OurGrid: a *free-to-join* grid for *bag-of-tasks* applications

Marco Aurélio Spohn

maspohn@dsc.ufcg.edu.br

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Laboratório de Sistemas Distribuídos
Campina Grande, PB - BRASIL

http://www.lsd.ufcg.edu.br/

Outline

- Motivation
- Design principles
- OurGrid architecture
  - Resource discovery and sharing
  - Programming interface
  - Application scheduling and user interface
  - Security
- Applications
- Current status and on-going research

e-Science

- Computers are changing scientific research
  - Enabling collaboration
  - As investigation tools (simulations, data mining, etc...)
- As a result, many research labs around the world are now computation hungry
- Buying more computers is just part of answer
- Better using existing resources is the other

Harvesting Computing Power

- Idle workstations in a LAN
  - CONDOR
- Idle desktops in the Internet
  - SETI@home
- Virtual organizations
  - Globus-based grids
- Plug-in-the-wall
  - “The” computational grid

Solution 1: Globus

- Globus-based solutions are the closest realization of “plug on the wall and solve your problem” vision
  - Deployed for dozens of sites in several initiatives
- But it requires highly-specialized skills and complex off-line negotiation
- Good solution for large labs that work in collaboration with other large labs
  - CERN’s LCG, GRID’5000, TeraGrid and some others

CERN's LCG

"Geneva, 25 September 2006. A milestone for scientific Grid computing was announced today at the launch of EGEE'06 hosted by CERN. The Enabling Grids for E-sciencE (EGEE) project maintains a global Grid infrastructure that has been able to sustain more than ***30000 jobs a day – over a million per month*** *– for a period of six months this year. These computing tasks were submitted by scientists from diverse fields of research, and range from simulations of molecular drug docking for neglected diseases to geophysical analysis of oil and gas fields. Clusters of hundreds and even thousands of PCs, in institutes and universities around the world, have been executing these calculations –* ***in total over 25000 central processor units*** *(CPUs) are involved. Several million gigabytes of data storage in disk and tape facilities also contribute to make EGEE the world's largest scientific Grid infrastructure."*

Solution 2: Voluntary Computing

- SETI@home, FightAIDS@home, Folding@home, YouNameIt@home
  - SETI@home reported the harvesting of more than 2.2 million years of CPU time, from over 5.3 million users, spread over 226 countries
- However, to use this solution, you must
  - have a very good support team to run “the server”
  - invest a good deal of effort in “advertising”
  - have a very high visibility project
  - be in a prestigious institution

And what about the many thousands of small and middle research labs throughout the world which also need lots of computing power?

Most labs in the world …

- Are small
- Focus their research on some narrow topic
- Do not belong to top Universities
- Cannot count on cutting-edge computer support team

Yet …

they increasingly demand large amounts of computing power, just as large labs and high-visibility projects do

Solution 3: *peer-to-peer grid*

- Each lab correspond to a **peer** in the system and contributes with its idle resources

Before going further …
an important note

- These are not competing technologies!
- Each is more appropriate to a particular class of the user base
- Each has virtues and drawbacks
- It is very likely that they will be able not only to co-exist, but also to interoperate

OurGrid Design Principles

- Labs can freely join the system without any human intervention
  - No need for negotiation; no paperwork
- Clear incentive to join the system
  - One can't be worse off by joining the system
  - Noticeable increased response time
  - Free-riding resistant
- Basic dependability properties
  - Configurable level of security
  - Resilience to faults
  - Scalability
- Easy to install, configure and program
  - No need for specialized support team

Bag-of-Tasks (BoT)

- To simplify the problem, we focus on Bag-of-Tasks (BoT) applications
  - No need for communication among tasks
    - Facilitates scheduling and security enforcement
  - Simple fail-over/retry mechanisms to tolerate faults
  - Best effort: No need for QoS guarantees
  - Script-based programming is natural
    - Facilitates use

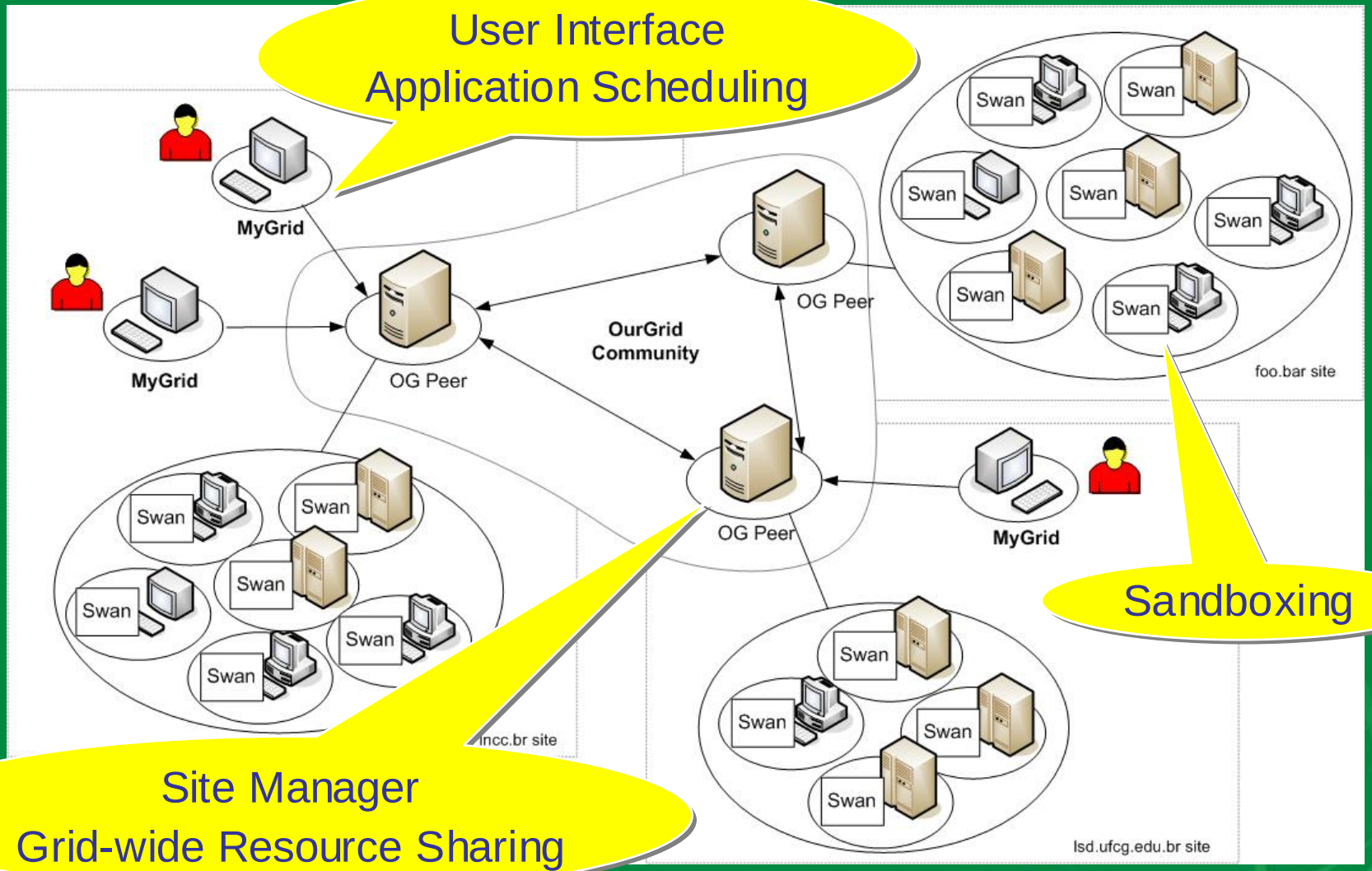
Fortunately, many important applications are BoT!

