Microsoft Research

Building Reliable Science Gateways: When can a Grid be a ~~Cloud~~ Data Center?

Dennis Gannon

Data Center Futures

Microsoft Research

The TeraGrid

- The US National Supercomputer Grid
  - CyberInfrastructure composed of a set of resources (compute and data) that provide common services for
    - Wide area data management (gridftp, gpfs, staged disk to tape.)
    - Single sign-on user authentication (globus toolkit)
    - Distributed Job scheduling and management. (in the works.)

Collectively
- 1Petaflop
- 20 Petabytes

Soon to triple.
Will add a new petaflop machine each year.

The TeraGrid Vision

- A large-scale operational cyberinfrastructure that
  - provides very high-end computational capabilities for leading-edge research in the national open science community (TeraGrid Deep)
  - while also reaching out to new users and communities to broaden the impact of cyberinfrastructure in research, education, and society (TeraGrid Wide)

TeraGrid becomes XD (eXtreme Digital)

- Telescopes of all types both terrestrial and space-based.
- A growing network of geo-sensors including
  - GPS equipped wireless-connected earthquake monitors,
  - fixed and autonomously roving undersea instruments,
  - atmospheric monitors including network of radars soon to be mounted on every cell tower,
  - urban instrumentation including cameras and traffic sensors.
- Medical instrumentation that will soon enable remote systems to monitor the health and well being of the entire population.

Science Examples: Data Access and Analysis

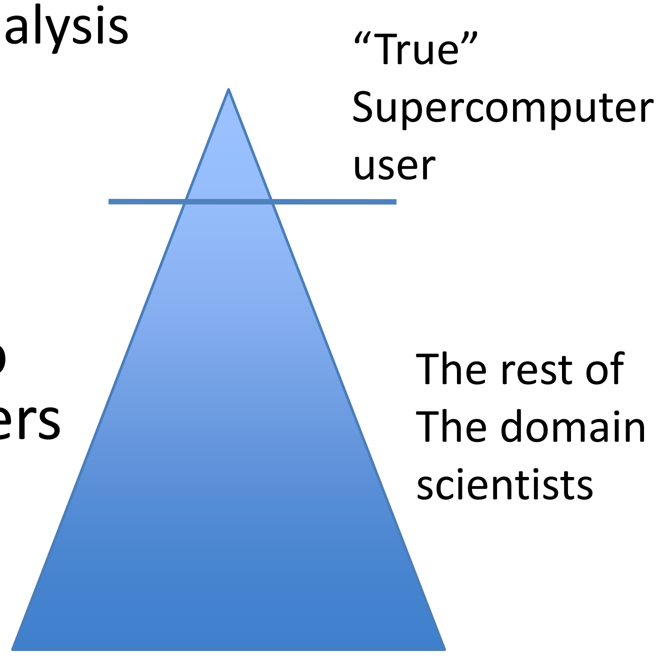
- SkyServer
  - Excellent Astronomy data access example.
    - Limited data analysis capability in current deployment.
- EGEE/OSG and the Large Hadron Collider
  - Massive data analysis requirements
  - Recent studies suggest data analysis may be better done in a large data center than in a distributed Grid.
- Earth Science
  - Polar Grid, LEAD (weather), ESG (climate) are all about providing service based access to data and analysis.
- Social Science/Medical Science/Biology
  - SidGrid, ICPSR (social science data archive), BioGrid, CaGrid are all data access and analysis systems that are well suited to cloud-like system design.

Real-Time, Data-Driven Applications

- Scientific advances are increasingly made by harvesting knowledge from streams of data.
  - Sensor networks are critical to geoscience, physics, engineering, economics …
- Given access to the right data streams and on-demand access to computation you can
  - Mange the energy consumption of a large city.
  - Monitor an active earthquake zone and provide warnings that can save lives
  - Predict tornados
  - Do the motion planning for swarms of remote robots exploring the ocean floor
  - Monitor the heath of the planet's food supply.
  - Find the Higgs boson

The Science Gateways

- How do we provide "scientific services" to communities (VOs) that have a need of high-end computation and data analysis, but no desire to use supercomputers.
- They want
  - Access to advanced domain specific analysis and simulation tools
  - Wide area data storage (indexing and curation)
  - Collaboration tools
- Start with a supercomputer user who wants to provide app services to others through a Science portals.
- Examples ....



