Queen's School of Computing

Queen's University

# Clouds vs Grids

KHALID ELGAZZAR
GOODWIN 531
ELGAZZAR@CS.QUEENSU.CA

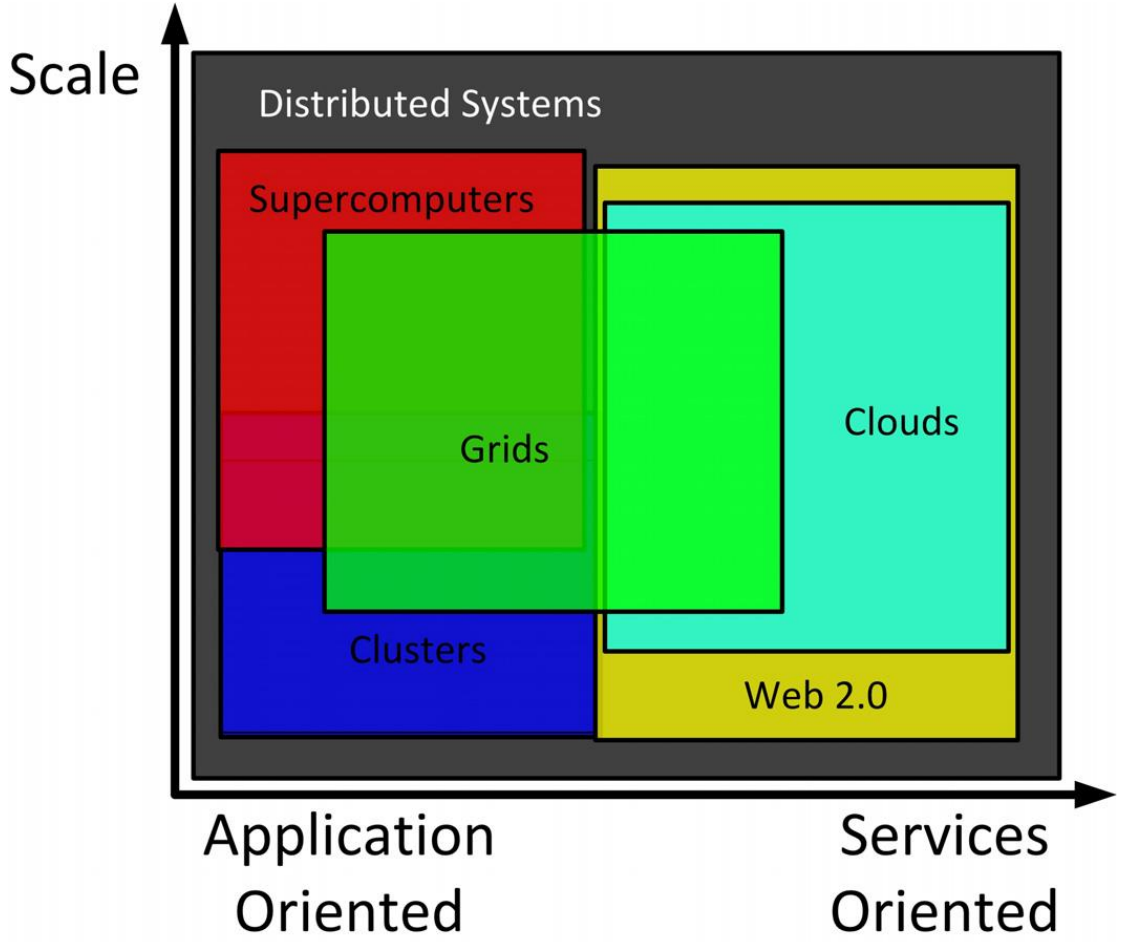
[REF] I Foster, Y Zhao, I Raicu, S Lu, “Cloud computing and grid computing 360-degree compared Grid Computing” Environments Workshop, 2008. GCE'08, 1-10

# Outline

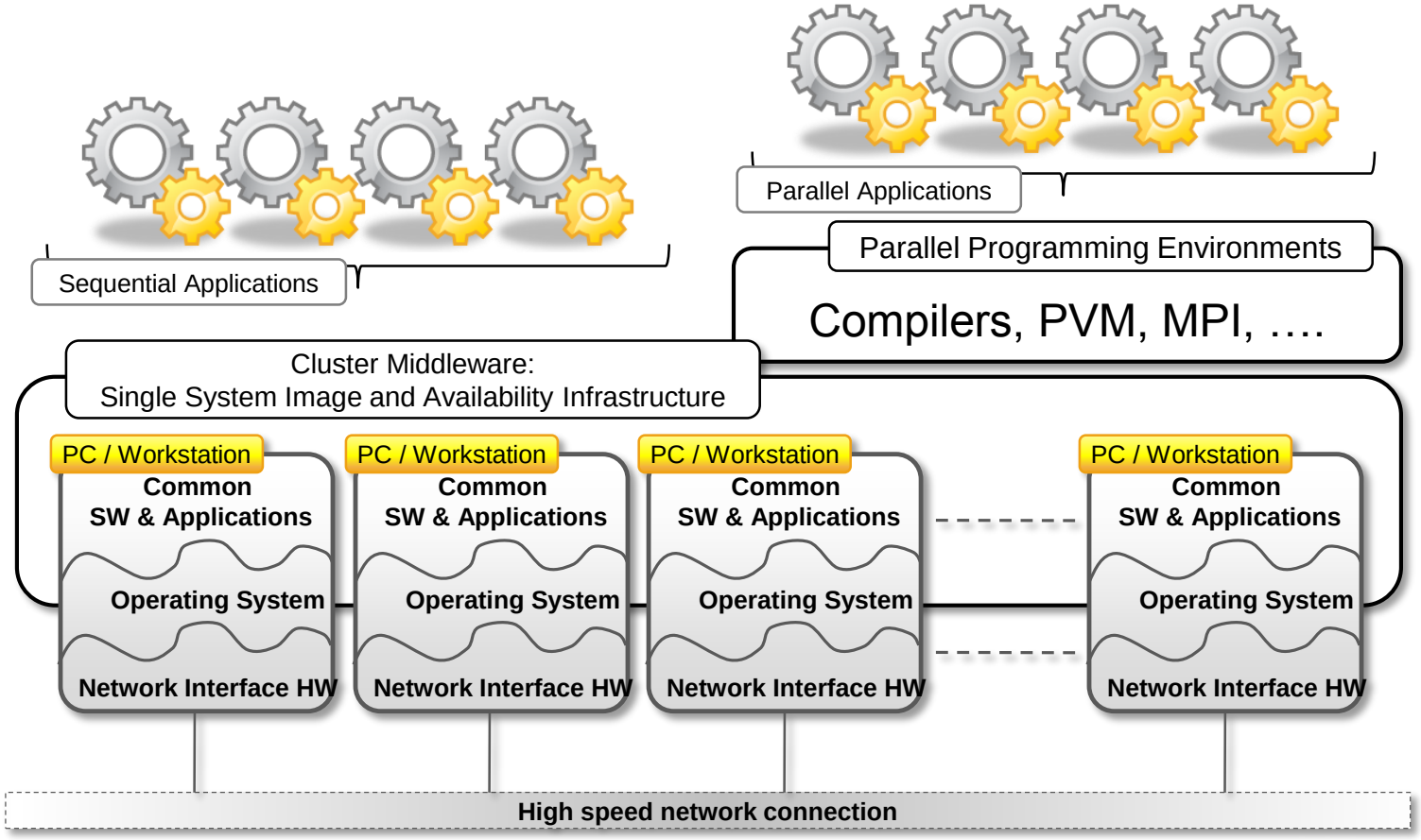
1. Clouds, Grids and Distributed Systems
2. Clouds VS. Grids (side-to-side)
   - Business model
   - Architecture
   - Resource management
   - Programming model
   - Application model
   - Security model