- Data mining
- Massive search
- Bio computing
- Parameter sweep
- Monte Carlo simulations
- Fractal calculations
- Image processing
- Computer Network simulations (e.g., MANETs)
- And many others…

OurGrid Architecture

- Components
  - OurGrid peer: A servant that takes part of a peer-to-peer network that performs fair resource sharing among unknown peers
  - MyGrid broker: A broker that provides a user interface and efficiently schedules BoT applications without requiring information neither on system load nor on applications and resources characteristics
  - Worker: An (*ideally*) secure environment for a peer to run a foreign computation

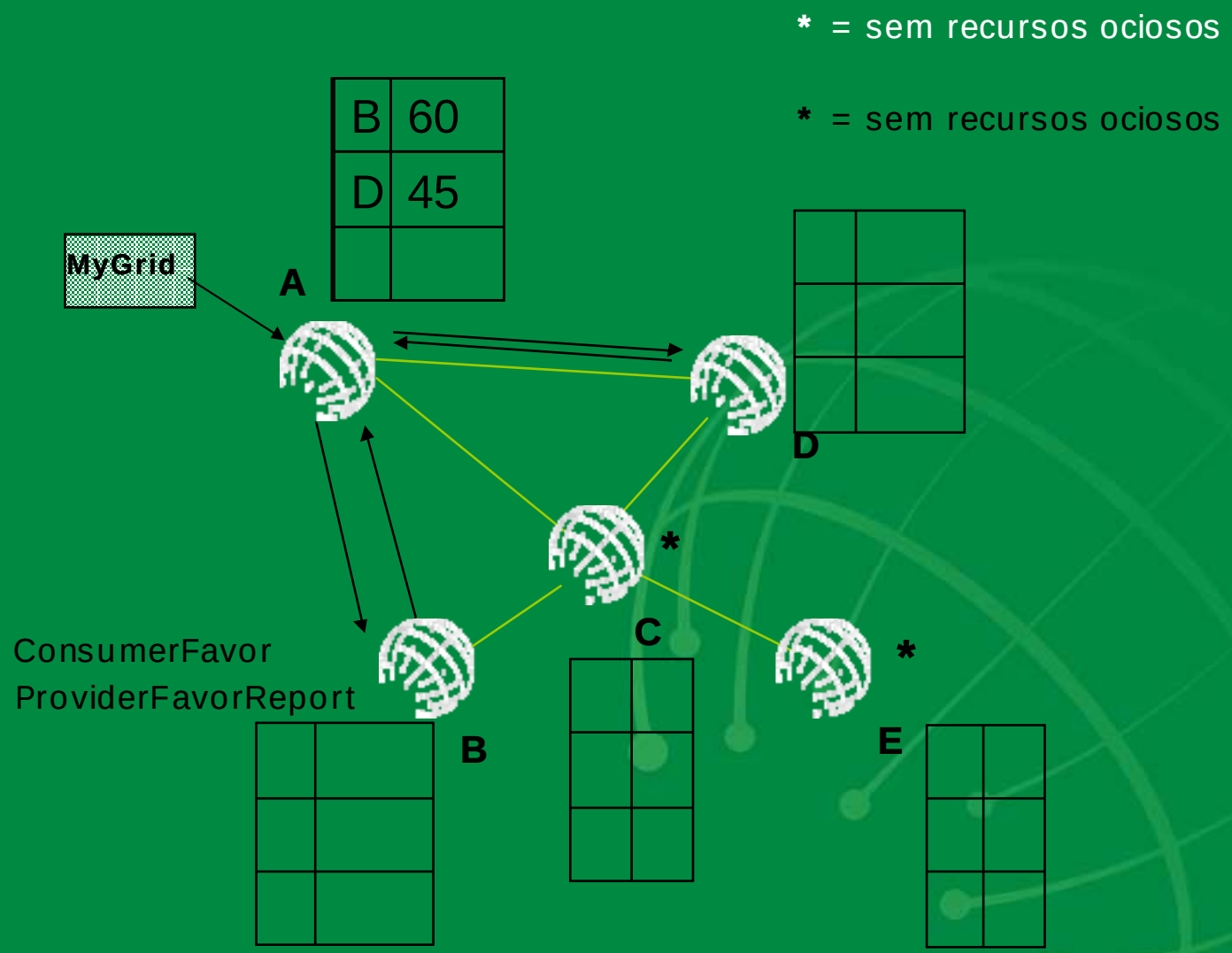
OurGrid Architecture



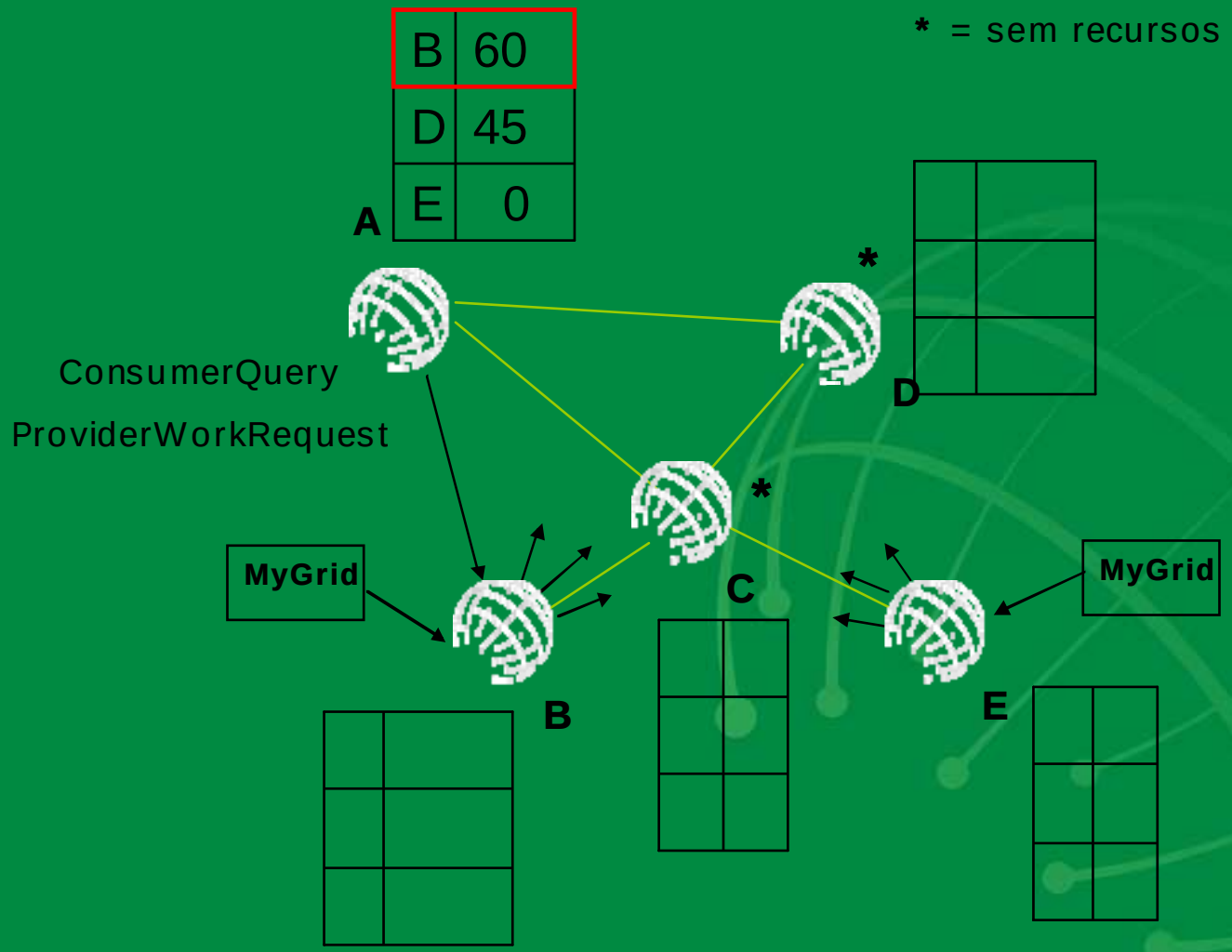
Network of Favors

- It is important to encourage collaboration within OurGrid (i.e. resource sharing)
  - In file-sharing, most users free-ride
- OurGrid uses the Network of Favors
  - All peers maintain a local balance for all known peers
  - Peers with greater balances have priority
  - Newcomers and peers with negative balance are treated equally
  - The emergent behavior of the system is that by donating more, one gets more resources back
  - No additional infrastructure is needed