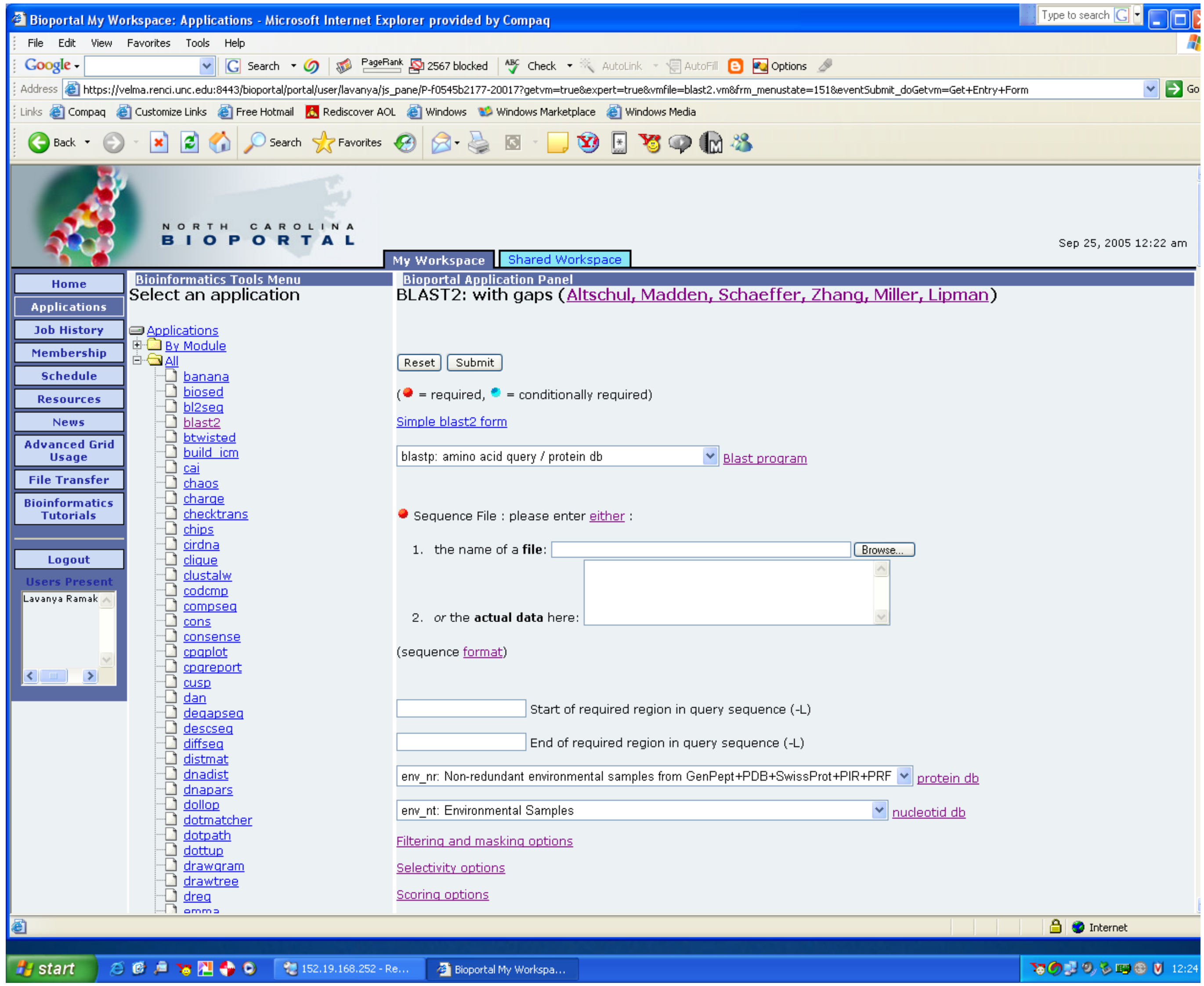
BIRN – Biomedical Information

Geological Information Grid Portal

Renci Bio Portal

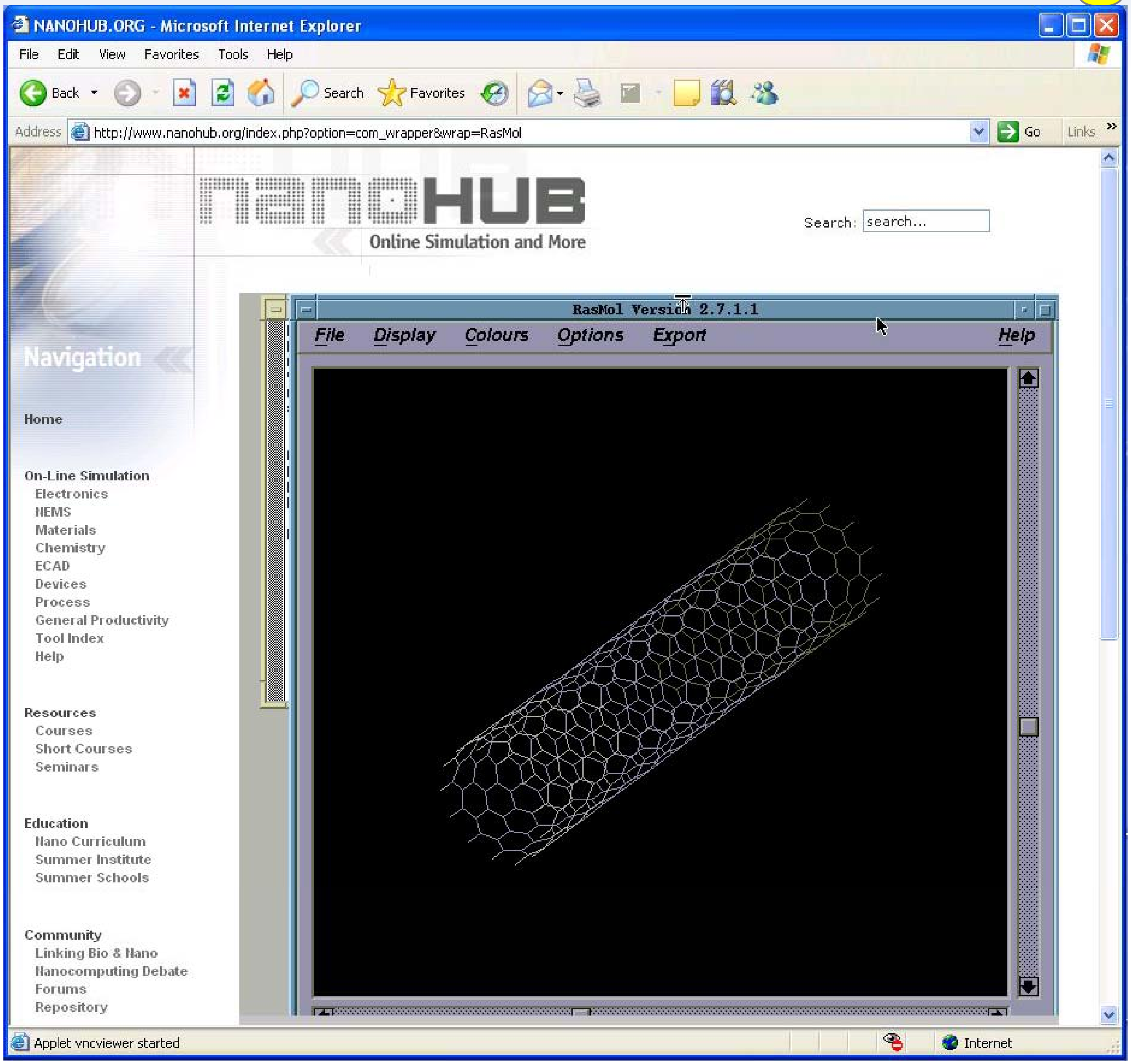
Providing access to biotechnology tools running on a back-end Grid.

- leverage state-wide investment in bioinformatics
- undergraduate & graduate education, faculty research
- another portal soon: national evolutionary synthesis center



NESCent

Nanohub - nanotechnology



X-Ray Crystallography

C.I.M.A.
Common Instrument Middleware Architecture

Welcome to the Crystallography Portal

Username:
Password:
Login
Remember me on this computer
Login Help

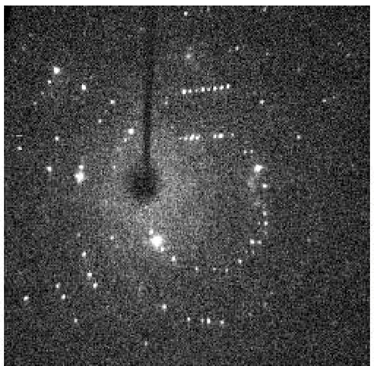
Home | **Current Status** | **Data Repository** | **About**

- **IUB IUMSC**
- **IUB Myers Hall**
- **Purdue Crystallography Center**
- **CSAF Sydney, Australia**
- **Minnesota X-ray Lab**
- **ChemMatCARS - Univ. of Chicago at APS**
- **Other collaborators**
- **NCS Southampton, UK**

