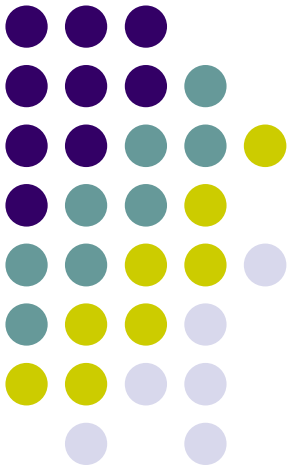
Cloud Design and Implementation

Cheng Li
MPI-SWS
Nov 9th, 2010

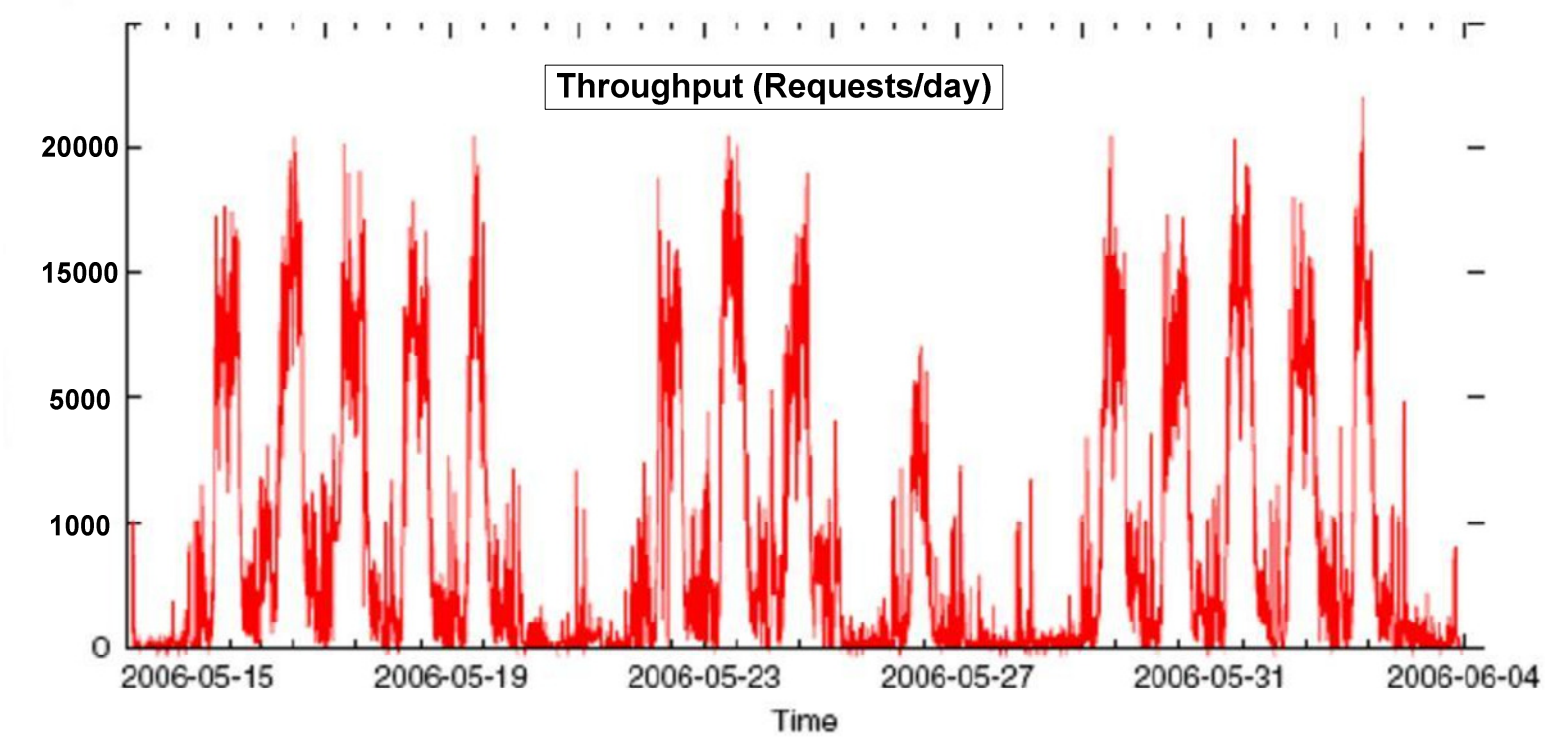


Modern Computing

- Academic computation
  - Chemistry, Biology
  - Large Data Set Analysis
- Online service
  - Shopping
  - Website
- Collaborative jobs
  - Google Docs
  - Online Games

Resource Demands

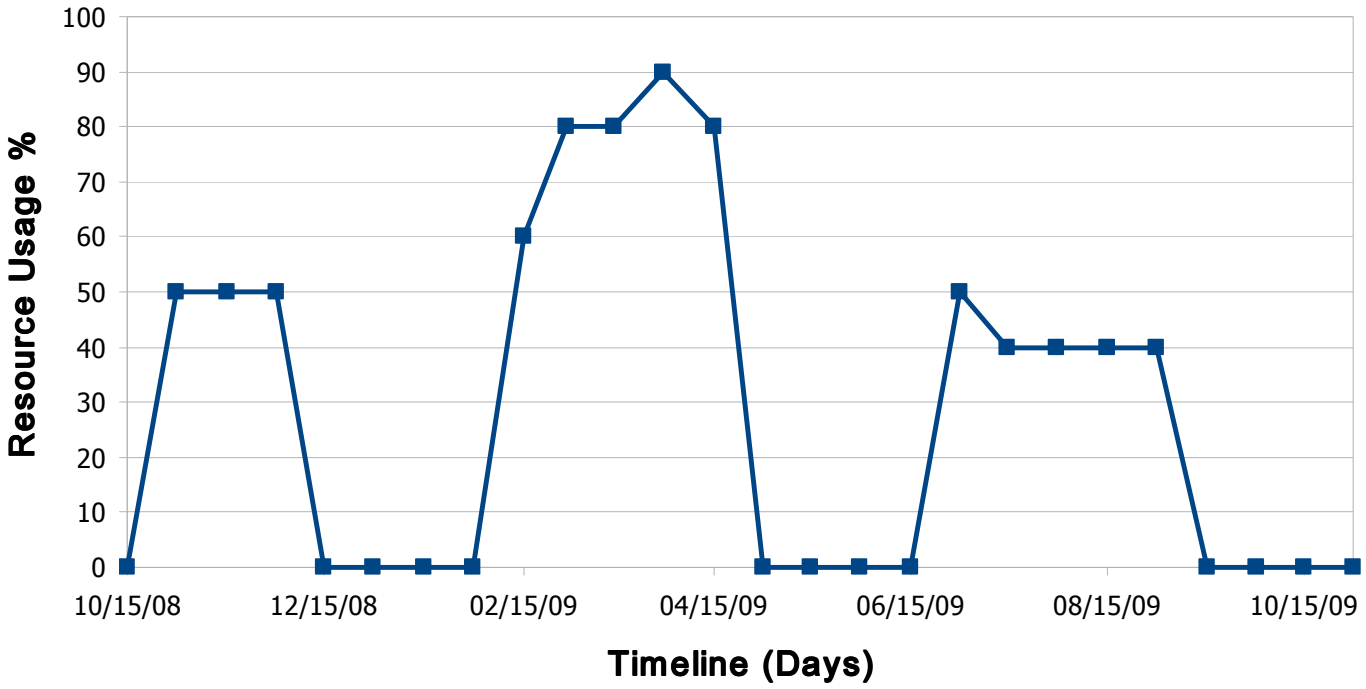
- Online Service Workload



The resource demands are changing significantly over time.

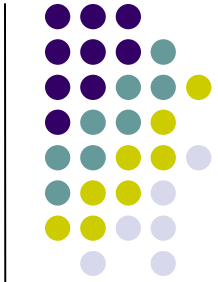
Resource Demands Con't

- Academic Computation Resource Usage



The resource demands are not constant, instead periodical and diverse.

