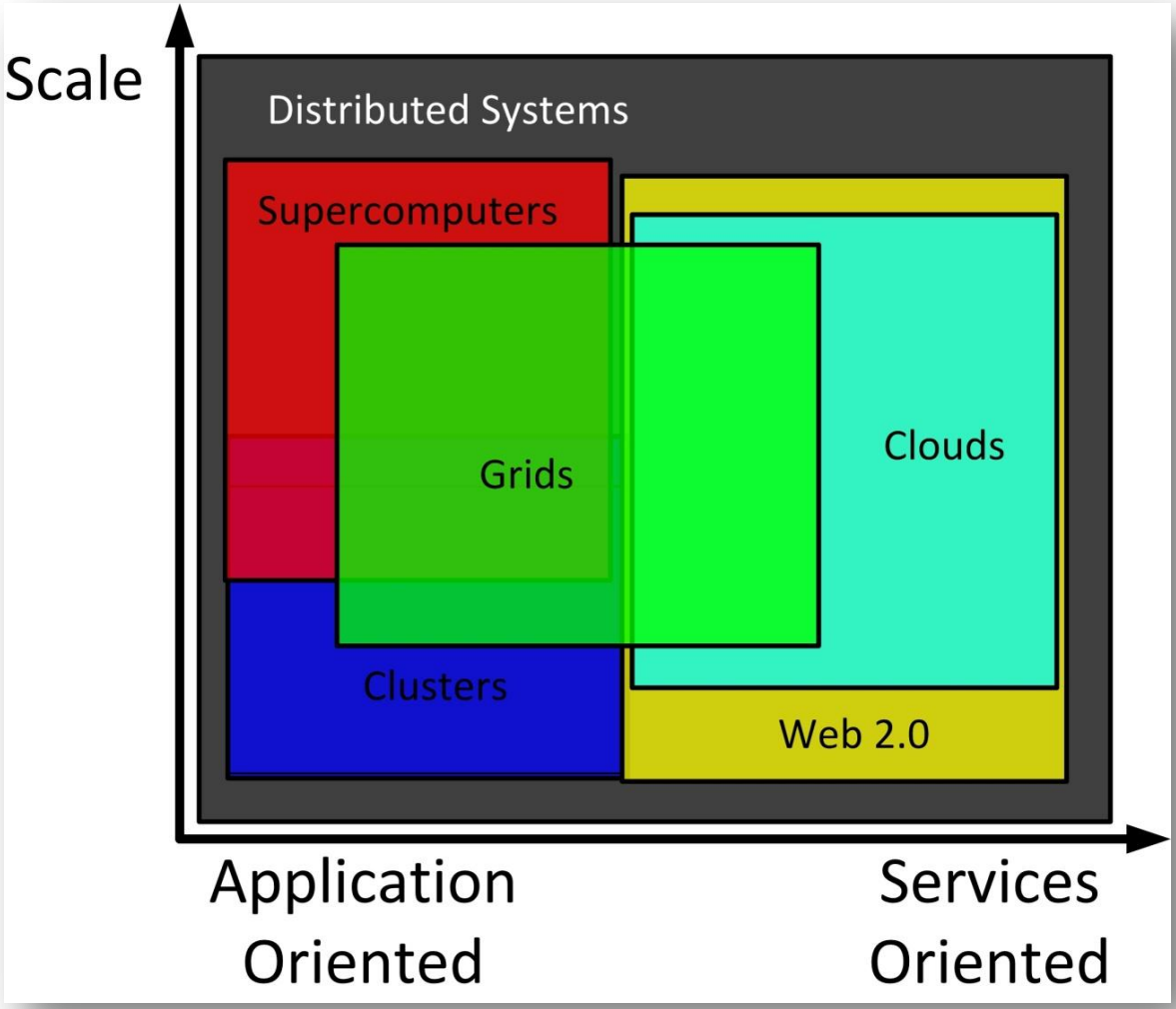
Distributed Systems:
Clusters, Supercomputers, Grids, and Clouds

Ioan Raicu
Center for Ultra-scale Computing and Information Security
Department of Electrical Engineering & Computer Science
Northwestern University

EECS 395 / EECS 495
Hot Topics in Distributed Systems: Data-Intensive Computing
January 14th, 2010

Clusters, Grids, Supercomputers

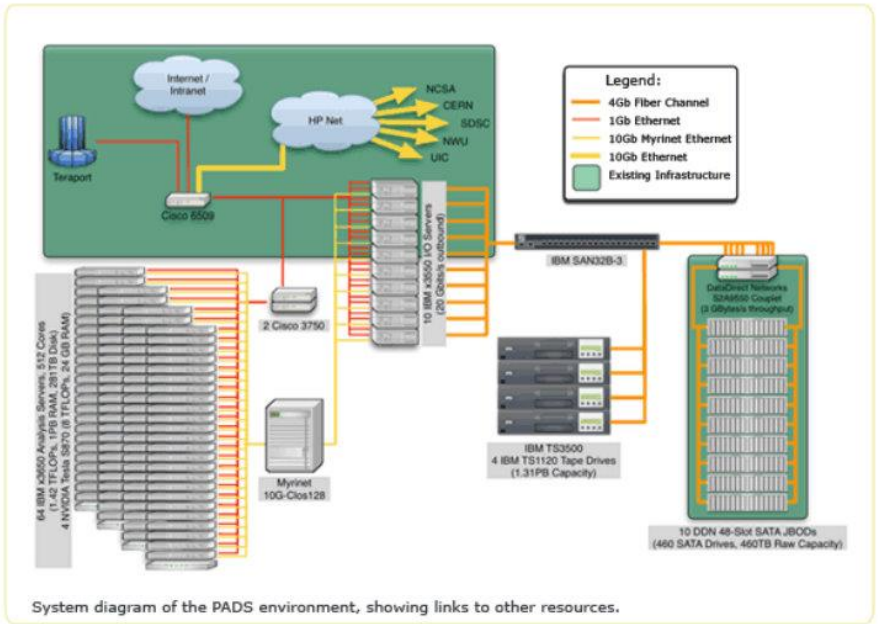


[GCE08] "Cloud Computing and Grid Computing 360-Degree Compared"

Cluster Computing: PADS

Computer clusters using commodity processors, network interconnects, and operating systems.

PADS



System diagram of the PADS environment, showing links to other resources.

PADS is a petabyte (10^{15}-byte)-scale online storage server capable of sustained multi-gigabyte/s I/O performance, tightly integrated with a 9 teraflop/s computing resource and multi-gigabit/s local and wide area networks. Its hardware and associated software enables the reliable storage of, access to, and analysis of massive datasets by both local users and the national scientific community.

