Module E:
Distributed Scientific Computing

Lecture E-3: Applications, Infrastructure Redux and Homework

Dr Shantenu Jha

Overview of Module E
Distributed Scientific Computing

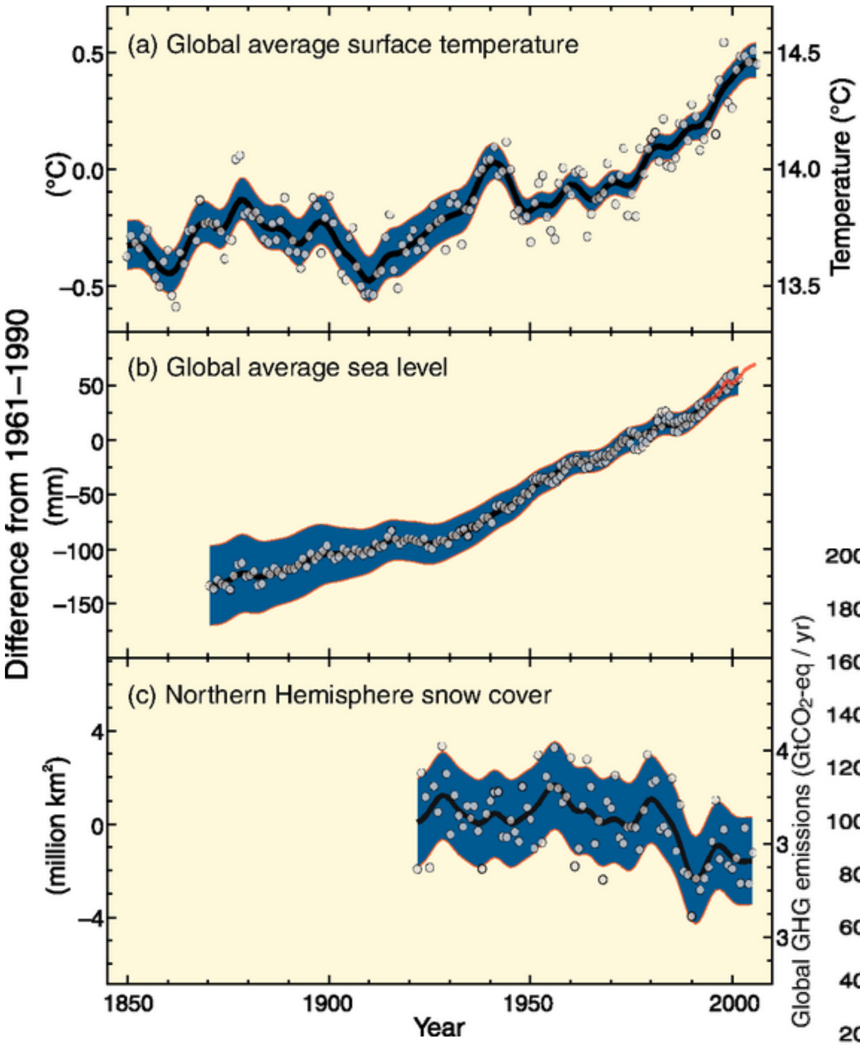
- E3: Applications, Infrastructure and Homeworks
  - Remaining Applications
    - Ensemble simulations, Replica-Exchange
  - Distributed Computing Infrastructure (DCI)
    - OSG, Amazon & Azure
  - Introduction to Pilot-Jobs
    - Discuss HW #0 (saga on futuregrid)
    - Introduce HW #1
  - Module E Project Discussion

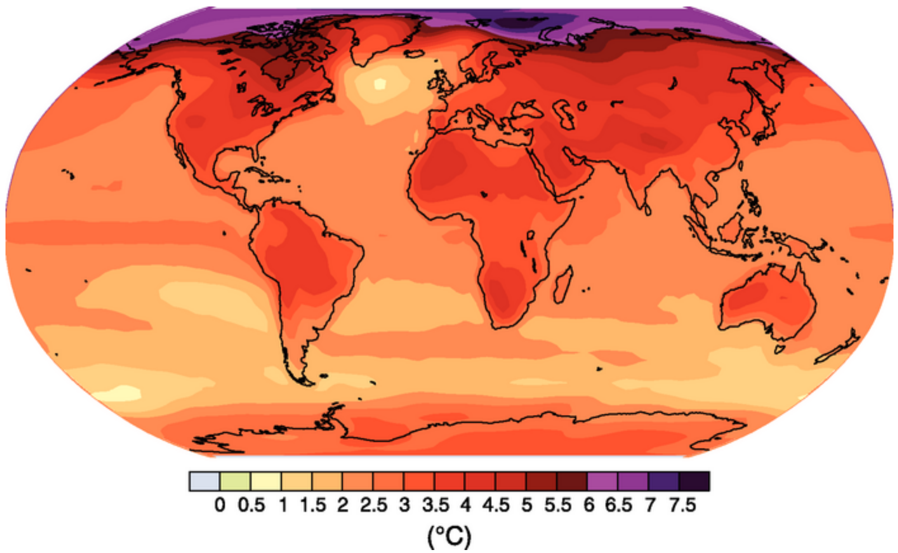
CENTER FOR COMPUTATION
& TECHNOLOGY

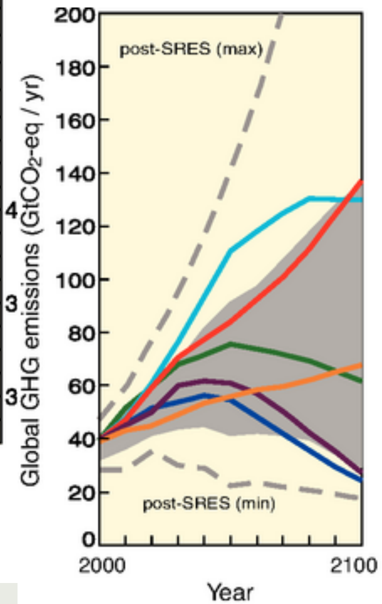
CLIMATEPREDICTION.NET

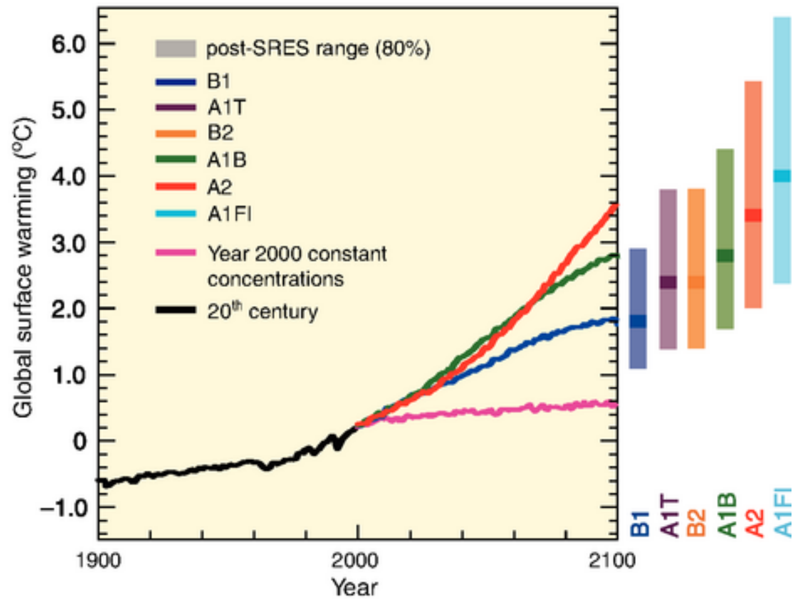
IPCC AR4:

http://www.ipcc.ch/publications_and_data/ar4/syr/en/contents.html





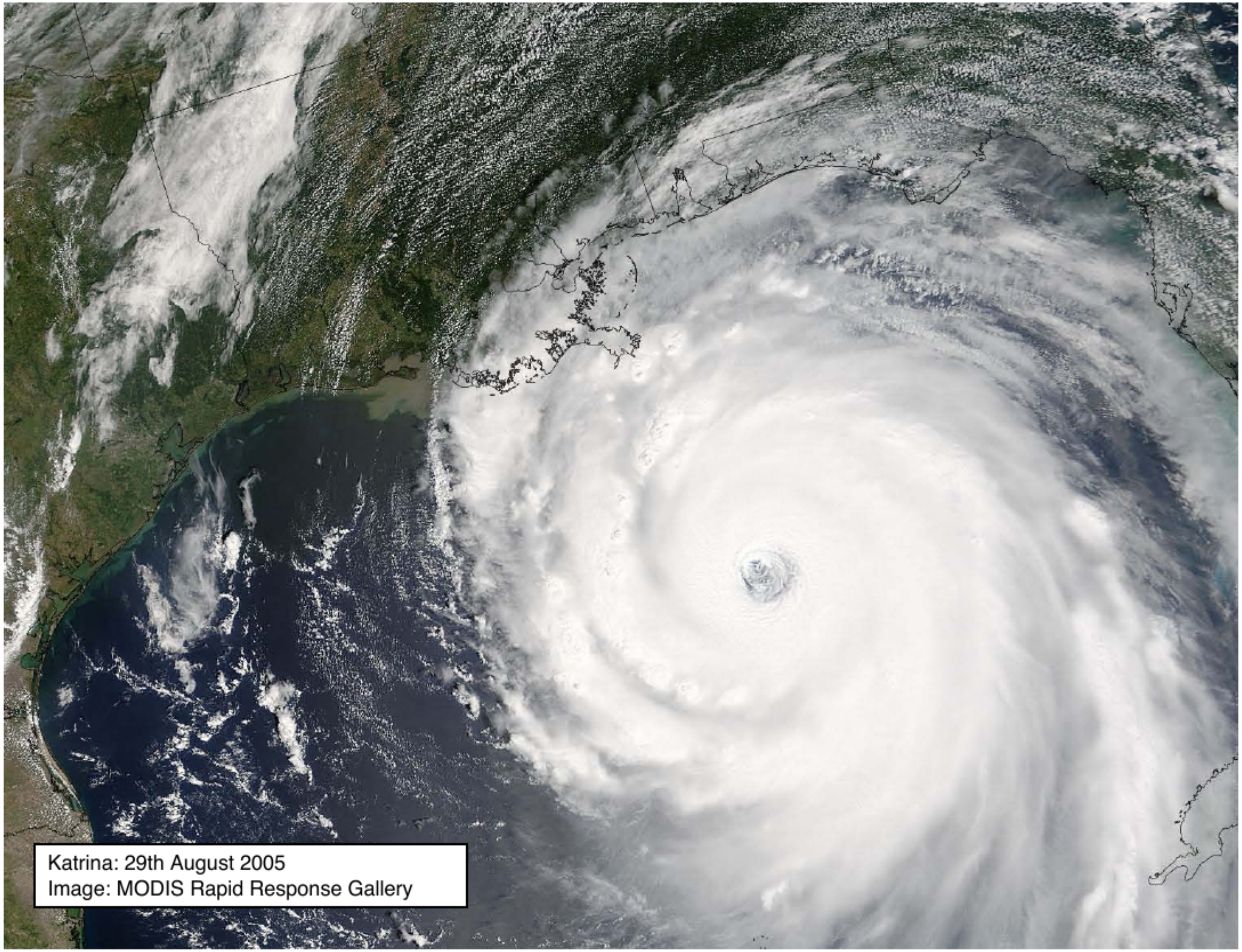




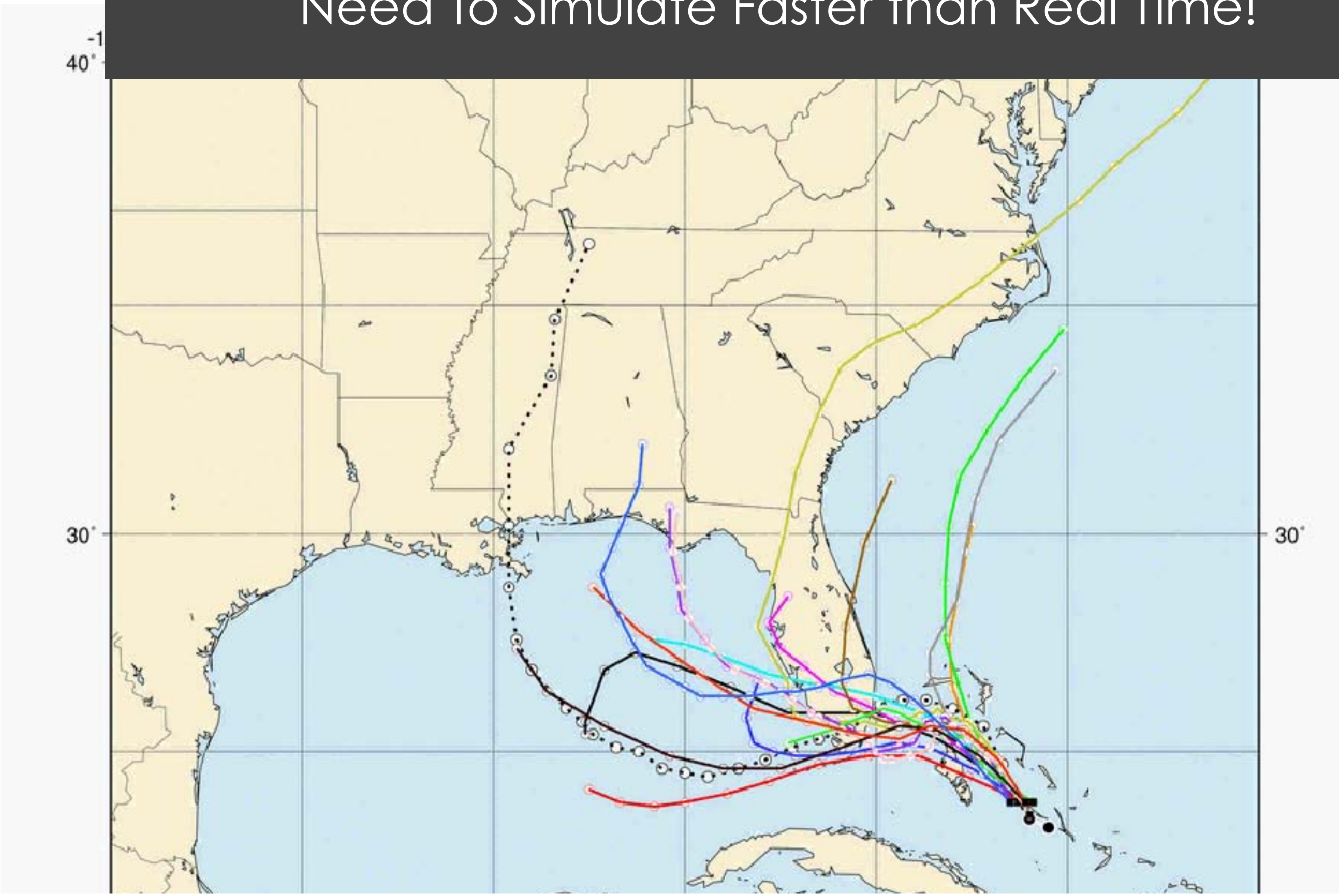
Understanding ClimatePrediction.net

- Why Distributed?
  - Many small indep. Comp. tasks – naturally decomposable
  - Access many more resources without owning
    - Petaflop computing years before Petaflop Computing era!?
- How Distributed?
  - BOINC -- basis for @HOME projects [Volunteer Computing]
  - "Trickles" – job reporting to the Master (project server)
  - Data too large to aggregate and analyze centrally
    - Hence must operate on data in-situ
- Limitations and Success?
  - Coordinating work across all the resources
  - Managing changing number of resources and failures

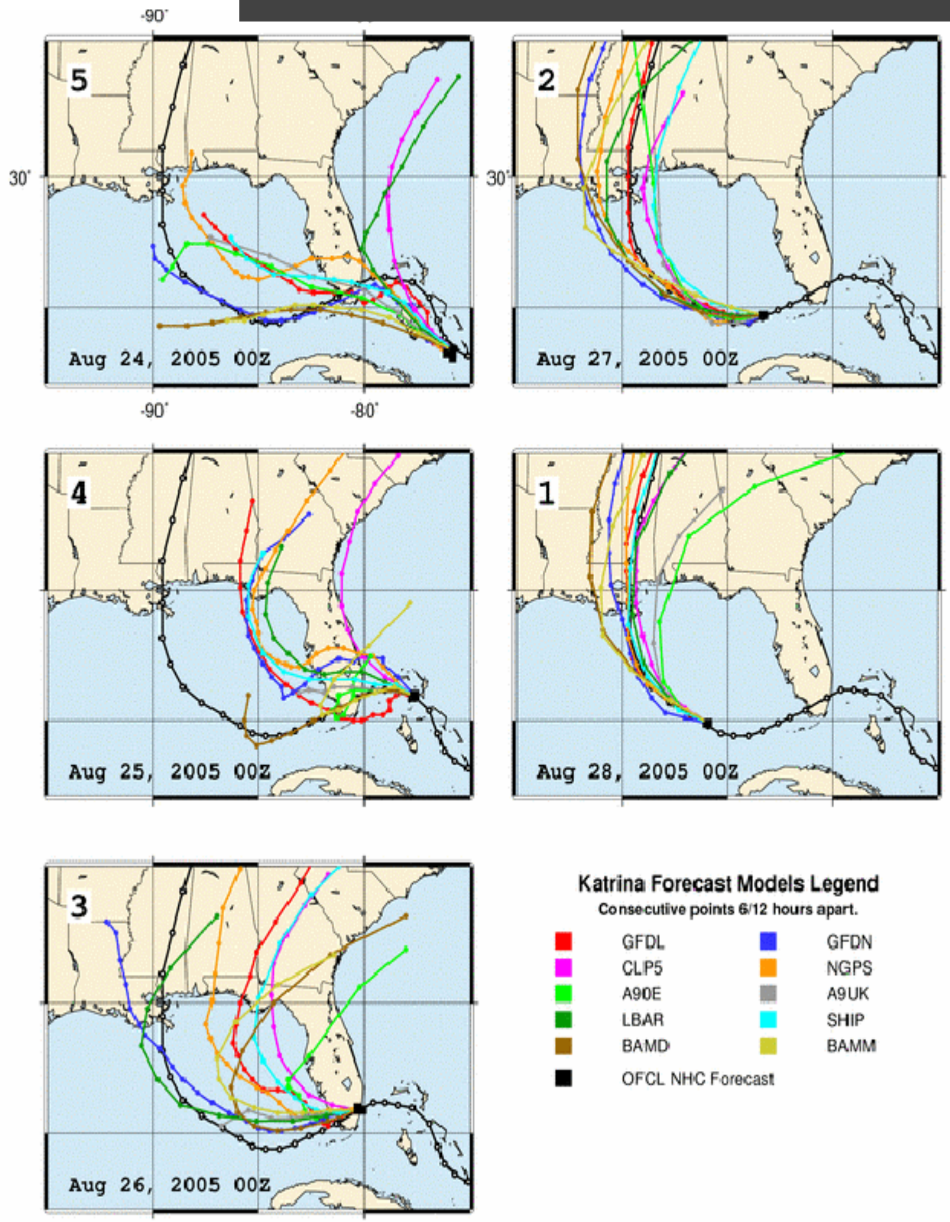
SCOOP..