Resource Provision



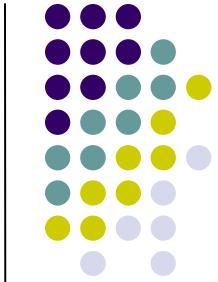
- Keep their own systems
  - High investments
    - Hardware & software
  - Significant maintenance cost
    - Power & Cooling
    - Technical supports
    - Upgrading
  - Hard to adapt to elastic demands
    - Scale up or down
    - Unnecessary to keep it always on

Resource Provision Con't

- Ideal Economic Model
  - Elastic Provision
  - Affordable Price
  - Robust
  - Easy to use
  - Buy whenever you want
  - Pay as you go

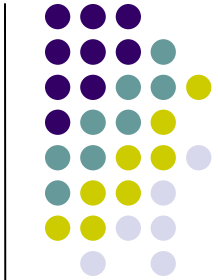
?

Resource Provision Con't



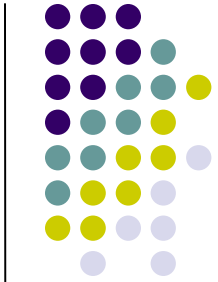
- Ideal Economic Model
  - Elastic Provision
  - Affordable Price
  - Robust
  - Easy to use
  - Buy whenever you want
  - Pay as you go

Outline



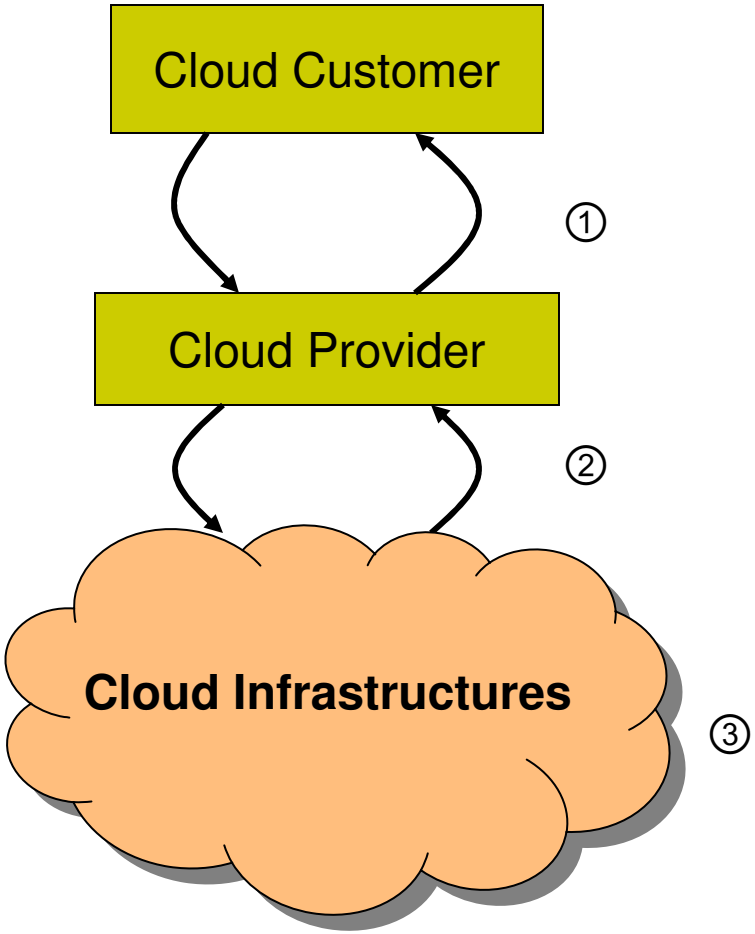
- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Cloud Computing



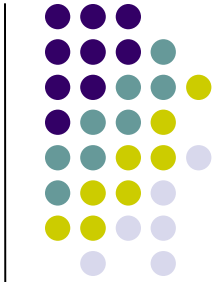
- The applications delivered as *services* over the Internet
  - e.g., *Google search, Docs, Amazon Web Services*
- The *hardware* and systems *software* in the datacenters that provide those services
  - e.g., *Network, disk, memory, CPU*
  - e.g., *Platforms*

Cloud Computing Con't



- ① Request Submission
- ② Resource Assignment
- ③ Offering services

Amazon Cloud Service



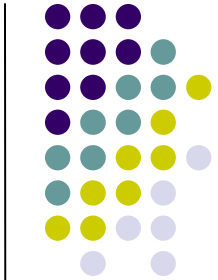
- Elastic Compute Cloud (EC2)

Standard Instances	Linux/UNIX Usage	Windows Usage
Small (Default)	$0.085 per hour	$0.12 per hour
Large	$0.34 per hour	$0.48 per hour
Extra Large	$0.68 per hour	$0.96 per hour

- Simple Storage Service (S3)

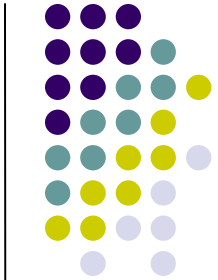
Tier	Price	Tier	Price
First 1 TB / month of Storage Used	$0.140 per GB	Next 49 TB / month of Storage Used	$0.125 per GB
Next 450 TB / month of Storage Used	$0.110 per GB	Next 500 TB / month of Storage Used	$0.095 per GB

Outline



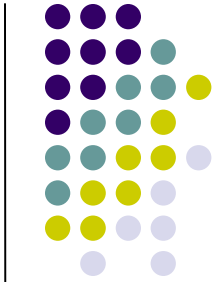
- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Design Goals



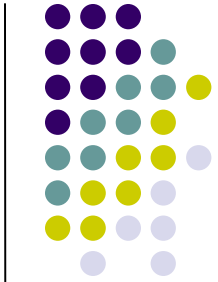
- Multi-tenancy
  - Hardware and software shared
  - Isolation problem
- Scalability
  - Precisely estimates demands
  - Quickly adapts to changes
    - Scale up or down
  - Cloud Growth

Design Goals Con't



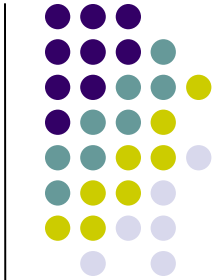
- Load and Tenant Balancing
  - Almost equal load distribution
  - Avoid overload
  - Take advantage of any available resource
- Availability
  - Failures are common
  - Outage leads to loss of money
  - Robust to tolerate failures

Design Goals Con't



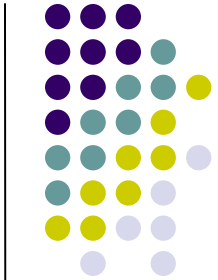
- Easy-of-use
  - Low admin efforts
  - Simple Interface (API)
  - Transparency (Challenge)
- User friendly
  - Close to user (e.g., geo-located datacenters)
  - Short instructions
  - Low latency