# 1. Clouds, Grids & Distributed Systems

# Clouds, Grids, Distributed Systems

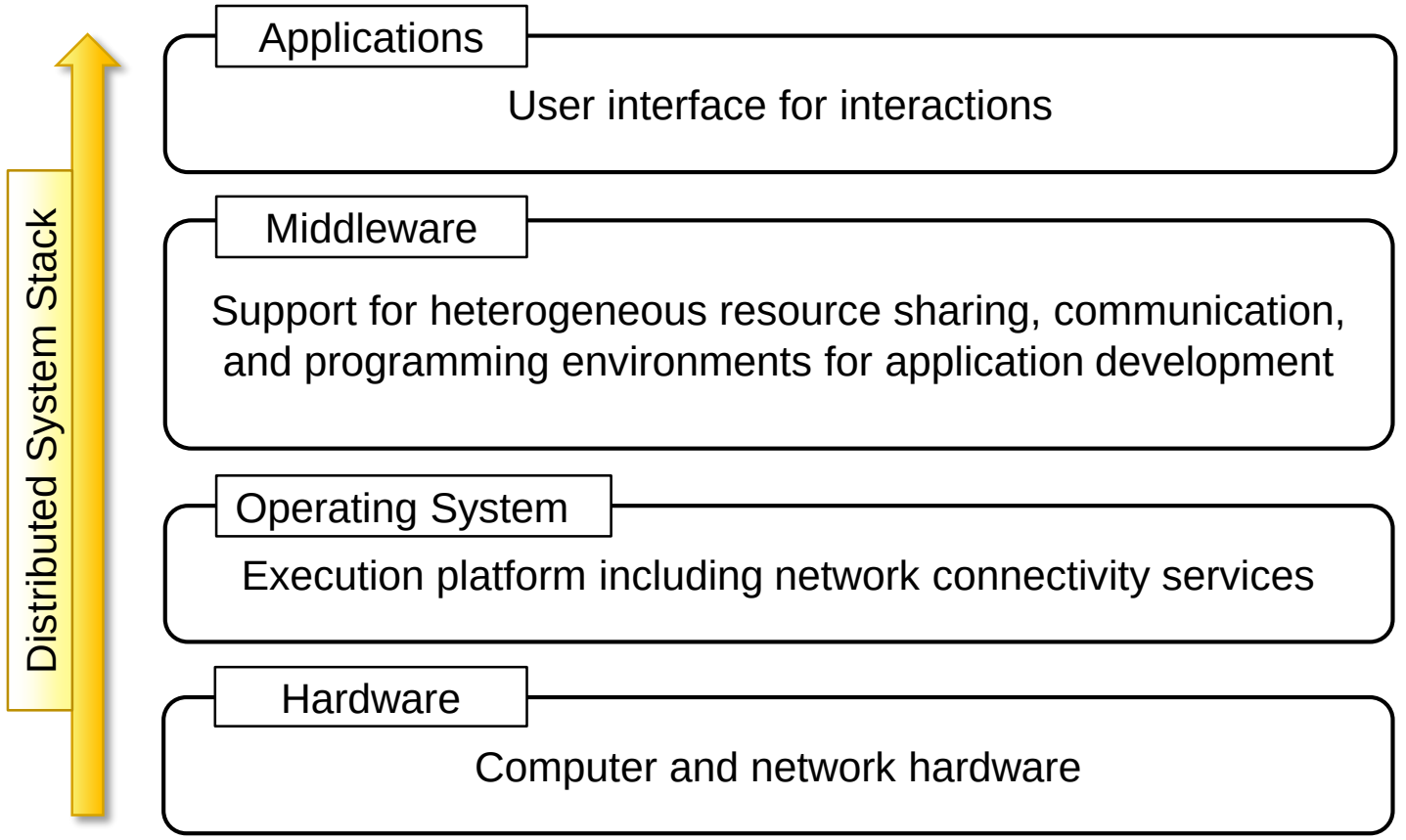


# Sequential Vs . Parallel Applications




R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing Morgan Kaufmann, 2013.