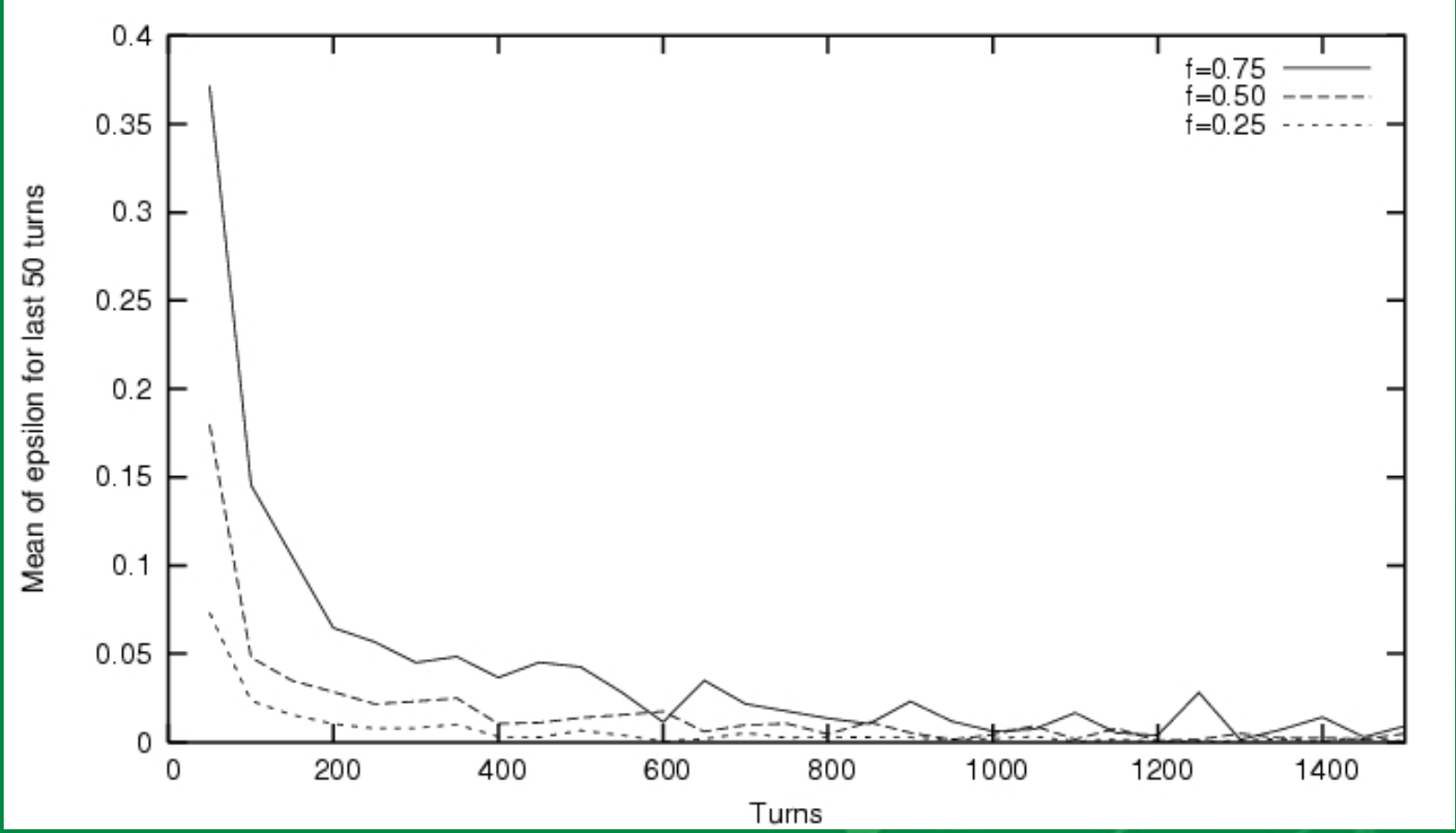
Network of Favors



Network of Favors

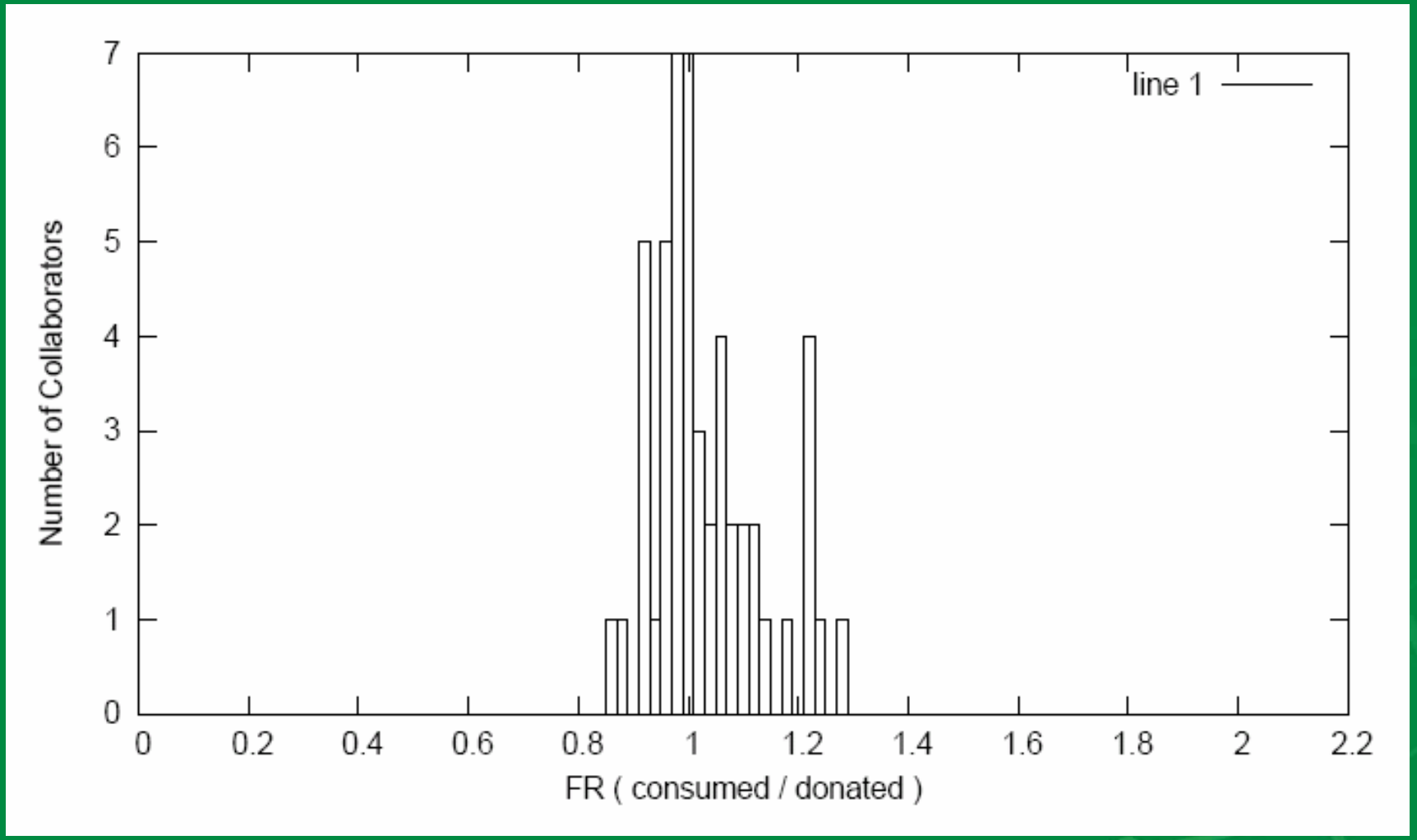


Free-rider Consumption



- Epsilon is the fraction of resources consumed by free-riders: epsilon becomes very small as the system reaches steady state

Equity Among Collaborators



100 peer community (after 3000 turns)