Modules of Cloud



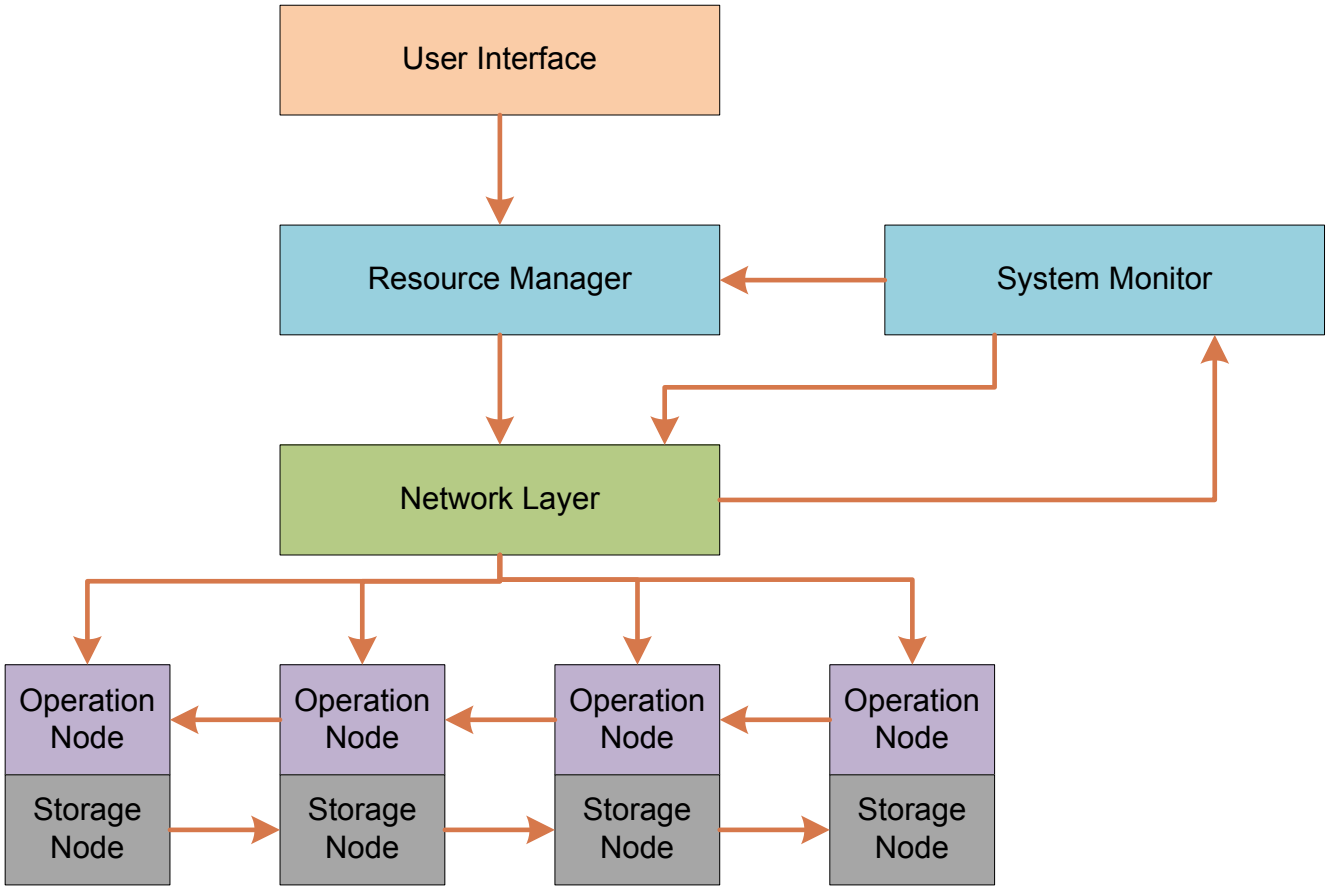
- External Interface
  - User & Admin access to their systems
- Resource manager
  - Allocate as required
  - Free & Clean up
- Operation node
  - Computing unit
- Storage Node
  - Dedicated to store data

Modules of Cloud

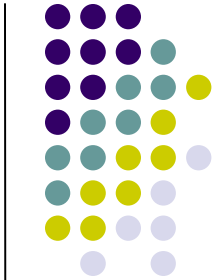


- Monitor
  - Inspect the whole system
  - Handle node failures
- Network Layer
  - Interconnectivity in one service
  - Isolation between services
- Load Balancer
- Security Guarantee
  - Access Control
  - Malicious attack