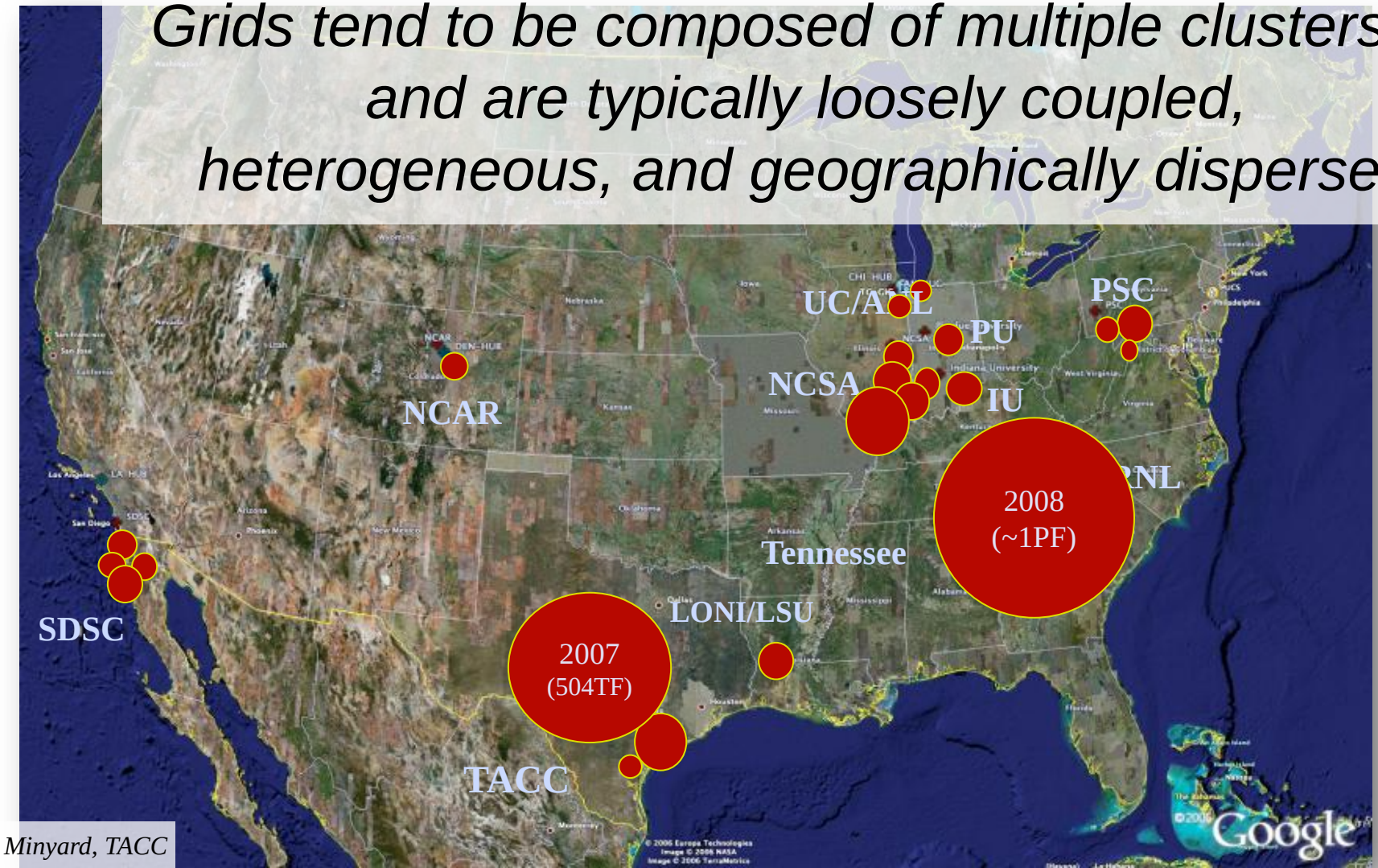
The PADS design results from a study of the storage and analysis requirements of participating groups in astrophysics and astronomy, computer science, economics, evolutionary and organismal biology, geosciences, high-energy physics, linguistics, materials science, neuroscience, psychology, and sociology. For these groups, PADS represents a significant opportunity to look at their data in new ways, enabling new scientific insights. The infrastructure also encourages new collaborations across disciplines. PADS is also a vehicle for computer science research into active data store systems, and provides rich data on which to investigate new techniques. Results will be made available as open source software.

The PADS project is supported in part by the National Science Foundation under grant OCI-0821678 and by The University of Chicago.

PADSstatus myPADS

Grid Computing: TeraGrid

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



Tommy Minyard, TACC

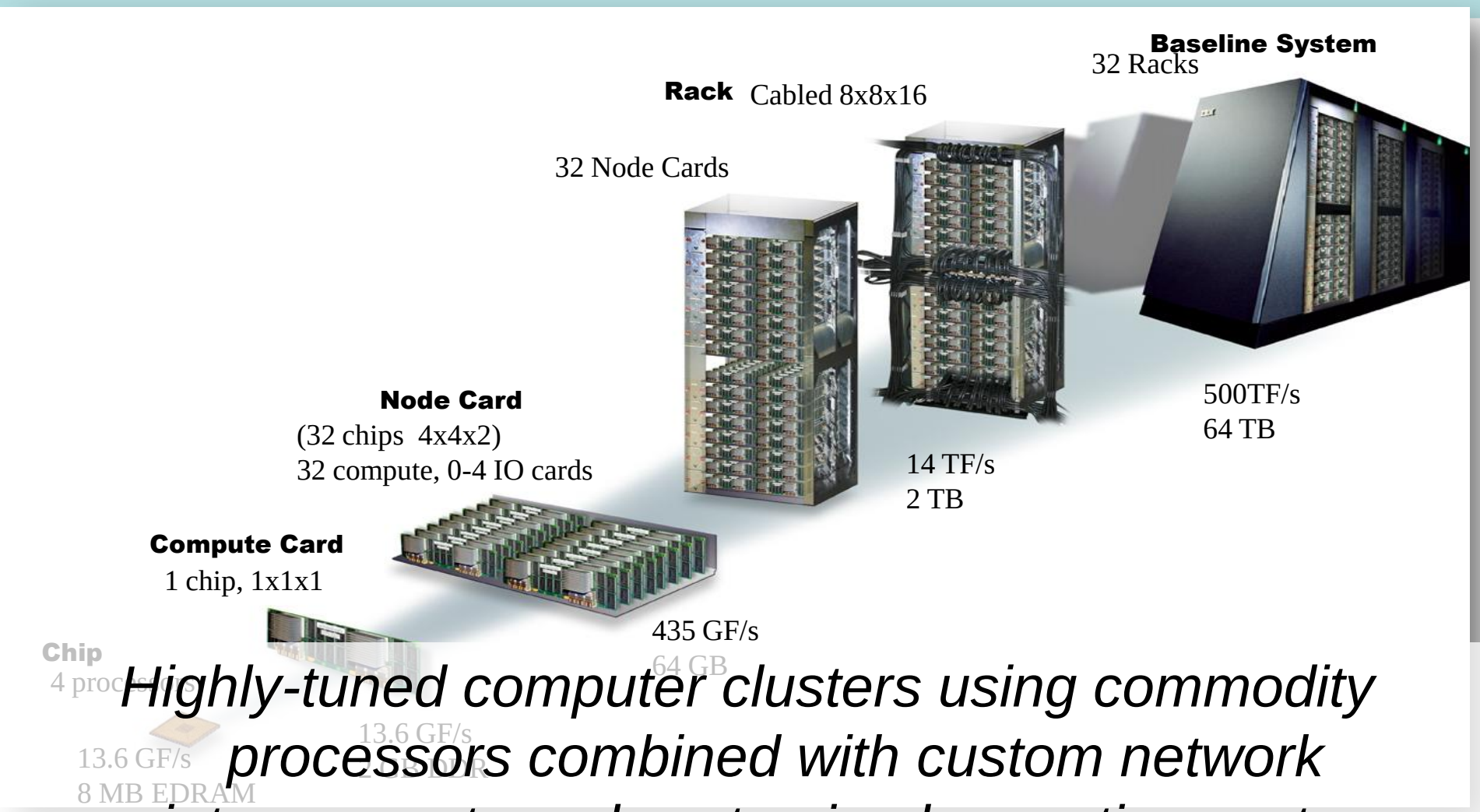
What is the TeraGrid?