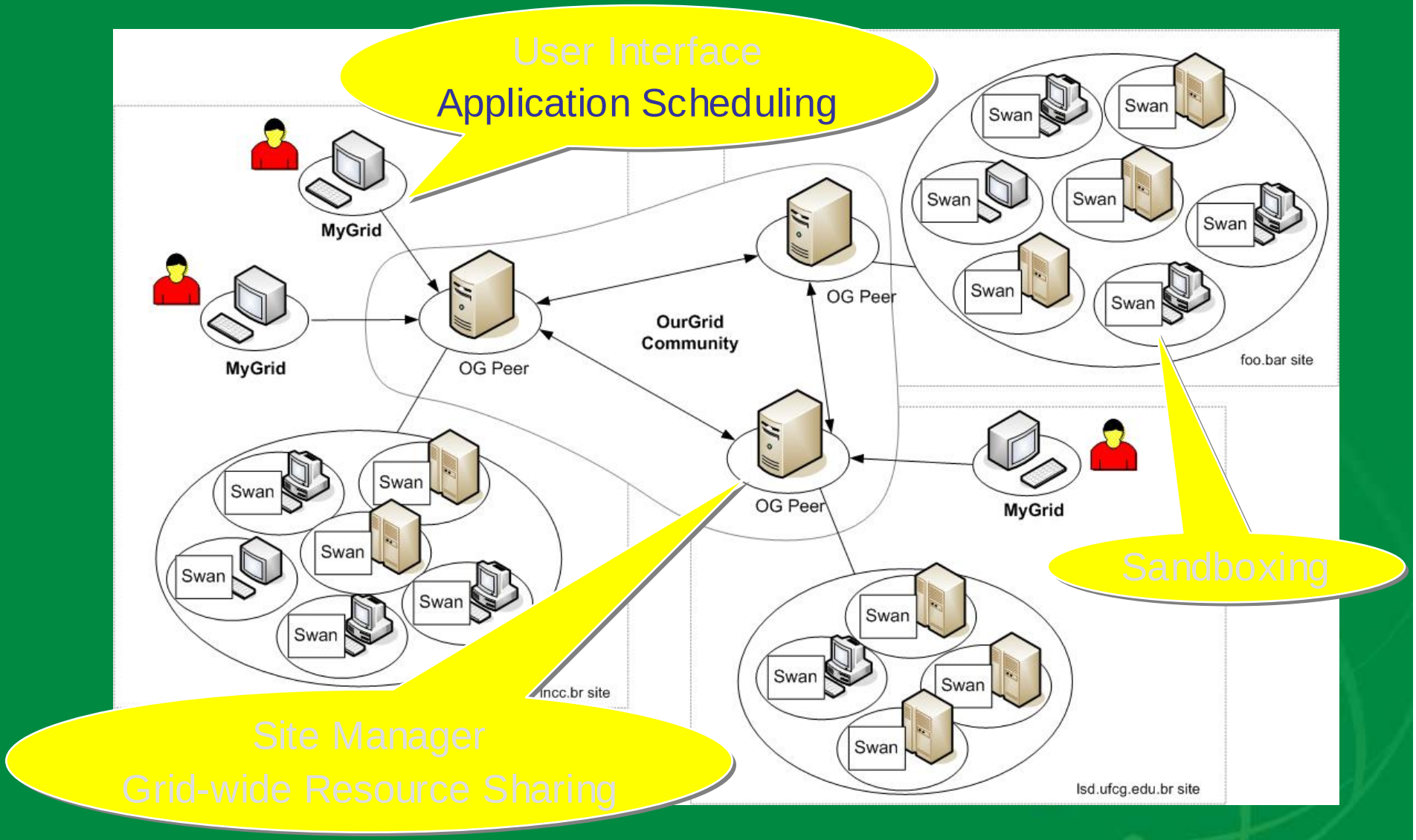
Autonomous Accounting

Slower: 2 tasks/machine; Faster: 1 task/machine

Some Pointers to the NoF

- OurGrid: An Approach to Easily Assemble Grids with Equitable Resource Sharing. Nazareno Andrade, Walfredo Cirne, Francisco Brasileiro, Paulo Roisenberg. *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing*. June, 2003.
- Discouraging Free Riding in a Peer-to-Peer CPU-Sharing Grid. Nazareno Andrade, Francisco Brasileiro, Walfredo Cirne, Miranda Mowbray. *Proceedings of the Thirteenth IEEE International Symposium on High-Performance Distributed Computing*. Honolulu, Hawaii. June, 2004.
- When Can an Autonomous Reputation Scheme Discourage Free-riding in a Peer-to-Peer System? Nazareno Andrade, Miranda Mowbray, Walfredo Cirne, Francisco Brasileiro. *Proceedings of the Fourth International Workshop on Global and Peer-to-Peer Computing*. April, 2004.
- Relative Autonomous Accounting for Peer-to-Peer Grids. Robson Santos, Alisson Andrade, Walfredo Cirne, Francisco Brasileiro, Nazareno Andrade. *Concurrency and Computation - Practice & Experience*. Accepted for publication.
- A Reciprocation-Based Economy for Multiple Services in Peer-to-Peer Grids. Miranda Mowbray, Francisco Brasileiro, Nazareno Andrade, Jaindson Santana, Walfredo Cirne. *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06)*. Cambridge, UK. August, 2006.

OurGrid Architecture



Grid-enabling an Application

- Write a script using a very simple language
  - Simple abstractions
    - File transfer (put, store, get)
    - Hide heterogeneity ($PLAYPEN, $STORAGE)
    - Define constraints (job requirements and grid machine attributes)
- Write a program that embeds the business logic and may make use of more complex features available through a Java API
- Deploy a Portal that embeds the application

An Example: Factoring with MyGrid

```
job:
    label: my_factorial_useless_example
    requirements: (os=linux)

task:
    init:  store ./Fat.class  $PLAYPEN
    grid:  java  Fat  3  18655  34789789799  output-$JOB-$TASK
    final:  get  $PLAYPEN/output-$JOB-$TASK  results
task:
    init:  store ./Fat.class  $PLAYPEN
    grid:  java  Fat  18656  37307  34789789799  output-$JOB-$TASK
    final:  get  $PLAYPEN/output-$JOB-$TASK  results
task:
    init:  store ./Fat.class  $PLAYPEN
    grid:  java  Fat  37308  55968  34789789799  output-$JOB-$TASK
    final:  get  $PLAYPEN/output-$JOB-$TASK  results
....
```

More interesting applications will be discussed later!!

Scheduling with no Information

- Grid scheduling typically depends on information about the grid (e.g. machine speed and load) and the application (e.g. task size)
- However, getting accurate information about all applications and resources is hard
- Can we efficiently schedule tasks without requiring access to information?
  - This would make the system much easier to deploy and simpler to use

Workqueue with Replication

- Tasks are sent to idle processors
- When there are no more tasks, running tasks are replicated on idle processors
- The first replica to finish is the official execution
- Other replicas are cancelled

Evaluation

- 8000 experiments
- Experiments varied in
  - grid heterogeneity
  - application heterogeneity (i.e., size of tasks)
  - application granularity (i.e., # tasks/machine)
- Performance summary:

	Sufferage	DFPLTF	Workqueue	WQR 2x	WQR 3x	WQR 4x
Average	13530.26	12901.78	23066.99	12835.70	12123.66	11652.80
Std. Dev.	9556.55	9714.08	32655.85	10739.50	9434.70	8603.06

WQR Overhead

- Obviously, the drawback in WQR is cycles wasted by the cancelled replicas
- Wasted cycles:

	WQR 2x	WQR 3x	WQR 4x
Average	23.55%	36.32%	48.87%
Std. Dev.	22.29%	34.79%	48.93%

  - Trading Cycles for Information: Using Replication to Schedule Bag-of-Tasks Applications on Computational Grids. Daniel Paranhos, Walfredo Cirne, Francisco Brasileiro. *Proceedings of the Euro-Par 2003: International Conference on Parallel and Distributed Computing*. August, 2003.

Data Aware Scheduling

- WQR achieves good performance for CPU-intensive BoT applications
- However, many important BoT applications are data-intensive
- These applications frequently reuse data
  - During the same execution
  - Between two successive executions

Storage Affinity

- Storage Affinity uses replication and just a bit of static information to achieve good scheduling for data intensive applications
- Storage Affinity uses information on which data servers have already stored a data item (*how close; i.e., how many bytes of the task input dataset are already stored at a specific site*).