Modules of Cloud Con't

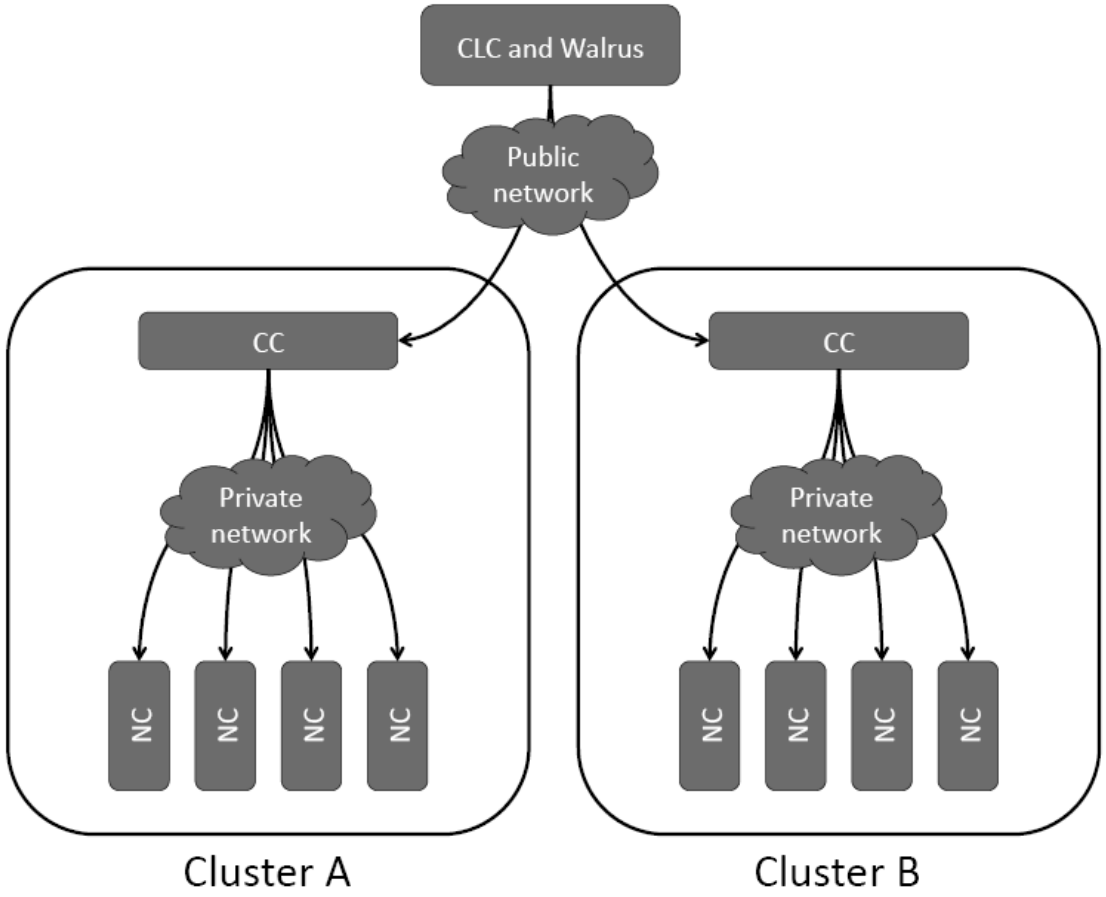


Outline



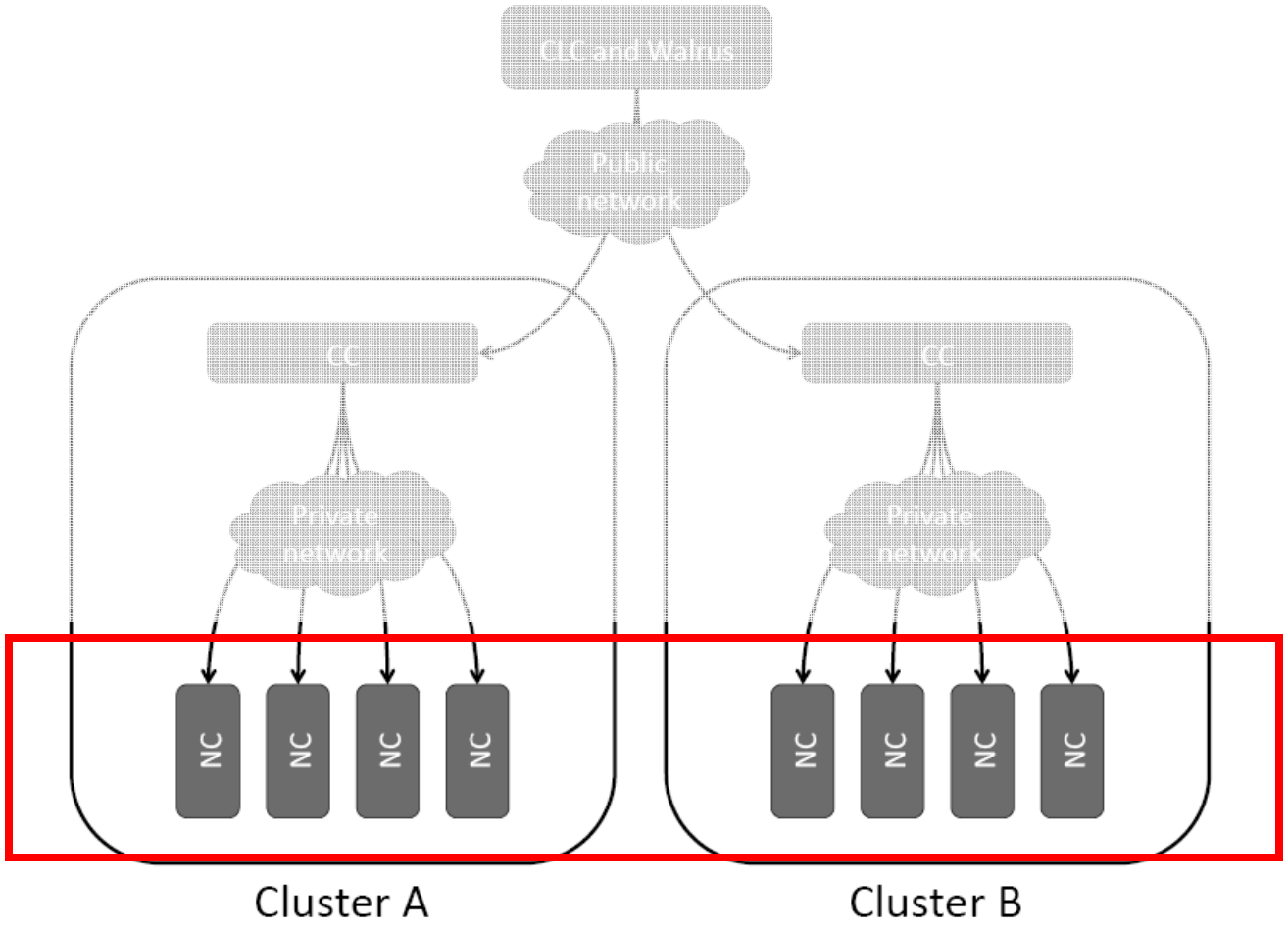
- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Eucalyptus

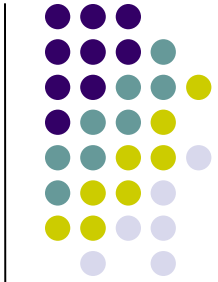


An open-source cloud system design

Node Controller



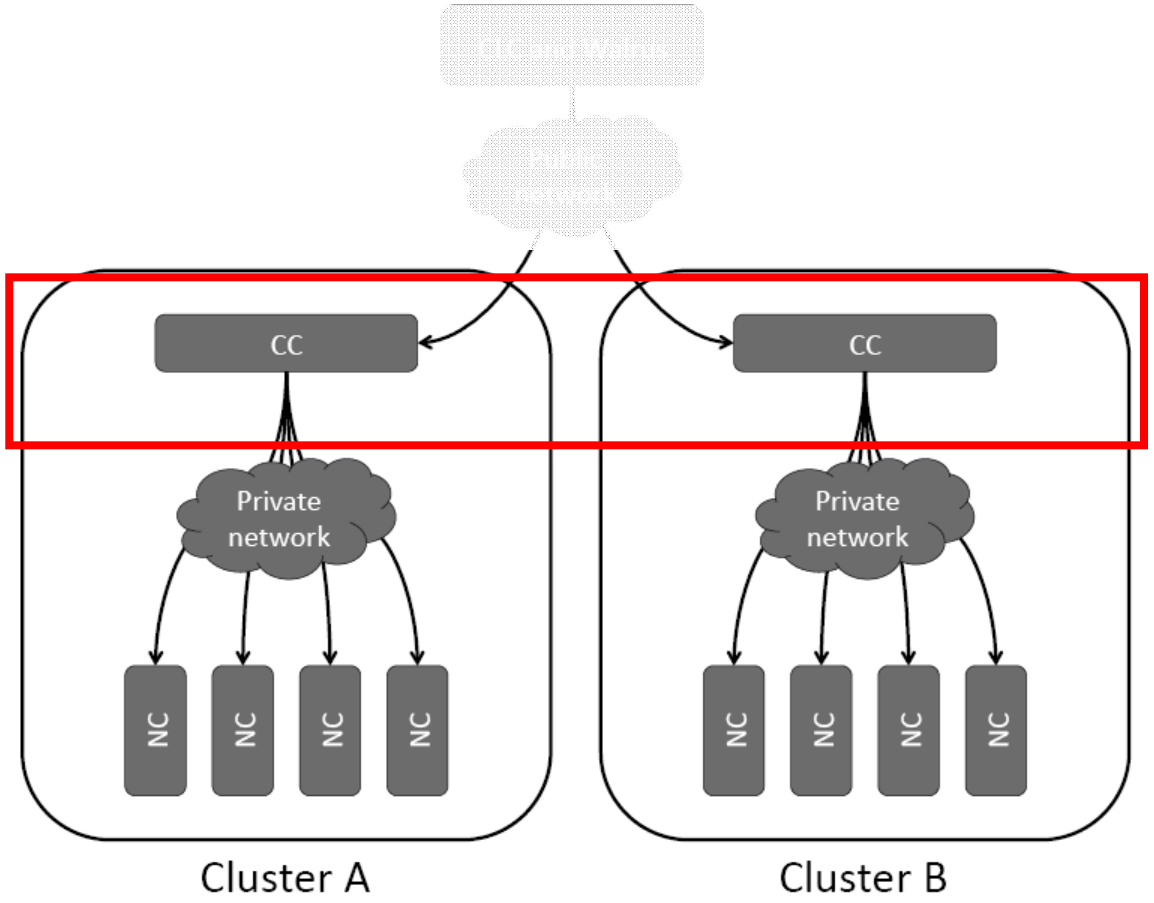
Node Controller



- Hosted at each physical node
- Serve upcoming requests
- Functionalities

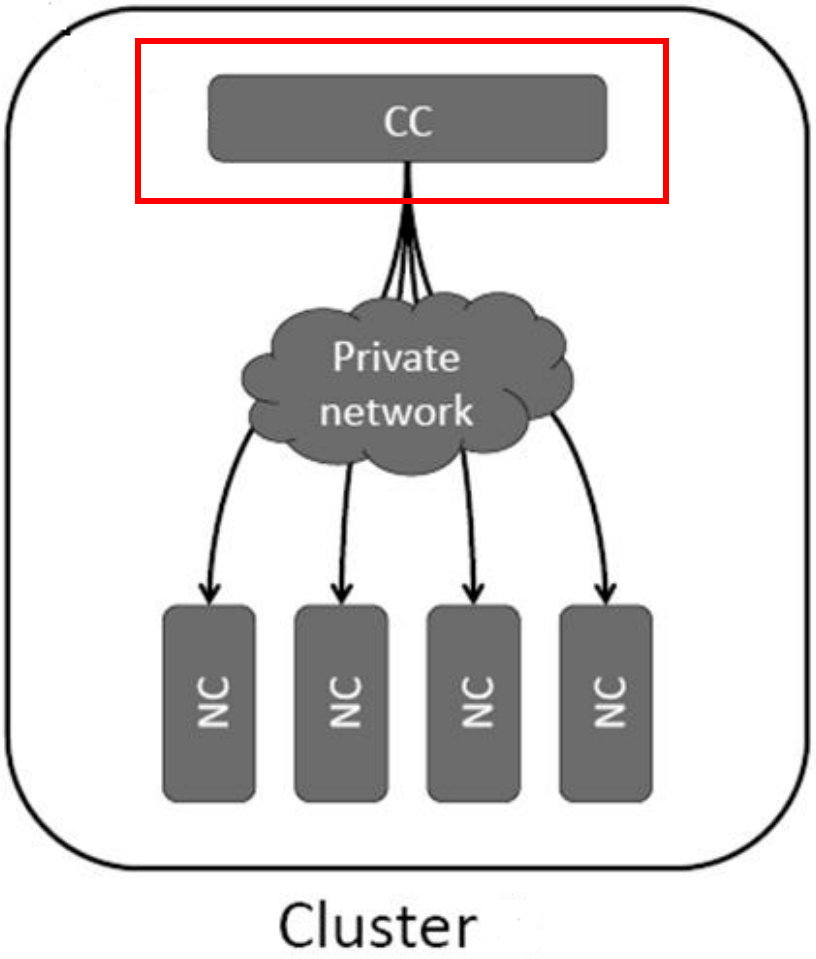
Tasks	API
Create a Virtual Machine Instance (VM) for services	runInstance()
Terminate VM if required	terminateInstance()
Inspect VM	describeInstance()
Discover physical resources (cores, memory)	describeResource()

