEN (VM Management solution)**
- **Case study 2: Openstack (cloud management solution)**

IaaS Layers

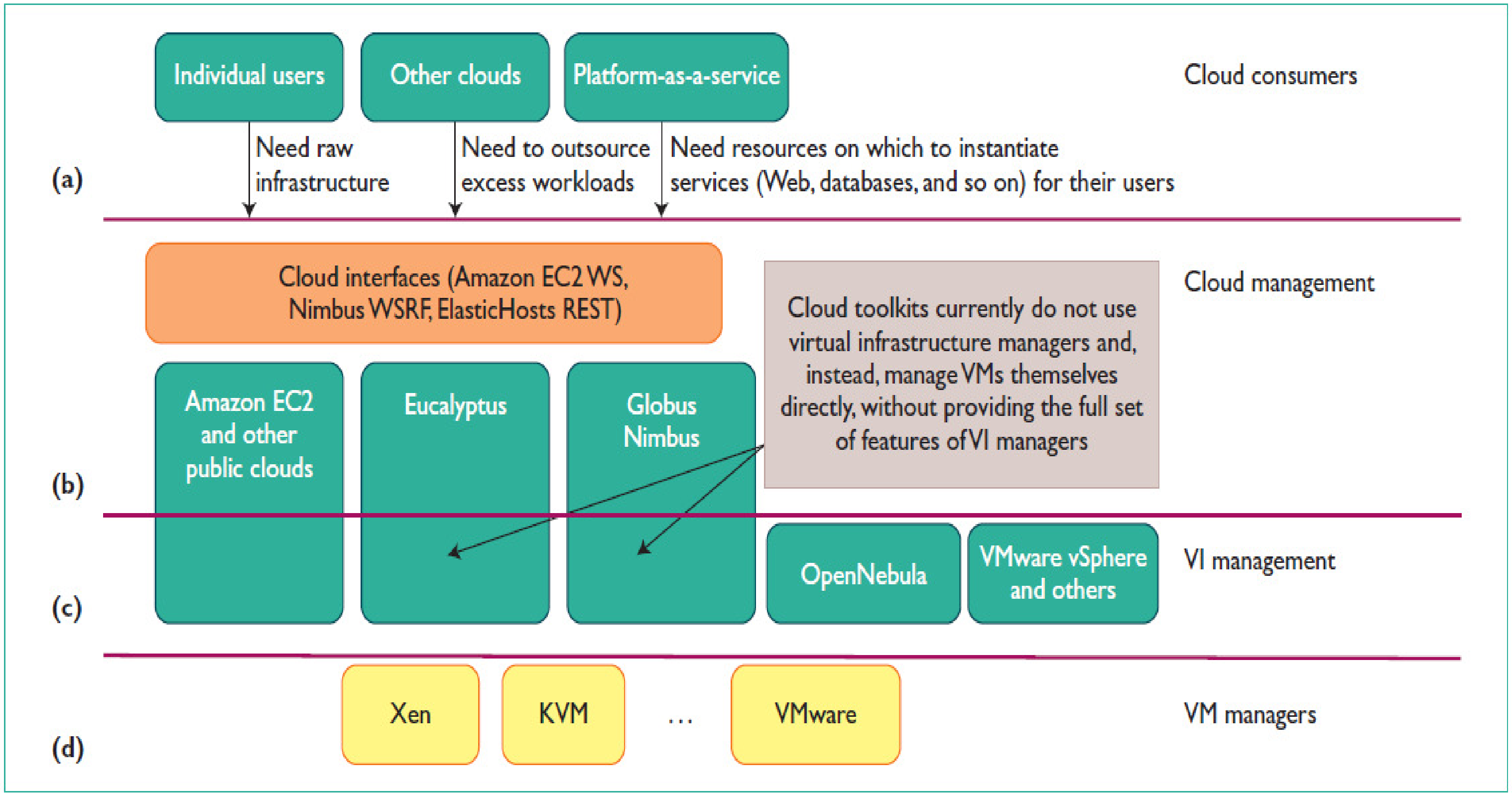


Figure 1. The cloud ecosystem for building private clouds. (a) Cloud consumers need flexible infrastructure on demand. (b) Cloud management provides remote and secure interfaces for creating, controlling, and monitoring virtualized resources on an infrastructure-as-a-service cloud. (c) Virtual infrastructure (VI) management provides primitives to schedule and manage VMs across multiple physical hosts. (d) VM managers provide simple primitives (start, stop, suspend) to manage VMs on a single host.

