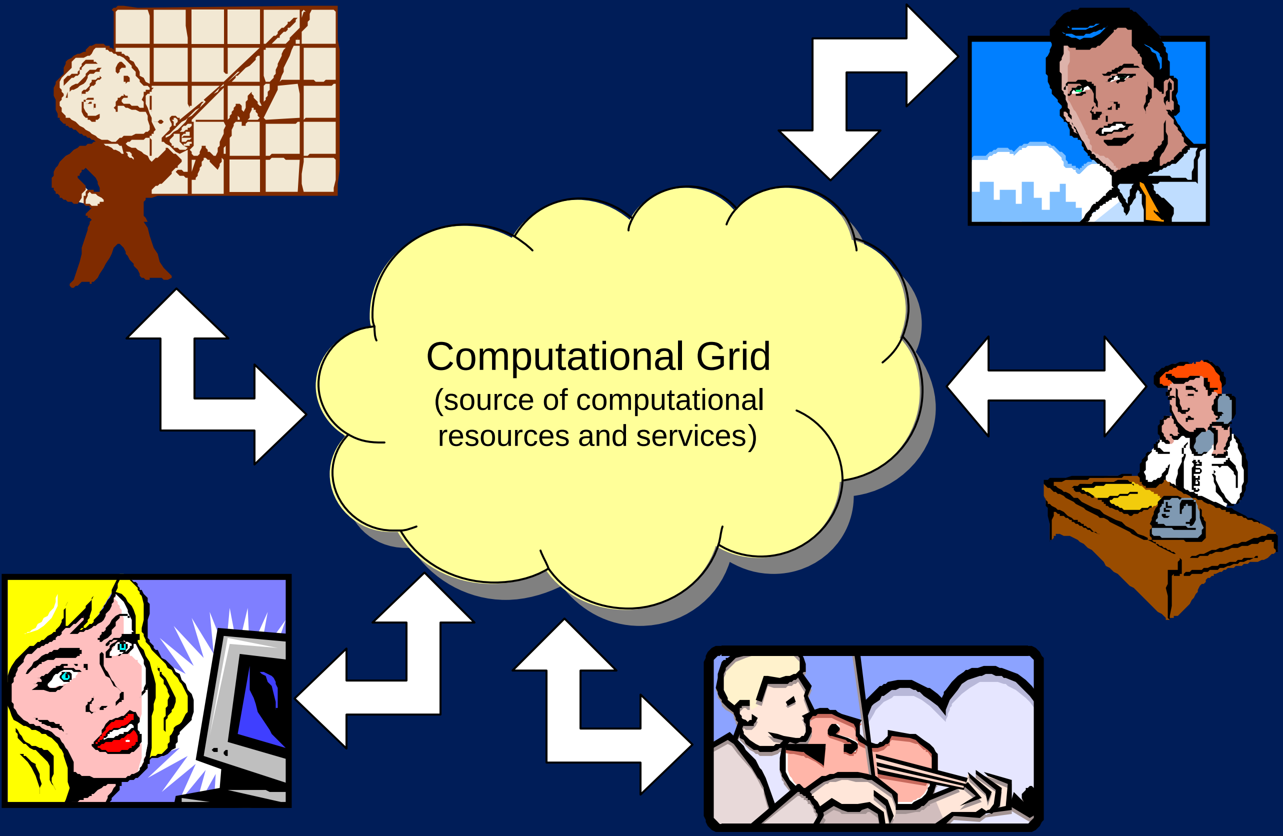
The OurGrid Project

www.ourgrid.org

Walfredo Cirne
walfredo@dsc.ufcg.edu.br
Universidade Federal de Campina Grande

What is a Grid?



Solving a real problem

- To finish my Ph.D., I had to run hundreds of thousands of independent simulations
- Since my simulations were independent, I had the perfect application for the grid
- I was in top grid research lab, but could not use the grid
  - Grid solutions are not in place yet

The motivation for MyGrid

- Users of loosely-coupled applications could benefit from the Grid now
- However, they don´t run on the Grid today because the Grid Infrastructure is not widely deployed
- What if we build a solution that does not depend upon installed Grid infrastructure?

