A Gentle Introduction to Grid Computing

Borja Sotomayor

CS/TTI Grad Student Cake Talk Series

February 15, 2006

A Gentle Introduction to Grid Computing

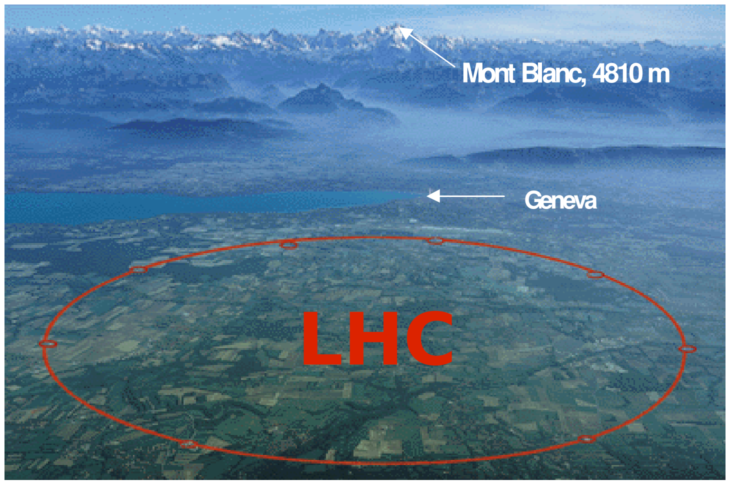
- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

A Gentle Introduction to Grid Computing

- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

A problem... (I)

A problem... (II)

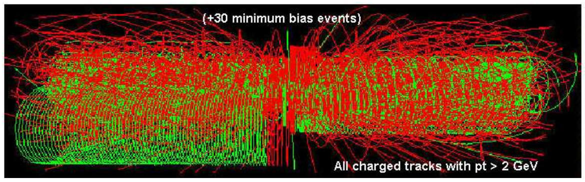


A problem... (III)

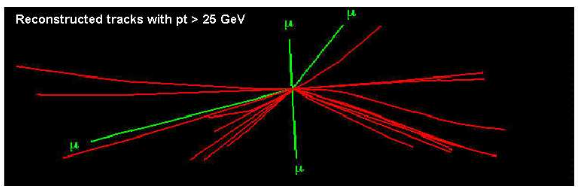
- El LHC (Large Hadron Collider), which is being built in CERN, is a particle accelerator/collider with a circumference of 27km (16.7mi).
- Will answer many interesting questions, specially: Does the Higgs boson exist?
- When it starts to work in 2007, it will produce *huge* amounts of information.

A problem... (IV)

From this event (1 event = 1 colission)...



We're searching for this characteristic signature:



1 in 10^{13}

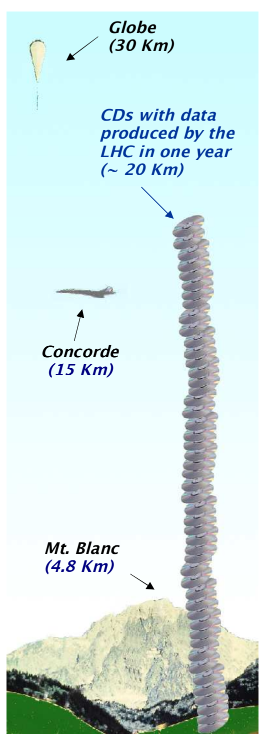
Like looking for one person in a thousand world populations.

A problem... (V)

- 40 million collisions per second
- After an initial filter, only 100 interesting collisions per second remain which must be stored and carefully analyzed.
- Each collision = 1MB
  - 100 MB/s. This information requires a (non-trivial) processing, and must be stored for future reference and study.
  - Largest single hard drive (as of 2006) can store 500GB: Almost 1h30m of LHC collisions.