XEN

P. Barham et al., Xen and the Art of Virtualization, Proceeding SOSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles, Pages 164-177

Design goals

- Unmodified application binaries
- Multi application operating systems
- Para-virtualization (instead of binary translations)

Architecture (Ref. 6)

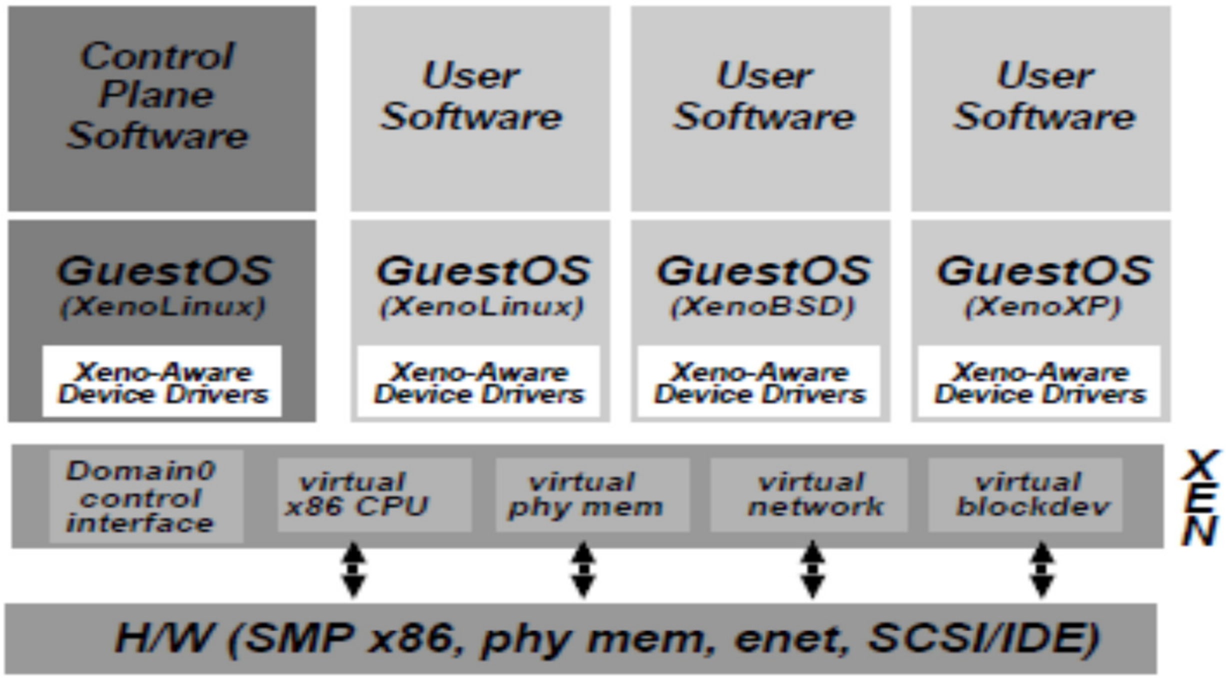


Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including *Domain0* running control software in a XenoLinux environment.

Openstack

T. Rosado and J. Bernardino, An Overview of Openstack, Proceeding IDEAS '14 Proceedings of the 18th International Database Engineering & Applications Symposium

On Cloud Management Solutions

- Highest level of abstraction towards consumers (e.g. Paas)
- Monolithic blocks integrating virtual infrastructure managers in most cases
  - No clean interface with virtual infrastructure managers

Openstack (A non comprehensive view – Reference 7)