Cluster Controller



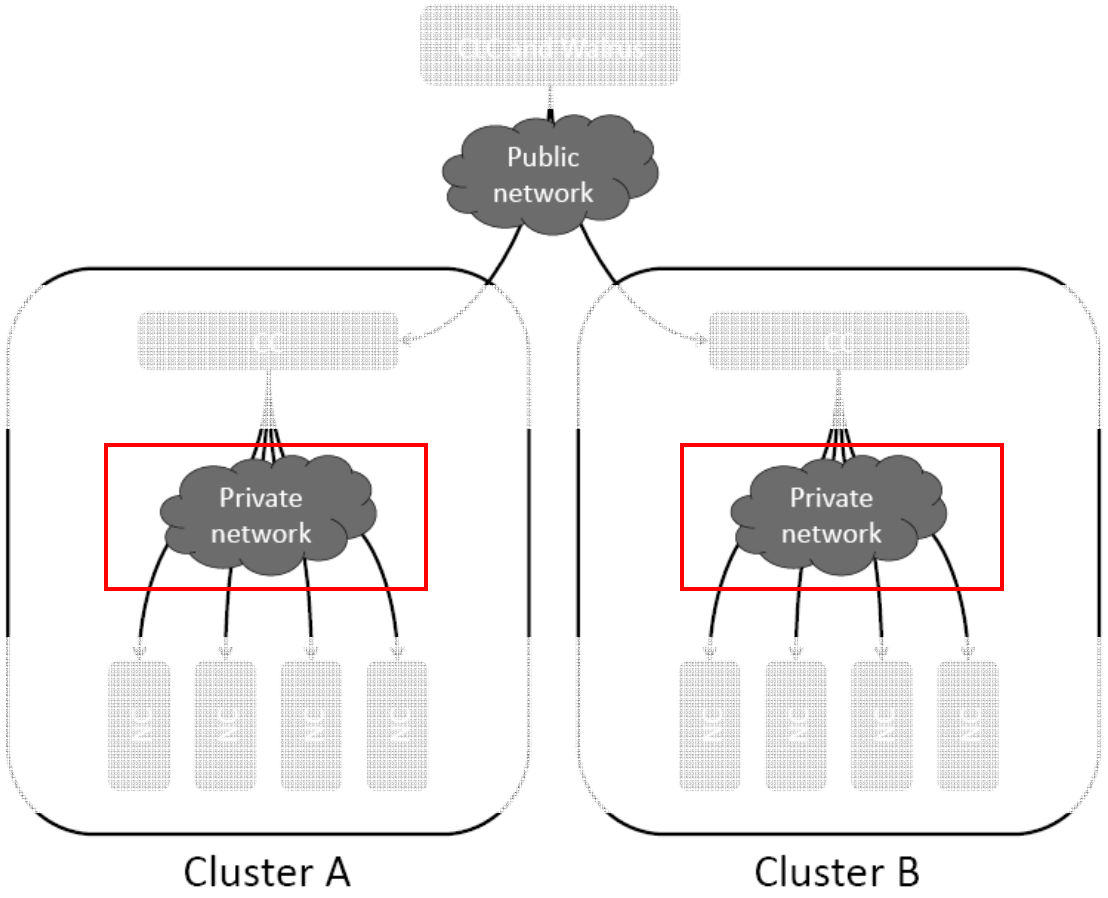
- Cluster definition

Cluster Controller



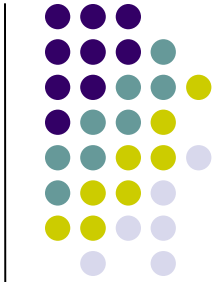
- Manage a set of NCs
- Functionalities similar to NC, but plural
  - runInstances
  - terminateInstances
  - describeInstances

Network Layer



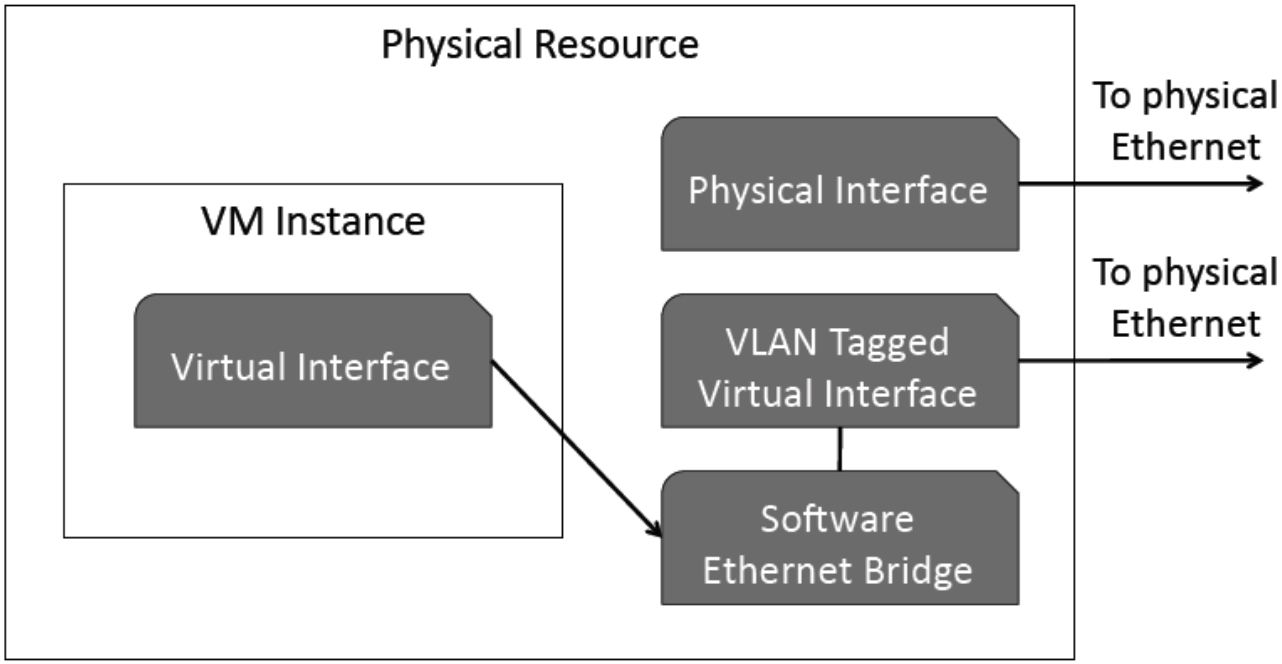
Network = Physical network + Virtual network

Virtual Network Overlay



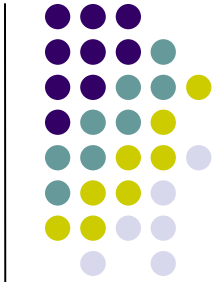
- Named network sets
  - Service owns its own network
  - Network contains a set of VM instances
- Network among VM instances
  - VMs interconnectivity
    - Exchange or share information
  - Isolation
    - Avoid interferences between two networks
  - Performance
    - Close to Physical connections