- An instrument (cyberinfrastructure) that delivers high-end IT resources - storage, computation, visualization, and data/service hosting - almost all of which are UNIX-based under the covers; some hidden by Web interfaces
  - 20 Petabytes of storage (disk and tape)
  - over 100 scientific data collections
  - 750 TFLOPS (161K-cores) in parallel computing systems and growing
  - Support for Science Gateways
- The largest individual cyberinfrastructure facility funded by the NSF, which supports the national science and engineering research community
- Something you can use without financial cost - allocated via peer review (and without double jeopardy)

Major Grids

- TeraGrid (TG)
- Open Science Grid (OSG)
- Enabling Grids for E-sciencE (EGEE)
- LHC Computing Grid from CERN
- Grid Middleware
  - Globus Toolkit
  - Unicore

Supercomputing: IBM Blue Gene/P



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

Top 10 Supercomputers from Top500

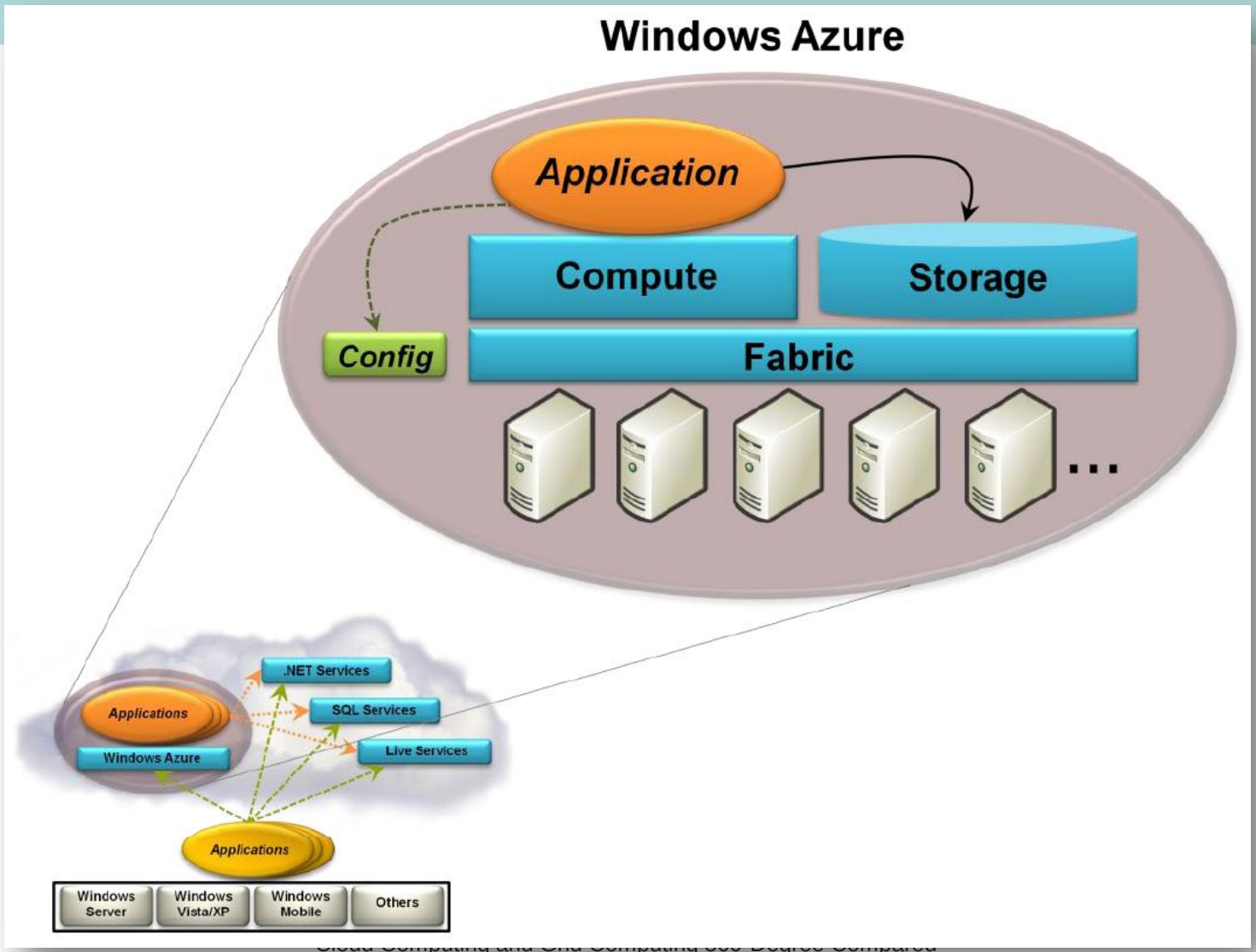
- Cray XT4 & XT5
  - Jaguar #1
  - Kraken #3
- IBM BladeCenter Hybrid
  - Roadrunner #2
- IBM BlueGene/L & BlueGene/P
  - Jugene #4
  - Intrepid #8
  - BG/L #7
- NUDT (GPU based)
  - Tianhe-1 #5
- SGI Altix ICE
  - Plaiedas #6
- Sun Constellation
  - Ranger #9
  - Red Sky #10

