# Cloud Computing and Grid Computing 360-Degree Compared

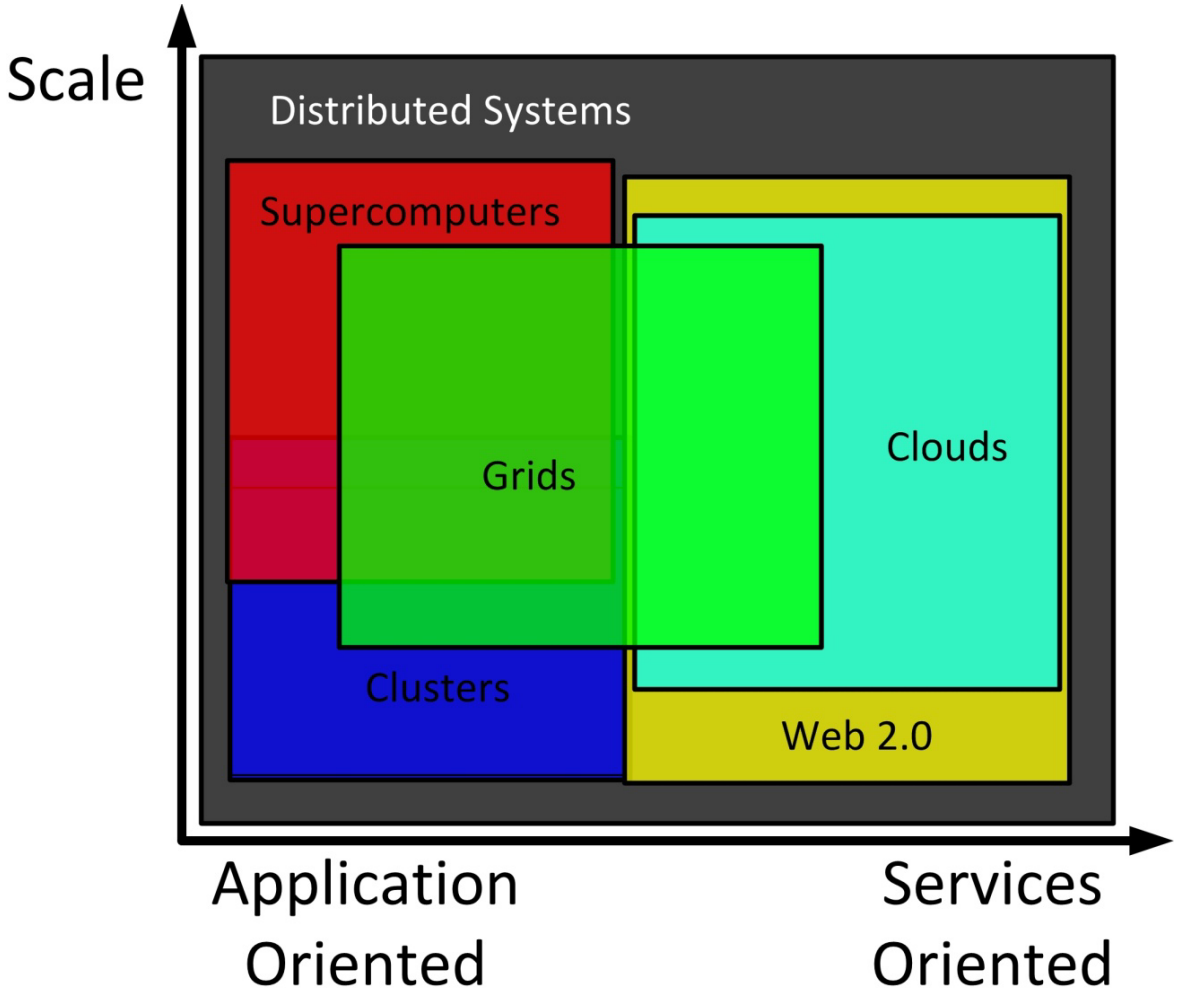

**Ioan Raicu**
Distributed Systems Laboratory
Computer Science Department
University of Chicago

**In collaboration with:**
**Ian Foster**, University of Chicago and Argonne National Laboratory
**Yong Zhao**, Microsoft Corporation
**Shiyong Lu**, Wayne State University