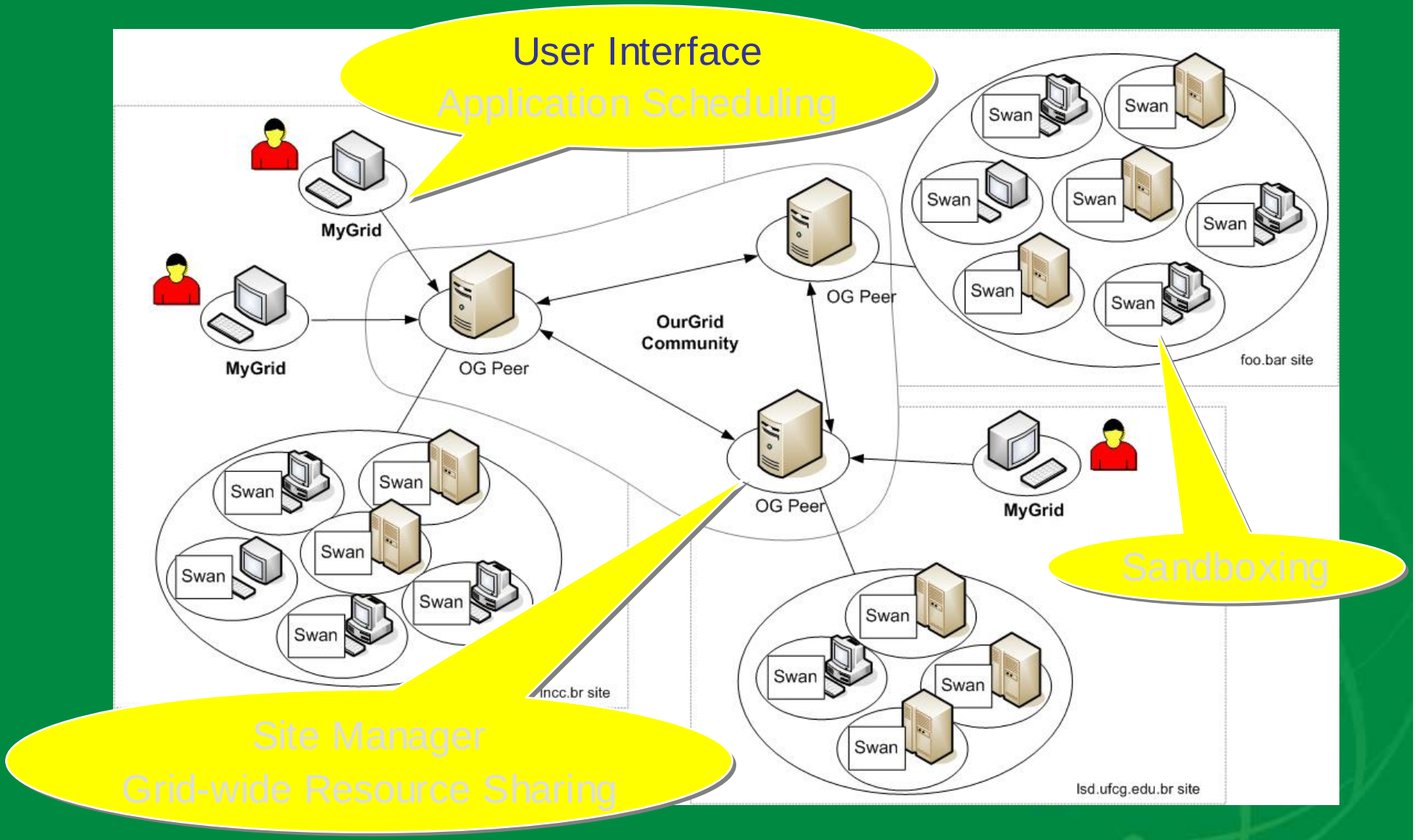
Storage Affinity Results

- Application was assumed to have 2GB input, and reuse was Inter-job
- Data transfer delays dominate the execution time of the application

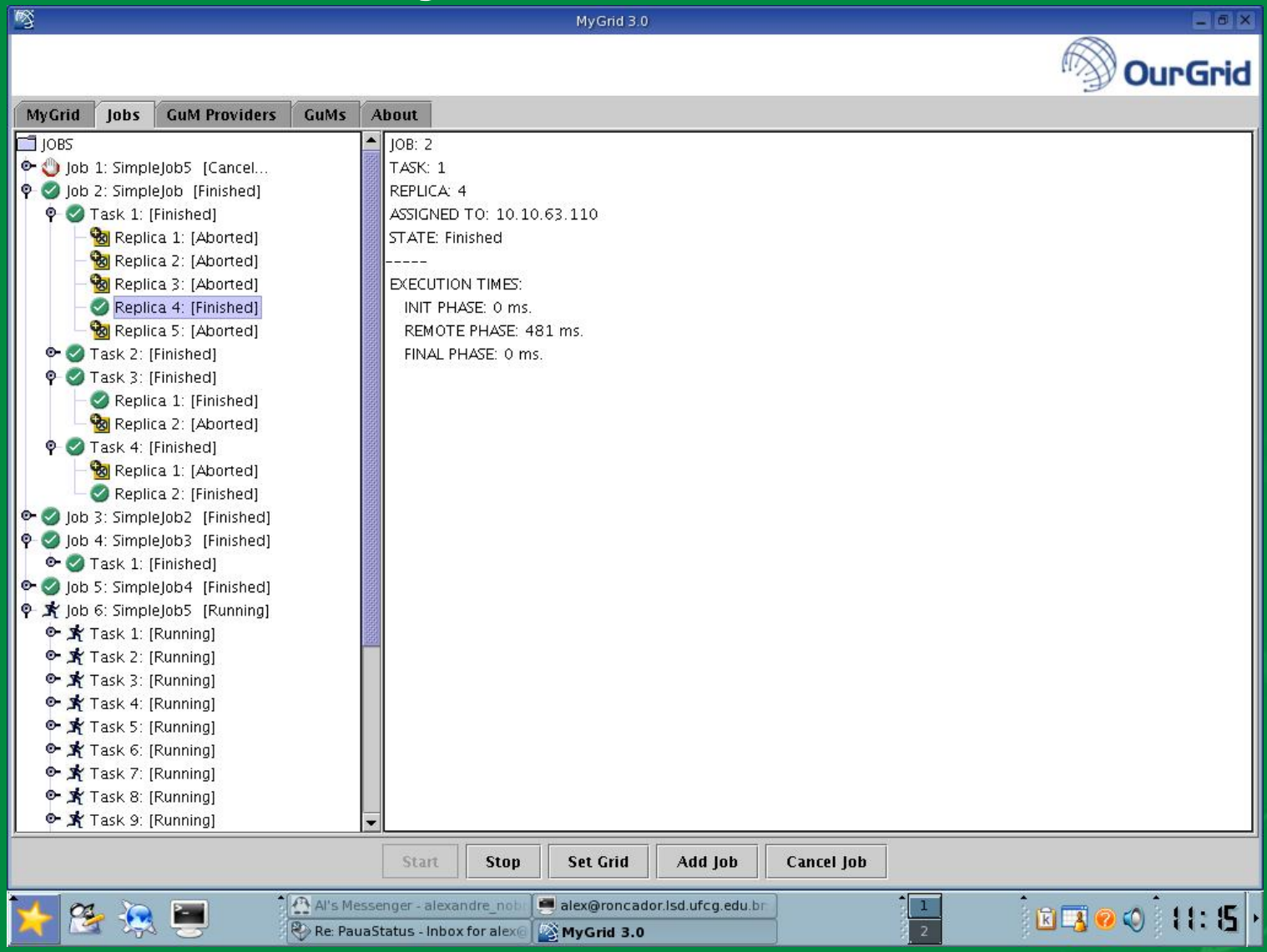
		Storage Affinity	WQR	XSufferage
Execution Time (sec)	Mean	14377	42919	14665
	Std Dev	10653	24542	11451

Exploiting Replication and Data Reuse to Efficiently Schedule Data-intensive Applications on Grids. Elizeu Santos-Neto, Walfredo Cirne, Francisco Brasileiro, Aliandro Lima. *Proceedings of the 10th Workshop on Job Scheduling Strategies for Parallel Processing*. June, 2004.