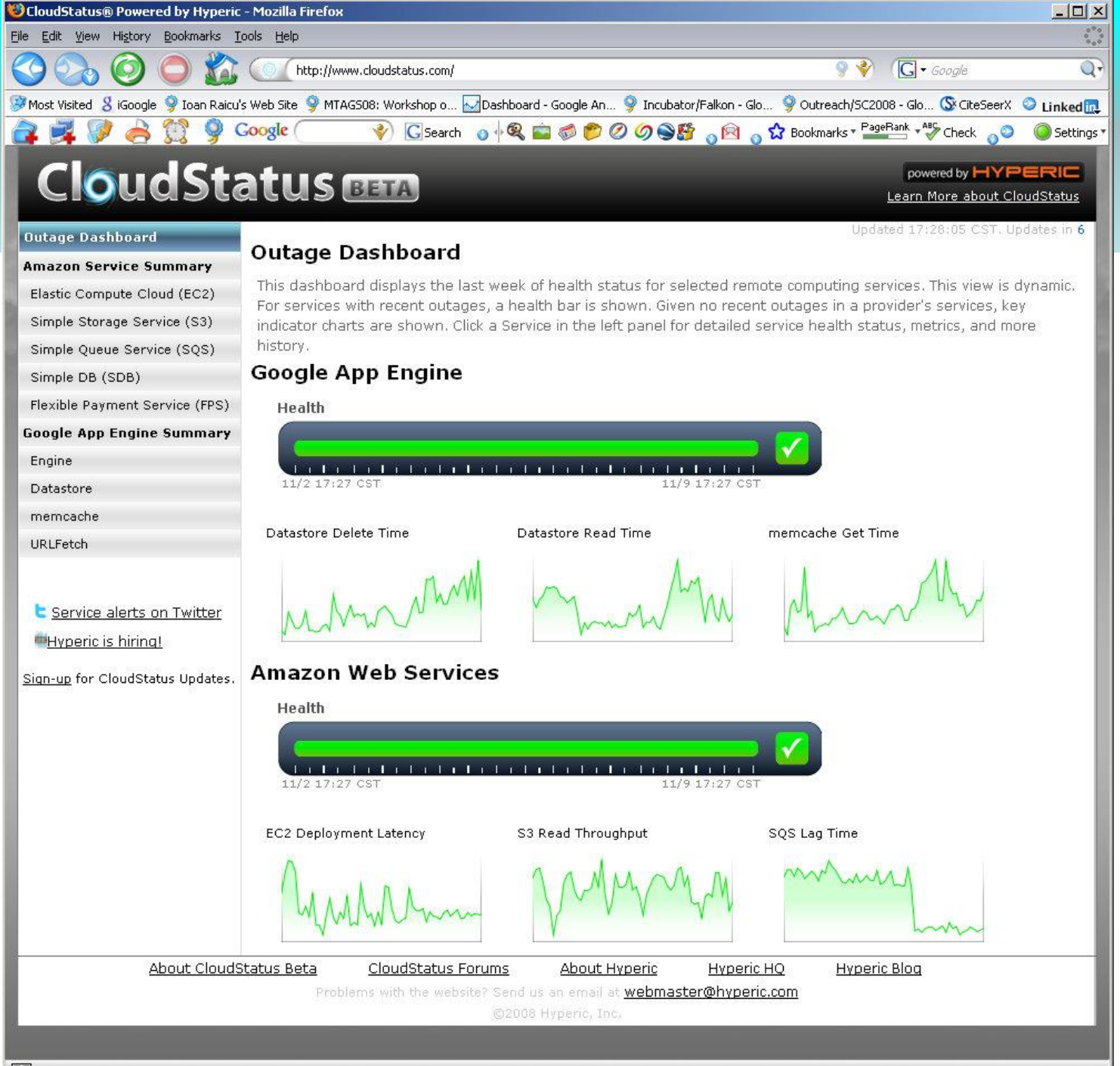
Cloud Computing

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****.*

e.g. Amazon EC2

Clouds: Windows Azure




CloudStatus® Powered by Hyperic - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.cloudstatus.com/
CloudStatus BETA
powered by HYPERIC
Learn More about CloudStatus
Outage Dashboard
Amazon Service Summary
Elastic Compute Cloud (EC2)
Simple Storage Service (S3)
Simple Queue Service (SQS)
Simple DB (SDB)
Flexible Payment Service (FPS)
Google App Engine Summary
Engine
Datastore
memcache
URLFetch
Service alerts on Twitter
Hyperic is hiring!
Sign-up for CloudStatus Updates.
Updated 17:28:05 CST. Updates in 6
Outage Dashboard
This dashboard displays the last week of health status for selected remote computing services. This view is dynamic. For services with recent outages, a health bar is shown. Given no recent outages in a provider's services, key indicator charts are shown. Click a Service in the left panel for detailed service health status, metrics, and more history.
Google App Engine
Health
11/2 17:27 CST
11/9 17:27 CST
Datastore Delete Time
Datastore Read Time
memcache Get Time
Amazon Web Services
Health
11/2 17:27 CST
11/9 17:27 CST
EC2 Deployment Latency
S3 Read Throughput
SQS Lag Time
About CloudStatus Beta
CloudStatus Forums
About Hyperic
Hyperic HQ
Hyperic Blog
Problems with the website? Send us an email at webmaster@hyperic.com
©2008 Hyperic, Inc.
Done

Major Cloud Middleware

- Google App Engine
  - Engine, Datastore, memcache
- Amazon
  - EC2, S3, SQS, SimpleDB
- Microsoft Azure
- Nimbus
- Eucalyptus
- Salesforce

So is "Cloud Computing" just a new name for Grid?

- IT reinvents itself every five years
- The answer is complicated…

- **YES**: the vision is the same
  - to reduce the cost of computing
  - increase reliability
  - increase flexibility by transitioning from self operation to third party

So is "Cloud Computing" just a new name for Grid?

- **NO**: things are different than they were 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
  - The prospect of needing only a credit card to get on-demand access to *infinite computers is exciting; *infinite<O(1000)

So is "Cloud Computing" just a new name for Grid?