IEEE Grid Computing Environments (GCE) 2008
November 16th, 2008

# Clusters, Grids, Clouds, …

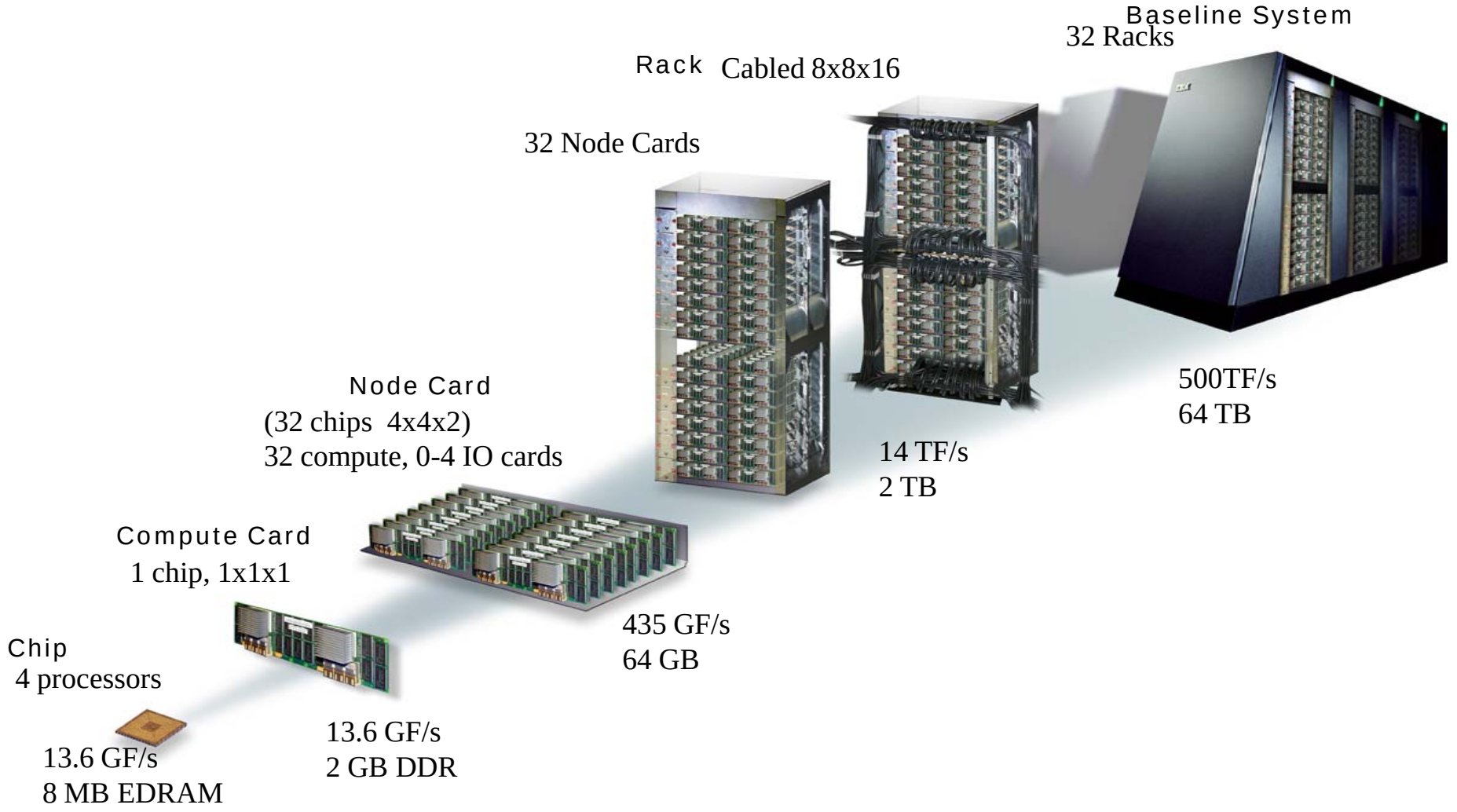


# Supercomputing

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

e.g. IBM Blue Gene/P

# IBM Blue Gene/P at ANL ALCF



# Cluster Computing

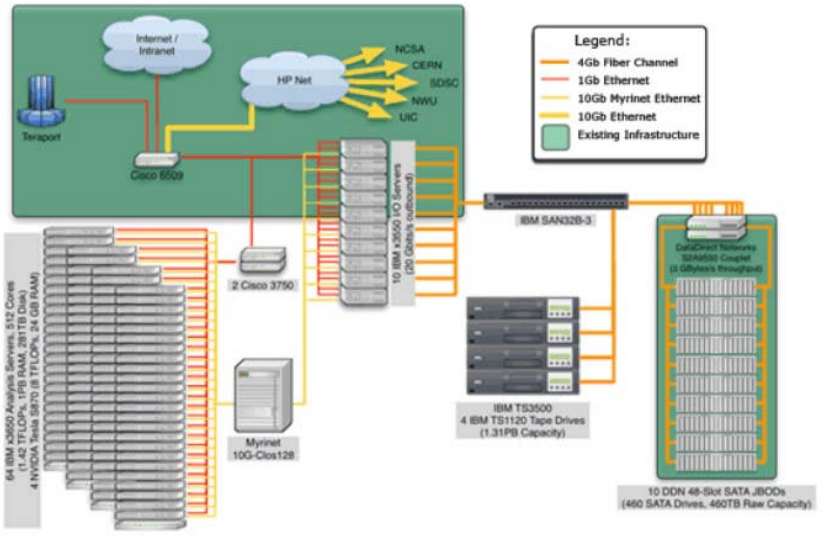
*Computer clusters using commodity processors, network interconnects, and operating system*

e.g. PADS

# Petascale Active Data Store (PADS)

PADS

home website in development




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.

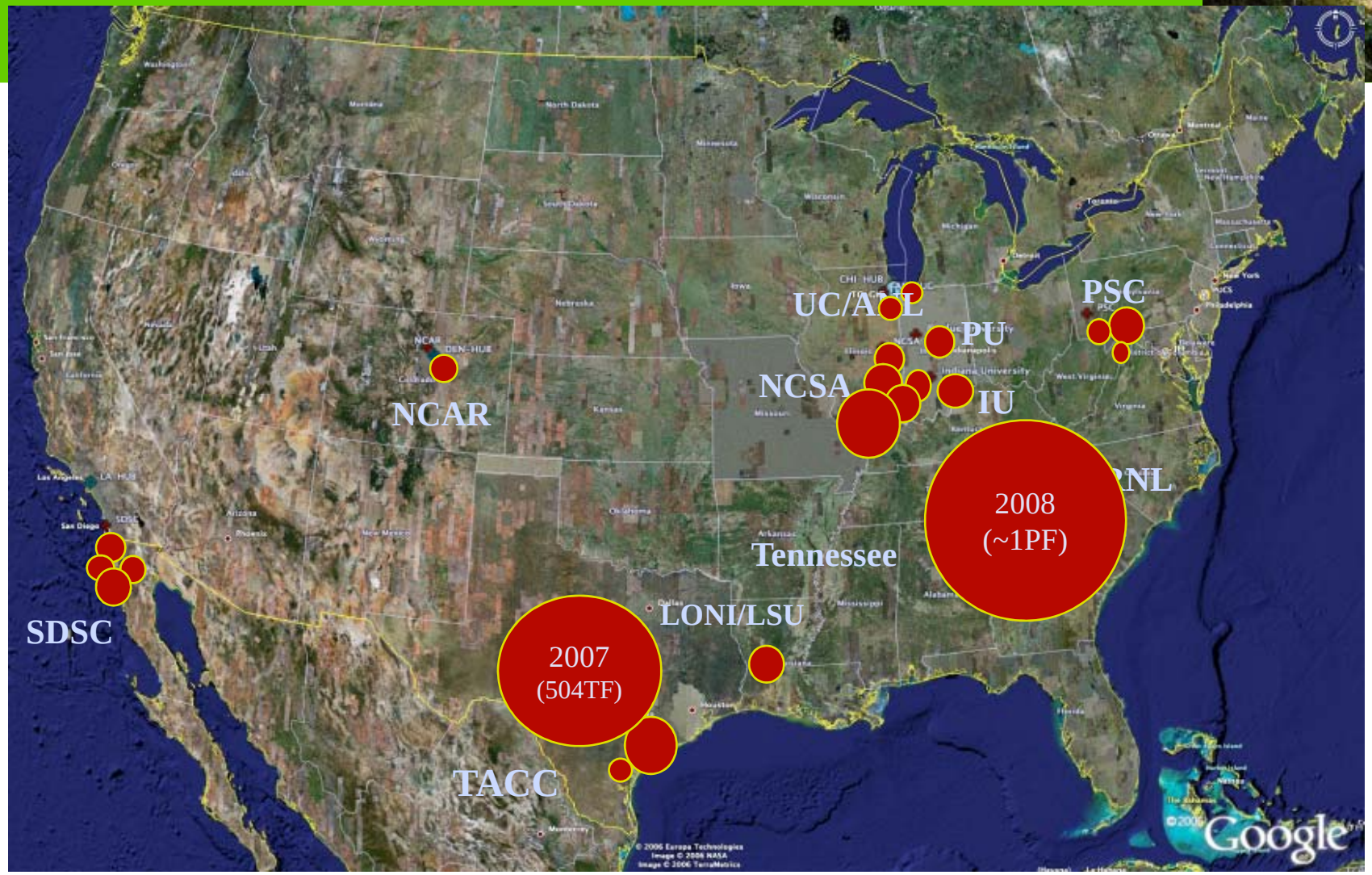
PADSstatus myPADS

# Grid Computing

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

e.g. TeraGrid

# TeraGrid High Performance Computing Systems 2007-8




Computational Resources
(size approximate - not to scale)