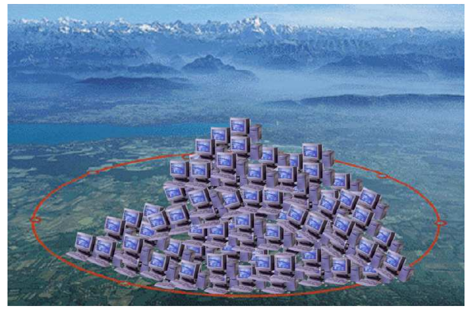
A problem... (VI)

- LHC will produce 10^{10} colisions each year.
  - 10 Petabytes of information per year!
- Just so we're clear:
  - 1 MB = A digital photograph.
  - 1 GB = 1024 MB = A CD-ROM and a half.
  - 1 TB = 1024 GB = Annual production of books all around the world.
  - 1 PB = 1024 TB = The information produced by an LHC experiment.
  - 1 EB = 1024 PB = Annual production of information all around the world.



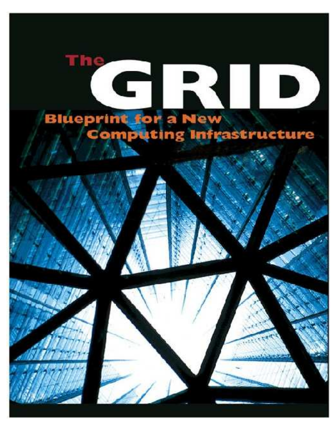
A problem... (VII)

- Using current technology, processing and storing all that data in a single site is impossible.
  - I kid you not, this seriously *cannot* be done.
- An estimated 100,000 high-tech processors would be needed to deal with the LHC's computational needs.
  - CERN 'only' has over 1,000 dual processor computers and 1 Petabyte of storage.

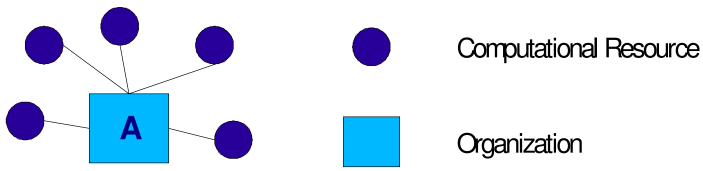


The Solution (I)

- Problem: A single node can't handle all that work.
  - But the combined power of *several* sites might be able to handle it.
- Solution: Achieving greater performance and throughput by pooling together resources from different organizations
  - In essence, this is what **Grid Computing** is all about.
  - A new distributed computing paradigm proposed by Ian Foster and Carl Kesselman in the mid-90s.

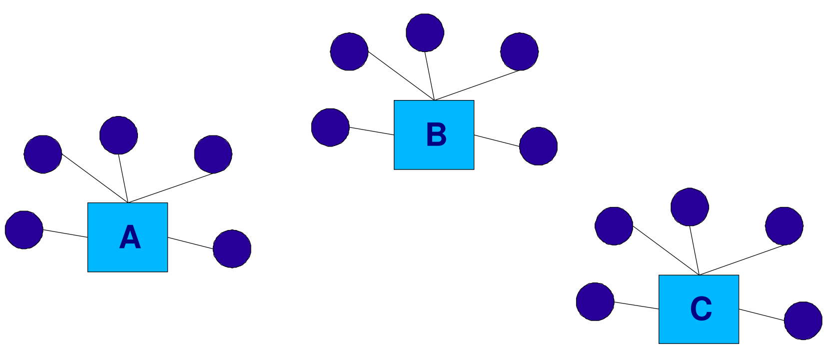


The Solution (II)



