1.6 On-demand Provisioning

- Resource Provisioning means the selection, deployment, and run-time management of software (e.g., database server management systems, load balancers) and hardware resources (e.g., CPU, storage, and network) for ensuring guaranteed performance for applications.
- Resource Provisioning is an important and challenging problem in the large-scale distributed systems such as Cloud computing environments.
- There are many resource provisioning techniques, both static and dynamic each one having its own advantages and also some challenges.
- These resource provisioning techniques used must meet Quality of Service (QoS) parameters like availability, throughput, response time, security, reliability etc., and thereby avoiding Service Level Agreement (SLA) violation.
- Over provisioning and under provisioning of resources must be avoided.
- Another important constraint is power consumption.
- The ultimate goal of the cloud user is to minimize cost by renting the resources and from the cloud service provider's perspective to maximize profit by efficiently allocating the resources.
- In order to achieve the goal, the cloud user has to request cloud service provider to make a provision for the resources either statically or dynamically.
- So that the cloud service provider will know how many instances of the resources and what resources are required for a particular application.
- By provisioning the resources, the QoS parameters like availability, throughput, security, response time, reliability, performance etc must be achieved without violating SLA

There are two types

- **Static Provisioning**
- **Dynamic Provisioning**

Static Provisioning

- For applications that have predictable and generally unchanging demands/workloads, it is possible to use "static provisioning" effectively.
- With advance provisioning, the customer contracts with the provider for services.
- The provider prepares the appropriate resources in advance of start of service.
- The customer is charged a flat fee or is billed on a monthly basis.

Dynamic Provisioning

- In cases where demand by applications may change or vary, “dynamic provisioning" techniques have been suggested whereby VMs may be migrated on-the-fly to new compute nodes within the cloud.
- The provider allocates more resources as they are needed and removes them when they are not.
- The customer is billed on a pay-per-use basis.
- When dynamic provisioning is used to create a hybrid cloud, it is sometimes referred to as cloud bursting.

Parameters for Resource Provisioning

- Response time
- Minimize Cost
- Revenue Maximization
- Fault tolerant
- Reduced SLA Violation
- Reduced Power Consumption

Response time: The resource provisioning algorithm designed must take minimal time to respond when executing the task.

Minimize Cost: From the Cloud user point of view cost should be minimized.

Revenue Maximization: This is to be achieved from the Cloud Service Provider’s view.

Fault tolerant: The algorithm should continue to provide service in spite of failure of nodes.

Reduced SLA Violation: The algorithm designed must be able to reduce SLA violation.

Reduced Power Consumption: VM placement & migration techniques must lower power consumption