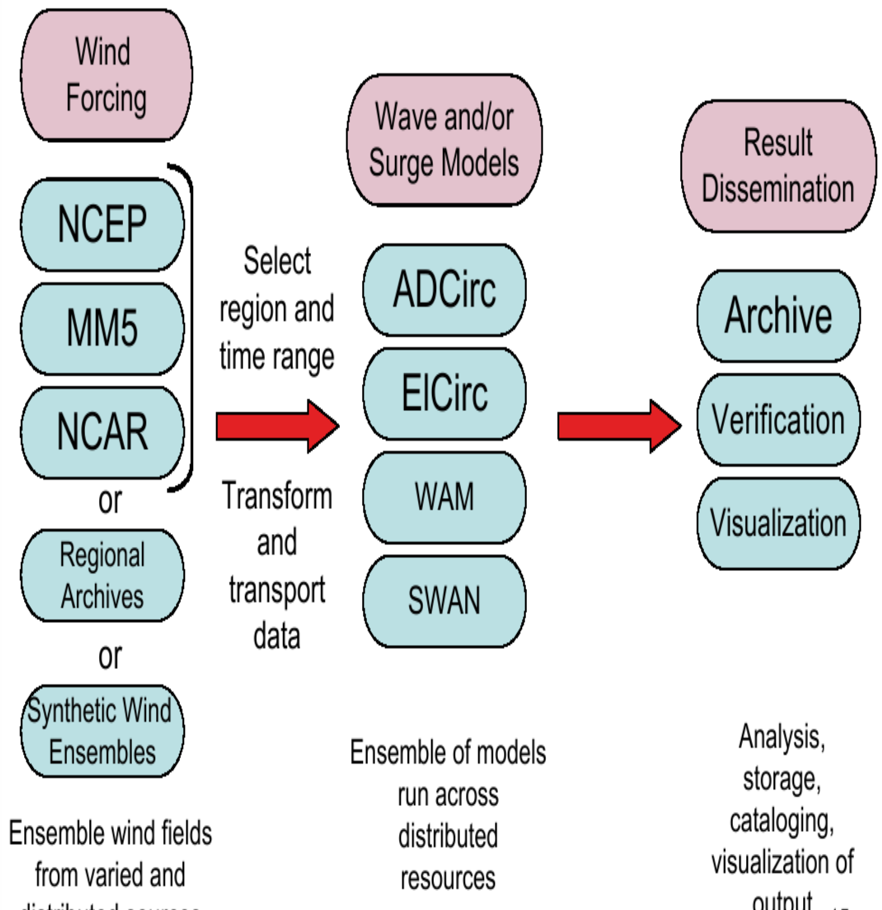

Katrina: 29th August 2005
Image: MODIS Rapid Response Gallery

Need To Simulate Faster than Real Time!