MyGrid

- MyGrid allows a user to run Bag-of-Tasks parallel applications on whatever resources she has access to
  - Bag-of-Tasks applications are those parallel applications formed by independent tasks
- One’s grid is all resources one has access to
  - No grid infrastructure software is necessary
  - Grid infrastructure software can be used (whenever available)

Bag-of-Tasks Applications

- Data mining
- Massive search (as search for crypto keys)
- Parameter sweeps
- Monte Carlo simulations
- Fractals (such as Mandelbrot)
- Image manipulation (such as tomography)
- And many others…

What is MyGrid?

- A broker (or application scheduler)
- A set of abstractions hide the grid heterogeneity from the user

An Example: Factoring with MyGrid

- init

```
mg-services put $PROC ./Fat.class $PLAYPEN
```

- grid1

```
java Fat 3 18655 34789789798 output-$TASK
```

- collect

```
mg-services get $PROC $PLAYPEN output-$TASK
```

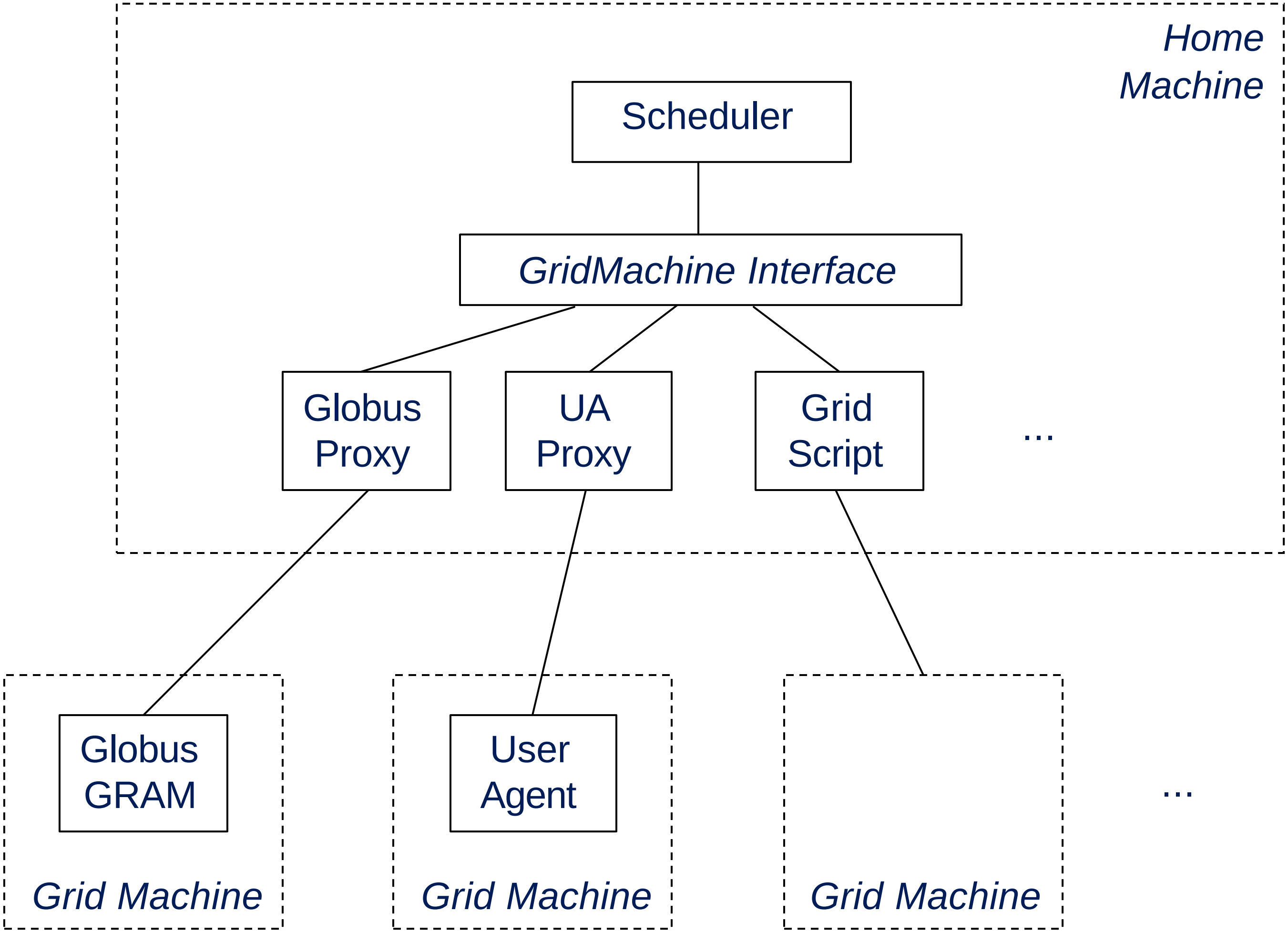
- grid2

```
java Fat 18655 37307 34789789798 output-$TASK
```