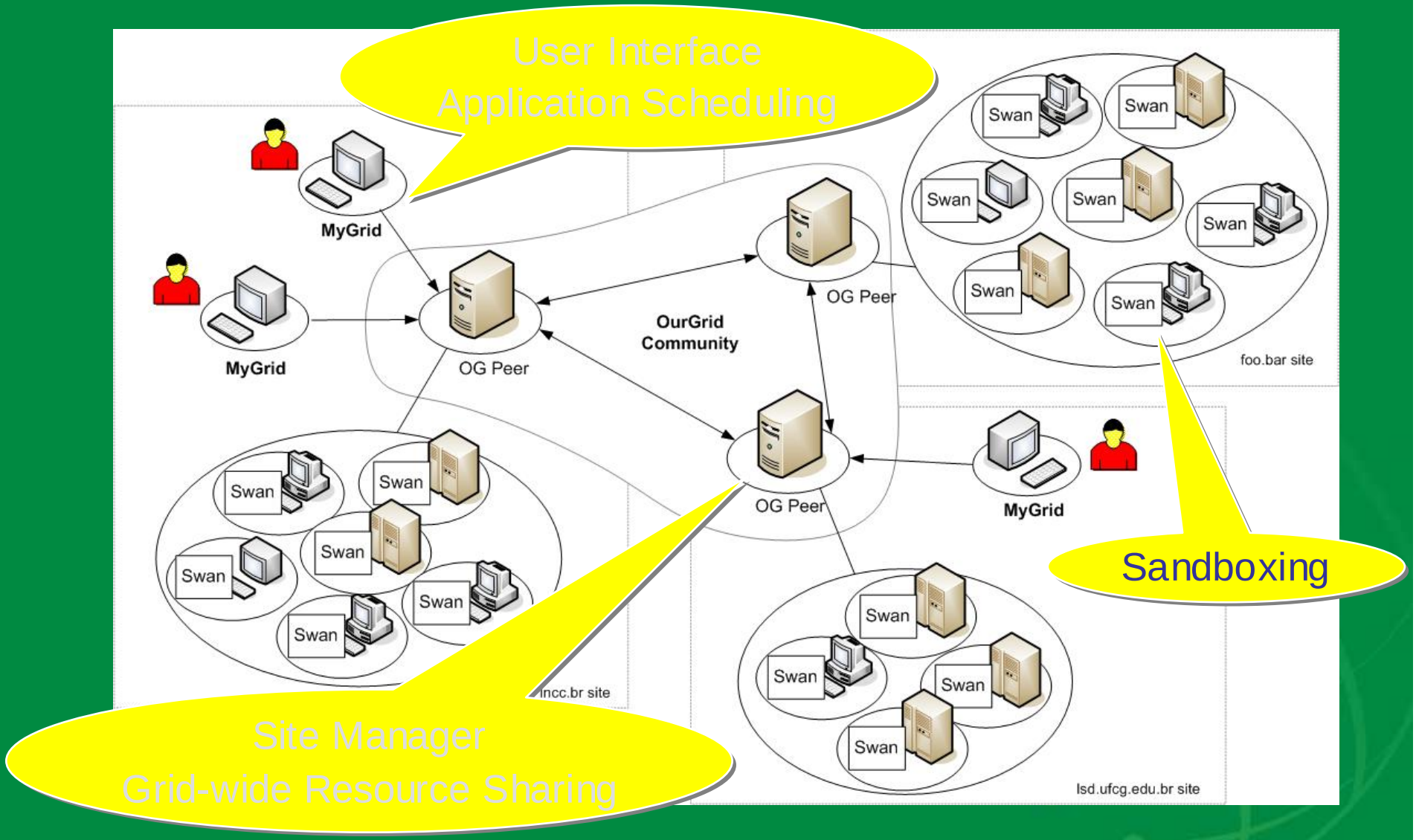
OurGrid Architecture



MyGrid GUI


MyGrid 3.0
OurGrid
MyGrid
Jobs
GuM Providers
GuMs
About
JOBS
Job 1: SimpleJob5 [Cancel...
Job 2: SimpleJob [Finished]
Task 1: [Finished]
Replica 1: [Aborted]
Replica 2: [Aborted]
Replica 3: [Aborted]
Replica 4: [Finished]
Replica 5: [Aborted]
Task 2: [Finished]
Task 3: [Finished]
Replica 1: [Finished]
Replica 2: [Aborted]
Task 4: [Finished]
Replica 1: [Aborted]
Replica 2: [Finished]
Job 3: SimpleJob2 [Finished]
Job 4: SimpleJob3 [Finished]
Task 1: [Finished]
Job 5: SimpleJob4 [Finished]
Job 6: SimpleJob5 [Running]
Task 1: [Running]
Task 2: [Running]
Task 3: [Running]
Task 4: [Running]
Task 5: [Running]
Task 6: [Running]
Task 7: [Running]
Task 8: [Running]
Task 9: [Running]
JOB: 2
TASK: 1
REPLICA: 4
ASSIGNED TO: 10.10.63.110
STATE: Finished
EXECUTION TIMES:
INIT PHASE: 0 ms.
REMOTE PHASE: 481 ms.
FINAL PHASE: 0 ms.
Start
Stop
Set Grid
Add Job
Cancel Job
MyGrid 3.0