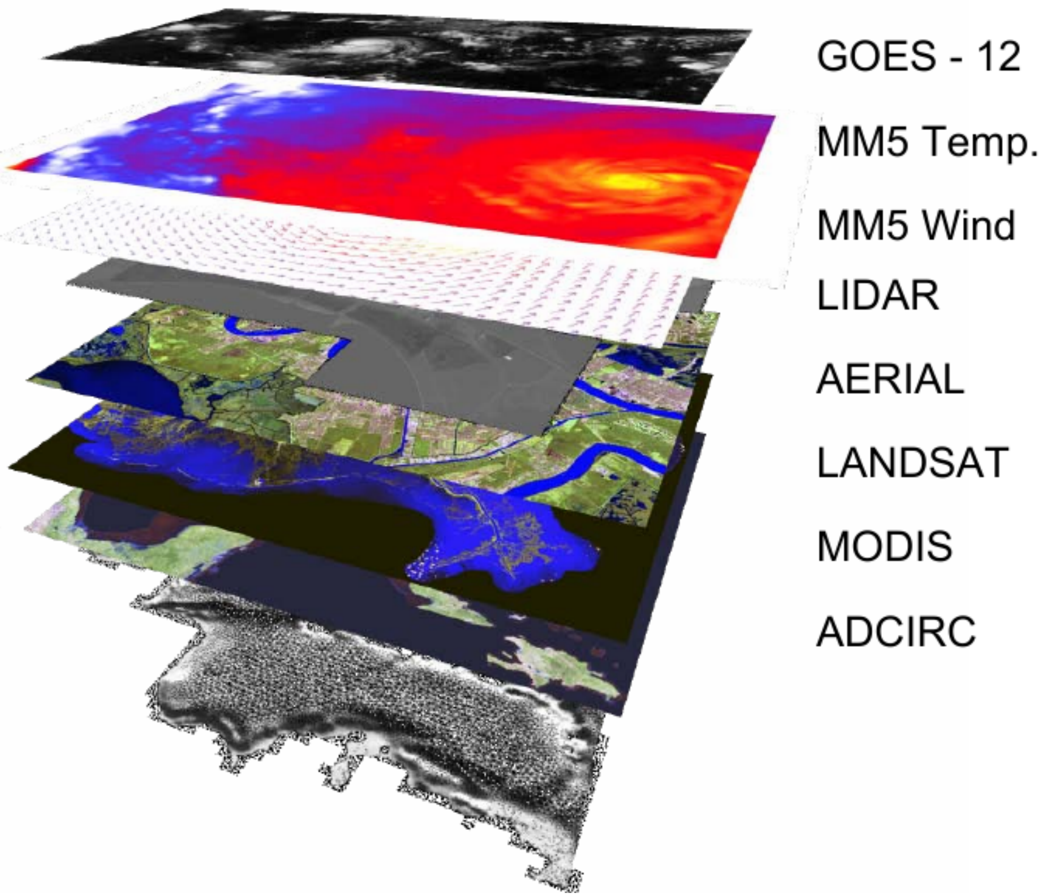
Heterogenous: Compute Models



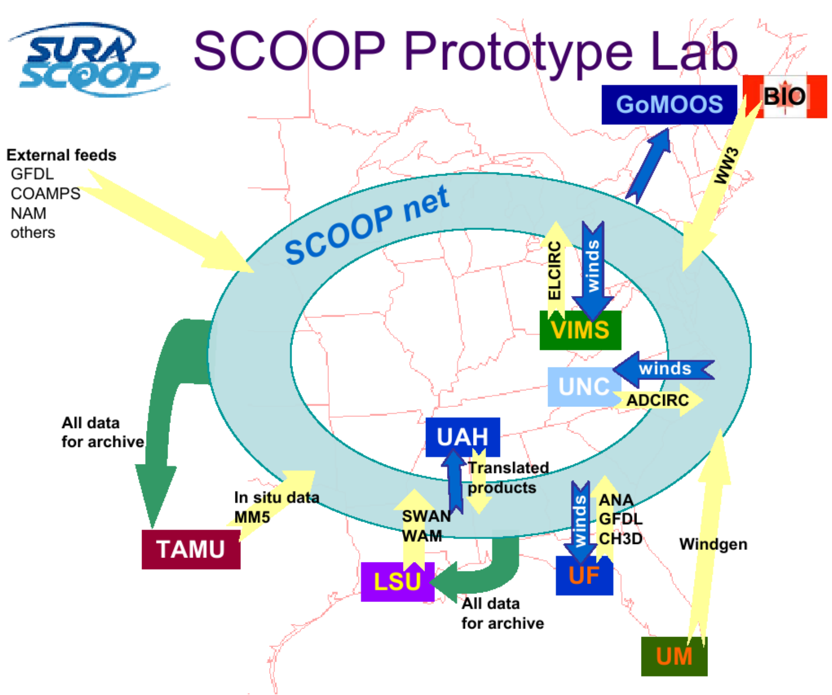


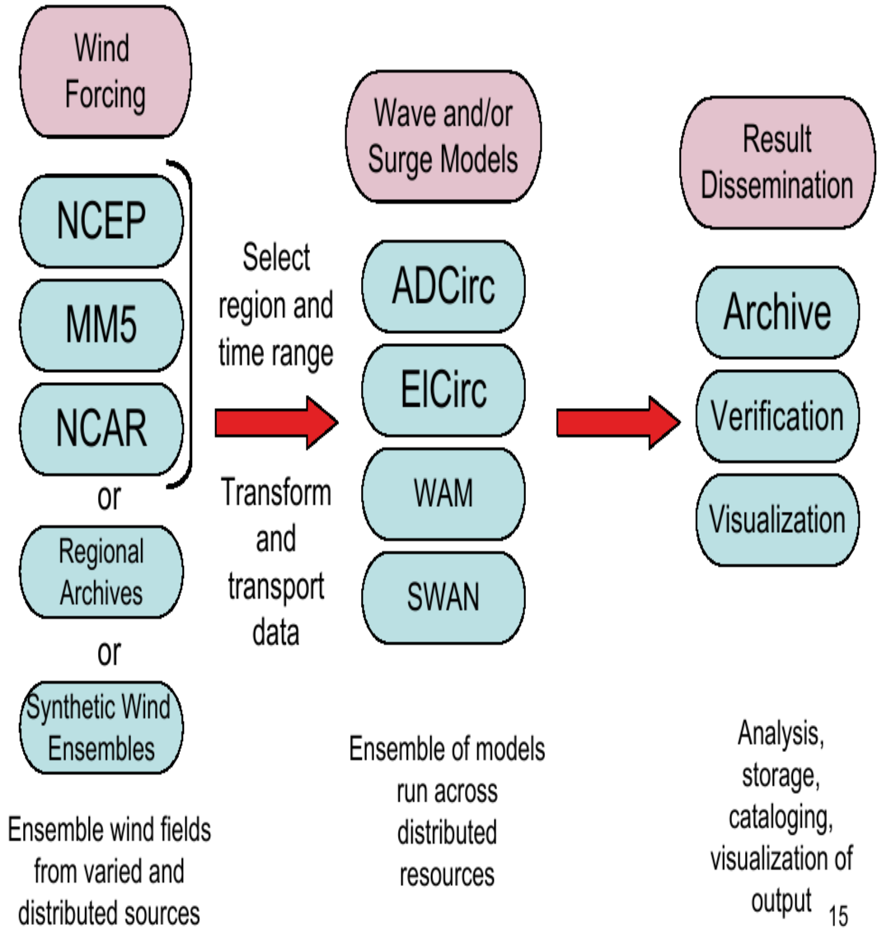
Many Heterogeneous Components

- Simulation data (forecast, nowcast, hindcast)
  - 2D, 3D
- Sensor data (time series)
- Remote imaging
- Aerial photography
- LIDAR
- Weather satellites
- GIS



Naturally Distributed..





Understanding SCOOP

- Why Distributed?
  - Naturally Distributed: Geographically distributed producers and end-users
  - High Peak Demand and quick response time
    - But with very low duty cycle --- Economic Argument !!
- How Distributed?
  - Customized workflows
- Limitations and Success?
  - Not Robust – Many components that need to come together
  - Coordinating work across all the resources
  - Co-scheduling / Advanced Scheduling / Prediciton a challenge

Distributed Applications Summary

	Why Distributed?	How Distributed?	Challenges & Issues	How different from || ?
Montage	Processing > local limits	Workflow enactor	Coordination	[1, 2]
NeKTAR	Processing > local limits (memory)	MPIg	Advanced/Co-reservation	[1?, 4]
Ensemble-based/RE	Many compute-intensive task	SAGA, "Advert"	Coordination	[2,3]
ClimatePrediction.net	Many small tasks	BOINC, Trickles	Failures, variable # workers	[1, 4]
SCOOP	Peak req., naturally, Economic	Customized workflows	Not robust, adv. reservations	[1, 3, 4]

Open Science Grid

http://www.openscience.org

- Bottom-Up Organization: OSG brings together computing and storage resources from campuses and research communities into a common, shared grid infrastructure over research networks via a common set of middleware

- Philosophy: OSG offers participating research communities low-threshold access to more resources than they could afford individually, via a combination of dedicated, scheduled and opportunistic alternatives

- Management: OSG is a consortium of software, service and resource providers and researchers, who together build and operate the OSG project

Open Science Grid

http://www.openscience.org

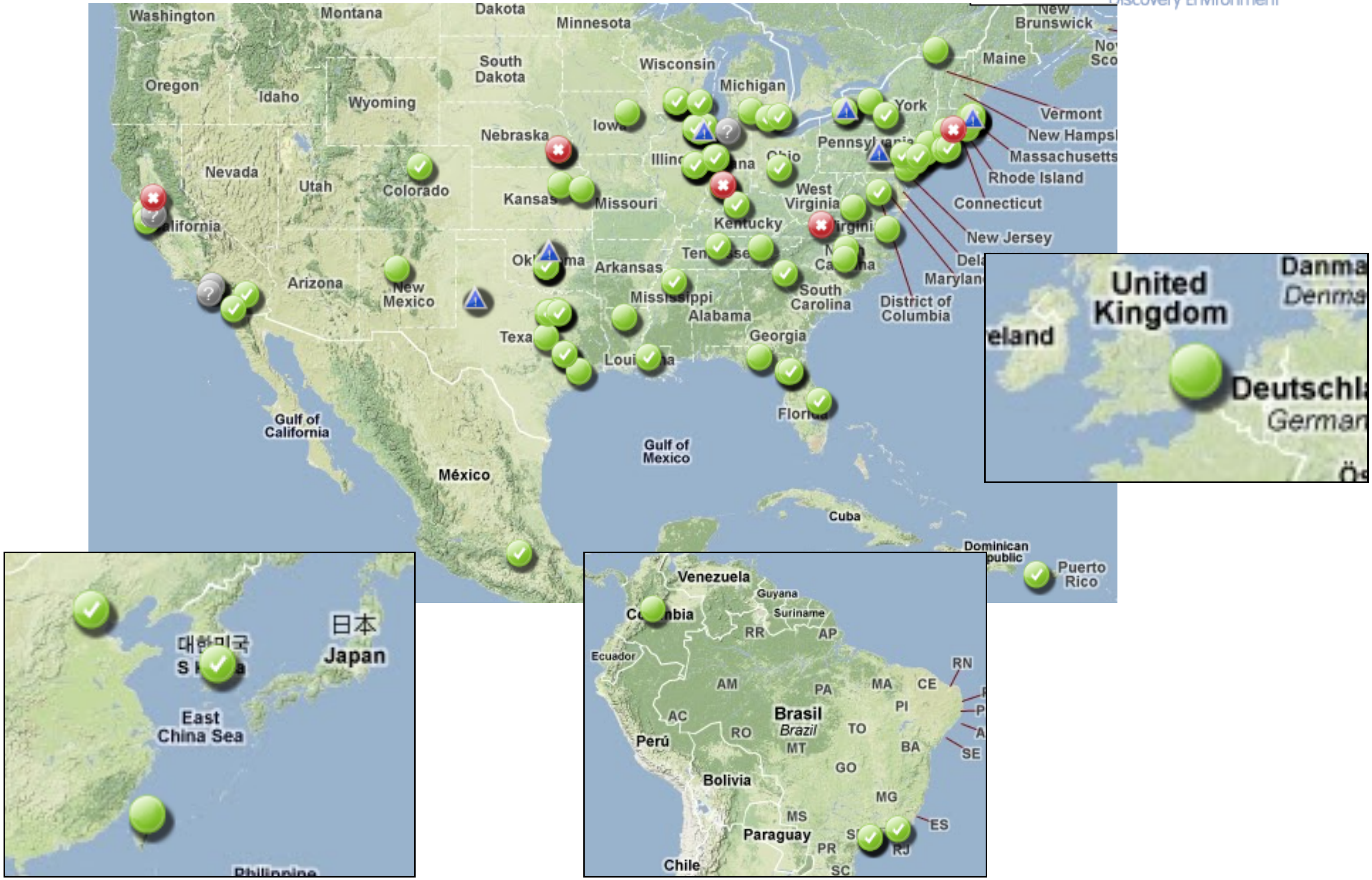
- OSG Consortium members' independently owned and managed resources make up the distributed facility, agreements between them provide the glue for it
  - Organized around Virtual Organizations

- Software: Virtual Data Toolkit provides packaged, tested and supported collections of software for installation on participating compute and storage nodes and a client package for end-user researchers.