- **YES:** the problems are mostly the same
  - How to manage large facilities
  - Define methods to discover, request, and use resources
  - How to implement and execute parallel computations
  - Details differ, but issues are similar

Outline

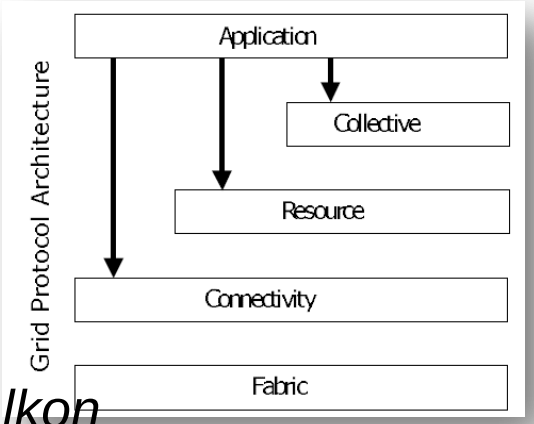
- Business model
- Architecture
- Resource management
- Programming model
- Application model
- Security model

Business Model

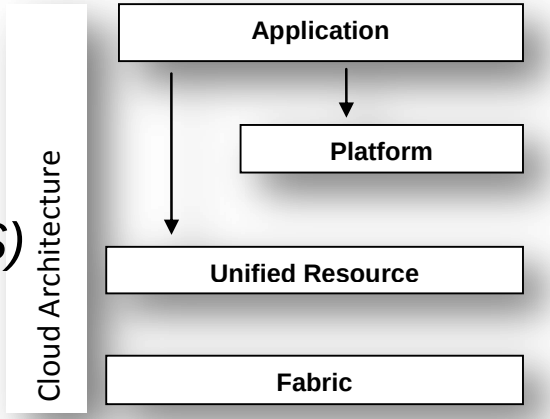
- Grids:
  - Largest Grids funded by government
  - Largest user-base in academia and government labs to drive scientific computing
  - Project-oriented: service units
- Clouds:
  - Industry (i.e. Amazon) funded the initial Clouds
  - Large user base in common people, small businesses, large businesses, and a bit of openn science research
  - Utility computing: real money

Architecture

- Grids:
  - Application: *Swift, Grid portals (NVO)*
  - Collective layer: *MDS, Condor-G, Nimrod-G*
  - Resource layer: *GRAM, Falkon, GridFTP*
  - Connectivity layer: *Grid Security Infrastructure*
  - Fabric layer: *GRAM, PBS, SGE, LSF, Condor, Falkon*



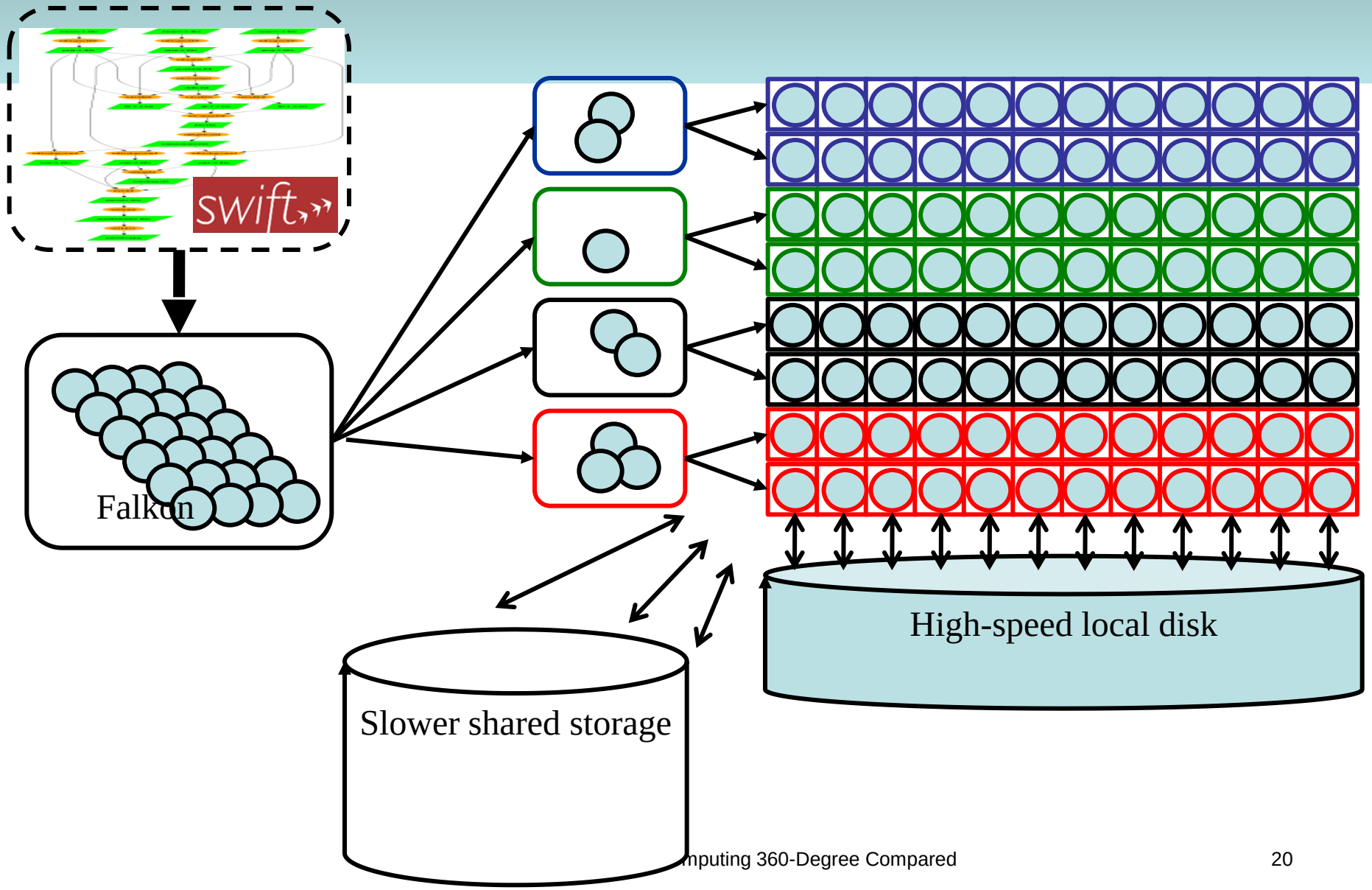
- Clouds:
  - Application Layer: *Software as a Service (SaaS)*
  - Platform Layer: *Platform as a Service (PaaS)*
  - Unified Resource: *Infrastructure as a Service (IaaS)*
  - Fabric: *IaaS*



Resource Management

- Compute Model
  - batch-scheduled vs. time-shared
- Data Model
  - Data Locality
  - Combining compute and data management
- Virtualization
  - Slow adoption vs. central component
- Monitoring
- Provenance