# Distributed System Stack



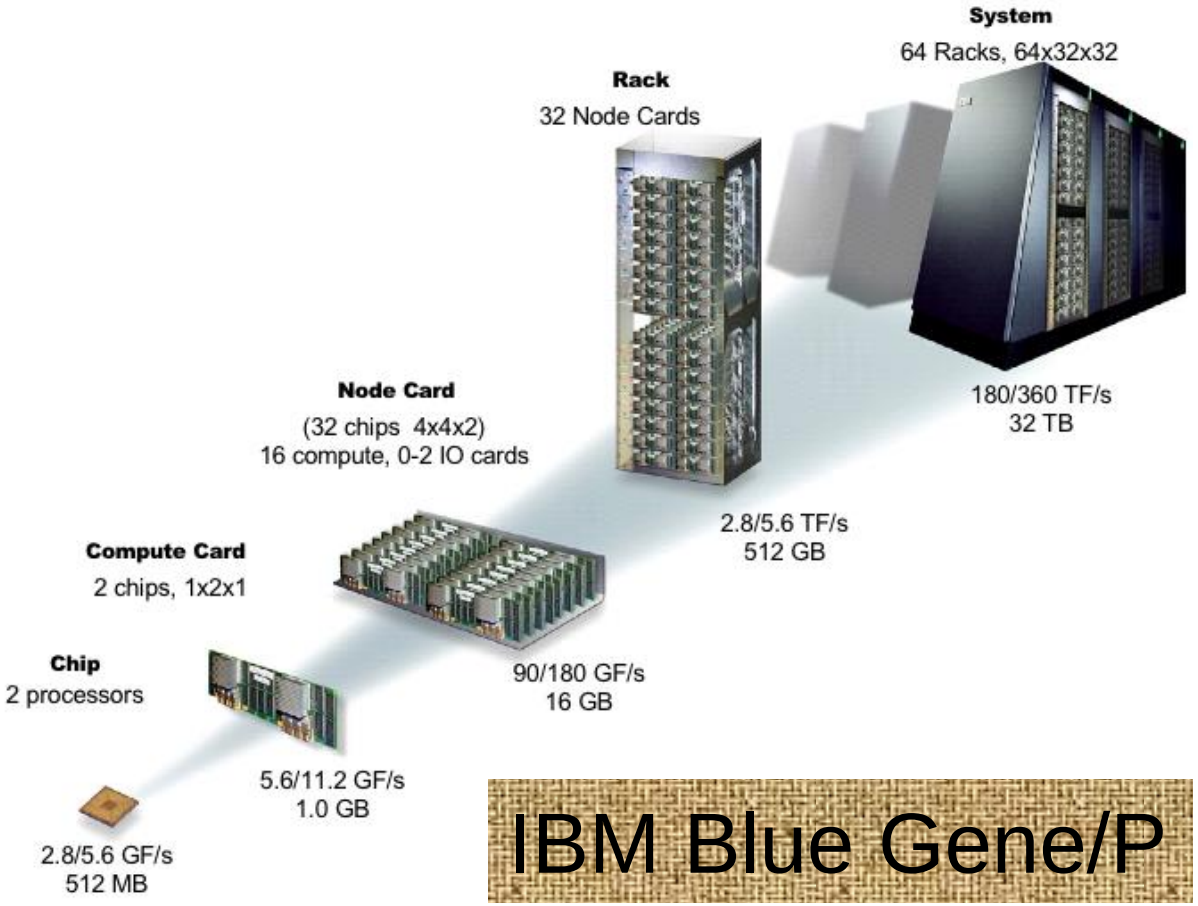

R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing Morgan Kaufmann, 2013.

# Supercomputing

Highly-tuned computer clusters using commodity processors combined with custom network interconnects and customized operating system

e.g. IBM Blue Gene/P

# Supercomputing




IBM Blue Gene/P

Tianhe-2 Supercomputer

# Cluster Computing

*Computer clusters using commodity machines, network interconnects, and operating system*

# Grid Computing

Grid Computing enables resource sharing and coordinated problem solving in virtual organizations (VO) where each VO can consist of either physically distributed institutions or logically related projects/groups.

Builds a uniform computing environment from diverse resources by defining standard network protocols and providing middleware to mediate access to a wide range of heterogeneous resources (eg *GlobusToolkit*).