*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

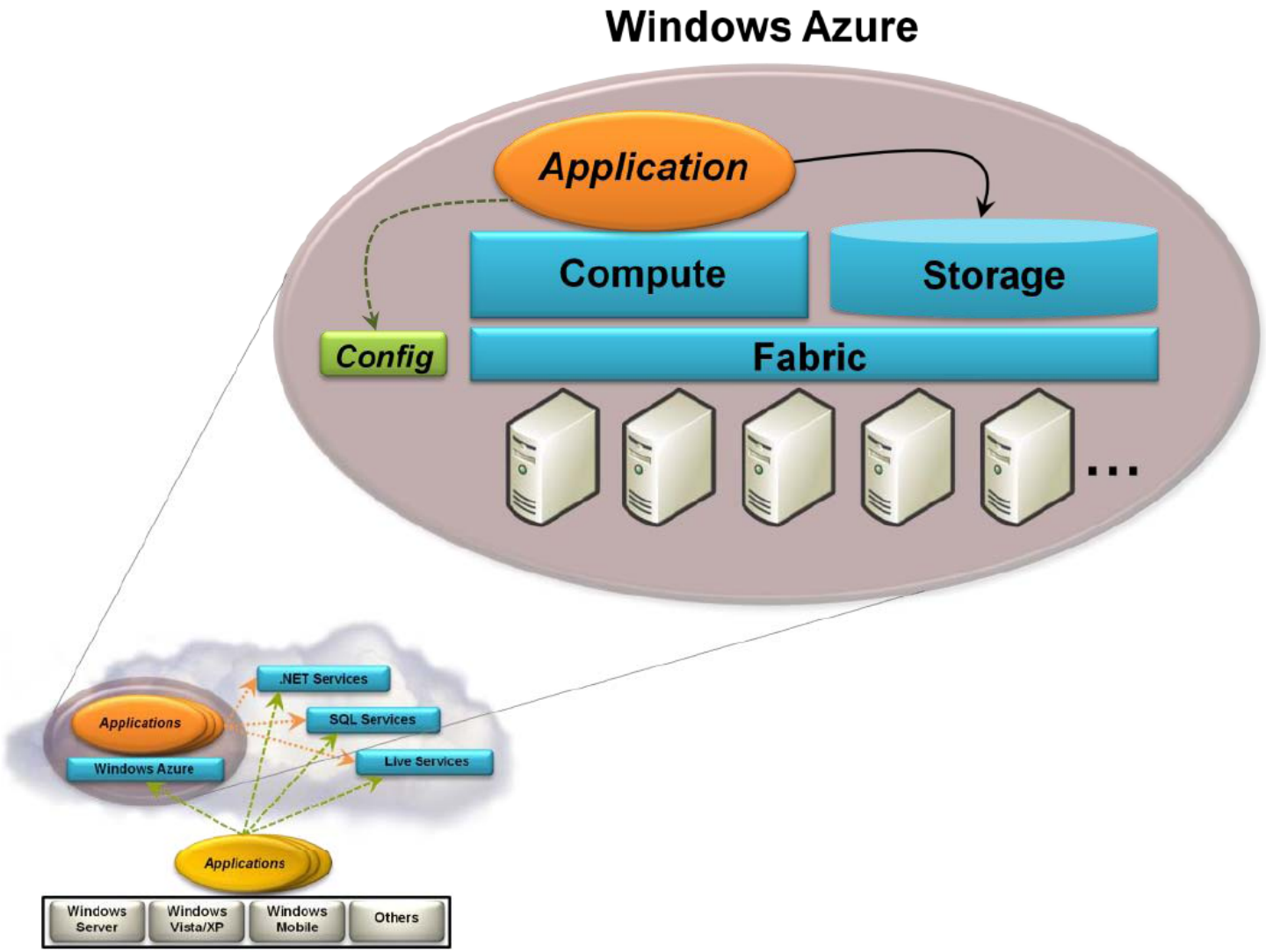
- TeraGrid (TG)
- Open Science Grid (OSG)
- Enabling Grids for E-sciencE (EGEE)
- LHC Computing Grid from CERN
- Grid Middleware
  - Globus Toolkit
  - Unicore