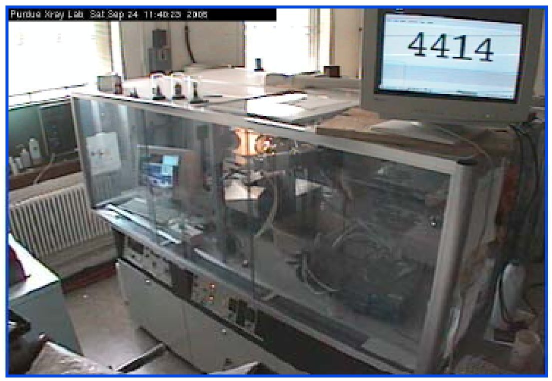
The Purdue Chemistry Crystallography Center

The Purdue Chemistry Crystallography Center

Disable your browser's cache to get the live stream!



Data from Nonius Kappa CCD detector
(Under development!)
Total Number of jpg: 10
Frame: s01f0010.jpg
All available jpg images
Browse the 20 latest jpg images



<□>

Streaming video from the lab showing the Nonius instrument

<□>

Streaming video from the crystal microscope on the Noniu diffractometer

Local date/time: 2005-09-24 11:36:54

These values are updated approx. every 60 sec. Times in UT(

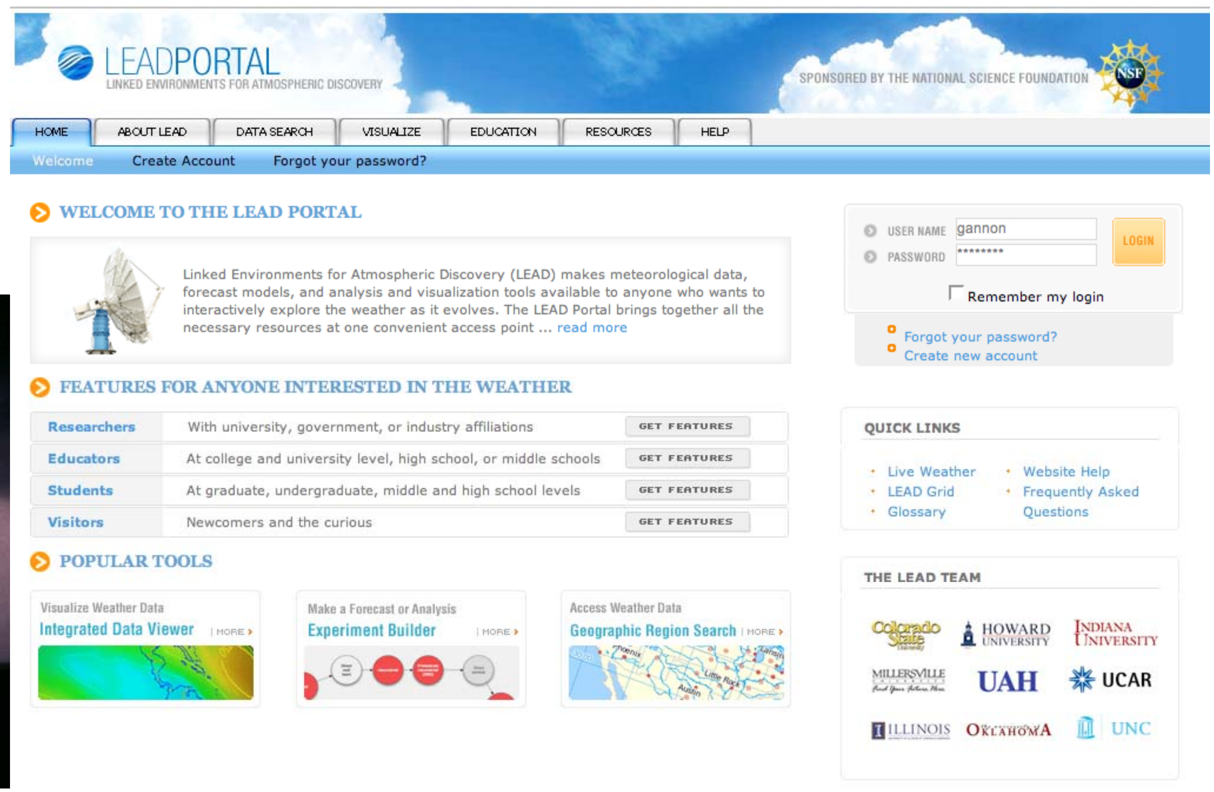
LabJack U12

Instrument Enclosure Temp. & Humidity:	23.4 C Rel. Humid. 43.1 %	2005-09-24 16:35:5(
Chill Water In:	16.4 C	2005-09-24 16:36:2(
Chill Water Out:	19.3 C	2005-09-24 16:36:2(
Generator Relay Voltage: All previous voltages	3.42 X-ray Generator is: **OFF**	2005-09-24 16:36:4(