Virtual Interfaces

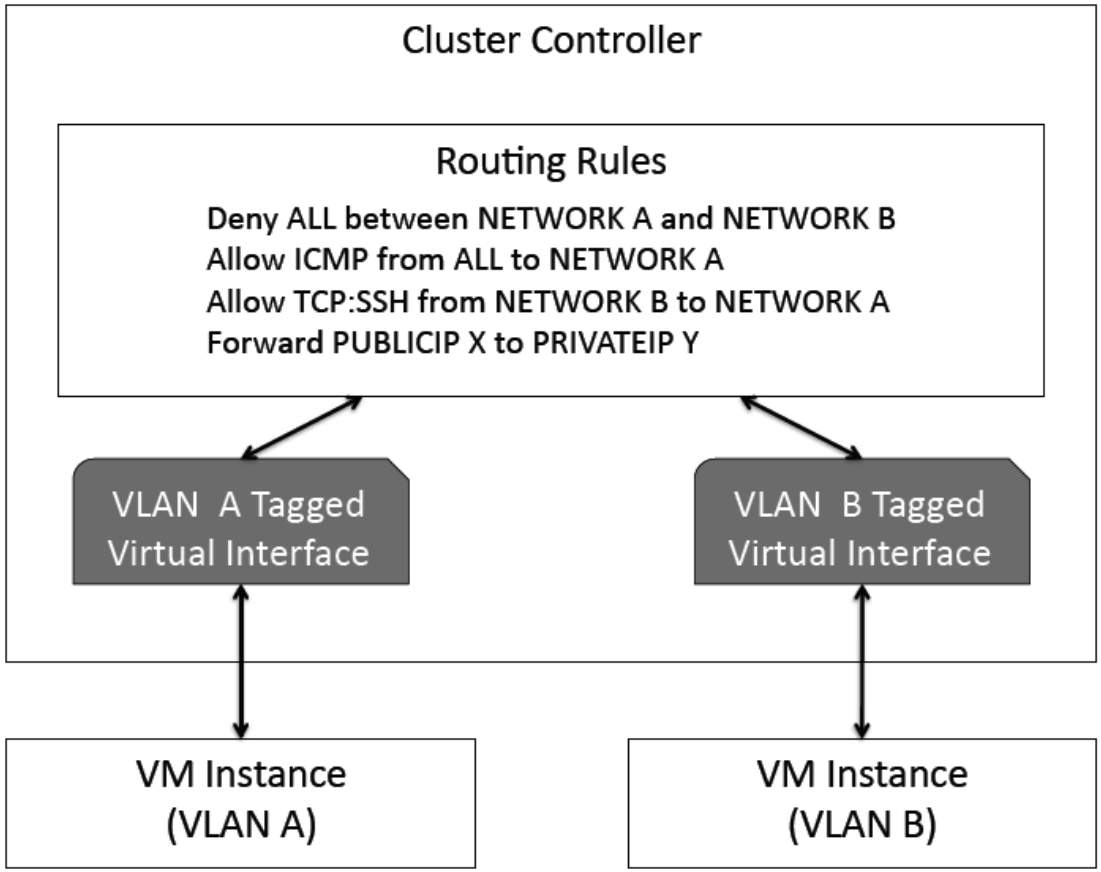


- VM interface connects to a named network
- Disjoint sets of VMs belong to different network - isolation

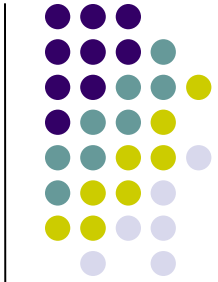
Communication across networks



- Cluster Controller acts as a router

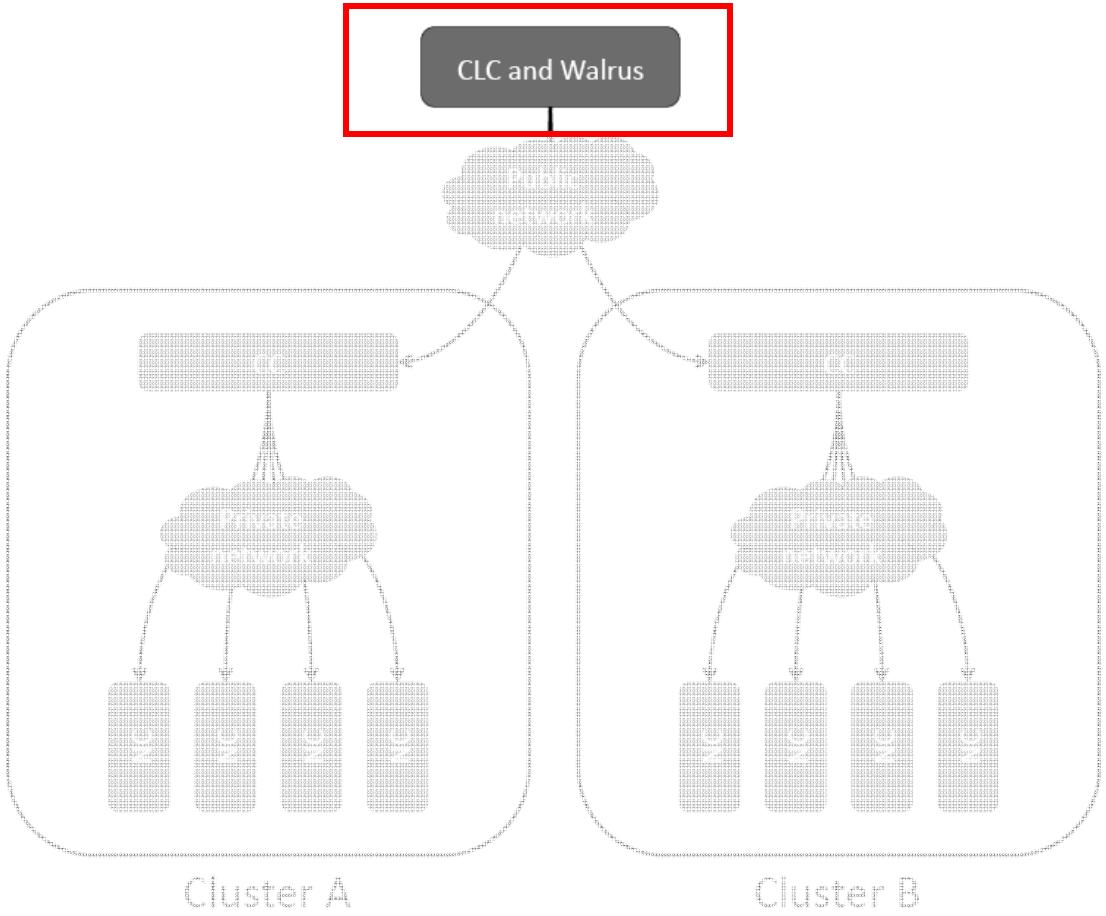


Storage Controller (Walrus)

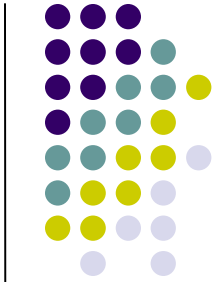


- Users
  - Stream data into/out of the cloud directly via Walrus
  - Manipulate data through communications between VM instances and Walrus
- Cloud System
  - Host root file system
  - Store VM images for initialization