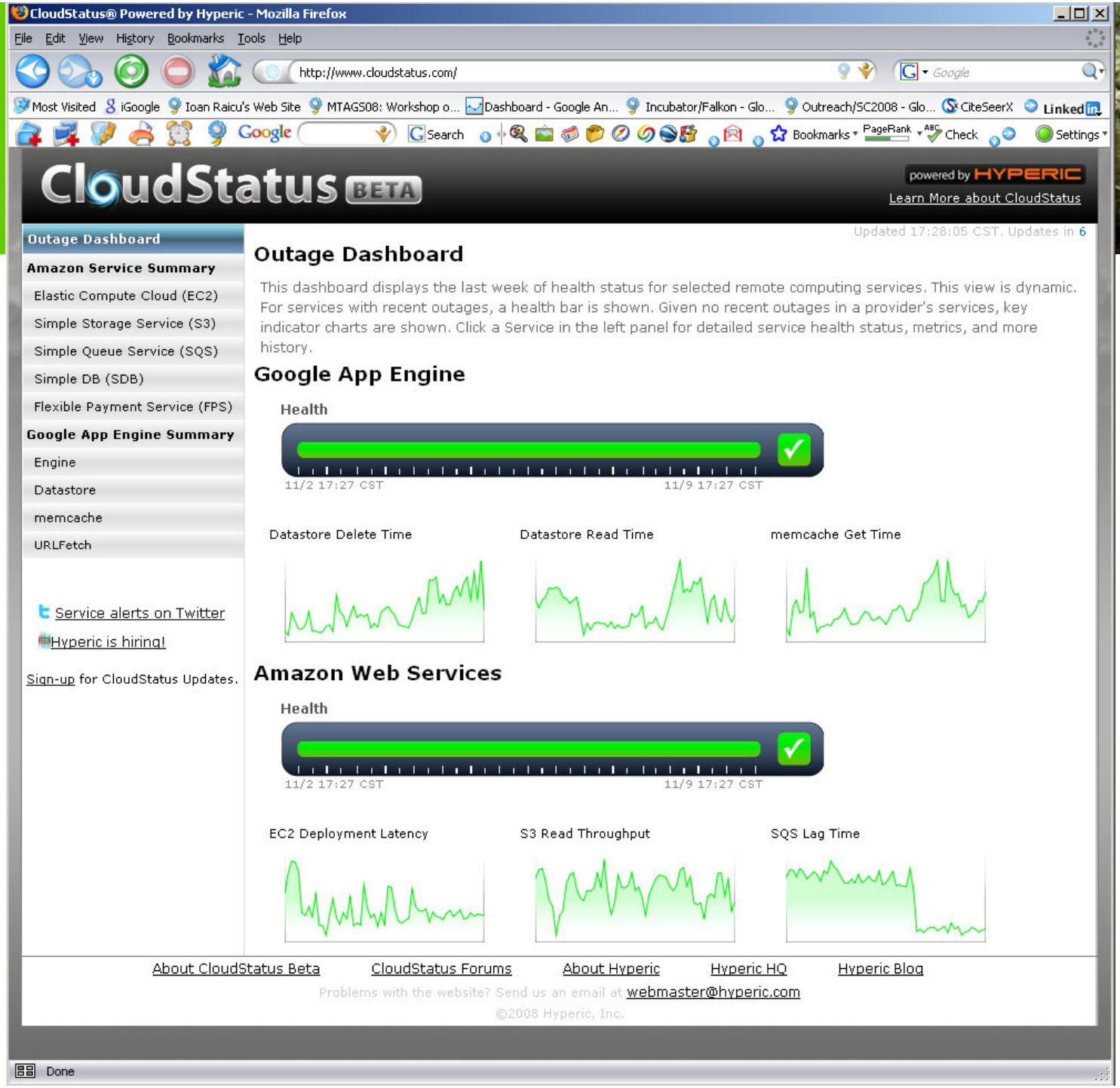
# Cloud Computing

*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

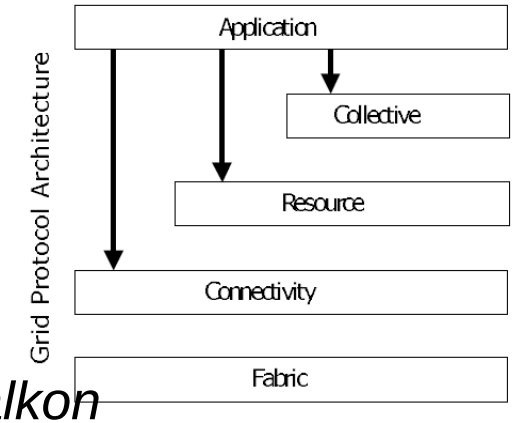
- Business model
- Architecture
- Resource management
- Programming model
- Application model
- Security model

# Business Model

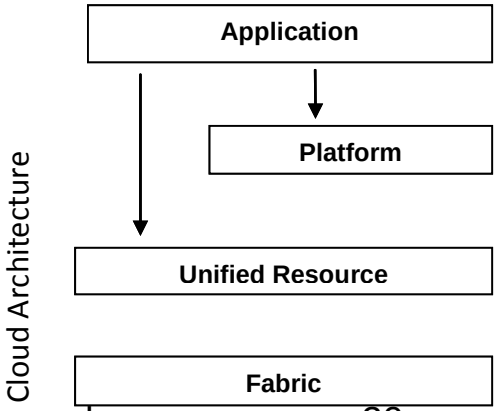
- Grids:
  - Largest Grids funded by government
  - Largest user-base in academia and government labs to drive scientific computing
  - Project-oriented: service units
- Clouds:
  - Industry (i.e. Amazon) funded the initial Clouds
  - Large user base in common people, small businesses, large businesses, 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*



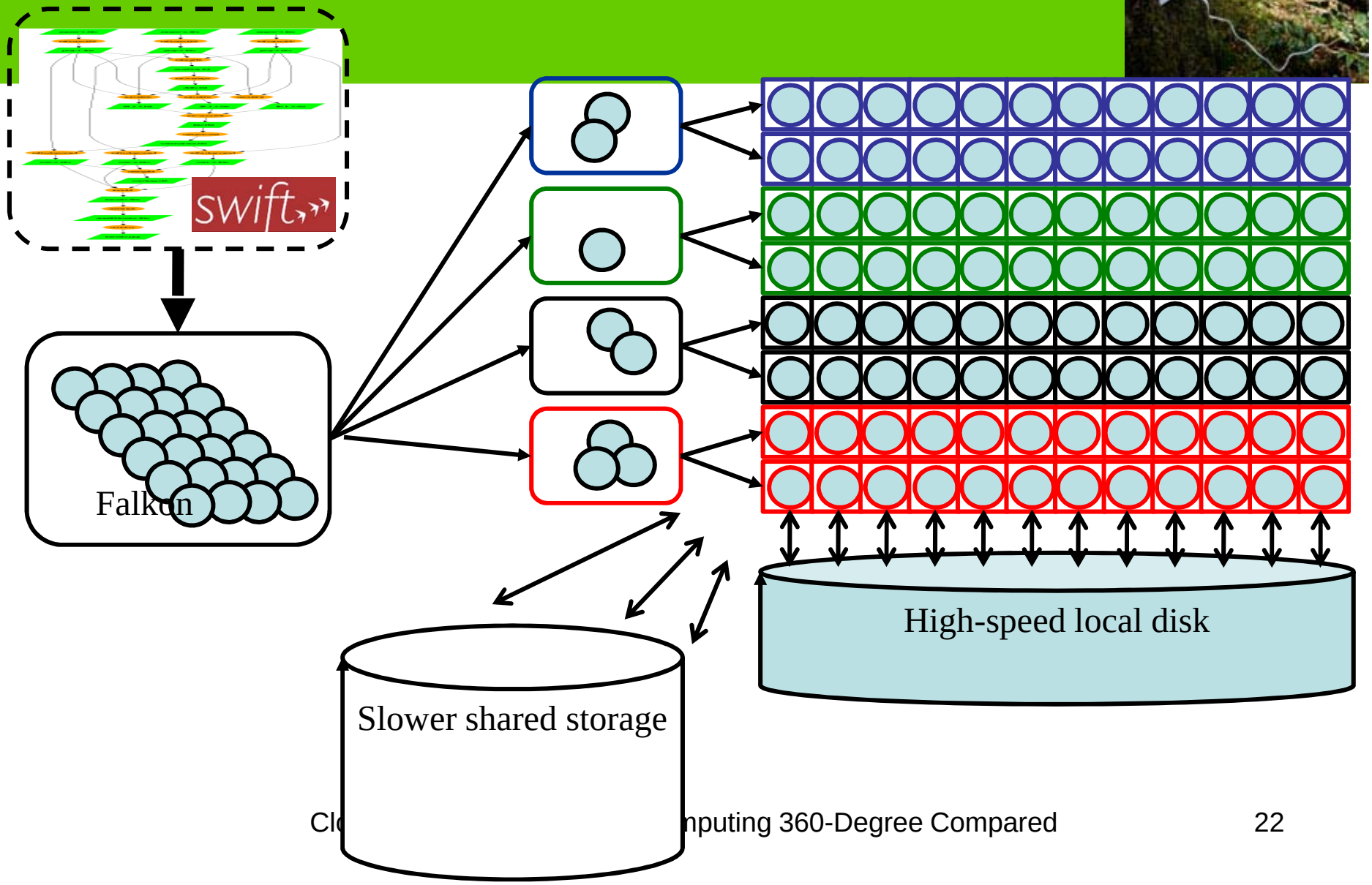

- Clouds:
  - Application Layer: *Software as a Service (SaaS)*
  - Platform Layer: *Platform as a Service (PaaS)*
  - Unified Resource: *Infrastructure as a Service (IaaS)*
  - Fabric: *IaaS*