The LEAD Gateway Portal

- On demand prediction of tornados and hurricanes
- To support three classes of users
  - Meteorology research scientists & grad students.
  - Undergrads in meteorology classes
  - People who want easy access to weather data.

Go to:
http://www.leadproject.org



Gateway Components

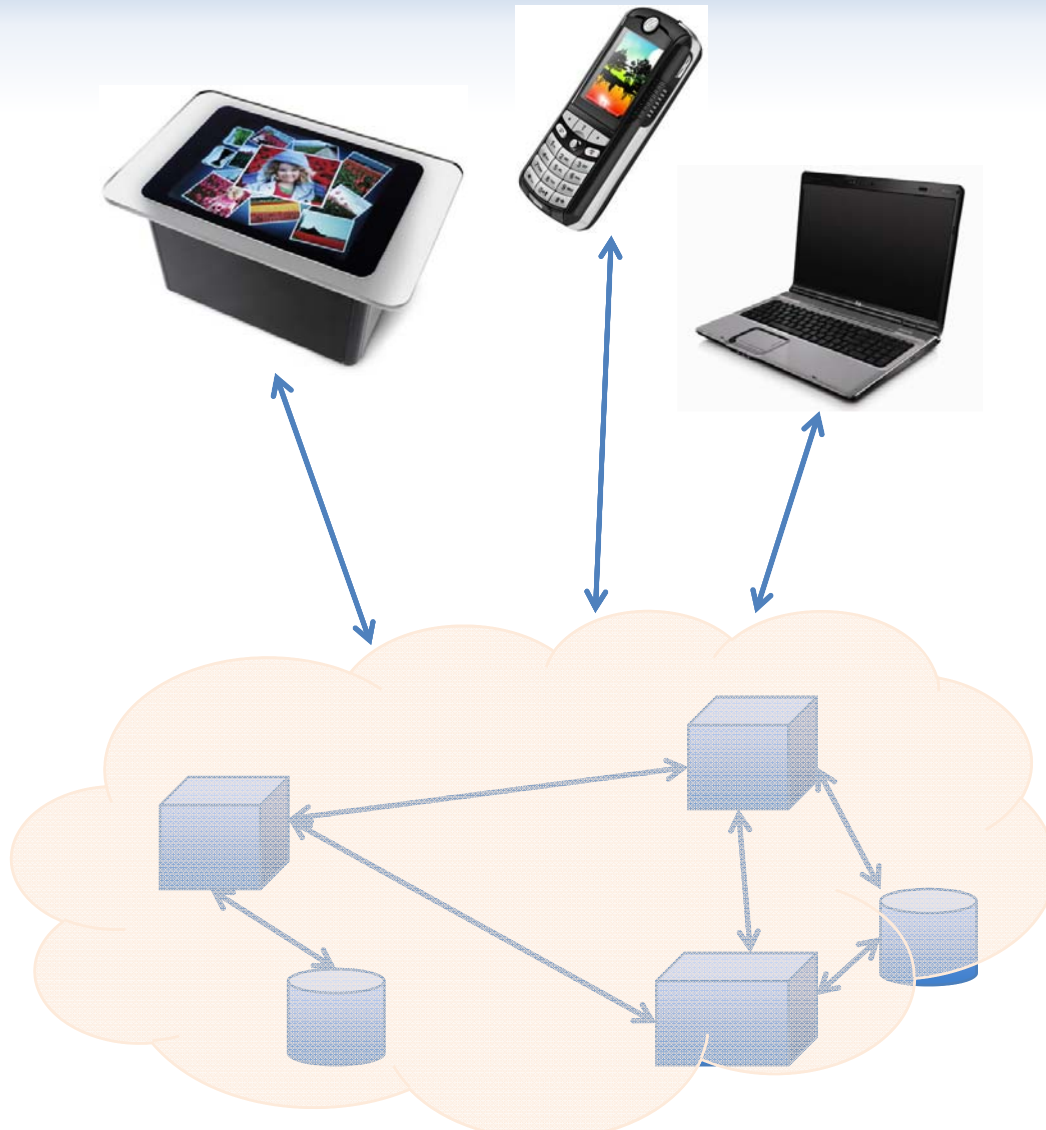
- A Framework for Discovery
  - Four basic components
- Data Discovery
  - Catalogs and index services
- The experiment
  - Computational workflow managing on-demand resources
- Data analysis and visualization
- Data product preservation,
  - automatic metadata generation and experimental data providence.