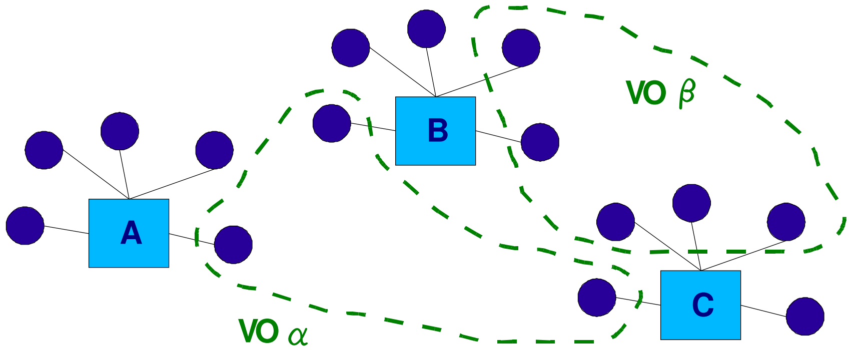
Without Grid computing, an organization is stuck with using only the resources it has direct control over

The Solution (III)



Using Grid Computing, resources from several different organizations are involved.

The Solution (IV)



These resources are dynamically pooled into *virtual organizations* (or VO) to solve specific problems.

- Parallelism (high throughput) and/or load balancing (high performance)

The Solution (V)

- Doing this is not trivial!
  - How do we decide what resources are part of each virtual organization?
  - Given a computational task, how do we decide what resources will be allocated to deal with that task? For how long?
  - How do we get the resources to communicate amongst themselves? Take into account that these are *heterogeneous* resources from *different* organizations!
  - If I want to "split up" a task so that it can be performed in parallel by several computers in different organization, how to I actually "split up" the program?
  - A lot of security challenges. For example, how can an organization make sure its resources are only being used by trusted users and that they are not being abused by malicious users?

The Solution (VI)

- Grid Computing aims to provide an answer to these questions (and many more!) by providing a set of protocols, technologies, and methodologies.
- Unfortunately, definitions of Grid Computing are like resources on a Grid:
  - Numerous and heterogeneous

A textbook definition

- Ian Foster provides an (open) definition in the paper *What is the Grid? A Three Point Checklist.*
- A grid is a system that:
  - coordinates resources that are not subject to centralized control...
  - ...using standard, open, general-purpose protocols and interfaces...
  - ...to deliver nontrivial qualities of service

LHC

- Back to the LHC...
  - The EGEE (Enabling Grids for E-science in Europe) project will pool computational resources from research centers all around Europe to provide enough computational power and storage space for the LHC.
  - EGEE will also be used for other purposes.
  - http://public.eu-egee.org/

A Gentle Introduction to Grid Computing

- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

What is it used for?

- LHC is a very large scale example
  - However, Grid Computing is not limited to gargantuan projects like LHC, and certainly not science fiction.
- There are a *lot* of applications that leverage Grid technologies to great effect.
  - Most originate in research centers or academia.
  - Out of reach of the layman, but it affects him indirectly.
  - There is no "The Grid", but there are a lot of small Grid systems around the world.

Grid Projects

USA

•NASA Information Power Grid
•DOE Science Grid
•NSF National Virtual Observatory
•NSF GriPhyN
•DOE Particle Physics Data Grid
•NSF TeraGrid
•DOE ASCI Grid
•DOE Earth Systems Grid
•DARPA CoABS Grid
•NEESGrid
•DOH BIRN
•NSF iVDGL

Europe

•EGEE (CERN, ...)
•DataGrid (CERN, ...)
•EuroGrid (Unicore)
•DataTag (CERN,…)
•Astrophysical Virtual Observatory
•GRIP
•GRIA (Industrial Applications)
•GridLab (Cactus Toolkit)
•CrossGrid (Infrastructure components)
•EGSO (Solar Physics)
•UK e-Science Grid

Applications (I)

- Applications that benefit from Grid Computing?
  - Computation-intensive applications
  - Data-intensive applications (with large data storage or data processing needs)
  - Collaborative applications.
- This list is not exhaustive

Applications (II)