# 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

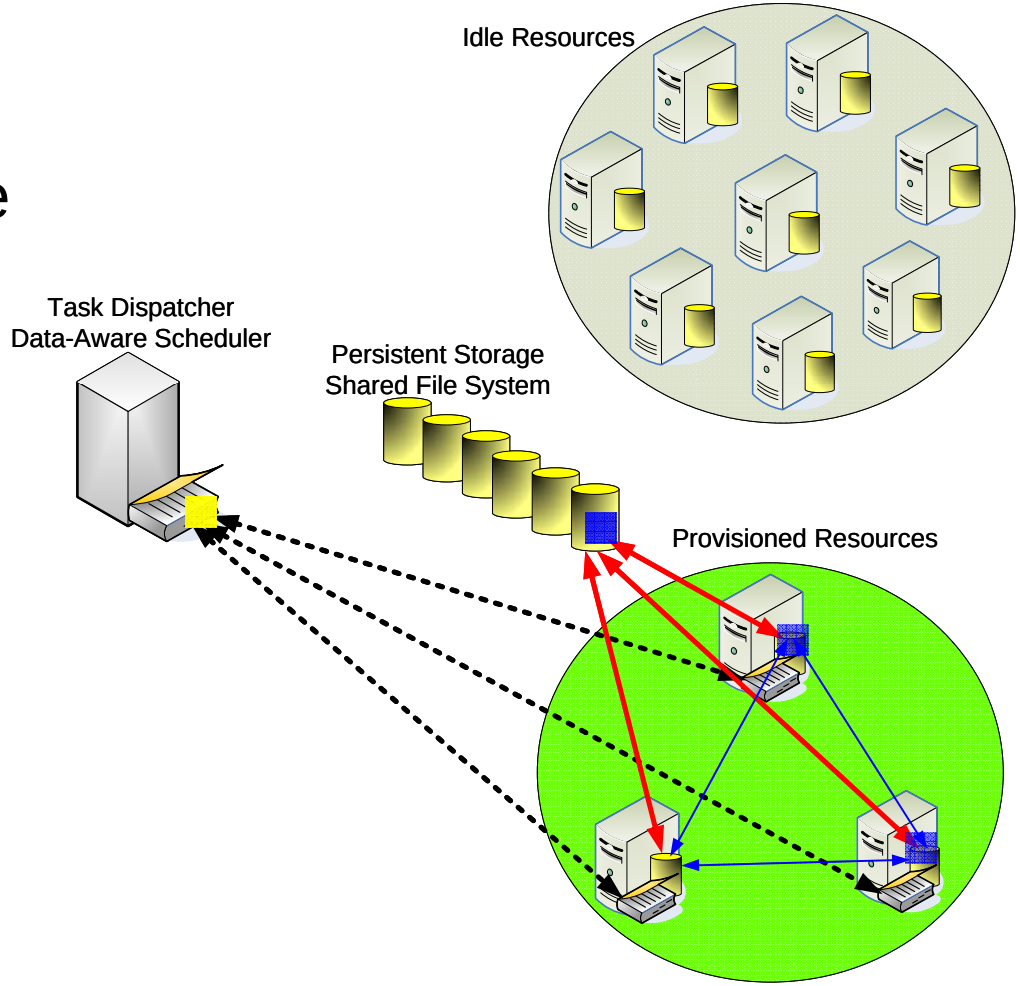


# Resource Management

- 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



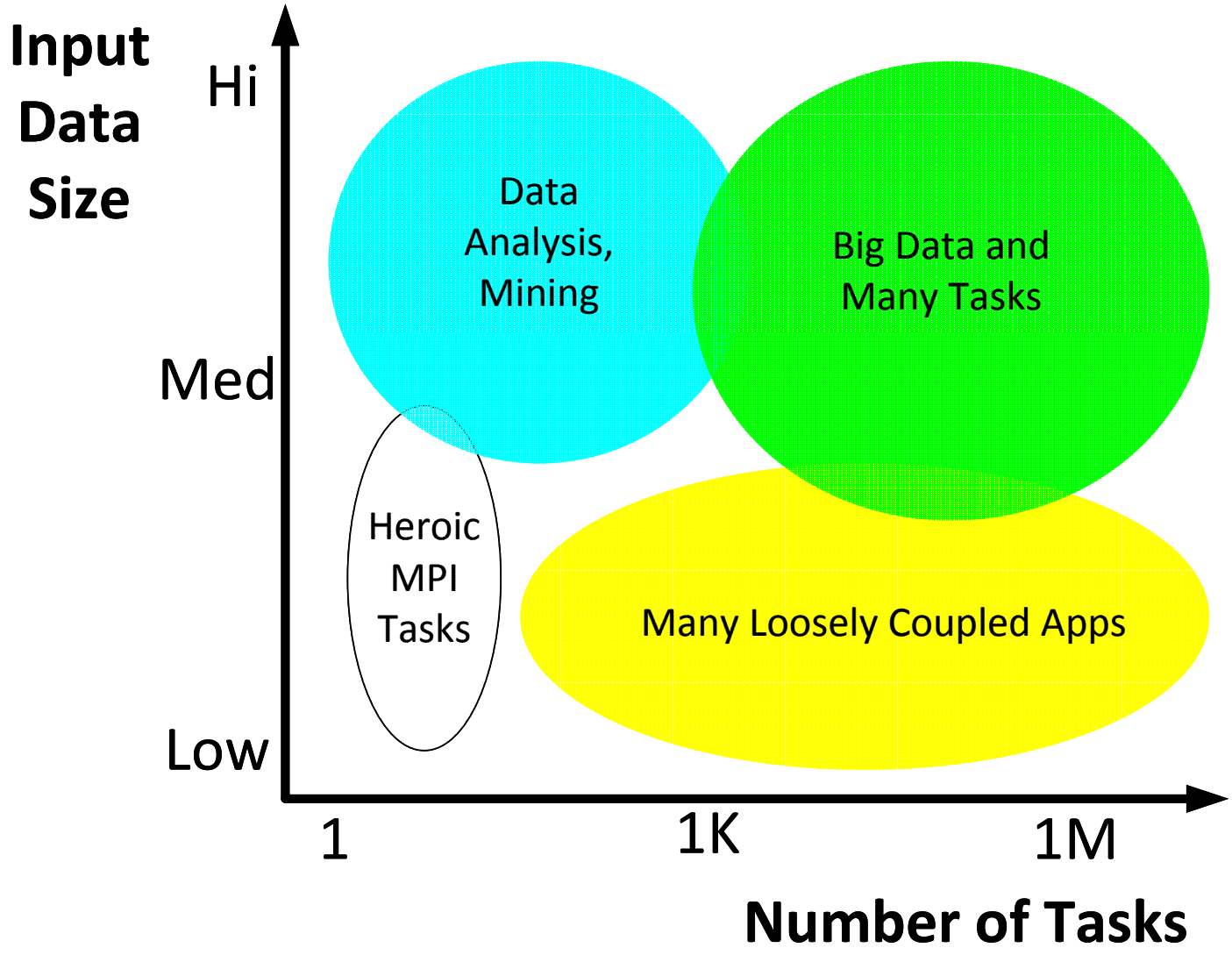
# Resource Management

- 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

**Single task, modest data**
MPI, etc…

**Many Tasks**
DAGMan+Pegasus
Karajan+Swift+Falkon

**Much Data**
MapReduce/Hadoop
Dryad