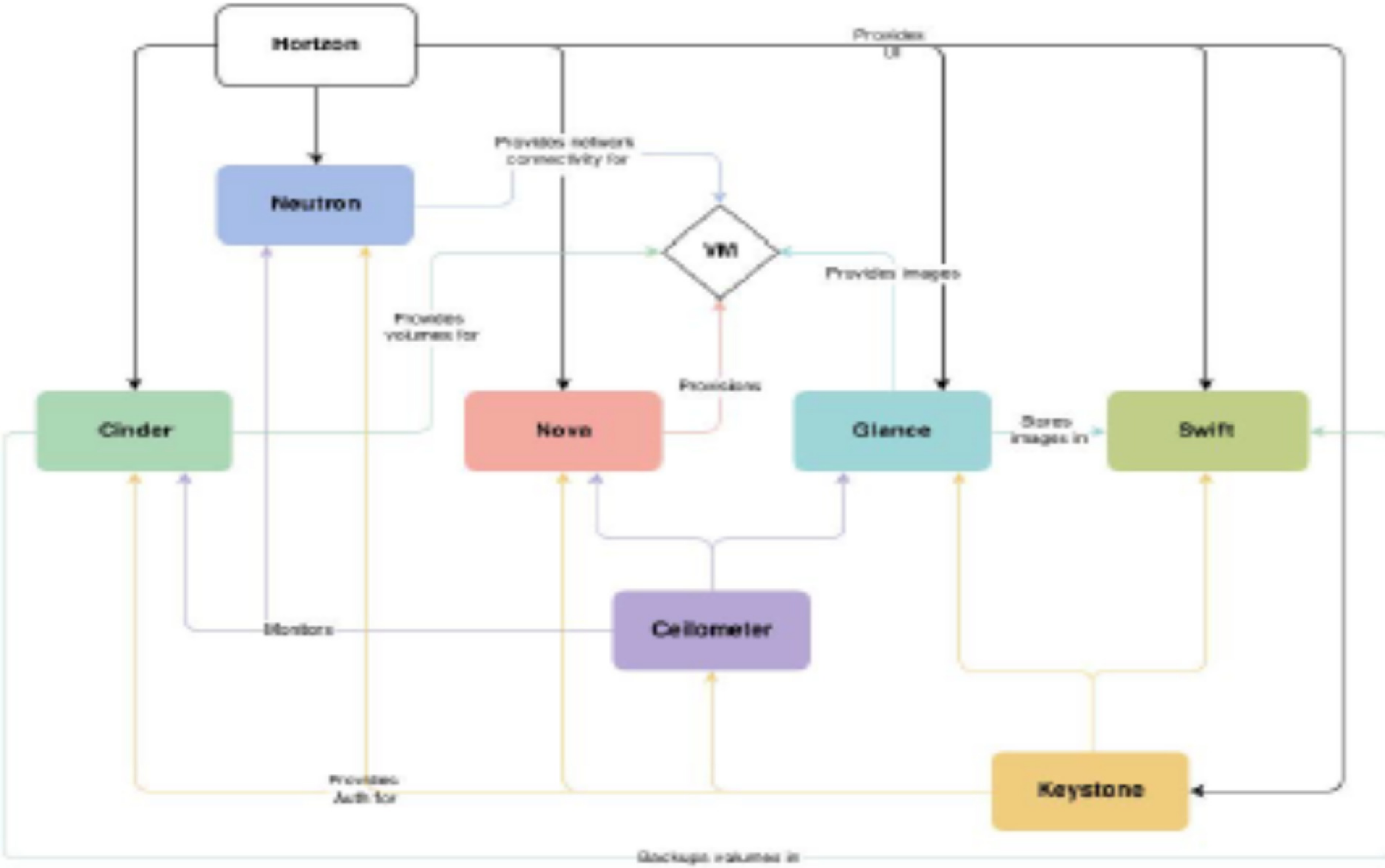


Figure 1 - Openstack conceptual architecture

Openstack

- Open source combination of related software projects, with a subset shown on previous figure:
- Some examples
  - Computing
    - Nova
      - Independence of specific VMs (e.g. XEN, KVM, VMWare)
    - Glance
      - Image service

  - Networking
    - Neutron

Openstack

- Storing
  - Swift
    - Highly available distributed object store
  - Cinder
    - Persistent block storage
    - VM back up by working with Swift