Managing 160K CPUs

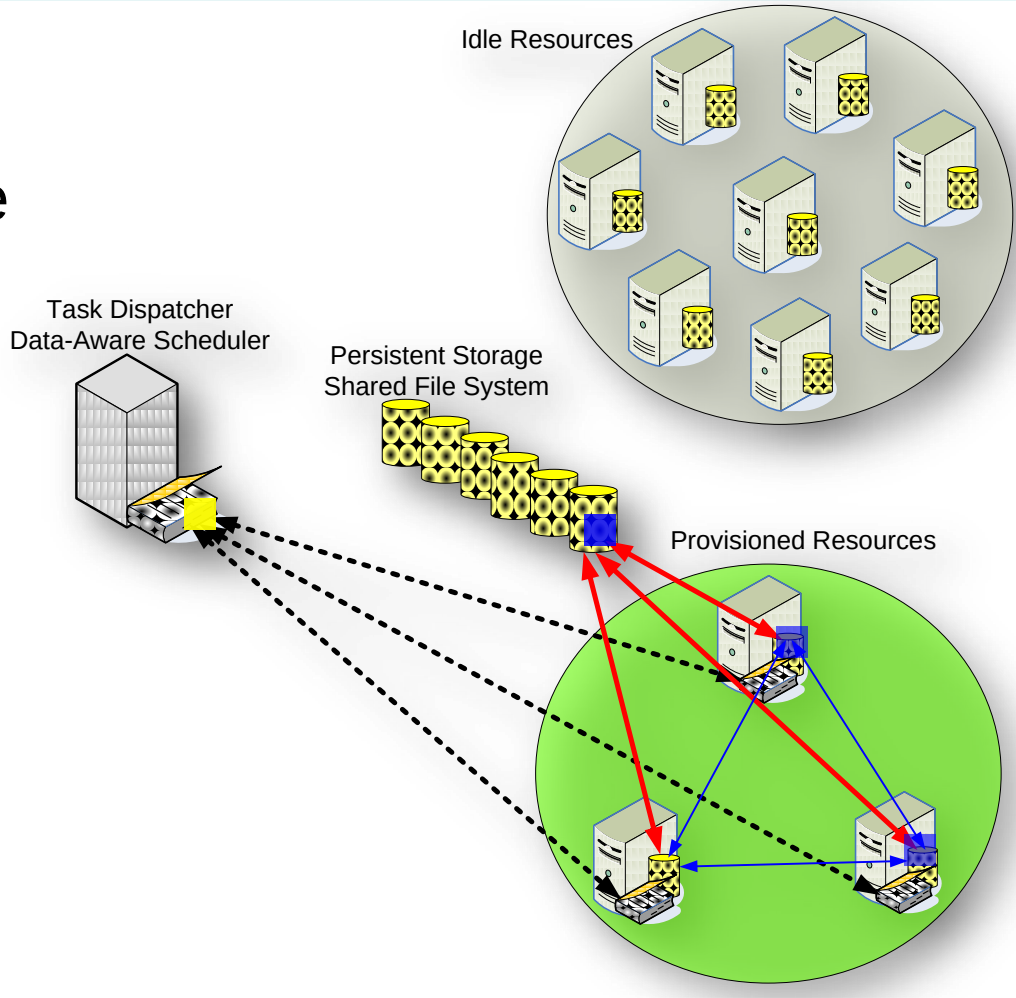


Resource Management

- Compute Model
  - batch-scheduled vs. time-shared
- Data Model
  - Data Locality
  - Combining compute and data management
- Virtualization
  - Slow adoption vs. central component
- Monitoring
- Provenance

Data Diffusion

- 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



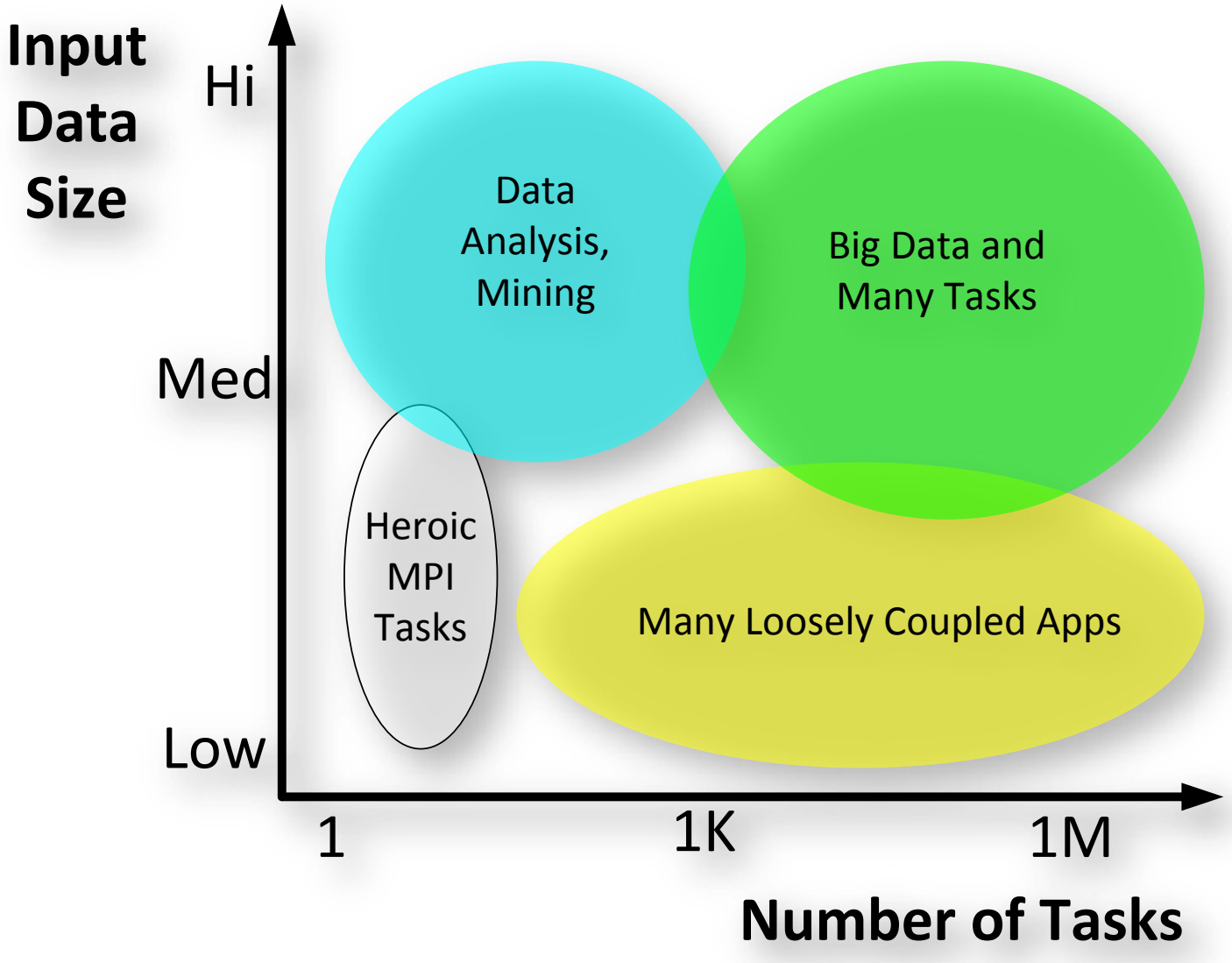
Resource Management

- Compute Model
  - batch-scheduled vs. time-shared
- Data Model
  - Data Locality
  - Combining compute and data management
- Virtualization
  - Slow adoption vs. central component
- Monitoring
- Provenance