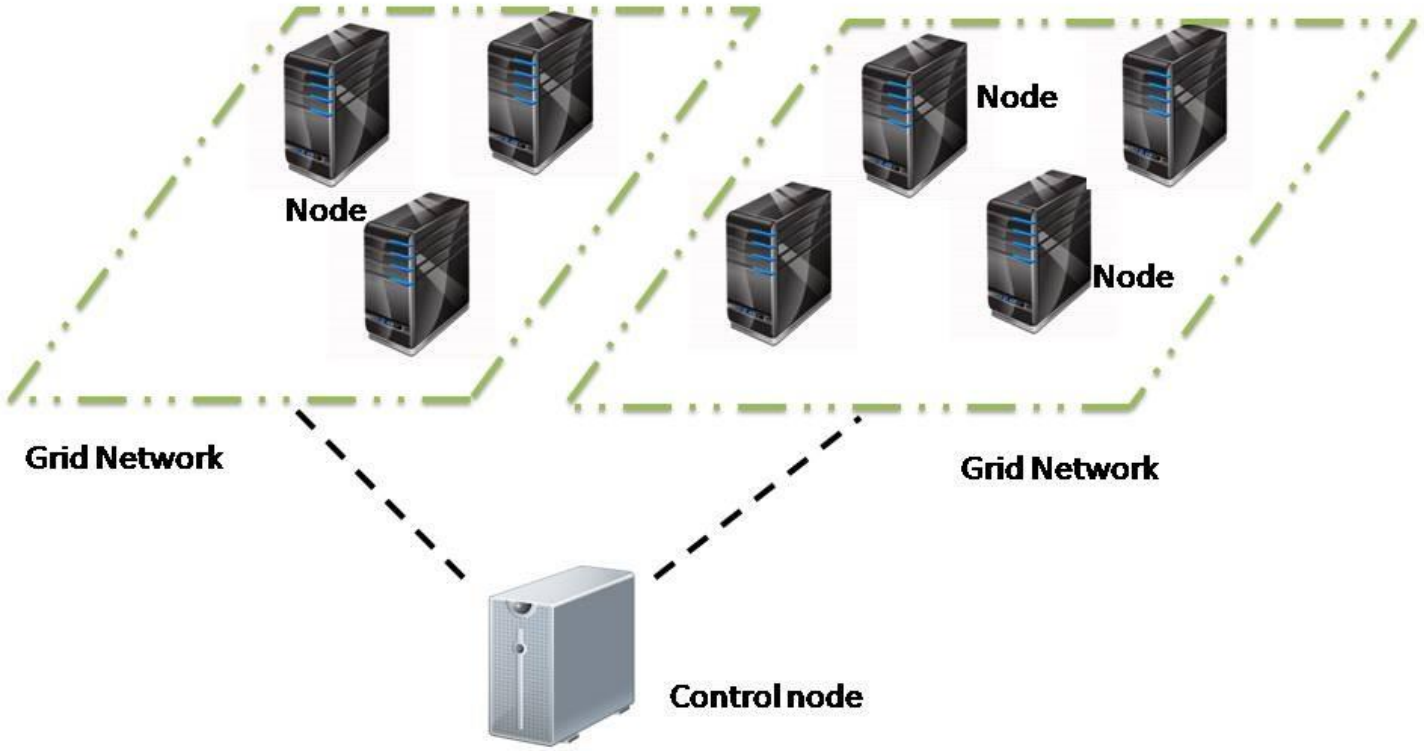
# Grid Computing

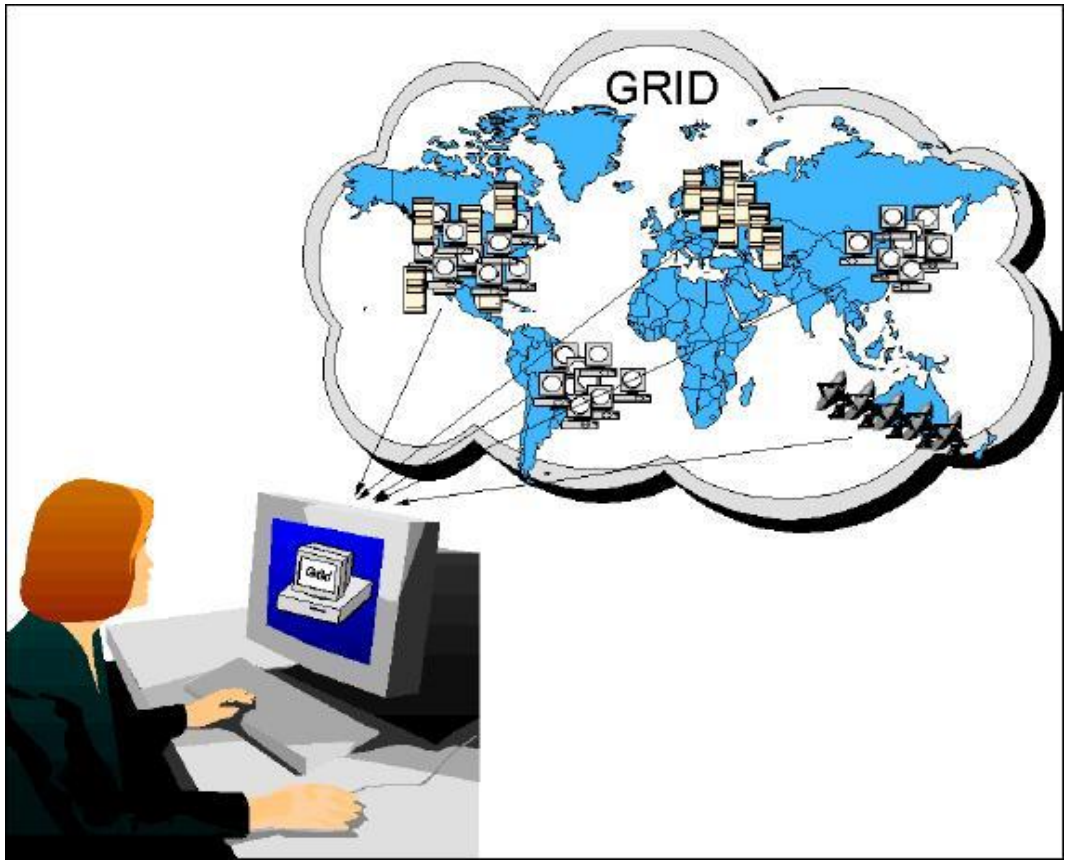
*Grids tend to be composed of multiple clusters, and are typically loosely coupled, heterogeneous, and geographically dispersed*