What did we learn?

- What did we learn?
  - Simple portal for data access is easy
  - Providing index and upload/catalog is harder
  - Providing on-demand scalable services for dozens or hundreds of users is very hard.
- Why?
  - TeraGrid is best effort.
  - Fault tolerance is very hard…. SC users learn to tolerate faults.
  - Basic services are extremely unreliable under load.

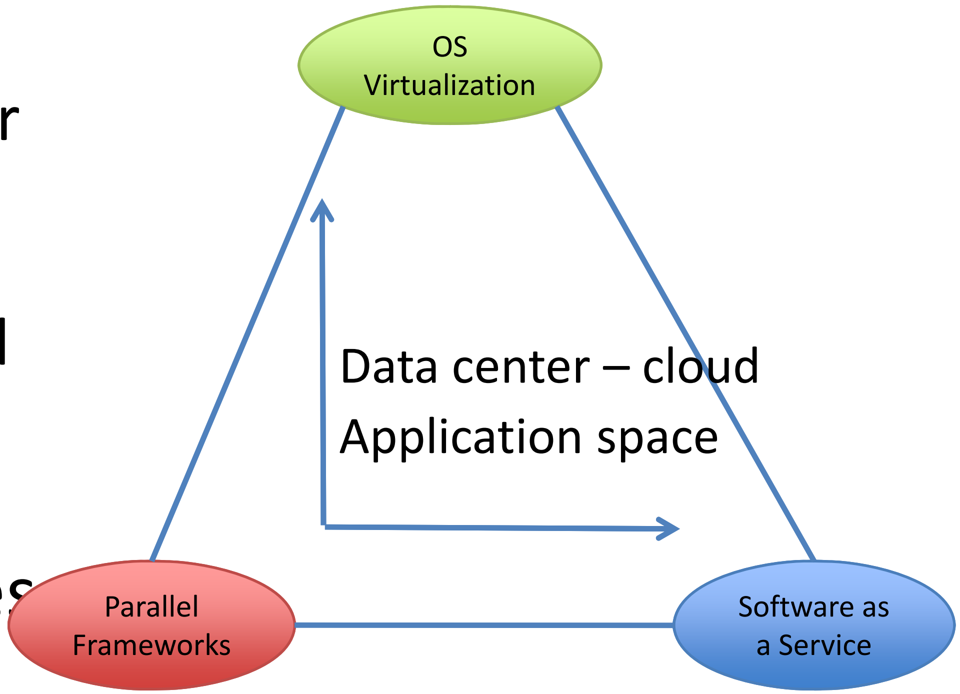
Grids vs. Data Center Cloud ?

- Grid standards focused on resource aggregation.
- Data Centers are about delivering "scalable" application services to users



Today's Cloud Programming Models

- Cloud definition
  - A data center plus a layer of system software services designed to support the creation and scalable deployment of application services.
- Current practices defines a space of approaches
  - OS Virtualization
  - Parallel Frameworks
  - Software as a Service

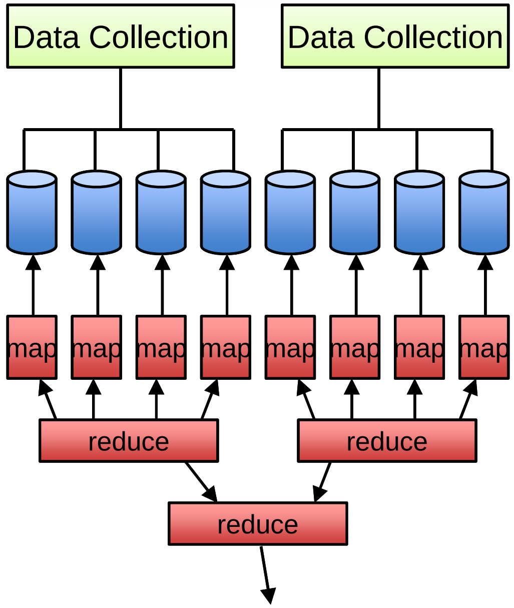


OS Virtualization

- Simple Idea (promoted by Amazon)
  - Provide a platform that can allow app designers to upload a VM image and store it and then instantiate copies on demand.
- Give app designers a menu of VM choices
  - Flavors of Linux and Windows with standard web servers and database components.
- Give them basic web services to manage instances and back-end data.
  - Requires sys admin-level management
  - 3rd party companies provide high level app config tools (RightScale, GigaSpaces, Elastra, 3Tera, …)