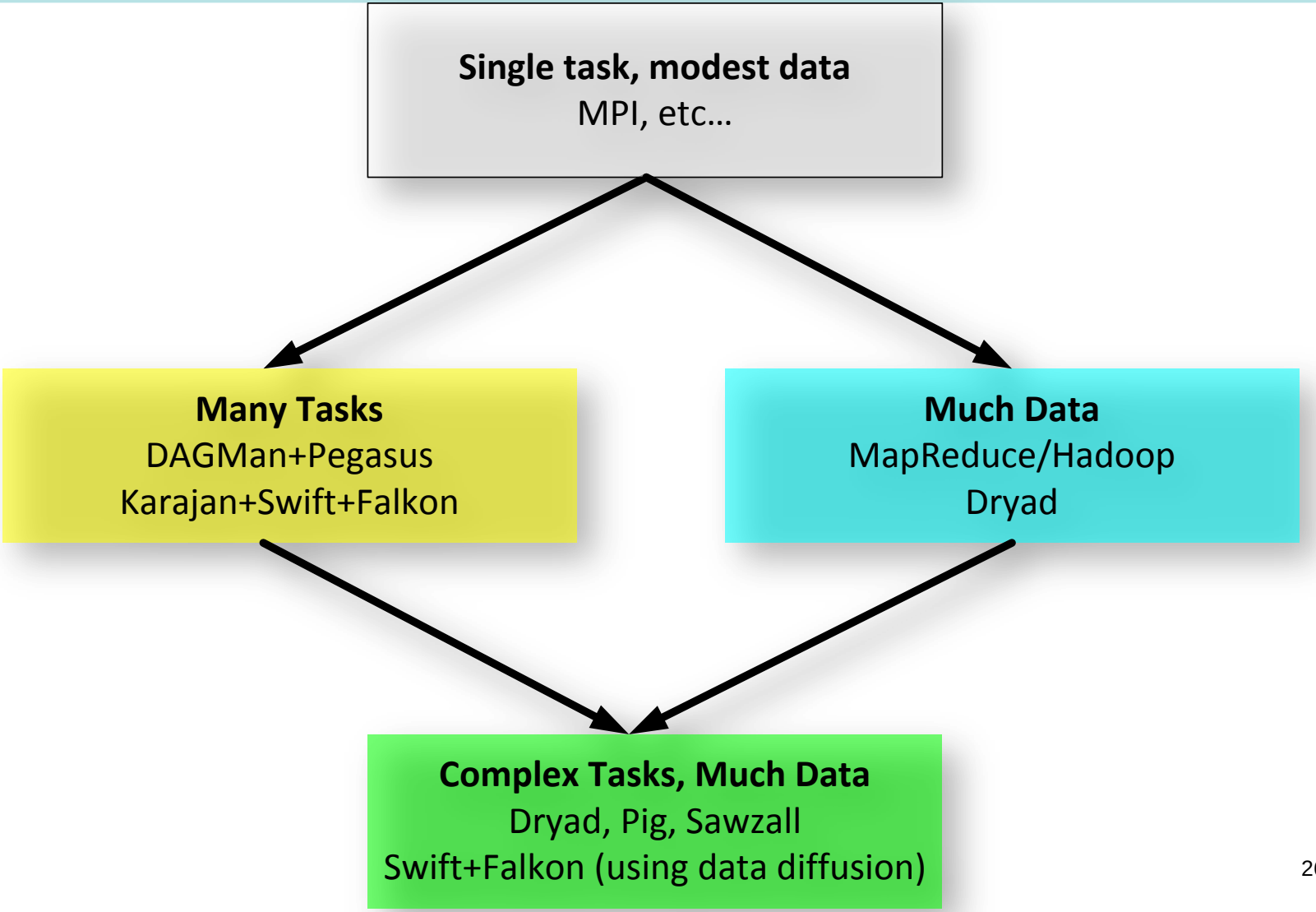
Programming and Application Model

- Grids:
  - Tightly coupled
    - High Performance Computing (MPI-based)
  - Loosely Coupled
    - High Throughput Computing
    - Workflows
  - Data Intensive
    - Map/Reduce
- Clouds:
  - Loosely Coupled, transactional oriented