**Complex Tasks, Much Data**
Dryad, Pig, Sawzall
Swift+Falkon (using data diffusion)

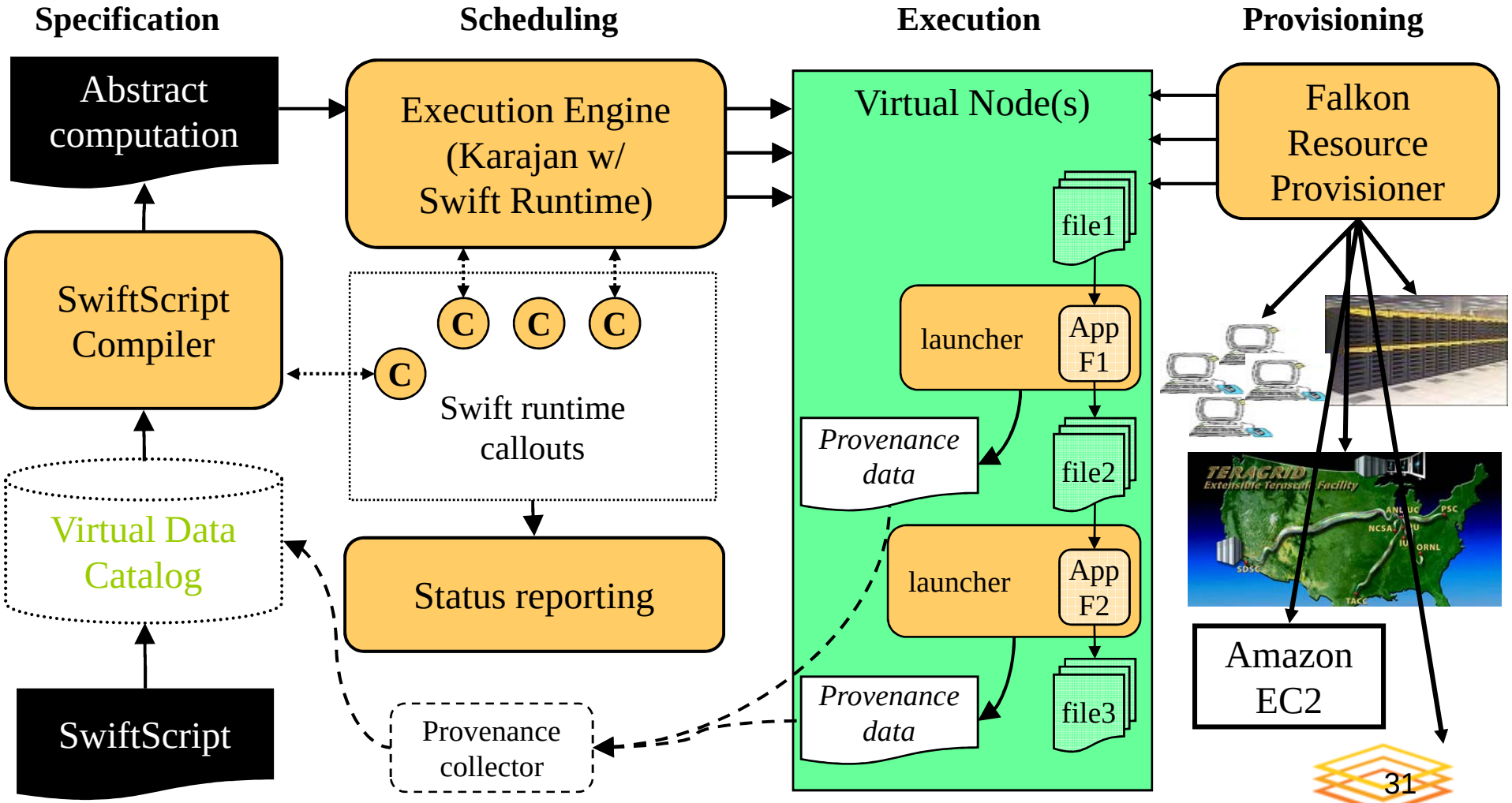
# MTC: Many Task Computing

- 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

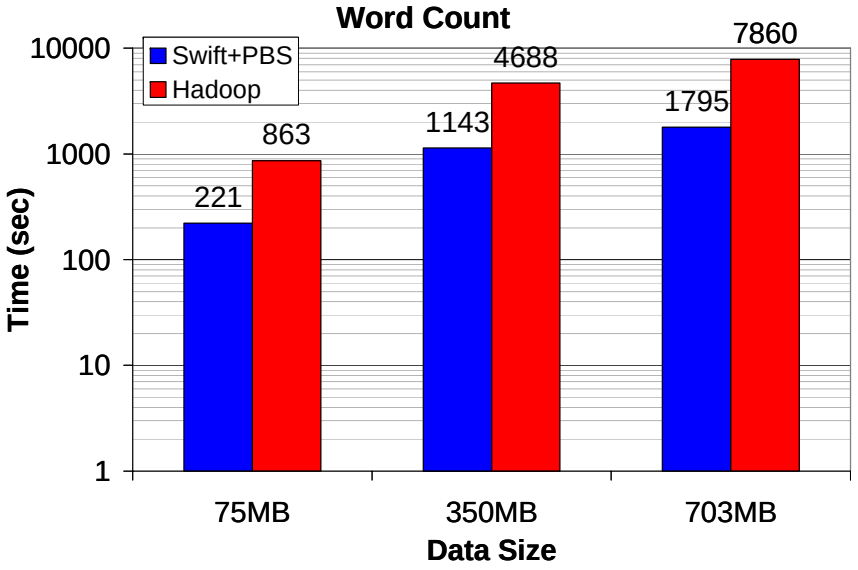
- **Multicore** processors
- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Dynamic task graphs**
- Documenting **provenance** of data products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