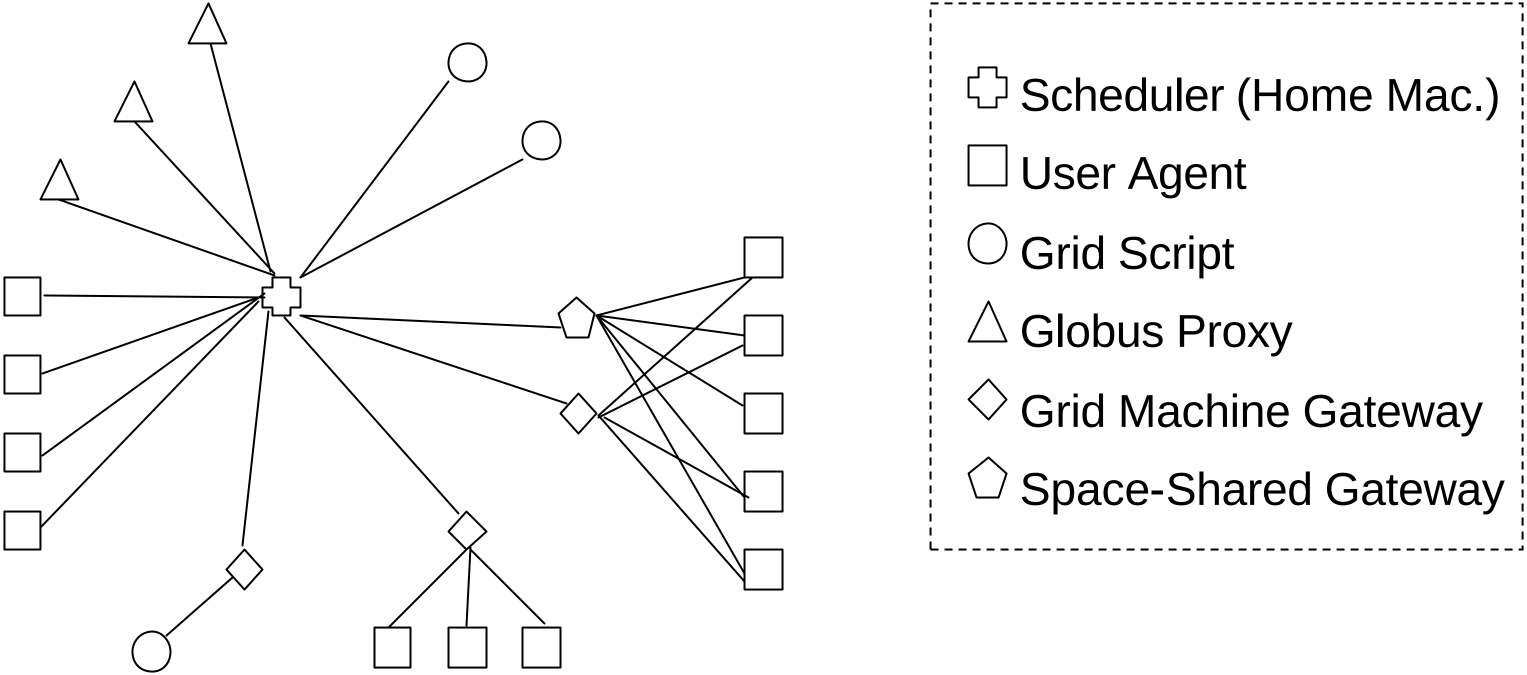
Defining my personal Grid

```
proc:
  name = ostra.lsd.ufcg.edu.br
  attributes = lsd, linux
  type = user_agent

proc:
  name = memba.ucsd.edu
  attributes = lsd, solaris
  type = grid_script
  rem_exec = ssh %machine%command
  copy_to = scp %localdir/%file %machine:%remotedir
  copy_from = scp %machine:%remotedir/%file %localdir

[...]
```