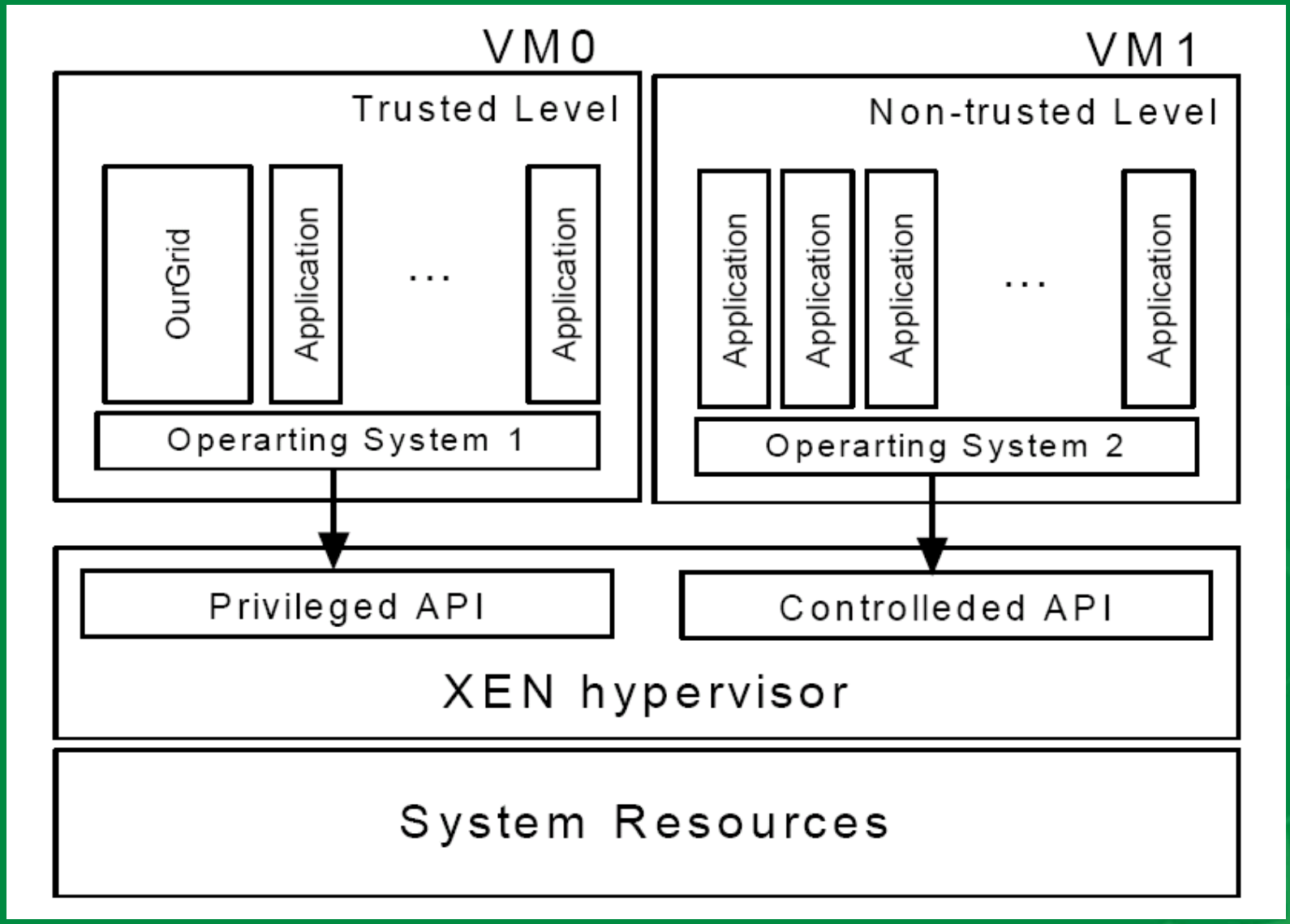
OurGrid Architecture



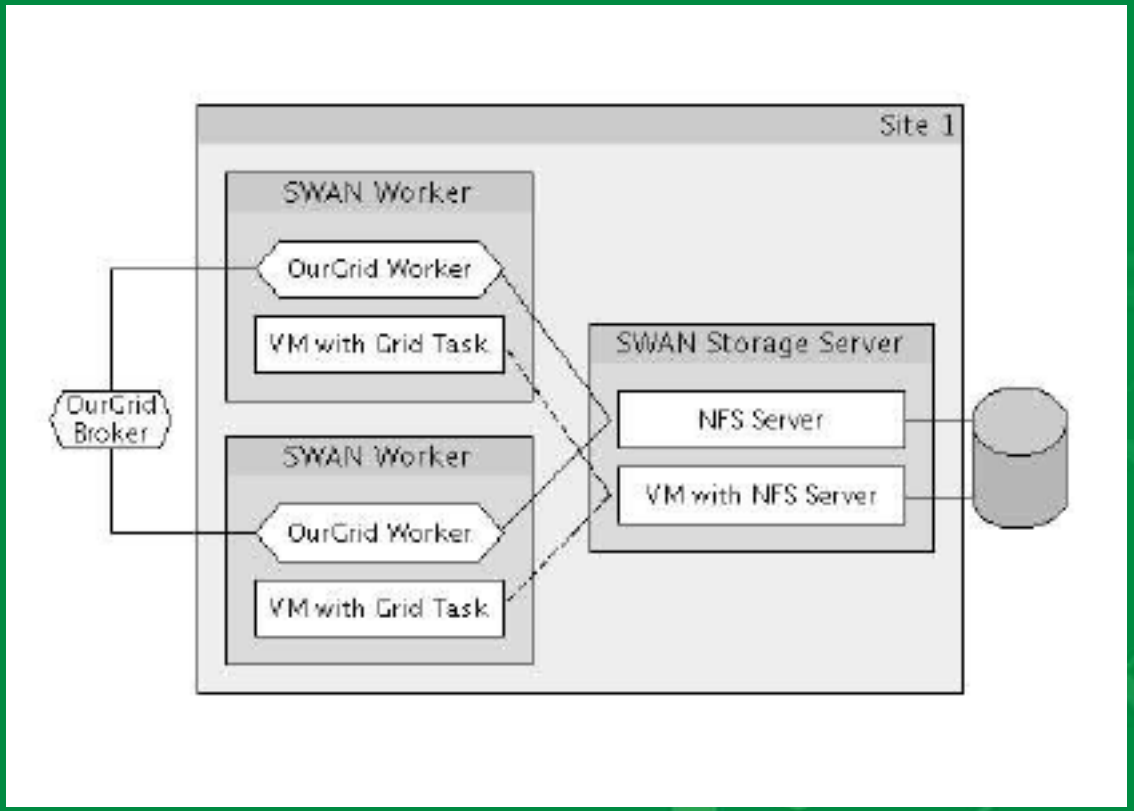
SWAN: OurGrid Security

- Running an unknown application that comes from an unknown peer is a clear security threat
- We leverage the fact that BoT applications only communicate to receive input and return the output
- The remote task runs inside a virtual machine, with no network access, and disk access only to a designated partition
- Input/output is done by OurGrid itself that runs in a privileged virtual machine
- Sanity checks are executed before a new task is run