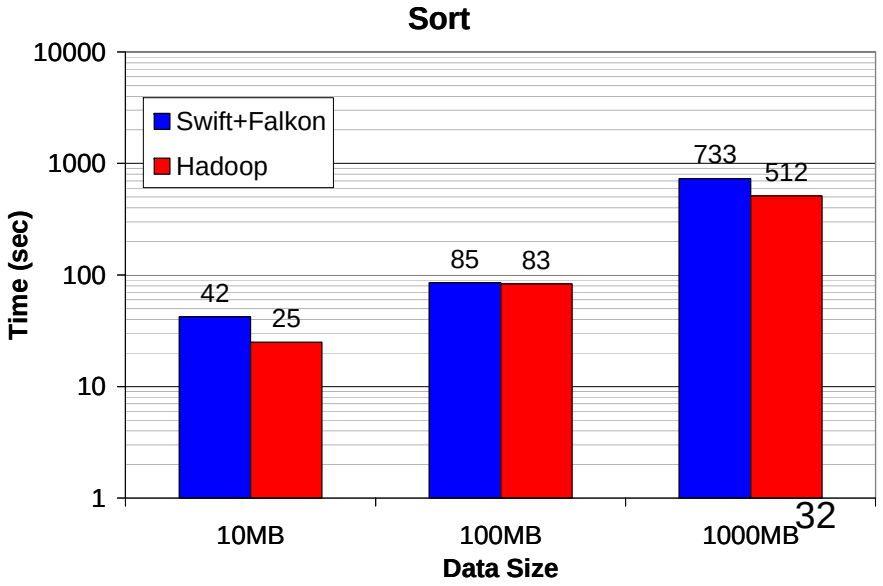
# Swift Architecture



# 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

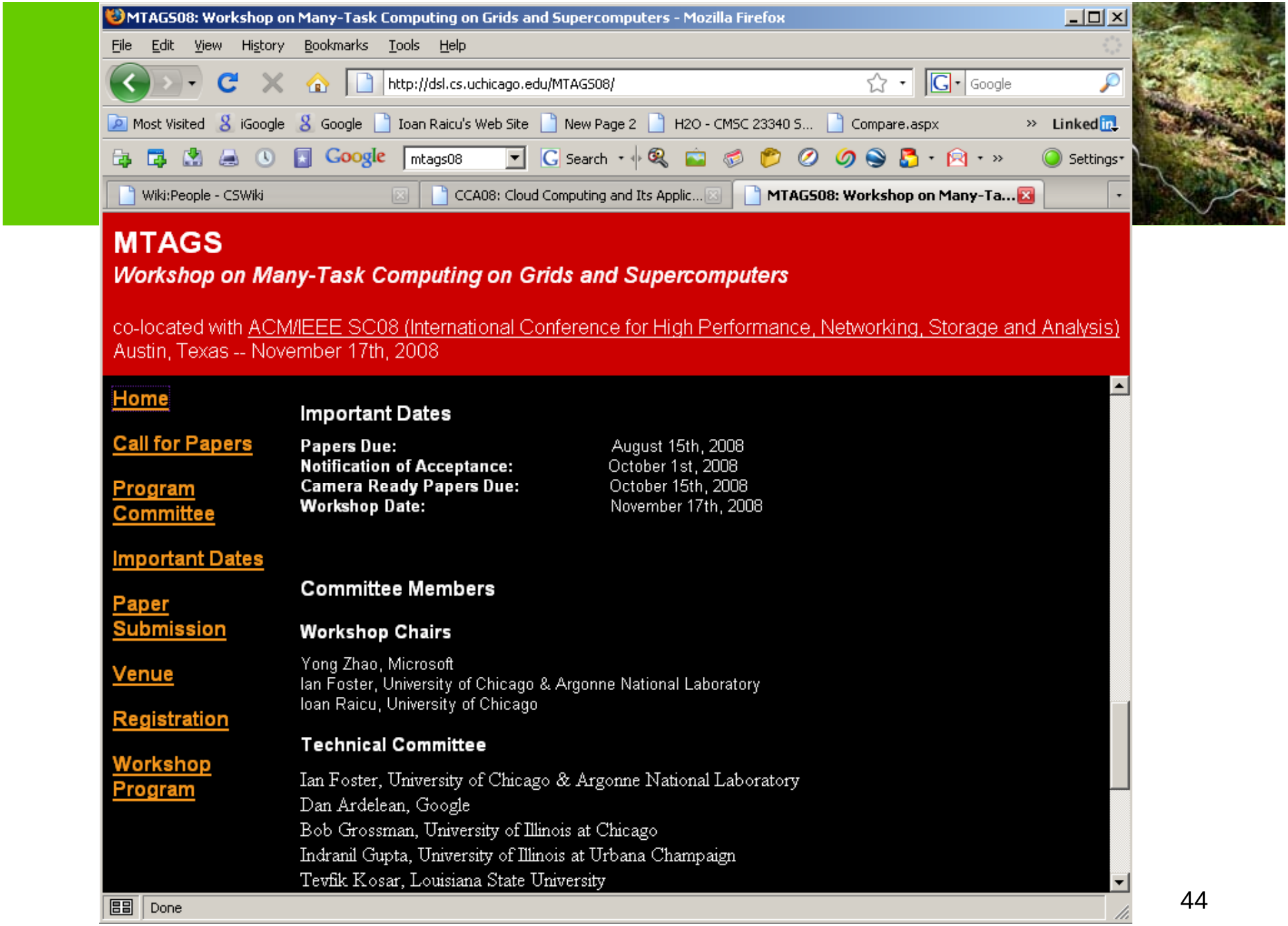
- Grids
  - Grid Security Infrastructure (GSI)
  - Stronger, but steeper learning curve and wait time
    - Personal verification: phone, manager, etc
- Clouds
  - Weaker, can use credit card to gain access, can reset password over plain text email, etc