Dynamic Provisioning Types

1. Local On-demand Resource Provisioning
2. Remote On-demand Resource Provisioning

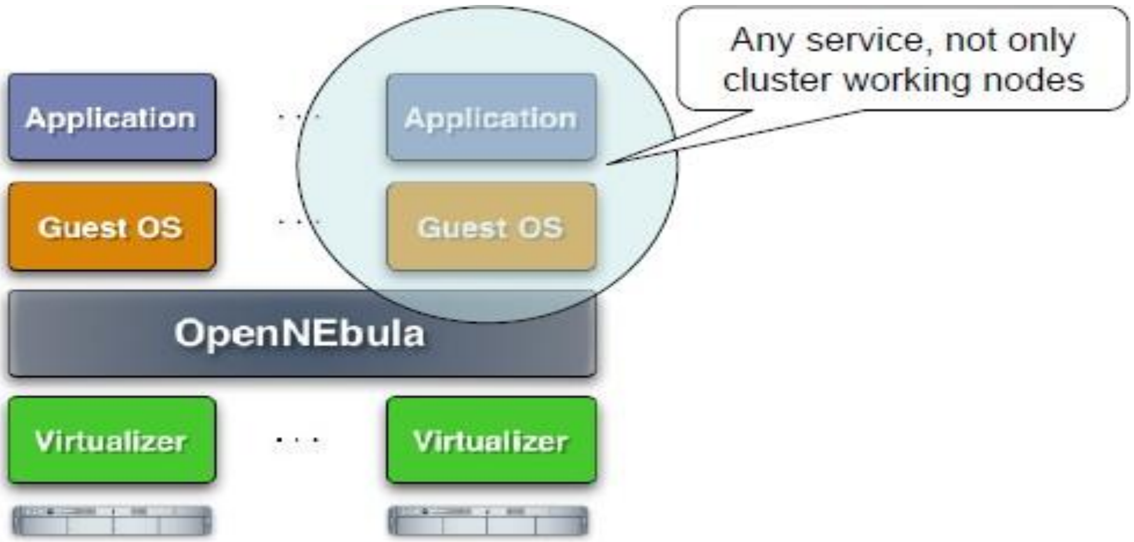
Local On-demand Resource Provisioning

1. The Engine for the Virtual Infrastructure

<u>The OpenNebula Virtual Infrastructure Engine</u>

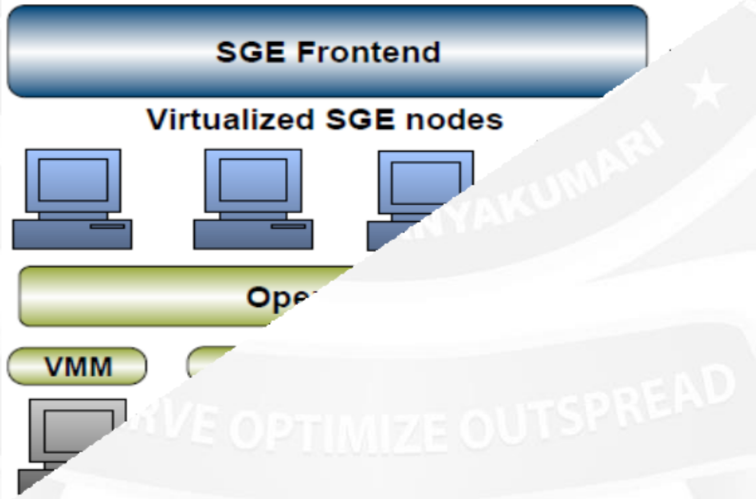
- OpenNEbula creates a distributed virtualization layer
    - Extend the benefits of VM Monitors from one to multiple resources
    - Decouple the VM (service) from the physical location
- Transform a distributed physical infrastructure into a flexible and elastic virtual

infrastructure, which adapts to the changing demands of the VM (service) workloads



Separation of Resource Provisioning from Job Management

- New virtualization layer between the service and the infrastructure layers
- Seamless integration with the existing middleware stacks.
- Completely transparent to the computing service and so end users



Cluster Partitioning

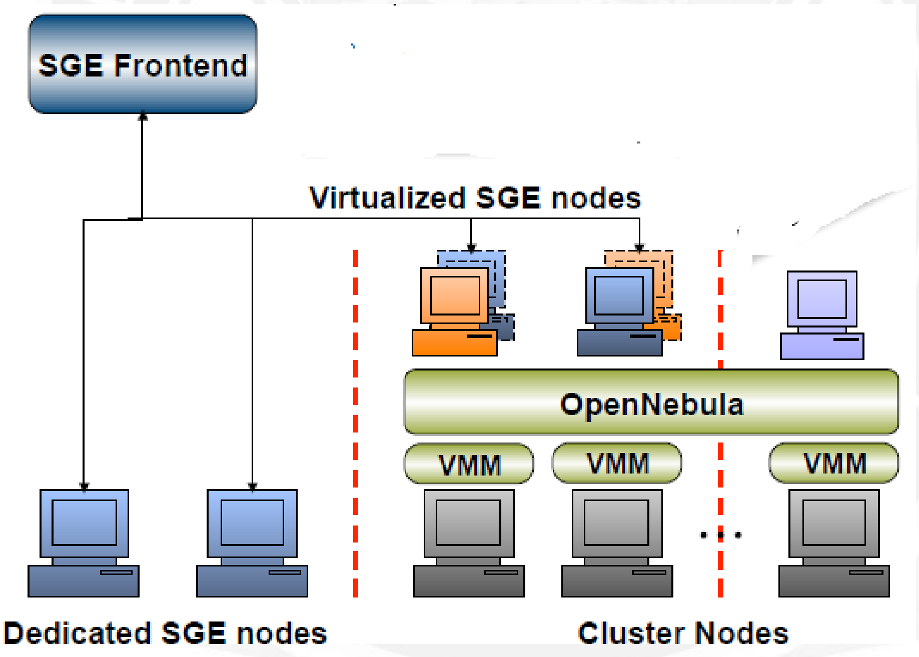
- Dynamic partition of the infrastructure
- Isolate workloads (several computing clusters)