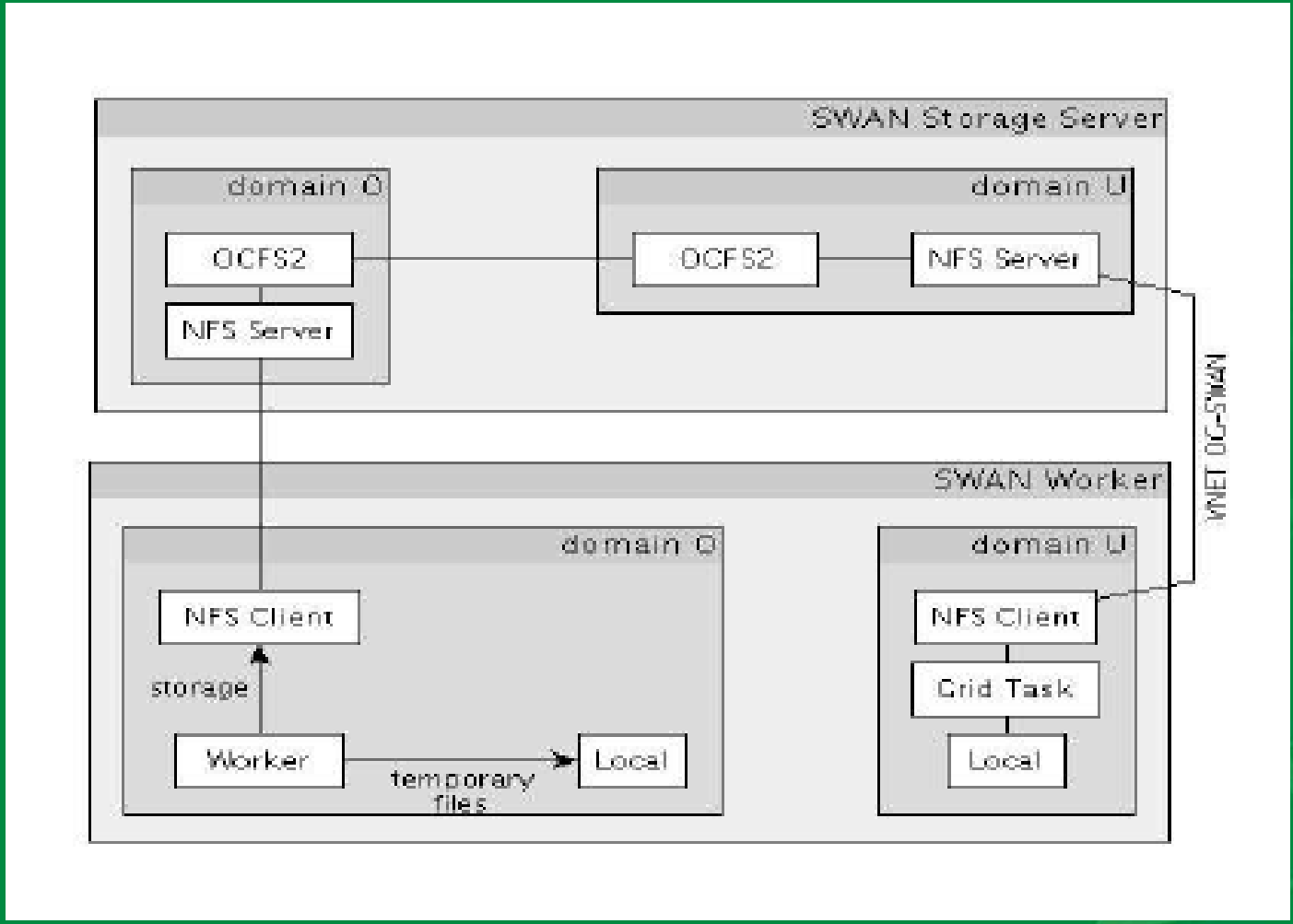
Xen's Architecture



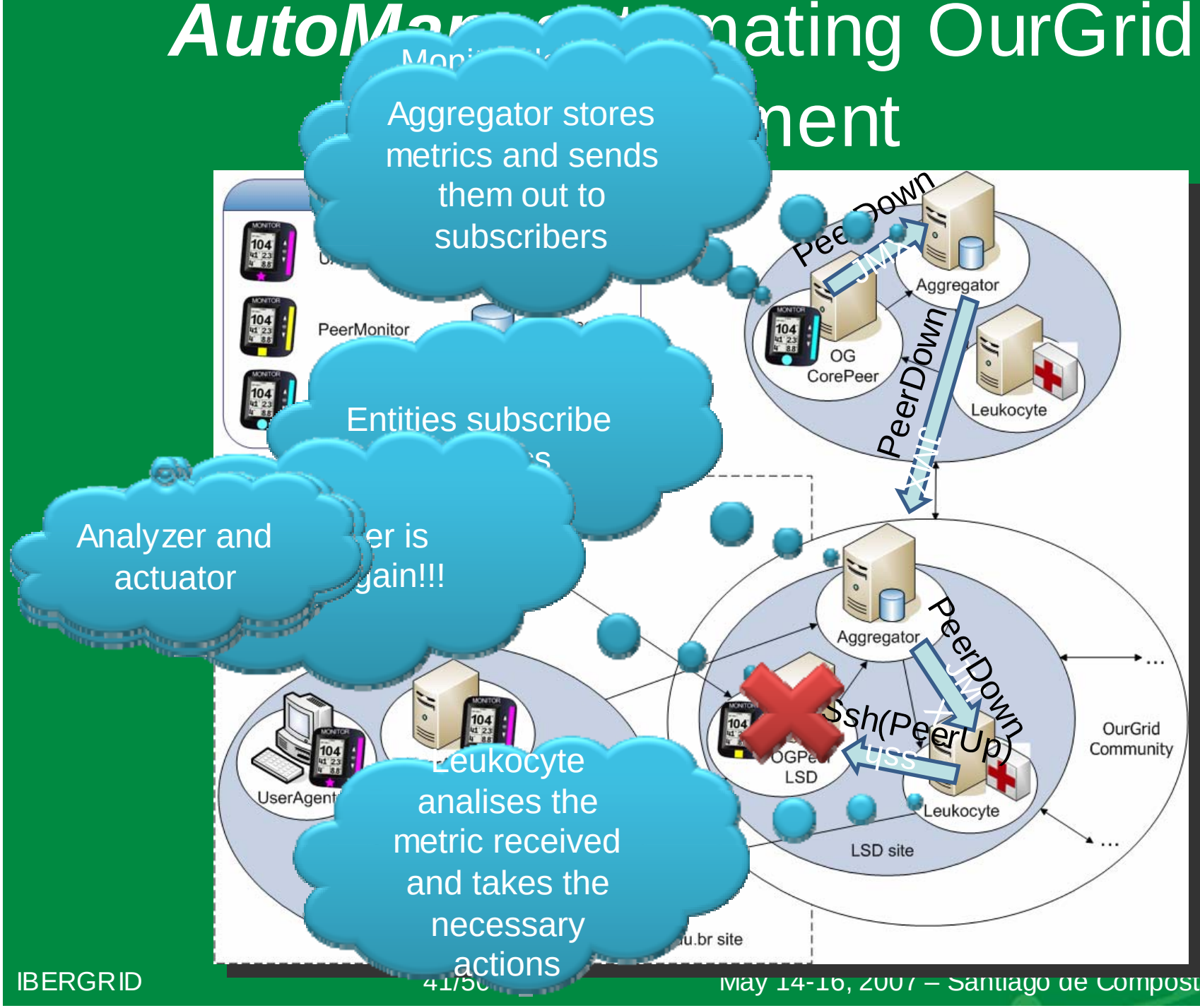
SWAN Architecture



SWAN Implementation



AutoMan… …mating OurGrid …ment



Some Applications

- Script-based
  - Risk assessment for agriculture loans (EMBRAPA)
  - Our own research on computer science
    - Simulations
    - Workload characterization
- API-based
  - SmartPumping (PETROBRAS)
    - Parallel execution of genetic algorithms for optimizing oil pipeline operation
  - EPANET-Grid (R&D project)
    - Grid-enabled version of the EPANET system for simulation of water supply systems
  - GridVida (R&D Project)
    - Image processing to support diagnosis by identifying similar cases in the archival database

Some Applications

- Portal-based
  - SegHidro (R&D project)
    - Several uses related to management of water resources in a Brazilian semi-arid area
    - Academic users and production users
      - Researchers developing and tuning new models
      - Operational centers providing general public with information generated from observations and simulations
    - Allow the configuration of different workflows of simulation models and the execution of them in ensembles
    - Sharing of computing resources, data and complementary expertise

Some Applications

- Framework-based
  - GridUnit (R&D project)
    - An extension of ***JUnit***
  - Features
    - Transparent and Automatic Distribution
    - Test Case Contamination Avoidance
    - Environmental coverage
    - Graphical user interface

Some Pointers to OurGrid Applications

- Otimização do Controle de Redes de Escoamento de Petróleo. Esther Brasileiro, Carlos Galvão, Francisco Brasileiro. *Petro&Química*. São Paulo, v. XXIII, n. 273, p. 153-157, 2005.
- Using the Computational Grid to Speed up Software Testing. Alexandre Nóbrega Duarte, Walfredo Cirne, Francisco Brasileiro, Patricia Machado. *Proceedings of the 19th Brazilian Symposium on Software Engineering (Anais do 19o Simpósio Brasileiro de Engenharia de Software)*. October, 2005.
- Fostering Collaboration to Better Manage Water Resources. Eliane Araújo, Walfredo Cirne, William Voorsluys, Carlos Galvão, Enio Souza, Enilson Cavalcanti. Accepted for publication in Concurrency and Computation: Practice and Experience
- The SegHidro Experience: Using the Grid to Empower a Hydro-Meteorological Scientific Network. Eliane Araújo, Walfredo Cirne, Gustavo Wagner, Nigini Abílio, Enio Souza, Carlos Galvão, Eduardo S. Martins. Proceedings of the eScience 2005: 1st IEEE International Conference on eScience and Grid Computing. December 2005.
- Multi-Environment Software Testing on the Grid. Alexandre Duarte, Gustavo Wagner, Francisco Brasileiro, Walfredo Cirne. *Proceedings of the Parallel and Distributed Systems: Testing and Debugging (PADTAD IV).* Portland, Maine, USA. July, 2006.
- GridUnit: Software Testing on the Grid. Alexandre Duarte, Walfredo Cirne, Francisco Brasileiro, Patrícia Machado. *Proceedings of the 28th International Conference on Software Engineering (ICSE 2006).* May, 2006.

OurGrid Status

- OurGrid free-to-join grid is in production since December 2004
  - We currently have about 300 machines spread around 15 sites
  - See status.ourgrid.org for the current status of the system
- OurGrid is open source (GPL) and version 3.3.2 is available at www.ourgrid.org
  - Please, have a go at it and send us your impression!

On-going research

- Sabotage-tolerance
- Checkpoint
- Autonomous management: AutoMan
- Better presenting OurGrid components in a more diverse SOA setting
- Going beyond BoT
- More diverse security portfolio
- Interoperability with other grid systems
- Adaptive scheduling
- Grid economy
- Leveraging the power of off-line communities
- Software engineering issues

OUR*adhoc*GRID

- Synergy among P2P, grid computing, and Mobile Ad Hoc Networks (MANETs)
- What about a totally distributed OurGrid?
  - No need for setting up peers, core-peer, etc
  - Peers coordinate among themselves
  - Workers find out about their peers
  - Peers get elected on demand (e.g., at least one peer per sub-network)

OUR*adhoc*GRID

- Initial target: a single administrative domain
  - Usually comprised of hundreds or thousands of machines sitting idle most of the time
  - Disseminate ***The Seed*** over the network:
    - **The Seed**: a single entity that can act as a peer or a worker (or both!)

Conclusions

- We have a free-to-join grid solution for BoT applications that is helping people to do:
  - Climate forecast and management of water resources
  - Development of drugs for the Brazilian HIV variant
  - Optimization of pipeline operation in oil industry
  - Support for efficient test-driven programming
  - Simulations, including those that support our own research
- Real users provide invaluable feedback for systems research
- Delivering results to real users has been really cool!

Contacts and URLs

- Francisco Brasileiro (fubica@dsc.ufcg.edu.br)
- Walfredo Cirne (walfredo@dsc.ufcg.edu.br)
- LSD site (http://www.lsd.ufcg.edu.br/)
- OurGrid site (http://www.ourgrid.org/)
- Related projects
  - SegHidro (http://seghidro.lsd.ufcg.edu.br/)
  - Bio Pauá (https://www.biopaua.lncc.br/ENGL/index.php)
  - SmartPumping (http://www.sp.lsd.ufcg.edu.br/)
  - GridUnit (http://gridunit.sourceforge.net/)
  - Portal GIGA (http://portalgiga.unisantos.edu.br/)