e.g. TeraGrid

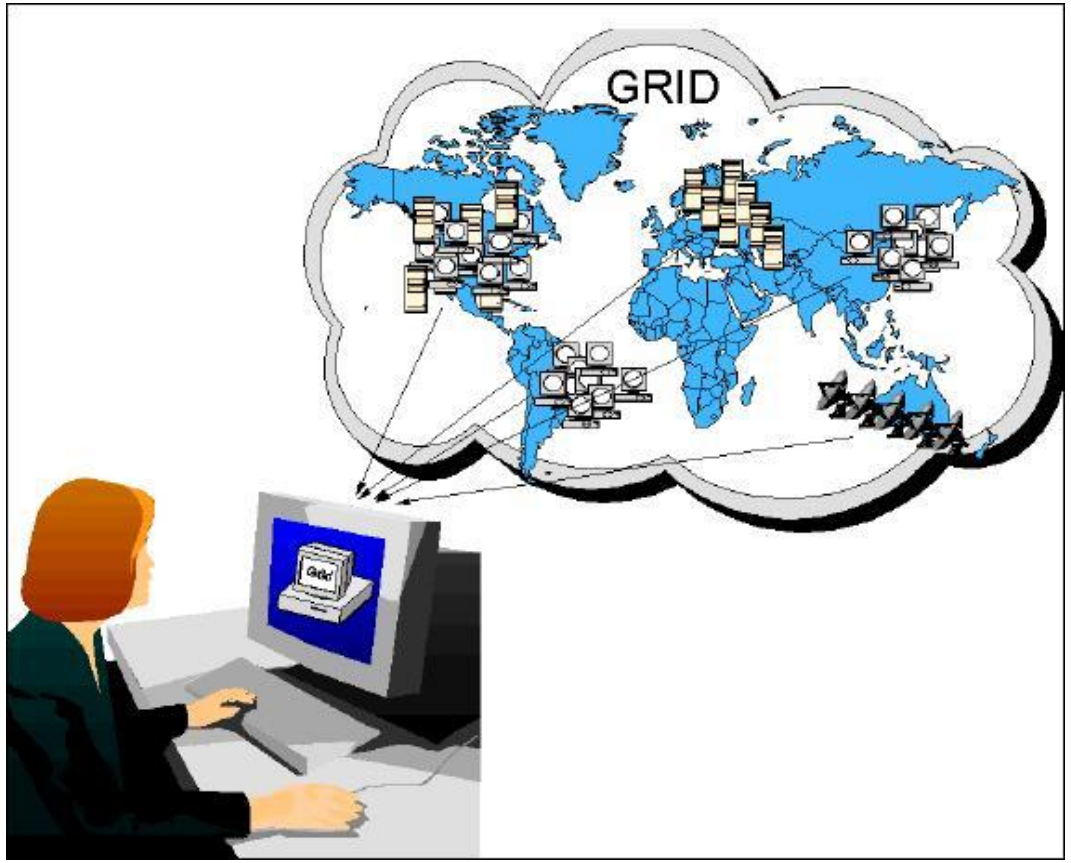
TeraGrid **was** a grid computing infrastructure combining resources at eleven partner sites. The project operated from 2004 through 2011.

# Grid Computing



# Grid VS. Cluster




# What is Cloud Computing?

*"A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet."*

[Foster et al., Cloud Computing and Grid Computing 360-Degree Compared, 2008]

e.g. AMAZON EC2


CBS
MORNING

# How technologists perceive the Cloud


Larry Ellison (Oracle CEO), Wall Street Journal, September 26, 2008

"The interesting thing about Cloud Computing is that we've redefined Cloud Computing to include everything that we already do. . . . I don't understand what we would do differently in the light of Cloud Computing other than change the wording of some of our ads."

# How technologists perceive the Cloud

Andy Isherwood (HP VP of sales), ZDnet News, December 11, 2008

"A lot of people are jumping on the [cloud] bandwagon, but I have not heard two people say the same thing about it. There are multiple definitions out there of "the cloud."

# How technologists perceive the Cloud

Richard Stallman (Advocator of Free Software), The Guardian, September 29, 2008

"It's stupidity. It's worse than stupidity: it's a marketing hype campaign. Somebody is saying this is inevitable — and whenever you hear somebody saying that, it's very likely to be a set of businesses campaigning to make it true."

# Key differences …

*From a hardware point of view, three aspects are new in Cloud Computing:*

- The illusion of infinite computing resources available on demand, thereby eliminating the need for Cloud Computing users to plan far ahead for provisioning.
- The elimination of an up-front commitment by Cloud users, thereby allowing companies to start small and go big on demand.
- The “Pay-As-You-Go” model, enables users to pay per use as needed (e.g., processors by the hour and storage by the day).

# So …

# Is Cloud a new name for Grids?

*IT reinvents itself every five years*


*The answer is complicated…*

**YES**: the vision is the same

- reduce the cost of computing
- increase reliability
- increase flexibility (transitioning from self-operation to third party)

# Is Cloud a new name for Grids?

**NO**: things are different than 10 years ago

- New needs to analyze massive data, increased demand for computing
- Commodity clusters are expensive to operate
- We have low-cost virtualization
- Billions of dollars being spent by Amazon, Google, and Microsoft to create real commercial large-scale systems with hundreds of thousands of computers
- Only need a credit card to get on-demand access to *infinite computers*

# Is Cloud a new name for Grids?

**NEVERTHELESS**: same problems but different details

- How to manage large facilities
- How to discover, request, and use resources
- How to implement and execute parallel Computations

# 2. Clouds VS. Grids (side-to-side)

Business model
Architecture
Resource management

Programming model
Application model
Security model

# Clouds VS. Grids

**Business model** | Programming model
Architecture | Application model
Resource management | Security model

- Industry (i.e. Amazon) funded the initial Clouds
- Large user base in common people, small businesses, large businesses, and some open science research
- Utility computing => real money