Openstack

- Shared services
  - Ceilometer
    - metering
  - Keystone
    - Identity management
  - Horizon
    - User interface for cloud infrastructure management

Openstack

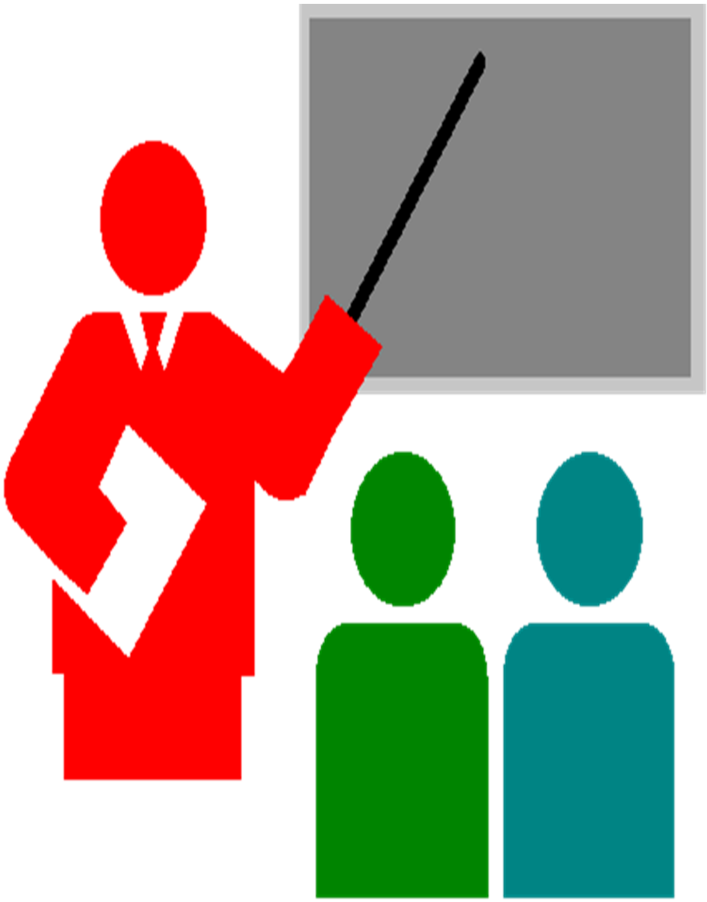
Examples of services which do not appear on the figure

- Orchestration
  - Heat
- Bare metal provisioning
  - Ironic

  Note: Bare metal provisioning in Openstack world means giving access to the bare hardware with no hypervisor
    - No virtualization
    - Single tenant !!!
      - » Might be useful in cloud environment if one wishes to use a specific specialized piece of hardware dedicated to a single tenant

Server-less computing

Server-less Computing (Function as a Service)



- Introduction
- Architecture
- Pros / Cons

Introduction

Server-less does not mean there is no server !!!

- There are indeed servers !!!
  - However the servers are completely transparent to the cloud users, unlike (Virtual Machine (VM), Containers, Uni-kernel)
    - Server-less computing might actual rely on VMs or containers or uni-kernels
  - Cloud users deal with functions
    - thus Functions as a Service (FaaS)

Architecture

Principles

1) Applications built as a set of functions
2) When there is a request for a given function, a run time environment (e.g. VM, container, uni-kernel) is launched with the function code + libraries
3) The run time is terminated after the execution of the function

Architecture (Reference 1)