- Computationally intensive applications
  - Simulations, prediction, real-time monitoring, ...
  - Crossgrid project
    - Modeling and simulating flood-susceptible regions to predict future floods and to provide real-time (processed) data to crisis management teams during a flood.
    - http://www.eu-crossgrid.org/
  - TeraGrid:
    - A Grid system providing a powerful infrastructure for open scientific research. As of 2006, TeraGrid had 40 teraflops of computing power and 2 petabytes of distributed storage.
    - http://www.teragrid.org/

Applications (III)

- Data-intensive applications
  - Applications that generate a large and steady flow of data. e.g. LHC
  - Applications that benefit from shared access to similar data in different organization. e.g. Distributed mammography analysis: http://www.ediamond.ox.ac.uk/

Applications (IV)

- Collaborative applications
  - Applications that, by their very nature, involve several organization and can benefit from a technology that facilitates communication and sharing between organizations.
  - Teleconferences, virtual meetings: http://www.accessgrid.org/
  - NEESit: Links together seismological laboratories http://it.nees.org/

Applications in Industry

- Grid technologies are also leveraged in industry
  - Novartis
    - Uses a Grid of desktop PCs to add 5+ additional teraflops of computing power to its existing computational resources.
  - Financial computing
    - Several Wall Street companies use Grid technologies for computation-intensive tasks (such as options pricing). e.g. Charles Schwab
  - BBC
    - Content distribution

A Gentle Introduction to Grid Computing

- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

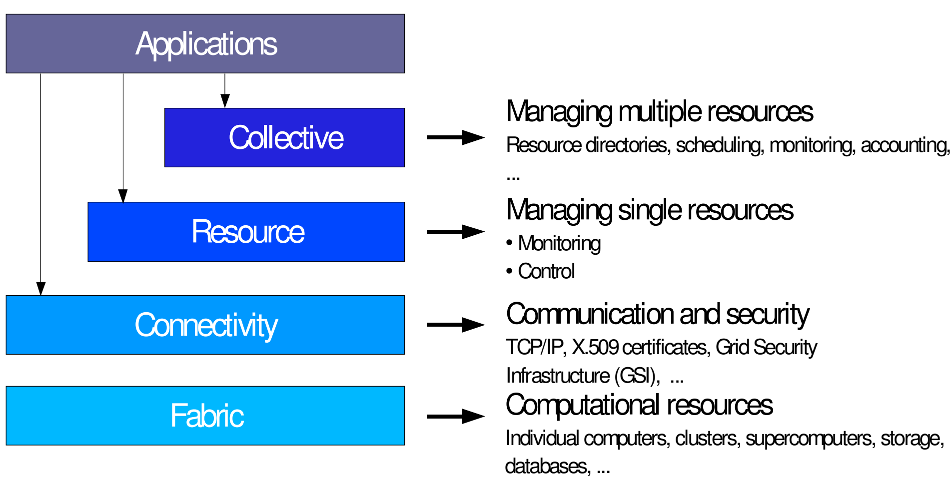
How does it work?

- Remember: Getting *heterogeneous* computational resources from *different* organizations to work together is not trivial.
- So how do Grid systems deal with this?
- We'll see:
  - The general Grid architecture
  - OGSA: Open Grid Services Architecture
  - WSRF: Web Services Resource Framework
  - Globus Toolkit 4

Grid Architecture (I)

- Creating a complete Grid system requires a wide variety of protocols, services, and software development kits. For example:
  - **VO Management Service**: To manage what nodes and users are part of each Virtual Organization.
  - **Resource Discovery and Management Service**: So applications on the grid can discover resources that suit their needs, and then manage them.
  - **Job Management Service**: So users can submit tasks (in the form of "jobs") to the Grid.
  - And a whole other bunch of services like security, data management, etc.
- We can categorize them into a general Grid architecture according to their function and purpose. *Anatomy of the Grid* (Foster, Kesselman, Tuecke)

Grid Architecture (II)



OGSA (I)

- Ok, we've cleared up what services are involved in a Grid system, but...
  - How does one service communicate with another service?
    - RPC? CORBA? RMI? Some ad-hoc protocol?
  - How is a job described?
    - How do I specify how many CPUs I need? And my memory requirements? etc.
  - How are files moved around in a Grid?
    - Using some sort of file transfer service? Plain old FTP?
- We could keep on asking questions *ad nauseam*.