## Clouds

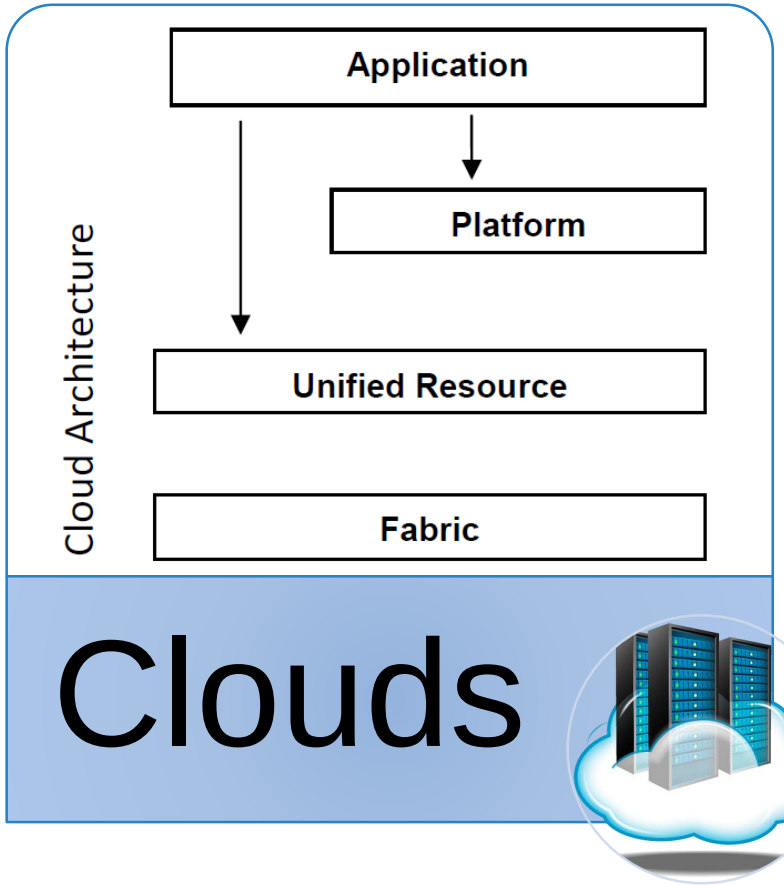
- Largest Grids funded by government
- Largest user-base in academia and government labs to drive scientific computing
- Project-oriented: assigned a number of service units

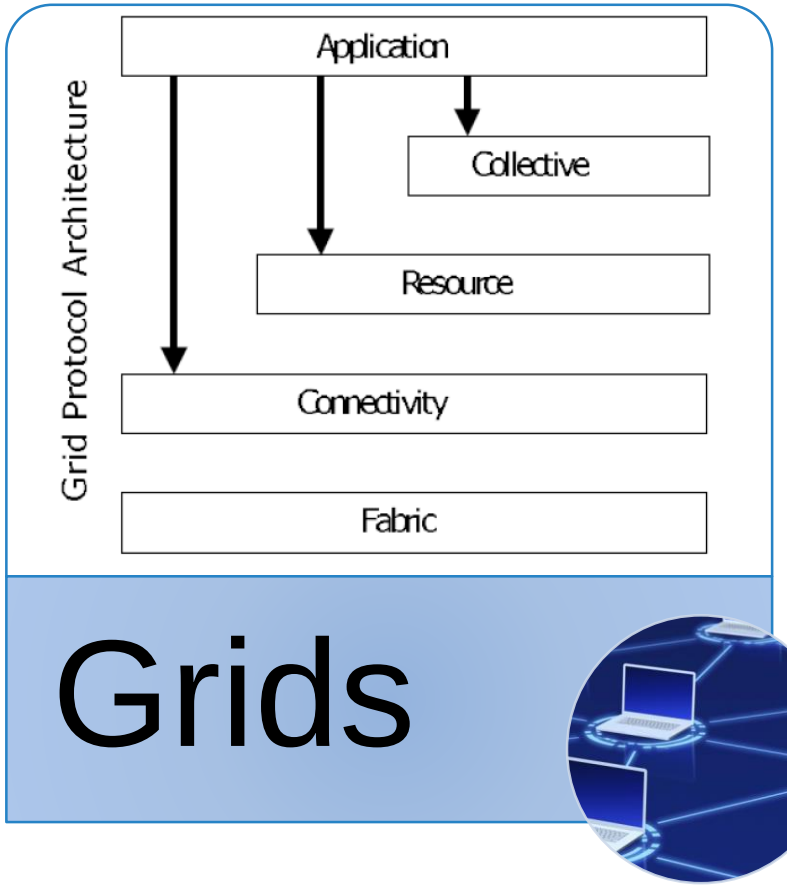
## Grids

# Clouds VS. Grids

Business model
**Architecture**
Resource management

Programming model
Application model
Security model



# Clouds VS. Grids

Business model
Architecture
**Resource management**
Programming model
Application model
Security model

Compute model
Data model
Data locality
Virtualization
Monitoring
Provenance

# Clouds VS. Grids

**RESOURCE MANAGEMENT**

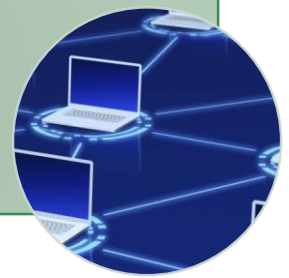
| | |
|---|---|
| **Compute model** | Virtualization |
| Data model | Monitoring |
| Data locality | Provenance |

- Shared resources acquired on demand
- Interactive applications can be supported by guaranteed QoS is a challenge!

Clouds

- Batch-oriented
- Required resources scheduled

Grids



# Clouds VS. Grids

**RESOURCE MANAGEMENT**

| | |
|---|---|
| Compute model | Virtualization |
| **Data model** | Monitoring |
| Data locality | Provenance |

- Specialized shared file systems emphasizing scalability and availability (automatic replication)
- Data locality supported so processing can go to data.

Clouds

- Data grids specifically designed for data-intensive applications
- Virtual data concept provides location, materialization & representation transparencies
- Shared file systems
- Data locality not easily supported