Cloud Controller

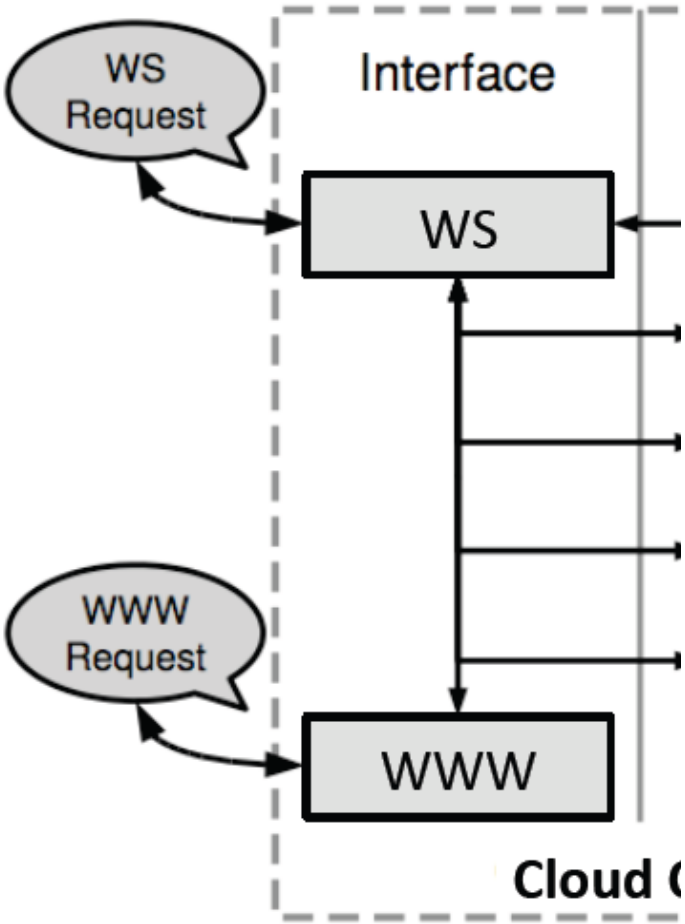


Cloud Controller Con't

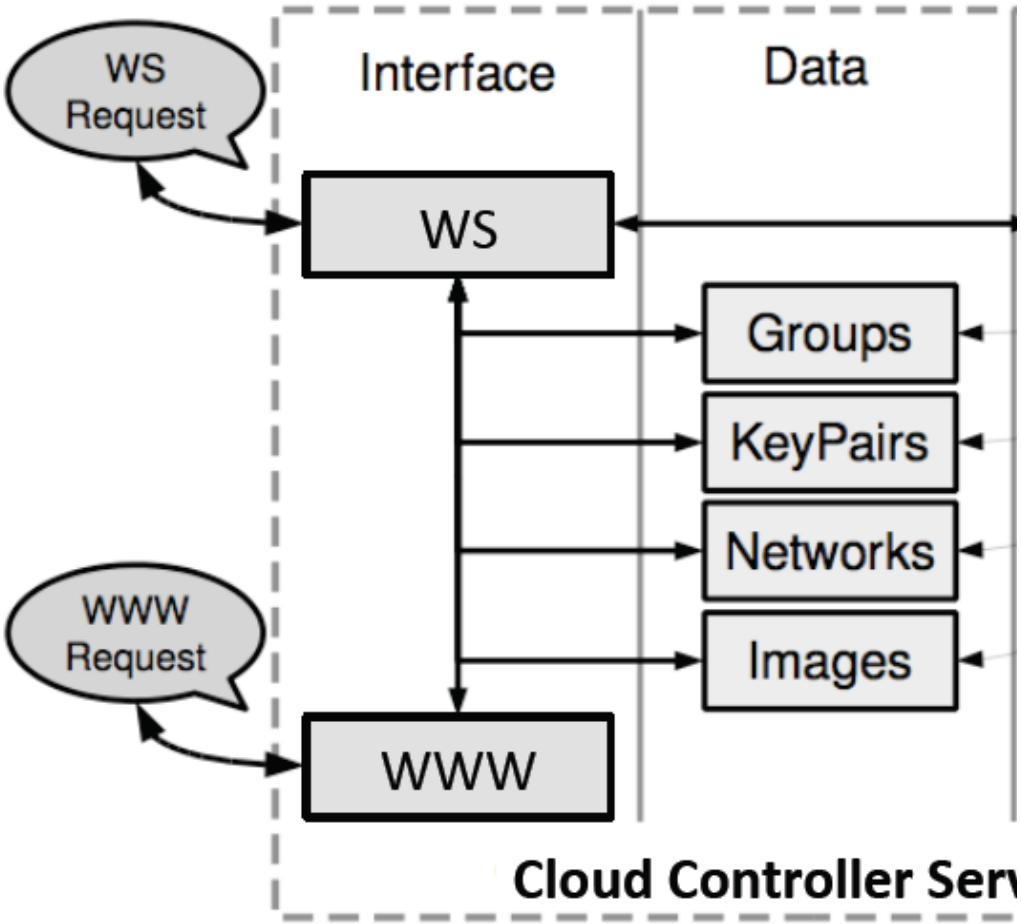


- Super administrator managing all states in the cloud, and exposing users:
  - Interface to control their VMs
  - Assure persistent meta data
    - System data: VM images
    - User data: password
  - Resource service process
    - High-level commands to allocate or free