OGSA (II)

- In the beginning was... the ad-hockery.
- Currently, there is a push towards standardization of the interfaces and behaviours of services one would expect to find on a Grid system:
  - Resource management
  - Job management
  - Security
  - Workflow management
  - Etc.

OGSA (III)

- The Open Grid Services Architecture (OGSA) is the grand unifying standard for Grid computing.
  - Aims to define a common, standard, and open architecture for grid-based applications
  - Although these standard interfaces are still in the works, OGSA already defines a set of requirements that must be met by these standard interfaces.
- It is being developed by the Global Grid Forum (http://www.ggf.org)

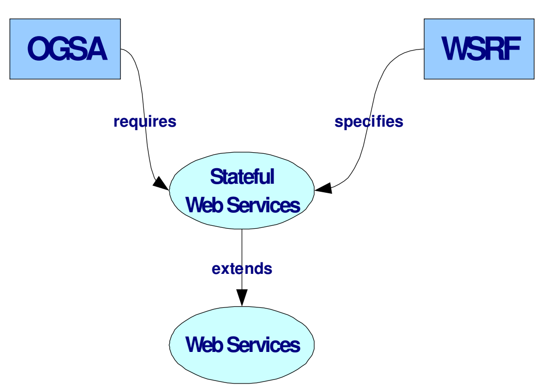
OGSA + WSRF (I)

- Some sort of distributed middleware is needed as a base for this architecture.
  - e.g. If OGSA defines that the *JobSubmissionInterface* has a *submitJob* operation, there has to be a common and standard way to invoke that operation if we want the architecture to be adopted as an industry-wide standard.
- This base for the architecture could, in theory, be any distributed middleware (CORBA, RMI, or even traditional RPC).

OGSA + WSRF (II)

- The powers-that-be chose *Web services*
  - Distributed middleware well suited for lowly coupled systems.
- However, Web services still don't meet one important OGSA requirement: OGSA requires *stateful* services.
  - Web services can be stateful, but there is no standard way of manipulating stateful Web services.
- Solution: WSRF (Web Services Resource Framework)
  - A collection of specifications under the auspices of OASIS

OGSA + WSRF (III)



Globus Toolkit 4 (I)

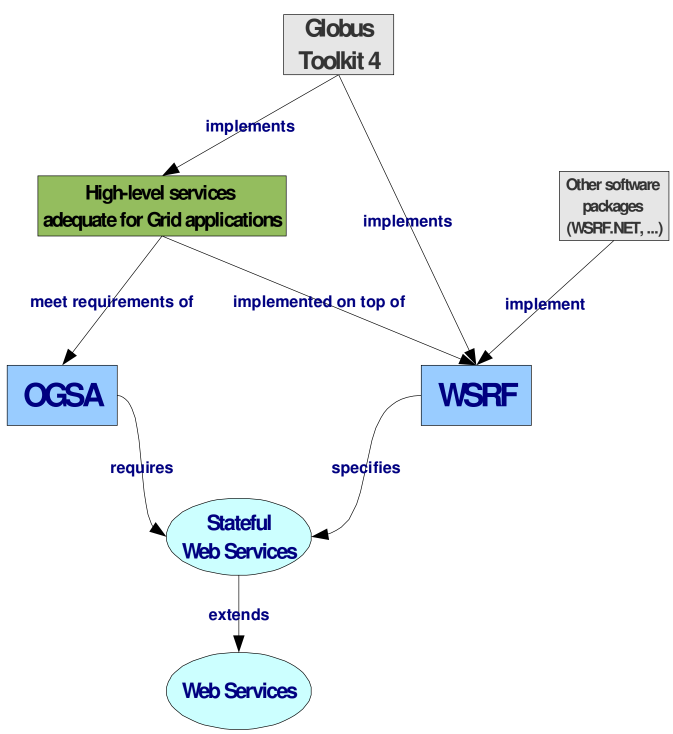
- The Globus Toolkit is a software toolkit, developed by The Globus Alliance (http://www.globus.org/), which we can use to create Grid systems.
- The toolkit, first and foremost, includes quite a few high-level services that we can use to build Grid applications.
  - These services, in fact, meet most of the abstract requirements set forth in OGSA.