Problem Types



An Incomplete and Simplistic View of Programming Models and Tools



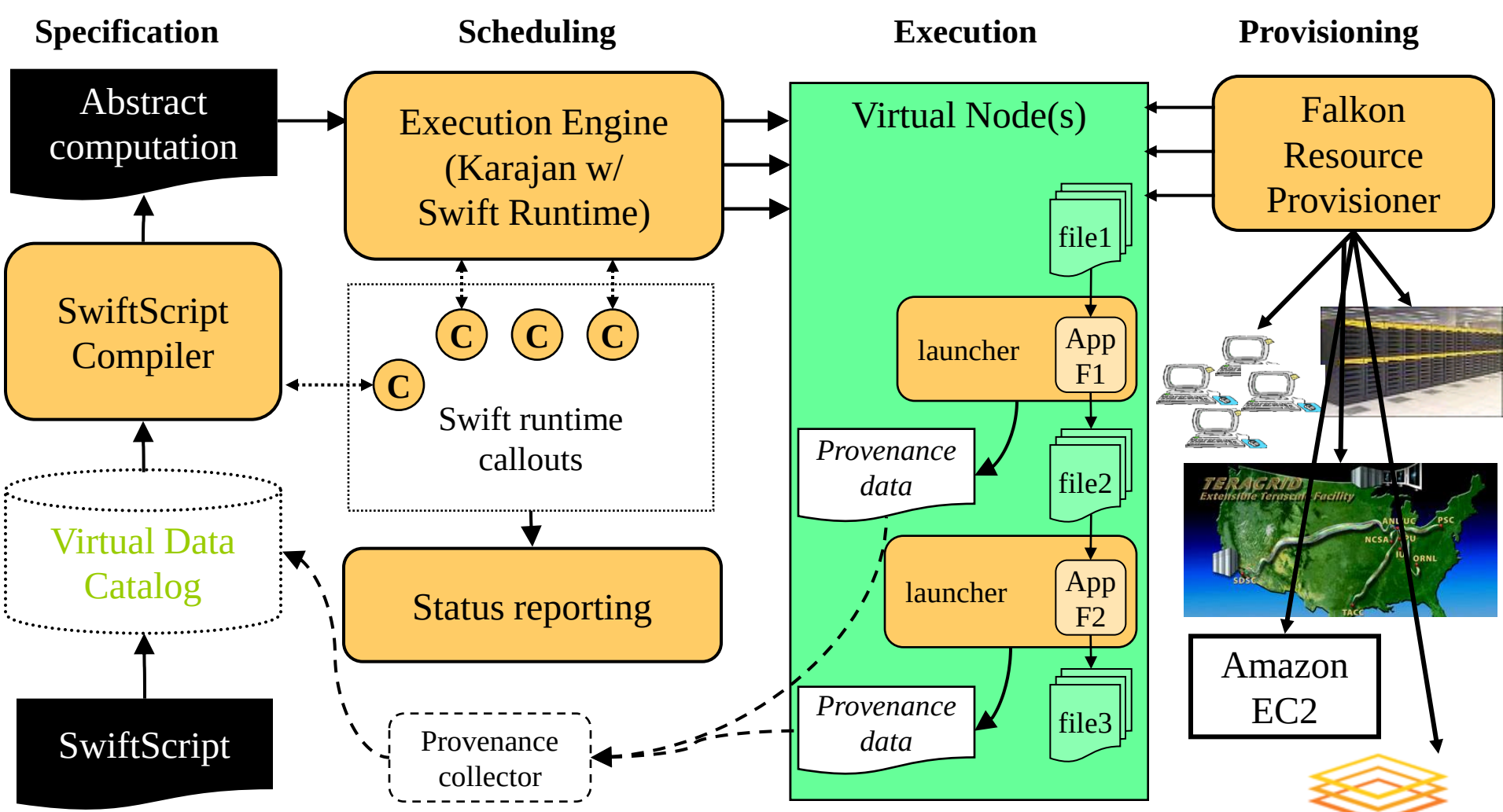
MTC: Many Task Computing

- Loosely coupled applications
  - High-performance computations comprising of multiple distinct activities, coupled via file system operations or message passing
  - Emphasis on using many resources over short time periods
  - Tasks can be:
    - small or large, independent and dependent, uniprocessor or multiprocessor, compute-intensive or data-intensive, static or dynamic, homogeneous or heterogeneous, loosely or tightly coupled, large number of tasks, large quantity of computing, and large volumes of data…

Programming Model Issues

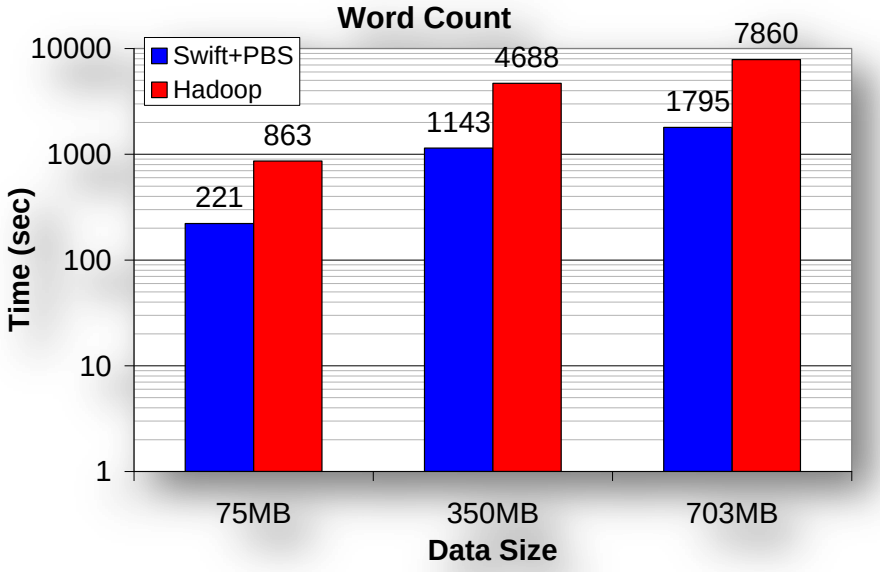
- **Multicore** processors
- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Dynamic task graphs**
- Documenting **provenance** of data products
- **Data management**: input, intermediate, output
- **Dynamic data access** involving large amounts of data

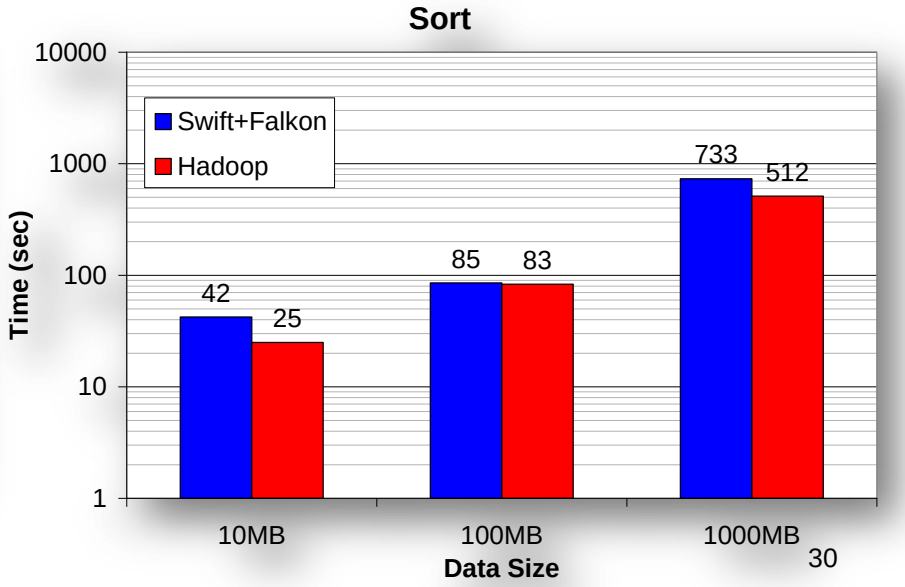
Swift Architecture



Comparing Hadoop and Swift

- Classic benchmarks for MapReduce
  - Word Count
  - Sort
- Swift performs similar or better than Hadoop (on 32 processors)





Gateways

- Aimed to simplify usage of complex resources
- Grids
  - Front-ends to many different applications
  - Emerging technologies for Grids
- Clouds
  - Standard interface to Clouds

Gateway to Grids

Gateway to Clouds

Gateway to Clouds

Gateway to Clouds

Gateway to Clouds

Gateway to Clouds

Gateway to Clouds

Security Model

- Grids
  - Grid Security Infrastructure (GSI)
  - Stronger, but steeper learning curve and wait time
    - Personal verification: phone, manager, etc
- Clouds
  - Weaker, can use credit card to gain access, can reset password over plain text email, etc

Conclusion

- Move towards a mix of micro-production and large utilities, with load being distributed among them dynamically
  - Increasing numbers of small-scale producers (local clusters and embedded processors—in shoes and walls)
  - Large-scale regional producers
- Need to define protocols
  - Allow users and service providers to discover, monitor and manage their reservations and payments
  - Interoperability
- Need to combine the centralized scale of today's Cloud utilities, and the distribution and interoperability of today's Grid facilities
- Need support for on-demand provisioning
- Need tools for managing both the underlying resources and the resulting distributed computations
- Security and trust will be a major obstacle for commercial Clouds by large companies that have in-house IT resources to host their own data centers

Questions