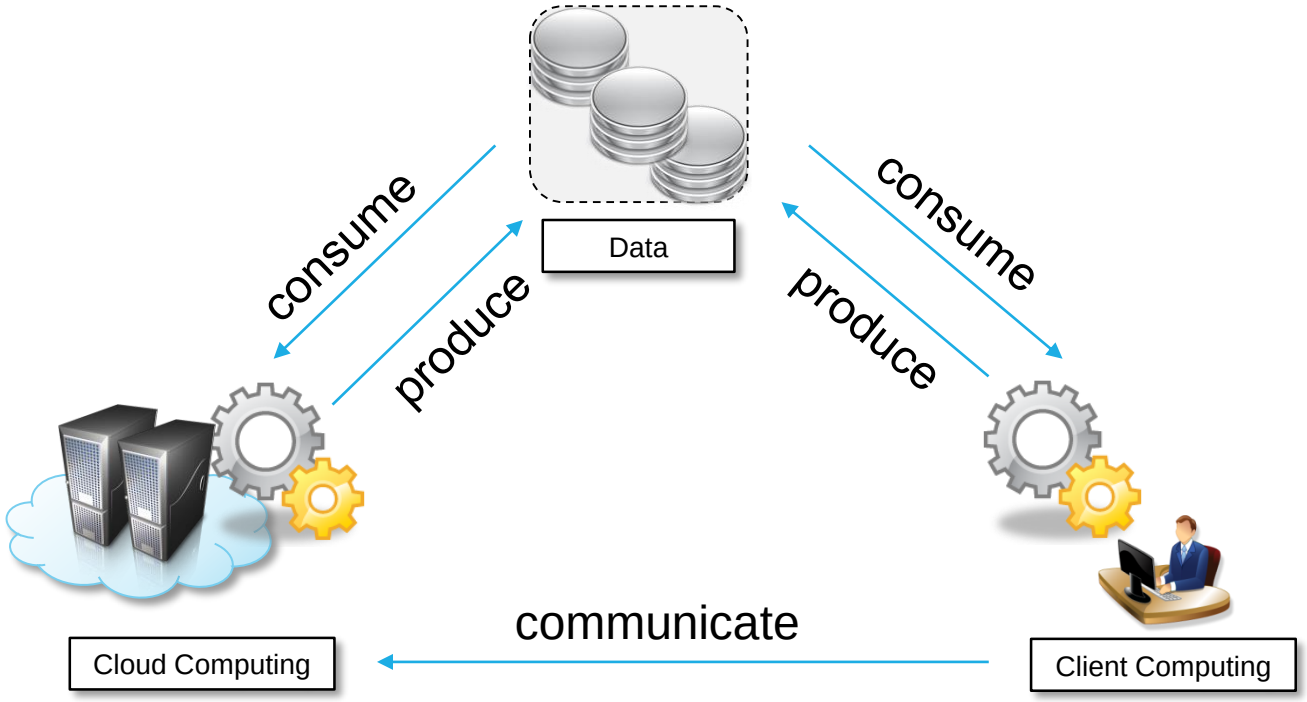
Grids

# Future Application Trend

For security reasons, people might not be willing to run mission-critical applications on the Cloud and send sensitive data to the Cloud for processing and storage

Users want to get their things done even when the Internet and Cloud are down or the network communication is slow

With the advances of multi-core technology, the coming decade will bring the possibilities of having a desktop supercomputer with 100s to 1000s of hardware threads/cores.



# Clouds VS. Grids

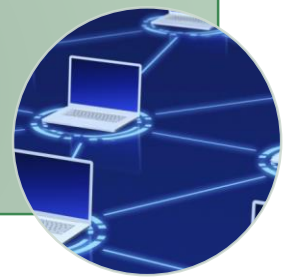


- High focus on storing and replicating data near to the associated compute unit

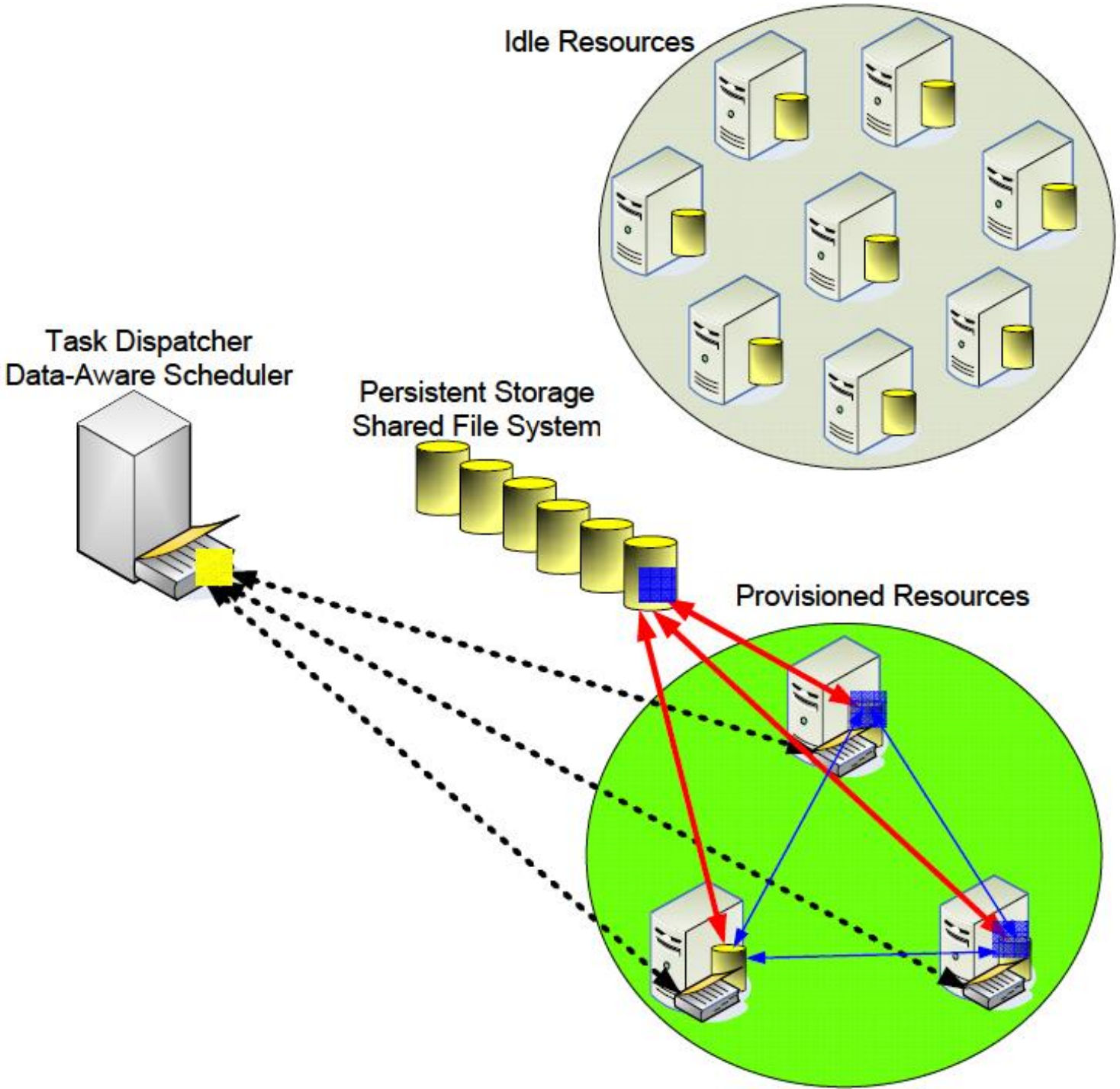
Clouds

- Data is stored in shared file systems, where data locality cannot easily be applied. However data-aware schedulers dramatically improve performance