Globus Toolkit 4 (II)

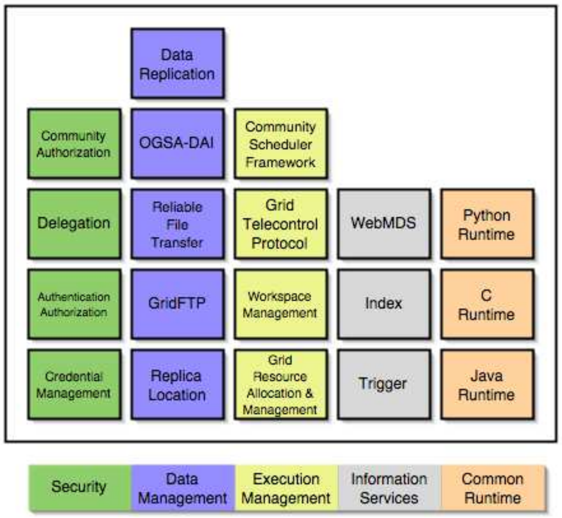
- However, not an *implementation* of OGSA.
  - Since the working groups at GGF are still working on defining standard interfaces for these types of services, we can't say (at this point) that GT4 is an implementation of OGSA (although GT4 does implement a few specifications defined by GGF).
  - However, it is a realization of the OGSA requirements and a sort of *de facto* standard for the Grid community while GGF works on standardizing all the different services.

Globus Toolkit 4 (III)

- Most of these services are implemented on top of WSRF.
  - The toolkit also includes some services that are not implemented on top of WSRF and are called the non-WS components.
- The Globus Toolkit 4, in fact, includes a complete implementation of the WSRF specifications.



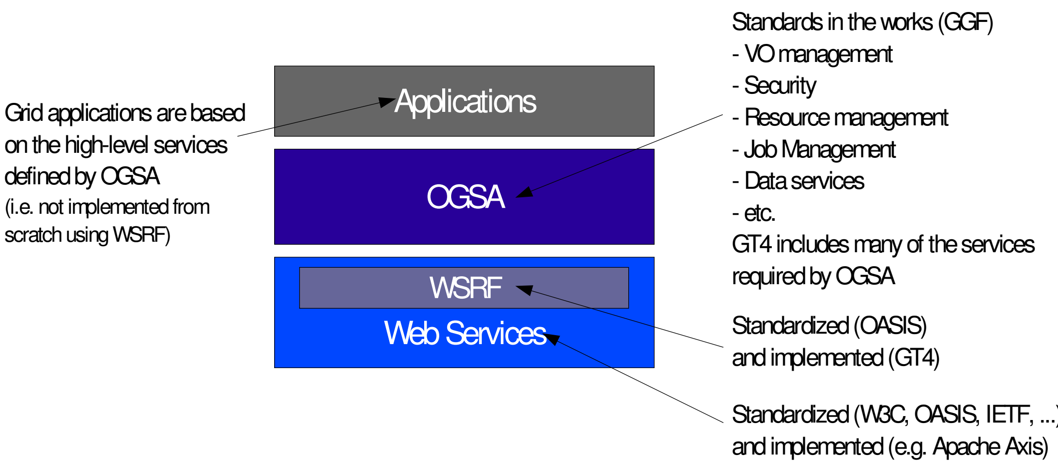
Globus Toolkit 4 (IV)