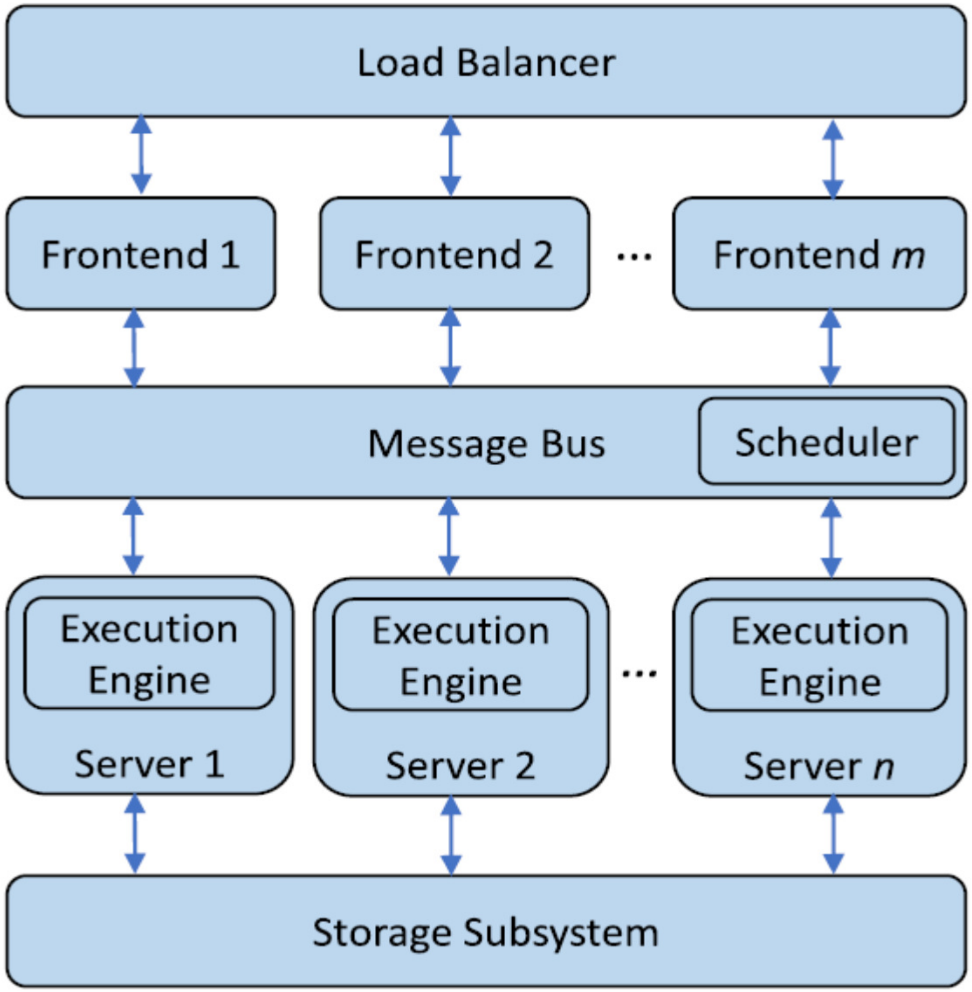


Fig. 1. *Serverless platform architecture.*

Architecture

Load balancer:

- Self explanatory

Front end:

- End user interface

Message bus and scheduler:

- Mediation between front ends and execution engines

Architecture

Load balancer:

- Self explanatory

Front end:

- End user interface

Message bus and scheduler:

- Mediation between front ends and execution engines
    - Relies on a publication / subscription principles

Architecture

Execution engine:

- Self explanatory
  - Might rely on VM, containers and uni-kernels

Storage sub-system:

- States
- Persistent data

Pros (Examples)

- No real / virtual server management by cloud users

- Resource Efficiency and low cost

- Built-in scalability

Cons (Examples)

- **Most cited:**
  - Start up latency
- **Others:**
  - Learning curve of the new programming model (e.g. stateless functions + events)