Grids



- Resource acquired in response to demand
- Data and applications diffuse from archival storage to newly acquired resources
- Resource “caching” allows faster responses to subsequent requests
  - Cache Eviction Strategies: RANDOM, FIFO, LRU, LFU
- Resources are released when demand drops



# Clouds VS. Grids

**RESOURCE MANAGEMENT**

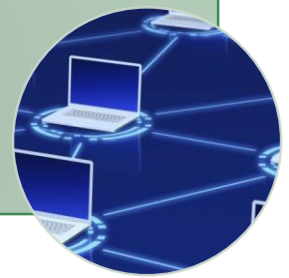
| | |
|---|---|
| Compute model | **Virtualization** |
| Data model | Monitoring |
| Data locality | Provenance |

- Heavy reliance on virtualization
- Provides abstraction & encapsulation needed for dynamic resource and application management
- Supports cost-effective use of cloud's physical resources

Clouds

- Not used much in grids
- Applications given physical resources on a scheduled basis

Grids



# Clouds VS. Grids

**RESOURCE MANAGEMENT**

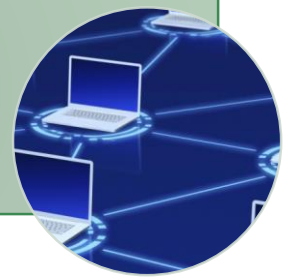
| | |
|---|---|
| Compute model | Virtualization |
| Data model | **Monitoring** |
| Data locality | Provenance |

- Virtualization poses challenges to fine-grained control over monitoring
- Service-oriented view means resources below service API are not visible
- Monitoring may not be as important because of abstractions

Clouds

- Grid trust model allows users via their identity delegation to access and browse resources at different sites
- Resources not highly abstracted & virtualized

Grids



# Clouds VS. Grids

**RESOURCE MANAGEMENT**

| | |
|---|---|
| Compute model | Virtualization |
| Data model | Monitoring |
| Data locality | **Provenance** |

**Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.** [Wikipedia]

# Clouds VS. Grids

**RESOURCE MANAGEMENT**

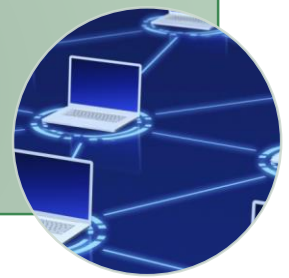
| | |
|---|---|
| Compute model | Virtualization |
| Data model | Monitoring |
| Data locality | **Provenance** |

- Still unexplored
- Scalable provenance querying and secure access to provenance info are still open problems for both grids and clouds

## Clouds

- Built into a workflow system to support discovery and reproducibility of scientific results (Chimera, Swift, Kepler, VIEW etc.)

## Grids



# Clouds VS. Grids



- MapReduce is most popular parallel programming model and runtime
- Mash-ups & scripting languages (Javascript, PHP, Python) used instead of workflows because of interoperability challenges
- AWS and Microsoft Azure use Web services APIs

**Clouds**

- Complicated by issues like multiple administrative domains, resource heterogeneity, etc
- MPI (Message Passing Interface)
- Heavy use of workflow tools to manage large sets of loosely-coupled tasks
- Focus on management rather than on interprocess communication

**Grids**

# Clouds VS. Grids

Business model | Programming model
Architecture | **Application model**
Resource management | Security model

- Traditionally can support same apps as grid except HPC (due to low latency needs) but this is changing
- Interactive, loosely-coupled, transaction-oriented apps

## Clouds

- Batch-oriented apps
- Support High-Performance Computing (HPC) through High Throughput Computing (HTC)
- Support workflows of loosely-coupled applications
- Scientific gateways are also popular

## Grids

# Clouds VS. Grids



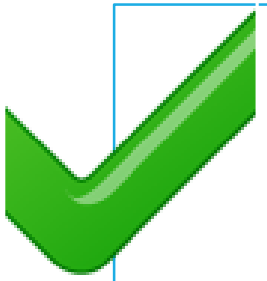
- Clouds currently more homogeneous and single provider so security simpler
- Virtualization adds level of security
- Still an important concern for cloud users
- Email address & credit card gets you an account

Clouds

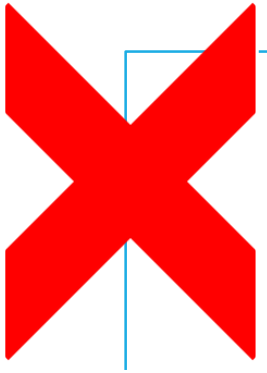
- Built on assumptions of heterogeneous and dynamic resources and multiple admin domains
- Key issues are single sign-on; privacy, integrity & segregation
- Stricter procedure to acquire an account

Grids

# Summary



Clouds and Grids share commonality in their vision, architecture and technology



Differ in aspects such as security, programming model, business model, compute model, data model, applications, and abstractions.

# Looking ahead…

*Parallel evolution in power and computing utilities*

*Need improved support for:*

- On-demand provisioning & configuration of "virtual systems"
- Dynamically manage applications across multiple providers
- Managing distributed computations & underlying resources