Globus Toolkit 4 (V)

- *Pitfall*
  - "If I install GT4, I can start sending off jobs to the Grid!"
  - No! GT4 is a *toolkit*: a collection of software components you can use as building blocks for a Grid application.
  - Those building blocks aren't going to piece themselves together on their own...
  - GT4 is for developers, *not* for users.
  - Even so, GT4 is still not a turnkey solution. We will generally need to integrate other software packages in our application to create a fully functional Grid application.

The mandatory layered diagram



A Gentle Introduction to Grid Computing

- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

My research (I)

- Grid Computing + Virtual Machines
  - Unholy union or match made in heaven?
  - There are many advantages to leveraging virtualization technologies in Grid systems.
    - *A Case for Grid Computing on Virtual Machines*. Figueiredo, R., P. Dinda, and J. Fortes. In 23rd International Conference on Distributed Computing Systems. 2003.

My research (II)

- One step towards the union of Grids and VMs is developing interfaces that allow for the dynamic deployment of virtual machines on Grid resources.
  - Or, more generally: the deployment of virtual execution environments.
- GT4 Workspace Service
  - http://workspace.globus.org/
  - Provides an abstraction for an execution environment. This abstraction is implemented with VMs.

My research (III)

- Fine-grained resource allocation for aggregate virtual workspaces
  - Aggregate workspace: Virtual workspace with several virtual nodes. e.g. One or several virtual clusters running on a single physical cluster.
  - In a nutshell: This makes it easier to run several applications (from different VOs) without having to deal with configuration conflicts + enforcing a resource allocation for each VO.
  - Master's thesis on fine-grained resource allocation for virtual clusters.

A Gentle Introduction to Grid Computing

- What is Grid Computing?
- What is it used for?
  - INTERMISSION
- How does it work?
- My research
- I want to know more!

I want to know more! (I)

- GridCafé: Very good introduction to Grid Computing.
  - http://gridcafe.web.cern.ch/
- Books
  - **Grid Computing**: "The Grid 2". Edited by Ian Foster and Carl Kesselman. Morgan Kaufmann, 2003.
  - **Grid Computing for Managers**: "Grid Computing: The Savvy Manager's Guide". Pawel Plaszczak, Richard Wellner, Jr. Morgan Kaufmann, 2005.
  - **Globus Toolkit 4**: "Globus Toolkit 4: Programming Java Services". Borja Sotomayor, Lisa Childers. Morgan Kaufmann, 2005.
  - Several other books on Grid Computing in general.

I want to know more! (II)

- Online:
  - Official Globus Documentation: http://www.globus.org/toolkit/docs/4.0/
  - The Globus Toolkit 4 Programmer's Tutorial http://gdp.globus.org/gt4-tutorial/
  - Lee Liming's (excellent) Globus primer: http://www-unix.mcs.anl.gov/~liming/primer/

I want to know more! (III)

- Grid Computing in the CS department
  - Faculty: Ian Foster <foster@cs.uchicago.edu>
  - Distributed Systems Laboratory
    - http://dsl.cs.uchicago.edu/
    - List of current projects and all our publications
    - DSL Seminar: http://dsl.cs.uchicago.edu/seminar/
      - Weekly meeting to discuss interesting papers on distributed systems and Grid Computing.

Questions? Comments? Grid skepticism?

Borja Sotomayor

Department of Computer Science

University of Chicago

borja@cs.uchicago.edu

Acknowledgements

- Some diagrams have been taken directly, with permission from the author, from the "Grid for Beginners" presentation available in the Grid Café:

http://gridcafe.web.cern.ch/gridcafe/demos/Grid-beginners.ppt