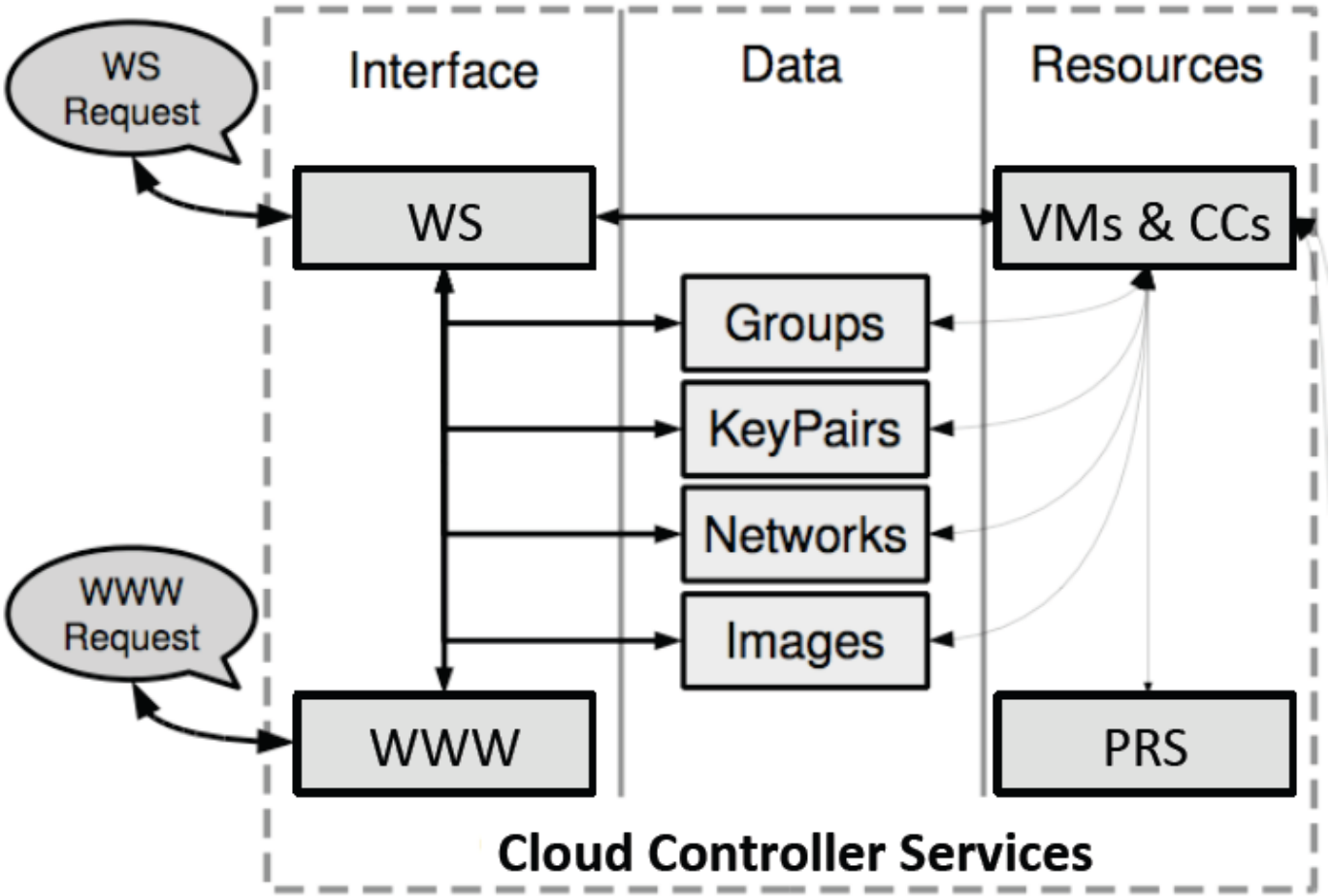
Cloud Controller Con't



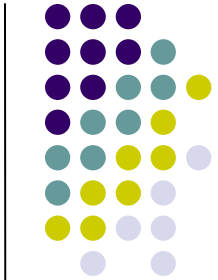
Cloud Controller Con't



Cloud Controller Con't

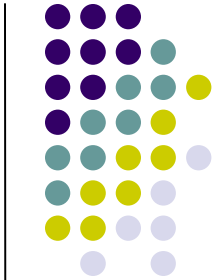


User Interfaces



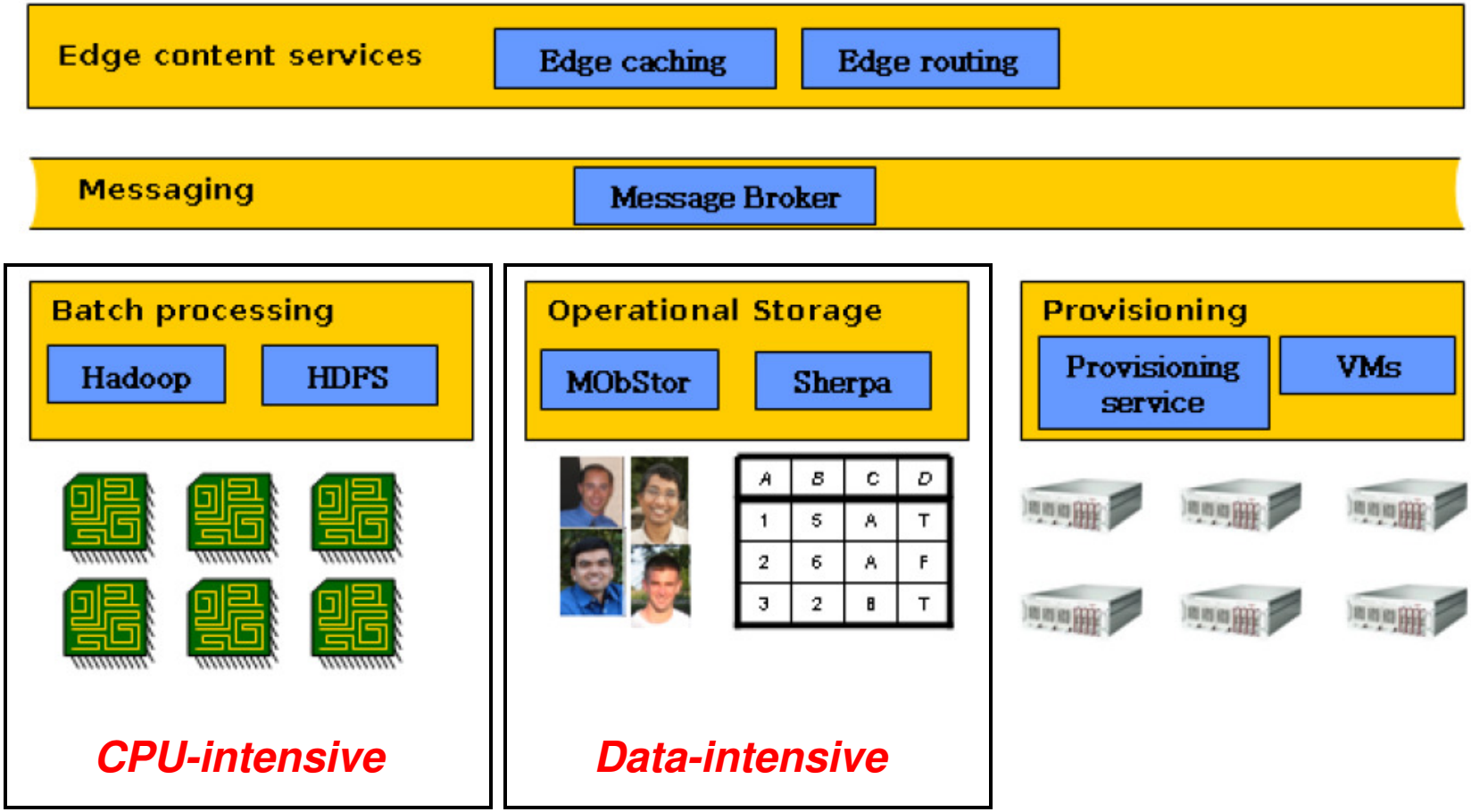
- Similar to EC2 & S3
- Simple Object Access Protocol (SOAP)
  - Use XML to express actions
- Query API
  - Only GET and POST
  - Embedded expressive queries with Parameters
- Representational State Transfer (REST)
  - Based on HTTP protocol
  - GET, DELETE, PUT, POST

Outline

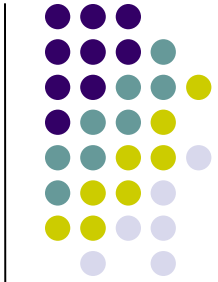


- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Yahoo! Private Cloud

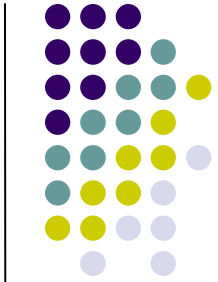


Yahoo! Private Cloud Con't



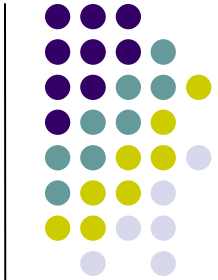
- Core Service
  - Batch process jobs
  - Operational Storage
  - Resource Provisioning
- Messaging Broker
  - Pub/Sub message pattern
- Edge services
  - Decomposition & aggregation
  - Routing
  - Caching

Yahoo! Private Cloud Apps



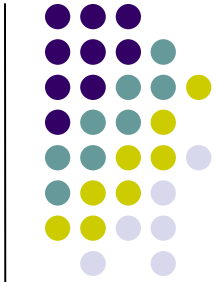
- Hadoop/MapReduce
  - Word count
  - Page ranking
  - Inverted index
- Sherpa
  - Massive data store
  - Schema-less data structure
  - Widely used by Yahoo! Web services

Outline



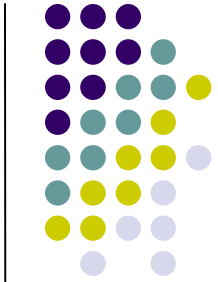
- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Open Questions:



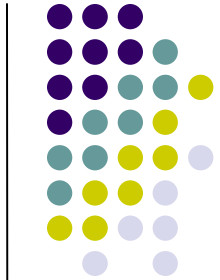
- Automated Service Provisioning
  - Predict future demands
  - Avoid outage caused by peak load
  - [INRIA, Tech Report'10]
- Scalability
  - Easy to add more resources
- Energy saving
  - Smart design
  - [Bruce Maggs, SIGCOMM'09]
- Traffic management
  - [James Willett, Invitation talk'09]

Open Questions Con't:



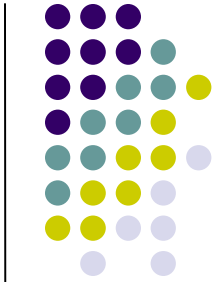
- Quality of Service (QoS)
  - Meets SLA for more applications
  - [QoS-Aware Clouds, Cloud Futures'10]
- Security
  - Malicious attacks
  - Blog: http://cloudsecurity.trendmicro.com/
- Transparency
  - User Privacy
  - Data Confidentiality and Integrity
  - [SPORC, OSDI'10]

Outline



- Motivation
- Introduction to Cloud Computing
- Design & Challenges
- Implementation
  - Eucalyptus
  - Yahoo!
- Open Questions & Discussion
- Conclusion

Conclusion

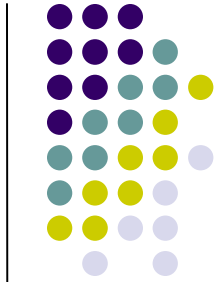


- Why we need Cloud Computing?
- What is Cloud Computing?
- What are the components?
- How to build a Cloud Computing System?
- What questions remain?

References

- [1] M. Armbrust et al., Above the clouds: A Berkeley View of Cloud Computing. *Technical Report, UC Berkeley, 2009.*
- [2] D. Nurmi et al., The Eucalyptus Open-source Cloud-computing System. In *IEEE CCGRID 2009*.
- [3] Brian F. Cooper et al., Building a Cloud for Yahoo!. In *IEEE ICDE 2009.*
- [4] Robert L. Grossman et al., On the Varieties of Clouds for Data Intensive Computing, In *IEEE ICDE 2009*
- [5] James Murty, *Programming Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB, O'Reilly Media*, 2008
- [6] Qi Zhang et.al. Cloud Computing: state-of-the-art and research challenges, In *Journal of Internet Services and Applications, 2010*.

The end!



Thanks!

Questions?