- Dedicated HA partitions

Benefits for Existing Grid Infrastructures

- The **virtualization of the local infrastructure** supports a virtualized alternative to contribute resources to a Grid infrastructure
    - Simpler deployment and operation of new middleware distributions
    - Lower operational costs
    - Easy provision of resources to more than one infrastructure
    - Easy support for VO-specific worker nodes

Performance partitioning between local and grid clusters



Other Tools for VM Management

- VMware DRS, Platform Orchestrator, IBM Director, Novell ZENworks, Enomalism, Xenoserver
- **Advantages**:
    - Open-source (Apache license v2.0)
    - Open and flexible architecture to integrate new virtualization technologies
    - Support for the definition of any scheduling policy (consolidation, workload balance, affinity, SLA)
    - LRM-like CLI and API for the integration of third-party tools

Remote on-Demand Resource Provisioning

Access to Cloud Systems

- Provision of virtualized resources as a service

VM Management Interfaces

The processes involved are

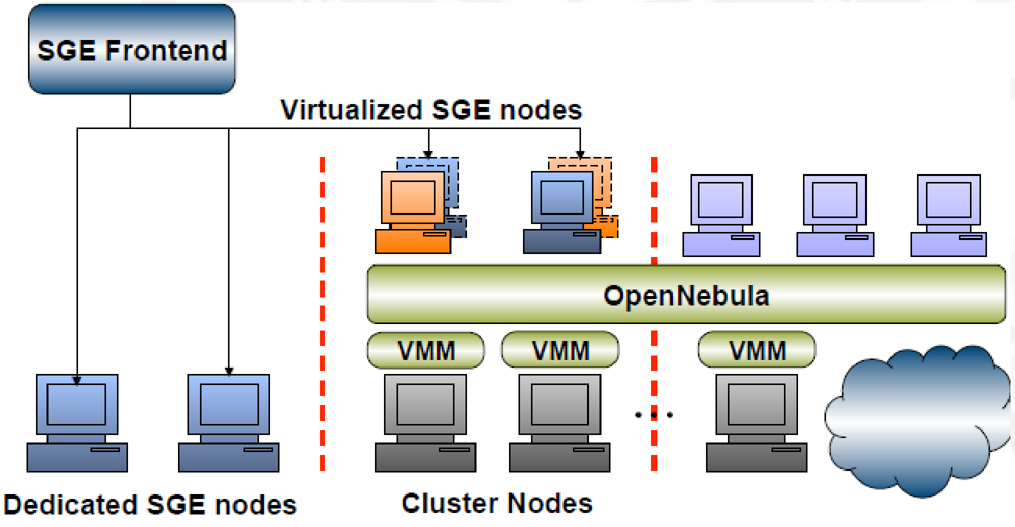
- Submission
- Control
- Monitoring

Infrastructure Cloud Computing Solutions

- Commercial Cloud: Amazon EC2
- Scientific Cloud: Nimbus (University of Chicago)
- Open-source Technologies
  - Globus VWS (Globus interfaces)
  - Eucalyptus (Interfaces compatible with Amazon EC2)
  - OpenNEbula (Engine for the Virtual Infrastructure)

On-demand Access to Cloud Resources

- Supplement local resources with cloud resources to satisfy peak or fluctuating demands


SGE Frontend
Virtualized SGE nodes
OpenNebula
VMM
VMM
VMM
Dedicated SGE nodes
Cluster Nodes