Parallel Frameworks

- Deploy a datacenter-wide application framework that makes it easy to build highly parallel data analysis application.
- Use simple parallel templates with "inversion of control" concept:
    - App designer provides kernel of data analysis application
    - The framework controls parallel execution and access to parallel file system and data structures.



Map: apply application kernel
Function to data chunks
in parallel

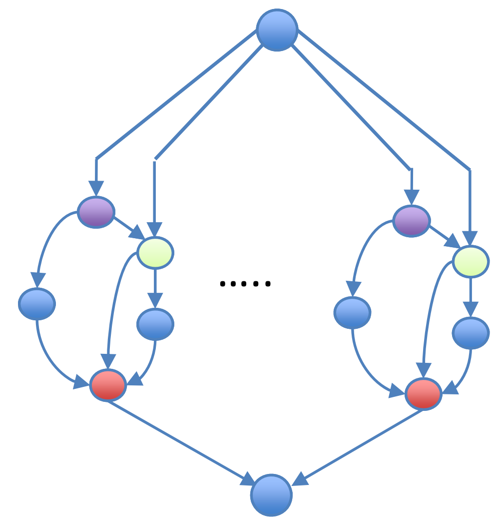
Reduce : apply application data
Reduction filter to map output.

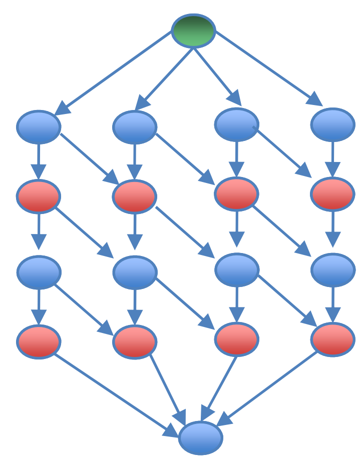
Parallel Frameworks, cont.

- Google has made MapReduce famous.
- Based on Google File System
  - Parallel, distributed, redundant "read often, write infrequently" file system.
- BigTable – a parallel data structure built on GFS
  - Two dimensional sparse map.
    - Cells are time-stamped, to allow for history
  - BigTable can be used as parallel input or output structure for map reduce computations.
- Open Source version: Hadoop created by Yahoo!
  - Part of NSF big data program.

Beyond MapReduce

- MapReduce is only one instance of many possible parallel execution templates.
- Simple parallel workflow/macro-dataflow/systolic constructs can be used to create arbitrarily nested, massively parallel execution patterns
- It is possible to build control & execution frameworks to run these on large data centers.
- The parallelism effectively exploits manycore.





Software As A Service

- The role of the "cloud" is to provide a place where application "suppliers" can make apps available to clients.
  - The applications are then hosted "services".
  - The cloud automatically scales to meet client demand
  - The cloud is reliable and robust.
- The data center provides the tools and "core" services that make it easy to build the apps.