Pros vs Cons

- **Decision to be made on case by case basis (Ref. 1)**

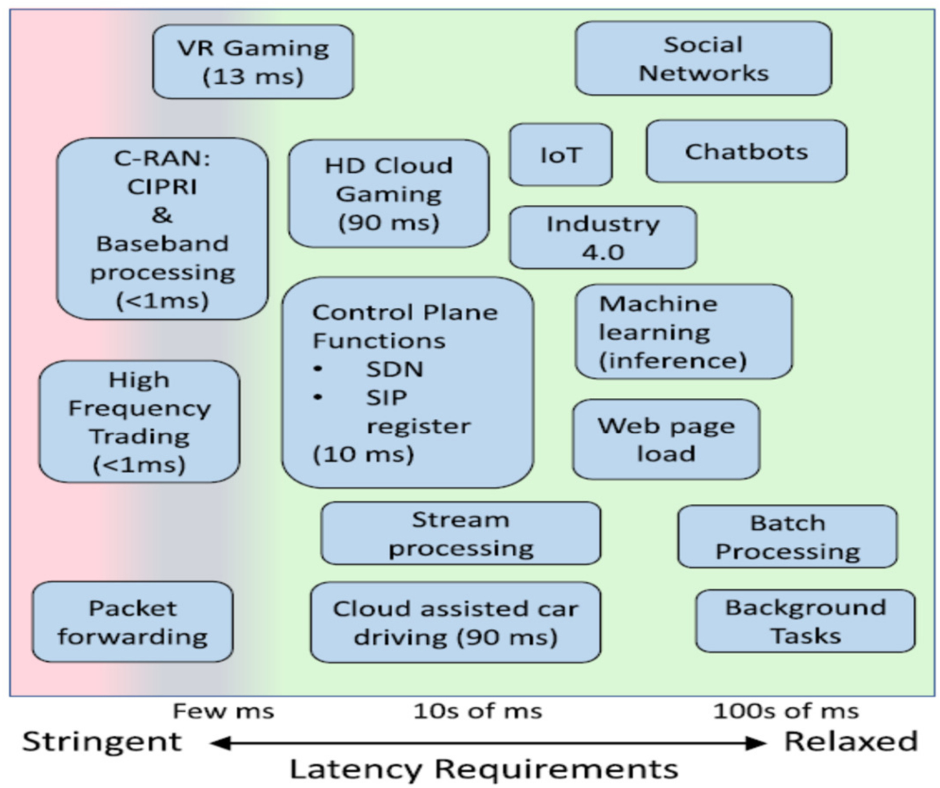


Fig. 3. *Latency requirement ranges for various applications.*

Pros vs Cons

- **Decision to be made on case by case basis (Ref. 1)**

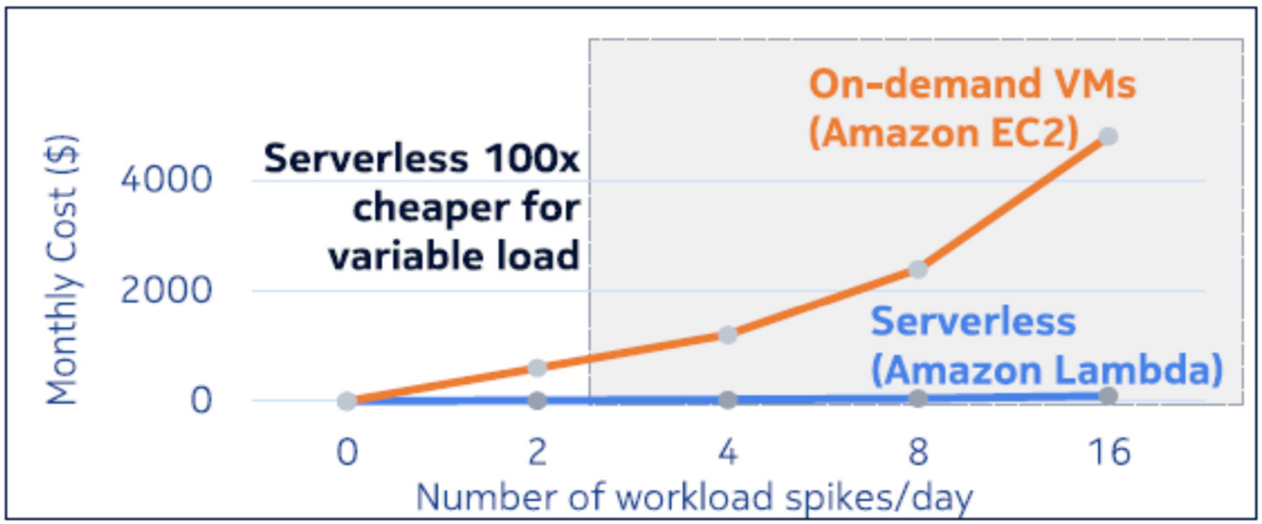


Fig. 4. *Cost comparison between Amazon Lambda (serverless) and Amazon EC2 (VMs) for spiky workload. In the gray region, serverless is 100x cheaper.*

The End