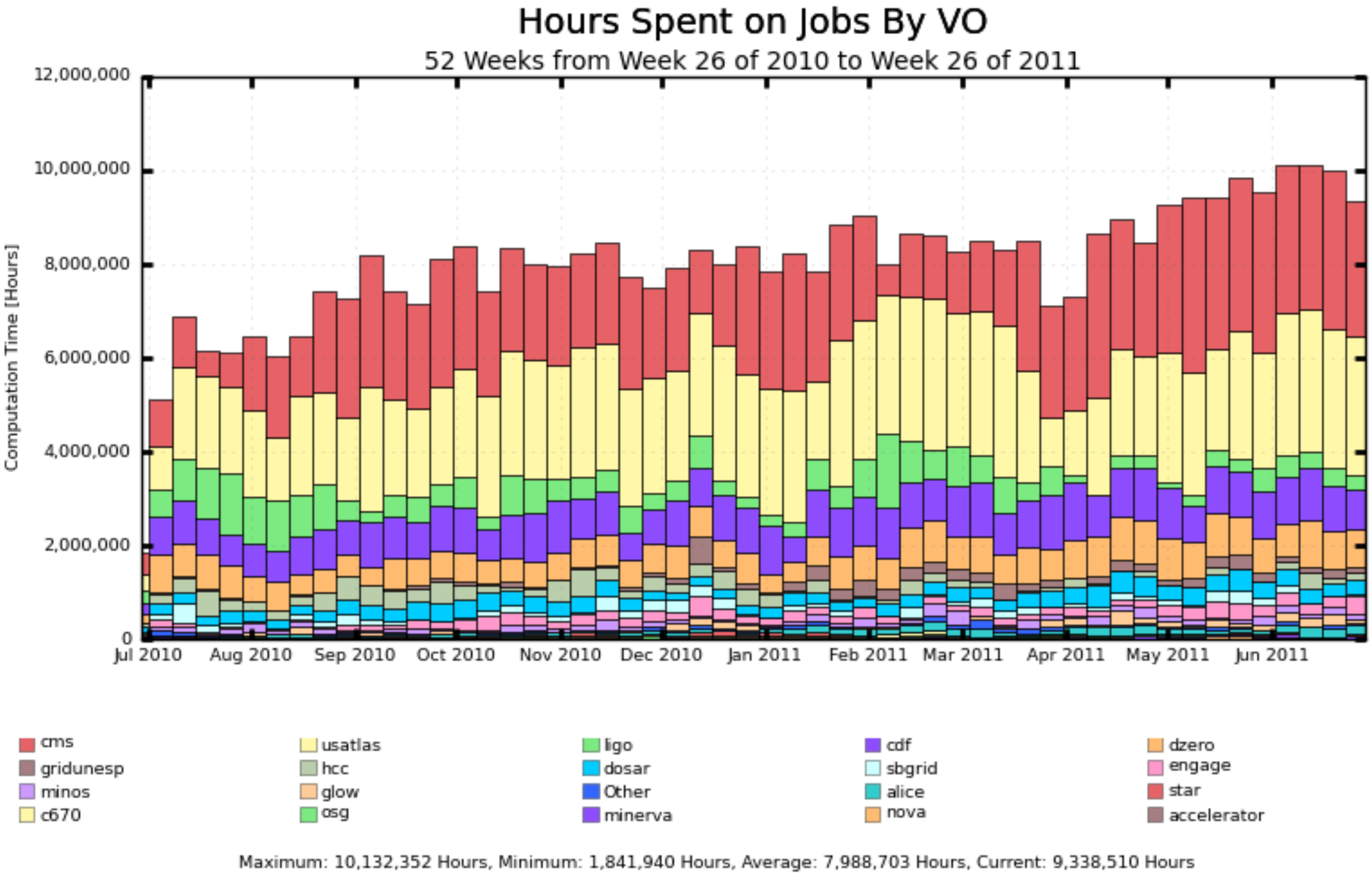
Some of the OSG Sites

Open Science Grid

XSEDE
Extreme Science and Engineering Discovery Environment



Some OSG Job stats



Average: ~3,500,000 jobs/week

OSG Middleware

User Science Codes and Interfaces

Applications

VO Middleware

Biology
Portals, databases etc

Astrophysics
Data replication etc

HEP
Data and workflow management etc

OSG Release Cache:

OSG specific configurations, utilities etc.

Infrastructure

Virtual Data Toolkit (VDT)

core technologies + software needed by stakeholders:many components shared with EGEE

Core grid technology distributions:

Condor, Globus, Myproxy: shared with TeraGrid and others

Existing Operating, Batch systems and Utilities.

It takes VOs to make OSG work!

cdf	Collider Detector at Fermilab
cms	Compact Muon Solenoid
compbiogrid	CompBioGrid
des	Dark Energy Survey
dosar	Distributed Organization for Scientific and Academic Research
dzero	D0 Experiment at Fermilab
engage	Engagement
fermilab	Fermi National Accelerator Center
fmri	Functional Magnetic Resonance Imaging
gadu	Genome Analysis and Database Update
glow	Grid Laboratory of Wisconsin
gpn	Great Plains Network
grase	Group Researching Advances in Software Engineering
gridex	Grid Exerciser (GEx)
grow	Grid Research and Education Group at Iowa
gugrid	Georgetown University Grid
i2u2	Interactions in Understanding the Universe Initiative
ligo	Laser Interferometer Gravitational-Wave Observatory
mariachi	Mixed Apparatus for Radar Investigation . . .
nanohub	nanoHUB Network for Computational Nanotechnology (NCN)
nwicg	Northwest Indiana Computational Grid
osg	Open Science Grid
osgedu	OSG Education Activity
sbgrid	Structural Biology Grid
sdss	Sloan Digital Sky Survey
star	Solenoidal Tracker at RHIC
usatlas	United States ATLAS Collaboration

OSG Usage Modes

Application Type	Characteristics & Examples
Simulation	CPU-intensive, large number of independent jobs; e.g., physics Monte Carlo event simulation
Production processing	Significant I/O of data from remote sources & long sequences of similar jobs passing through data sets; e.g., processing of physics raw event data
Complex workflow	Use of VO specific higher-level services & dependencies between tasks; e.g., analysis, text mining
Real time response	Short runs & semi-guaranteed response times; e.g., grid operations and monitoring
Small-scale parallelism	Allocation of multiple CPUs simultaneously & use of MPI libraries; e.g., protein analysis, MD

OSG versus TG/XSEDE

- Worth a quick read:
  - http://tinyurl.com/3by3t79

European Grid Initiative
http://www.egi.eu/

- The objective of EGI.eu (a foundation established under Dutch law) is to create and maintain a pan-European Grid Infrastructure in collaboration with National Grid Initiatives (NGIs) in order to guarantee the long-term availability of a generic e-infrastructure for all European research communities and their international collaborators
- Coordinating activities between European NGIs EGI.eu will
  - Operate a secure integrated production grid infrastructure that federates resources from providers around Europe
  - Work with software providers within Europe and worldwide to provide high-quality innovative software solutions that deliver the capability required by our user communities

European Grid Initiative
http://www.egi.eu/

- Management Model: EGI Council with representatives from all National Grid Projects
- EGI: Follow-on project to EGEE, EGEE-II and EGEE-III
- Usage Modes:
  - Mostly HTC but not confined to HTC
  - All of OSG Usage Modes and more
  - Many diverse research areas and not just particle-physics

Amazon AWS

http://aws.amazon.com

- Story goes: Build capacity for X-mas. What do with spare capacity year around?
- “Utility Computing”
  - Around long before Amazon EC2
  - $0.10 per CPU-hour, plus bandwidth cost
- *aaS Model:
  - * = Infrastructure, Software, almost anything
- AWS: A set of APIs which give users access to Amazon technology and content
  - IaaS, but also “people as a service” – Mechanical Turk

Amazon Simple Storage Service (S3)

- Data Storage in Amazon Data Center
- Web Service interface
- No set-up fee, No monthly minimum
- Storage: $0.15 per GB/Month
- Data Transfer: $0.20/GB to transfer data
- Private and public storage
- Each object up to 5GB in size

Amazon Elastic Compute Cloud

- A Web service that provides resizable compute capacity in the cloud. Designed to make Web-scale computing easier
- A simple Web service interface that provides complete control of your computing resources
- Quickly scales capacity, both up and down, as your computing requirements change
- Changes the economics of computing:
  - Pay only for capacity that used; no cost of ownership
    - $a + bc$ becomes just bc

Amazon Elastic Compute Cloud

- No start-up, monthly, or fixed costs
  – $0.10 per CPU hour
  – $0.20 per GB transferred across Net
- No cost to transfer data between Amazon S3 and Amazon EC2
- More when we do Cloud Computing…

Azure

- Description: Microsoft's "Platform as a Service" (Paas) offering
  - Platform that is "Available" and "Scalable"
  - Cloud Based around virtualization
- Explicit Cost to Use
  - No cost to transfer data, only to use/store
- "Democratization of Infrastructure"
- Rich Data Abstractions
  - Large user data items: blobs
  - Service state: tables
  - Service workflow: queues
  - Simple and Familiar Programming Interfaces
    - REST: HTTP and HTTPS

Each VM Has…

At Minimum

- CPU: 1.5-1.7 GHz x64
- Memory: 1.7GB
- Network: 100+ Mbps
- Local Storage: 500GB