Implementing SaaS

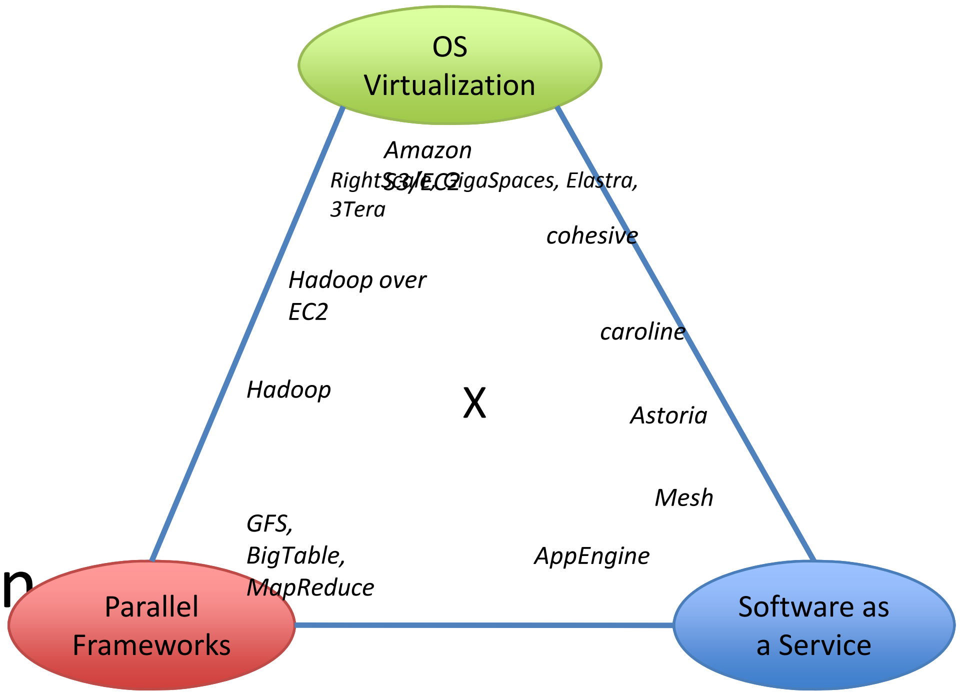
- Focus the virtualization concept
- One solution:
  - Provide a high level language VM and a rich library of core services.
- Client applications can access the functionality of the remote program through automatically generated WS or REST service interfaces.
- A local version of the same program can have some functionality when the client is off line.

Examples

- Cohesive has a Ruby on Rails engine for cloud app deployment.
- Google AppEngine is a Python runtime with APIs to access things like BigTable
- Microsoft Astoria is ADO.net based
  - Expose any data object as a URL to an ATOM or JSON representation.
- SUN's Project Caroline is based on spawning remote Java VMs

The Space of Solutions

- It is possible that the best cloud model for science lies somewhere in the middle (X).
- It should
  - Exploit OS Virtualization
  - Allow for a wide expression of parallelism
  - Be easily deployed as a service.



Conclusions - Some Requirements for X

- Simple Service API for Data
  - Support for very large, heterogeneous collections
  - Indexing, metadata, search & discovery, access
  - Streams and virtual data
  - Notification
- Tools for rapid application deployment
  - Scalability in two dimensions: parallel apps and multiple instances.
  - Turn an application into a service visible in a catalog.
- Tools for application composition (workflow/mashups)
- Community tools
  - Security (authentication & authorization)
  - Desktop integration: Portal and Web 2.0

Software Implications – High Level

- Dynamic code placement and tier splitting.
  - Given a smart client device and a variable bandwidth to the data center, how can we dynamically partition the split of the tasks in the application between the client and the data center?
    - In some cases the client side of the application may migrate from one client to another or it may be shared between multiple clients simultaneously.
- Quality of Service.
  - Can the application/runtime “sense” load spikes and communicate to the lower level of the system about an impending drop in performance so that additional resources can be added.

High Level Issues

- Can a single application adapt dynamically to use additional resources?
- Scalability and Geo-distribution?
  - Two types of scaling:
    - Capability per user
    - Number of concurrent users.
- Building application services that can fail.
  - How well does the development environment encourage the construction of applications that know how to recover from low-level failure or loss of resources?
- To what extent can an application understand how to minimize its own energy needs?

Dimensions of the Space

- The software stack needs to support the space of DC apps
  - From current apps (search, mail, messaging) to futures
  - Same stack for science!
- Client well beyond browser
  - Multiple concurrent streams between the client and data center
    - Sensors streams (cameras, voice etc.), user keyboard input.
  - A single user may have multiple clients
    - Operating in the same “session” concurrently or
    - Hand off session from Desktop to Laptop to phone to car to TV.

More Dimensions

- A clean programming model for handling session state and update.
  - State is updated by agents as well as user.
- “State is bad” so what is a better idea?
- Need a service composition model
  - Future apps may be mashups
- Agent creation and management
- Large scale parallel search/analysis is used.

Metrics

- How do we auto-instrument applications so they can be remotely measured and managed?
- How can we tell when the programming model works?
  - Is there a correctness model for a cloud app?
- How do we design apps that understand a service level agreement or QOS?

Conclusions

- "Data centers and supercomputers are twins separated at birth" – Dan Reed.
- A Grid can be the physical underpinning of a cloud, but providing cloud apps requires more than basic grid services.
- We need a programming model that will allow us to build apps that
  - Scale (in complexity and in the number of users)
  - Are reliable (i.e. tolerate their own failure)
  - Are debug-able, manageable
  - Are energy efficient.

Microsoft®

Your potential. Our passion.™