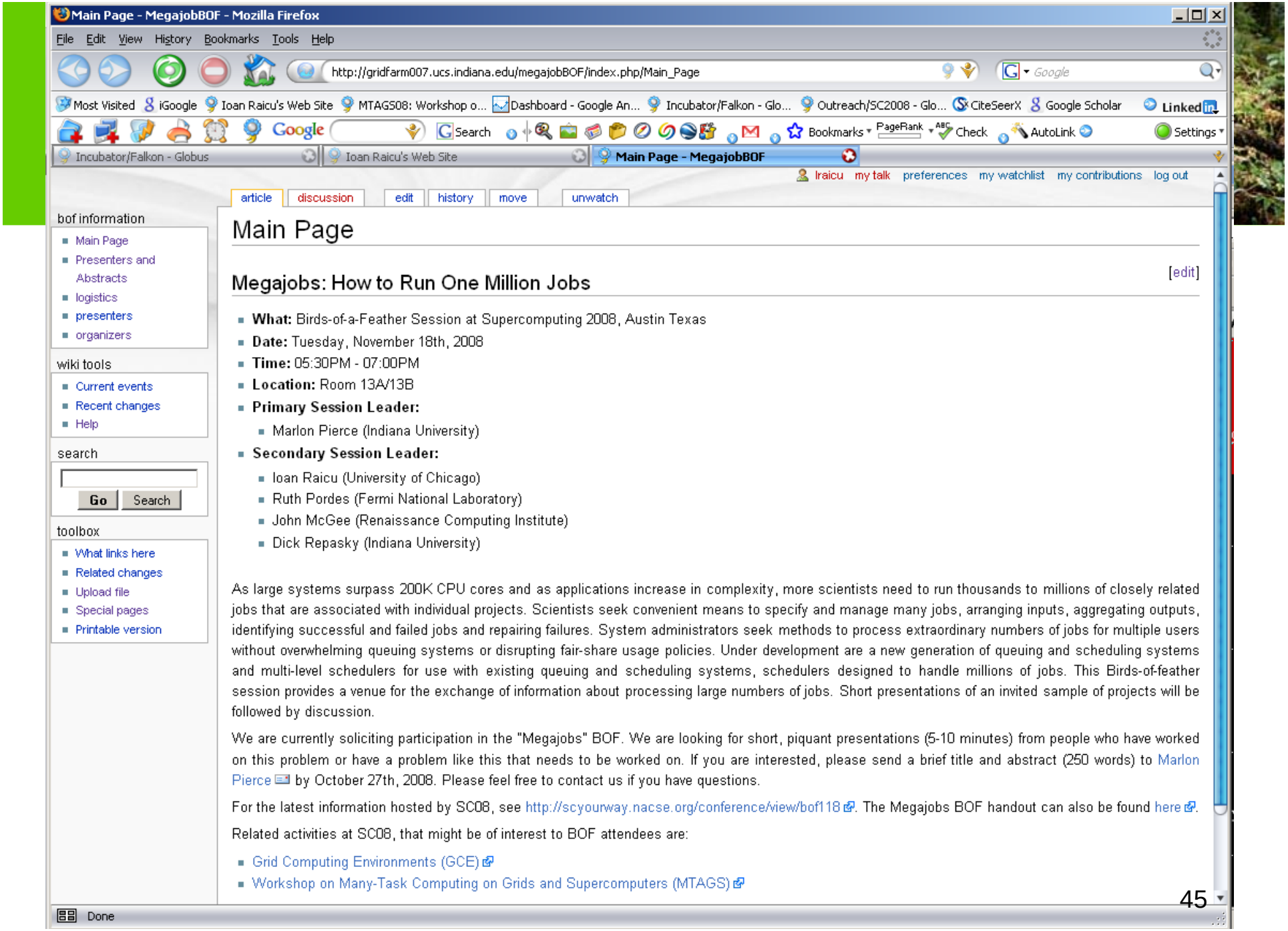
# Conclusion

- 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

# More Information

- Ioan Raicu
  - http://people.cs.uchicago.edu/~iraicu/
  - iraicu@cs.uchicago.edu
- Original Blog from Ian Foster
  - There's Grid in them thar Clouds*
  - http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html
- Paper published in IEEE Grid Computing Environments 2008
  - http://people.cs.uchicago.edu/~iraicu/publications/2008_GCE08_Clouds_Grids.pdf
- Cloud Computing and Its Applications Workshop
  - http://www.cca08.org/


MTAGS08: Workshop on Many-Task Computing on Grids and Supercomputers - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://dsl.cs.uchicago.edu/MTAGS08/
Google
Most Visited iGoogle Google Ioan Raicu's Web Site New Page 2 H2O - CMSC 23340 S... Compare.aspx
Linked
mtags08
Search
Settings
Wiki:People - CSWiki
CCA08: Cloud Computing and Its Applic...
MTAGS08: Workshop on Many-Ta...
MTAGS
Workshop on Many-Task Computing on Grids and Supercomputers
co-located with ACM/IEEE SC08 (International Conference for High Performance, Networking, Storage and Analysis)
Austin, Texas -- November 17th, 2008
Home
Call for Papers
Program Committee
Important Dates
Paper Submission
Venue
Registration
Workshop Program
Important Dates
Papers Due: August 15th, 2008
Notification of Acceptance: October 1st, 2008
Camera Ready Papers Due: October 15th, 2008
Workshop Date: November 17th, 2008
Committee Members
Workshop Chairs
Yong Zhao, Microsoft
Ian Foster, University of Chicago & Argonne National Laboratory
Ioan Raicu, University of Chicago
Technical Committee
Ian Foster, University of Chicago & Argonne National Laboratory
Dan Ardelean, Google
Bob Grossman, University of Illinois at Chicago
Indranil Gupta, University of Illinois at Urbana Champaign
Tevfik Kosar, Louisiana State University
Done


Main Page - MegajobBOF - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://gridfarm007.ucs.indiana.edu/megajobBOF/index.php/Main_Page
Most Visited iGoogle Ioan Raicu's Web Site MTAGS08: Workshop o... Dashboard - Google An... Incubator/Falkon - Glo... Outreach/SC2008 - Glo... CiteSeerX Google Scholar LinkedIn
Incubator/Falkon - Globus
Ioan Raicu's Web Site
Main Page - MegajobBOF
Iraicu my talk preferences my watchlist my contributions log out
article discussion edit history move unwatch
bof information
Main Page
Presenters and Abstracts
logistics
presenters
organizers
wiki tools
Current events
Recent changes
Help
search
Go Search
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Main Page
Megajobs: How to Run One Million Jobs
[edit]
What: Birds-of-a-Feather Session at Supercomputing 2008, Austin Texas
Date: Tuesday, November 18th, 2008
Time: 05:30PM - 07:00PM
Location: Room 13A/13B
Primary Session Leader:
Marlon Pierce (Indiana University)
Secondary Session Leader:
Ioan Raicu (University of Chicago)
Ruth Pordes (Fermi National Laboratory)
John McGee (Renaissance Computing Institute)
Dick Repasky (Indiana University)
As large systems surpass 200K CPU cores and as applications increase in complexity, more scientists need to run thousands to millions of closely related jobs that are associated with individual projects. Scientists seek convenient means to specify and manage many jobs, arranging inputs, aggregating outputs, identifying successful and failed jobs and repairing failures. System administrators seek methods to process extraordinary numbers of jobs for multiple users without overwhelming queuing systems or disrupting fair-share usage policies. Under development are a new generation of queuing and scheduling systems and multi-level schedulers for use with existing queuing and scheduling systems, schedulers designed to handle millions of jobs. This Birds-of-feather session provides a venue for the exchange of information about processing large numbers of jobs. Short presentations of an invited sample of projects will be followed by discussion.
We are currently soliciting participation in the "Megajobs" BOF. We are looking for short, piquant presentations (5-10 minutes) from people who have worked on this problem or have a problem like this that needs to be worked on. If you are interested, please send a brief title and abstract (250 words) to Marlon Pierce by October 27th, 2008. Please feel free to contact us if you have questions.
For the latest information hosted by SC08, see http://scyourway.nacse.org/conference/view/bof118. The Megajobs BOF handout can also be found here.
Related activities at SC08, that might be of interest to BOF attendees are:
Grid Computing Environments (GCE)
Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)
Done