Up to

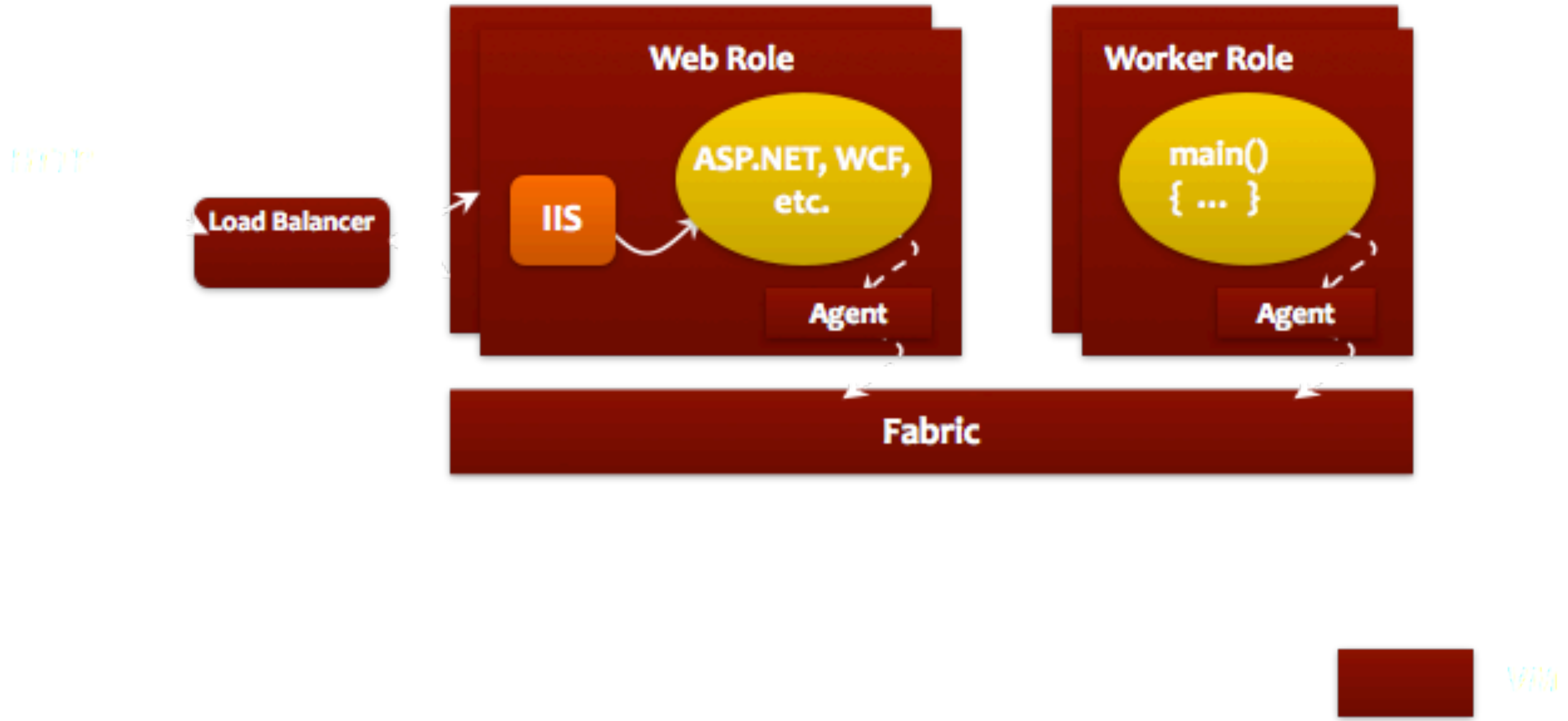
- CPU: 8 Cores
- Memory: 14.2 GB
- Local Storage: 2+ TB

Microsoft .NET Framework

Windows Server Internet Information Services 7.0

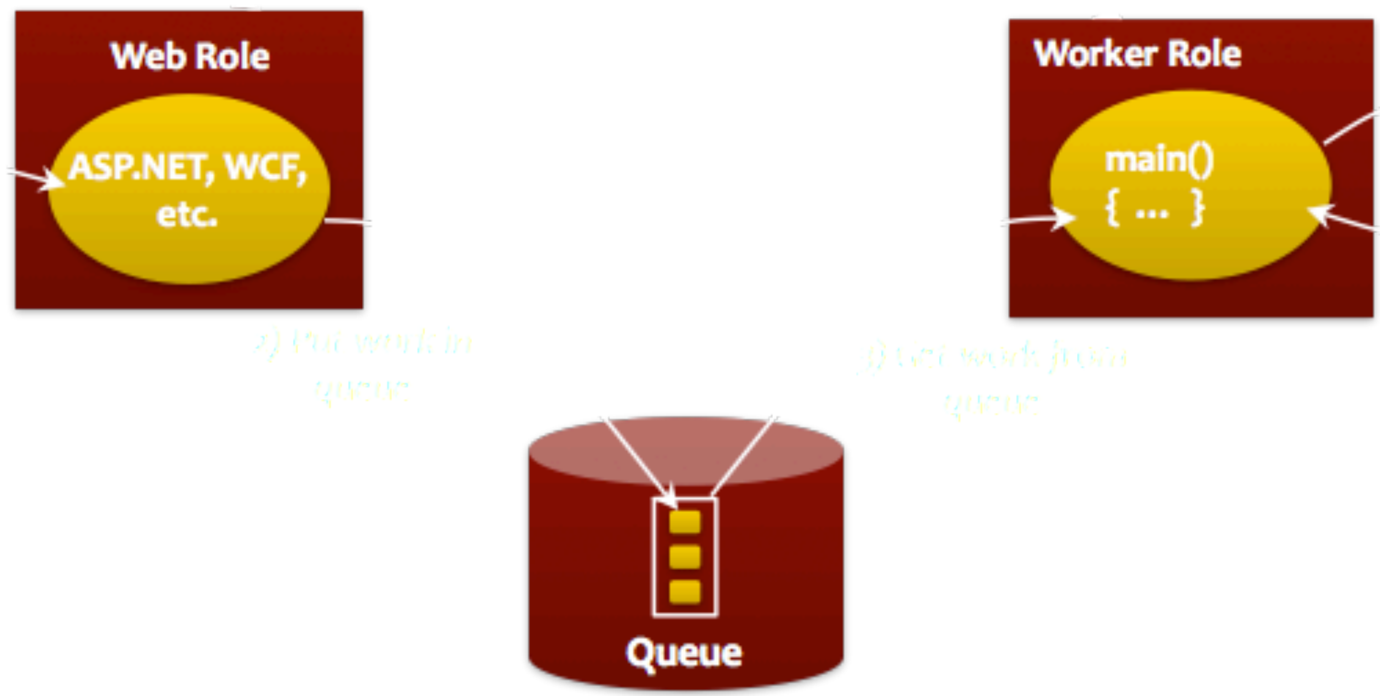
Windows Server 2008 Enterprise

Windows Azure Compute Service
A closer look

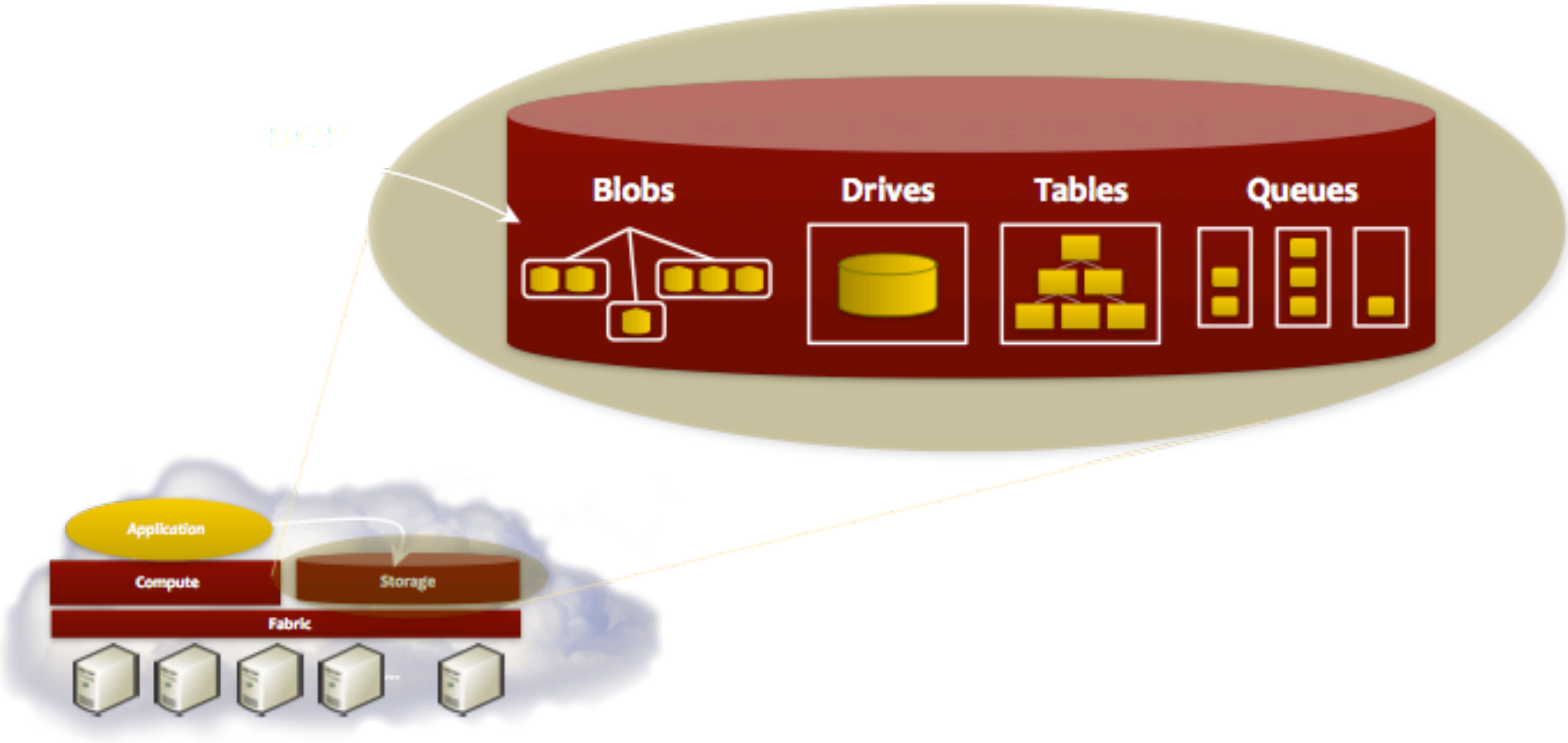


Suggested Application Model

Using queues for reliable messaging



Azure Storage Service
A closer look



DI - Summary

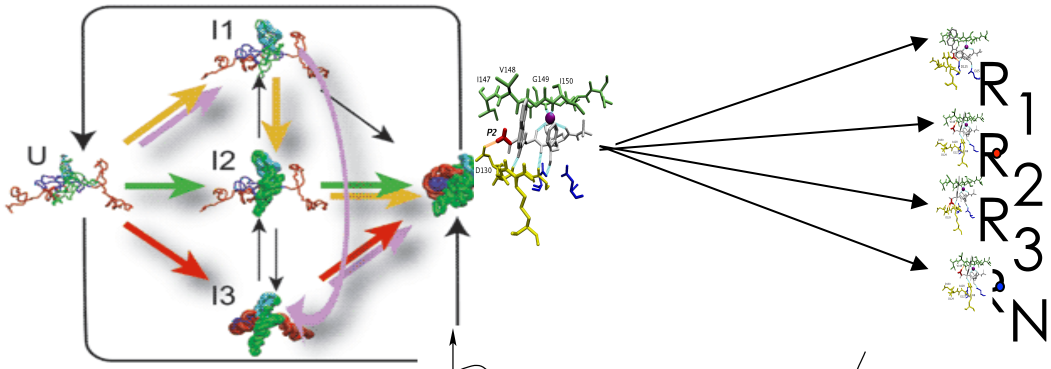
- A rich space of infrastructure & capabilities
- Not all DI have delivered on “their Grid” role
  - Why is difficult to answer, as many co-dependent factors, e.g. policy, social/technical/software ecosystem, external pressures..
  - But in general, “Narrow grid” (e.g OSG) have done better versus “Broad grids” (eg Teragrid)
- Interesting developments in the commercial sector: Google, Microsoft and Amazon
  - First time ever, Academic Research being done on commerical infrastructure!
  - Due to rise of data-intensive computing but also well designed infrastructure (Azure) and effective abstractions for applications (MapReduce)

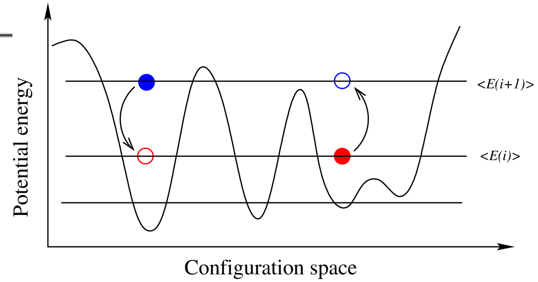
INTRODUCTION TO PILOT-JOBS

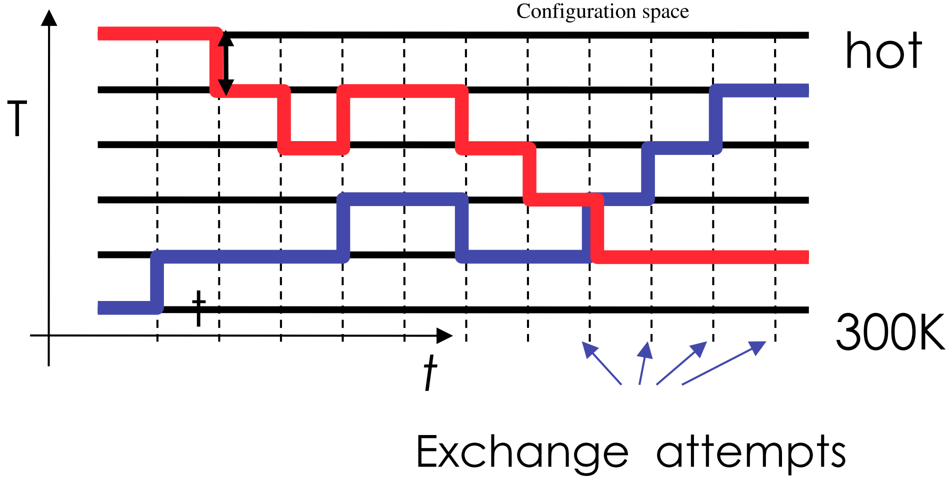
LSU
CENTER FOR COMPUTATION & TECHNOLOGY

HTC of HPC Jobs
Ensembles Can Enhance Sampling

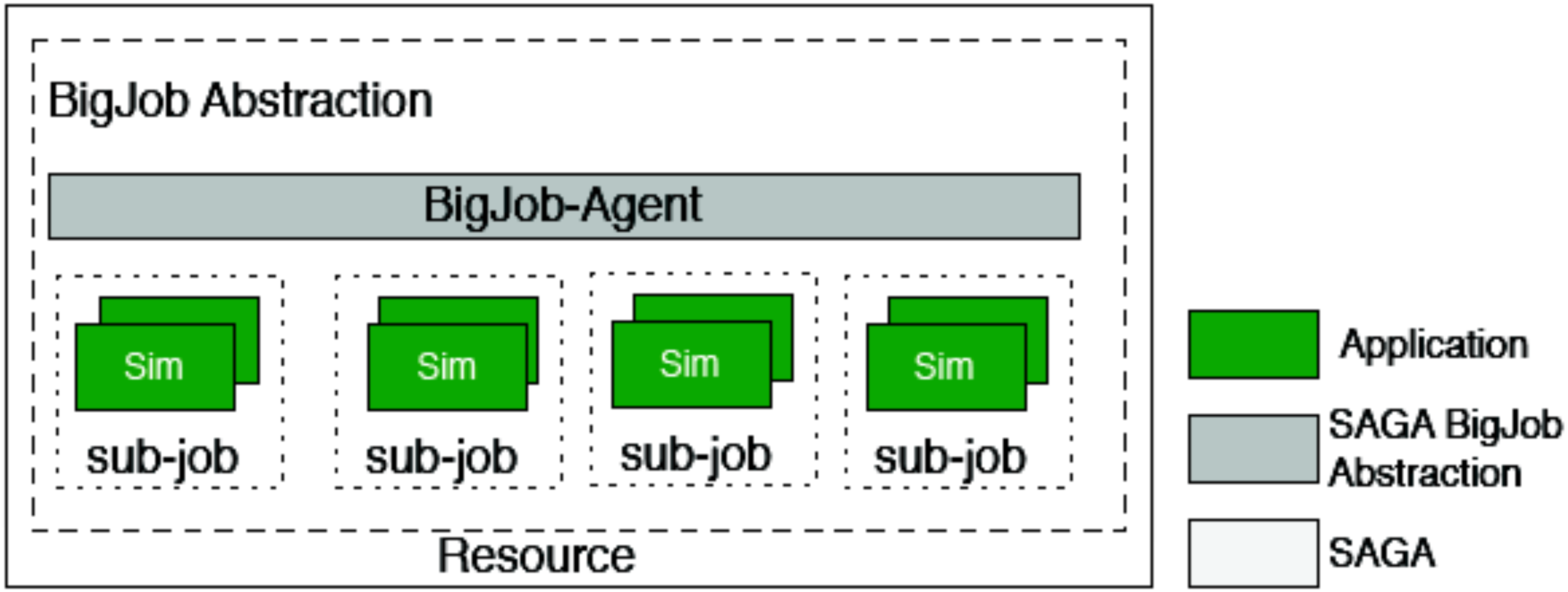
- Sampling is the challenge:
  - Long run vs multiple short runs?
- Task Level Parallelism
  – Embarrassingly distributable!
- Create replicas of initial configuration.
- Spawn 'N' replicas over different machine
- RE: Run for time *t ;* Attempt configuration swap. Run for further time t; Repeat till finish
  - We will not study Exchanges here





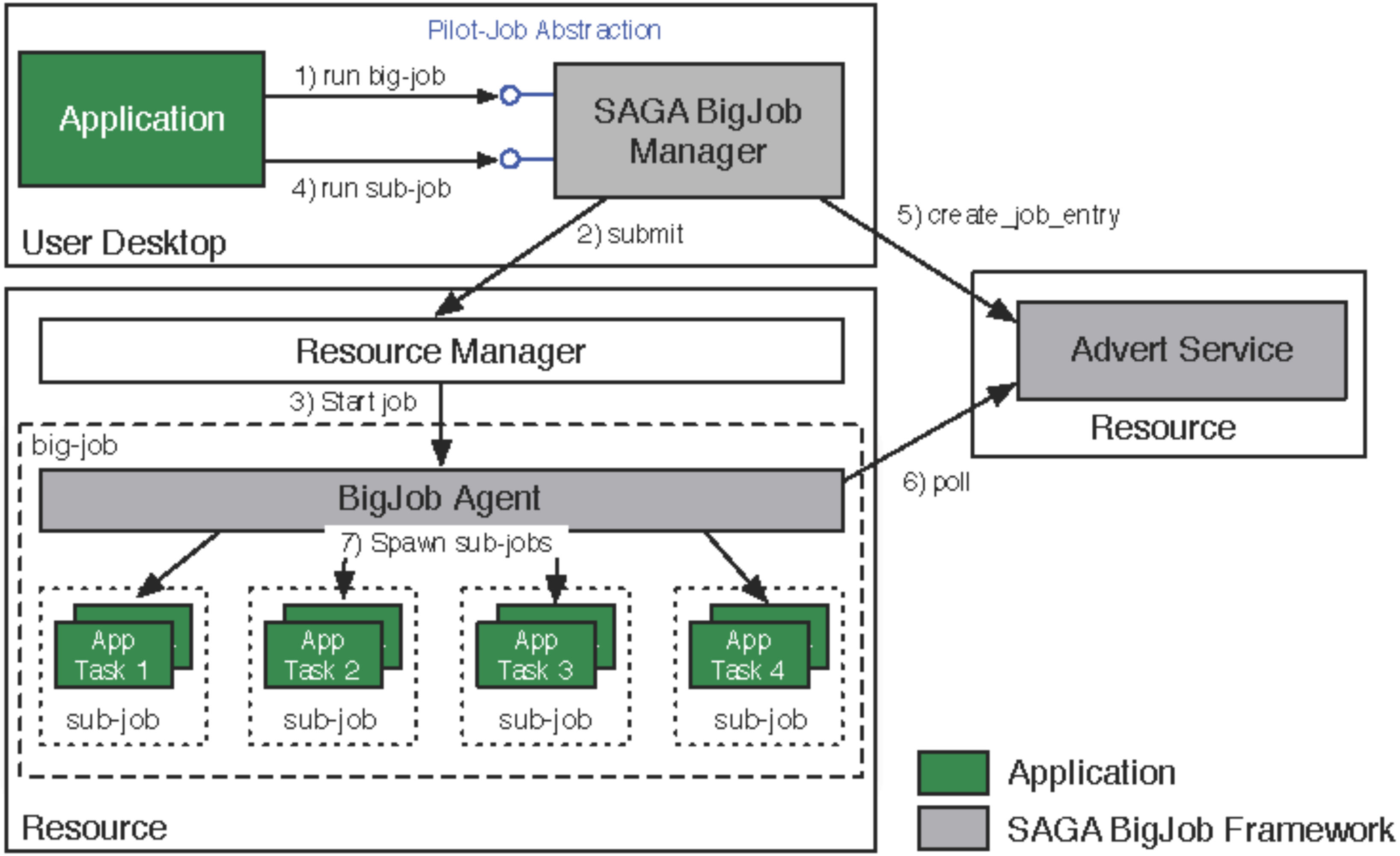


Abstractions for Dynamic Execution (1) Container Task



LSU
CENTER FOR COMPUTATION & TECHNOLOGY

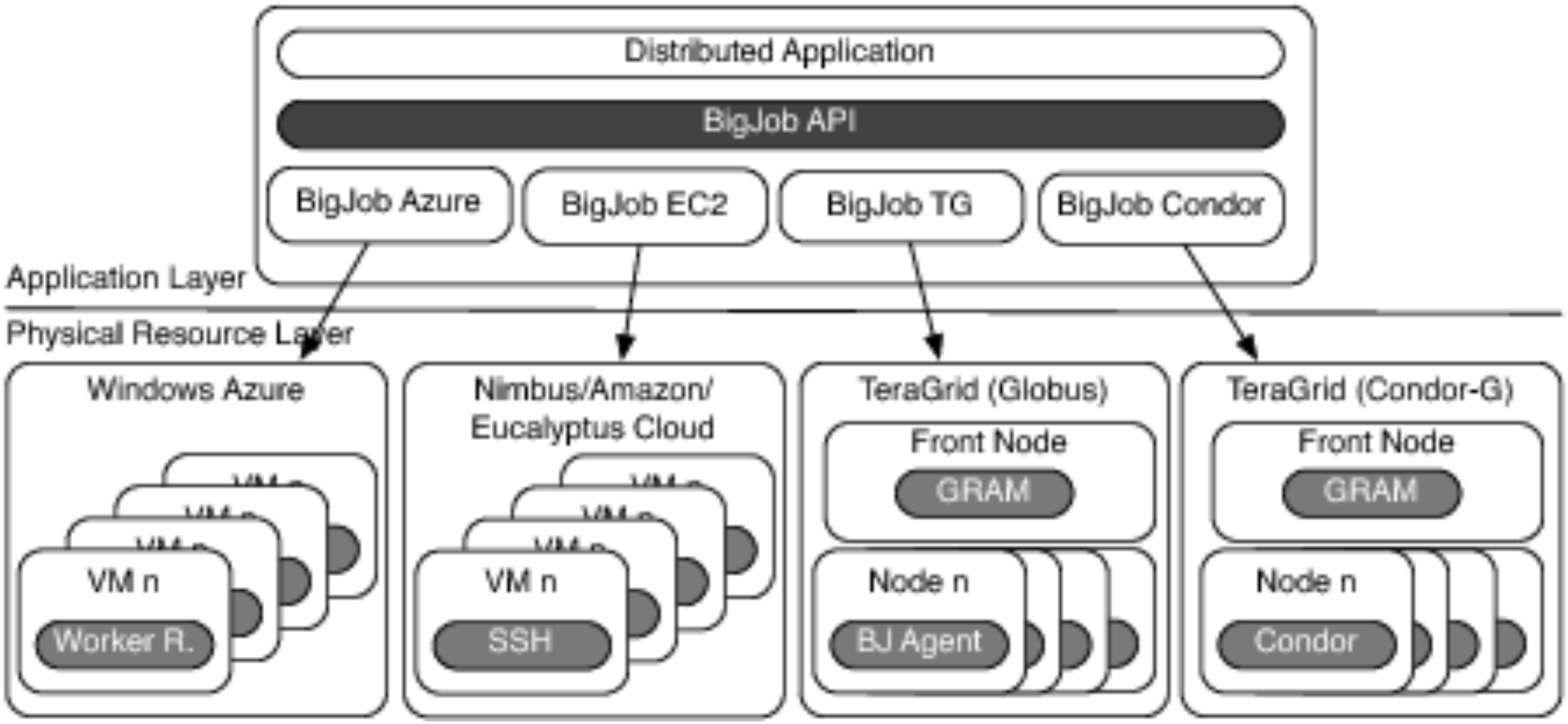
Abstractions for Dynamic Execution (2)
SAGA Pilot-Job (BigJob)



Pilot-Jobs: Commonly Used Abstraction

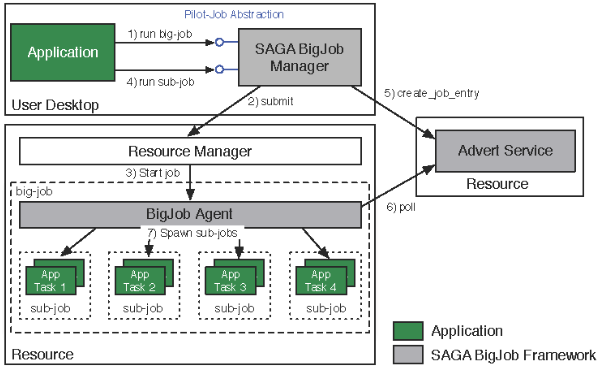
- Pilot-Jobs: Decouple Resource Allocation from Resource-Workload binding
- Pilot-Jobs are used for:
  - Enhancing resource utilisation
  - Lowering wait time for multiple jobs (better predictibility)
  - Facilitate high-throughput simulations
  - Basis for Application-level Scheduling Resource binding
- Falkon, Condor Glide-in
  - Do not support MPI
  - All of the above are coupled/bound to specific back-ends
- Ganga-Diane (EGEE/EGI), DIRAC/WMS, PANDA
  - Frameworks based upon Pilot-Jobs (pull model) for specific PGI

Deployment & Scheduling of Multiple Infrastructure Independent Pilot-Jobs



Abstractions for Dynamic Execution (3)
SAGA Pilot-Job Framework

- SAGA Pilot-Job Framework has three main components:
  - BigJob Manager:
    - Provides the Pilot-Job abstraction to end-user
    - BigJob launches a job, which is a BigJob agent
    - Sub-jobs are submitted to BigJob Manager;
    - Ensures sub-job gets right resource based (Job Id)
  - BigJob Agent:
    - Represents the Pilot-Job; Application-level Resource Manager
  - Communication Board (Advert Service):
    - Communication and Coordination between BigJob Manager and Agent



- BigJob and Sub-Job classes/interfaces (Higher-level Functionality)
  - Extension of SAGA job API
    - Uses SAGA job state-model; regular SAGA job description
  - Uses Advert service and files API

Homework #1

- SAGA-based Pilot-Job (BigJob):
  - http://faust.cct.lsu.edu/trac/bigjob
- Installation:
  - http://faust.cct.lsu.edu/trac/bigjob/wiki/install
- Tutorials:
  - http://faust.cct.lsu.edu/trac/bigjob/wiki/Tutorials
- Think of suitable sub-job?
- Execute BigJob (BJ):
  1. Multiple sub-jobs for 1 BJ
  2. Multiple sub-jobs for 2 BJ on same machine
  3. *Multiple sub-jobs for 2 BJ on different machines*

Module E: Project Suggestions

- Gain sufficient proficiency with SAGA to write a M-W application that uses > 1 FutureGrid resource?
- Extend Mandelbrot Set? Use > 1 FutureGrid Resource?
- PySAGA M-W? Implement another pattern?
- See FG Tutorial on Nimbus and Eucalyptus
  - Can use SAGA to submit jobs to Clouds
  - M-W to submit to FG-Bare Metal & Clouds?
- Written Project. Ideas?
- *Teamwork is acceptable provided: (i) effort is acknowledged, (ii) clear intellectual contribution from each*

SAGA On FutureGrid/XSEDE

1. "Something" will appear on:
   - http://www.saga-project.org/documentation/installation/cyberinfrastructure/xsede-fg