Making MyGrid Encompassing



Dealing with Firewalls, Private IPs, and Space-Shared Machines



✚ Scheduler (Home Mac.)

□ User Agent

○ Grid Script

△ Globus Proxy

◇ Grid Machine Gateway

⬠ Space-Shared Gateway

The Scheduling Challenge

- Grid scheduling typically depends on information about the grid (e.g. machine speed and load) and the application (e.g. task size)
- However, getting grid information makes it harder to build an encompassing system
  - The GridMachine Interface would have to be richer, and thus harder to implement
- Moreover, getting application information makes the system harder to use and more complex
  - The user would have to provide task size estimates

Scheduling with no information

- Work-queue 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
  - Replication may have a limit
- The key is to avoid having the job waiting for a task that runs in a slow/loaded machine

Work-queue with Replication

- 8000 experiments
- Experiments varied in
  - grid heterogeneity
  - application heterogeneity
  - application granularity
- 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%

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
- There are knowledge-dependent schedulers that explore data reutilization

Storage Affinity

- The “affinity” between a task and a site is the number of bytes within task input that is already stored at there
  - The heuristic is based on easy-to-get static information (size and location of data)
- The task with largest “affinity” is prioritized
  - The idea is avoid unnecessary data transfers
- Replication is used to cope with mistakes

Storage Affinity Results

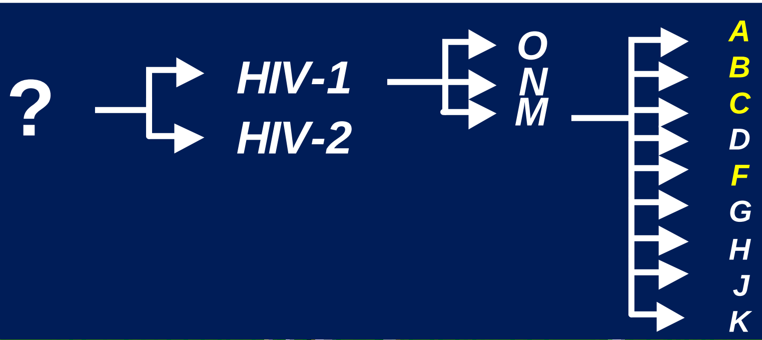
- 3000 experiments
- Experiments varied in
  - grid heterogeneity
  - application heterogeneity
  - application granularity
- Performance summary:

	Storage Affinity	XSufferage	WQR
Average (seconds)	57.046	59.523	150.270
Standard Deviation	39.605	30.213	119.200

Proof of Concept

- During a 40-day period, we ran 600,000 simulations using 178 processors located in 6 different administrative domains widely spread in the USA
- We only had GridScript and WorkQueue
- MyGrid took 16.7 days to run the simulations
- My desktop machine would have taken 5.3 years to do so
- Speed-up is 115.8 for 178 processors

HIV research with MyGrid



?

HIV-1

HIV-2

O

N

M

A prevalent in Africa

B prevalent in Europe and Americas

C majority in the world

D

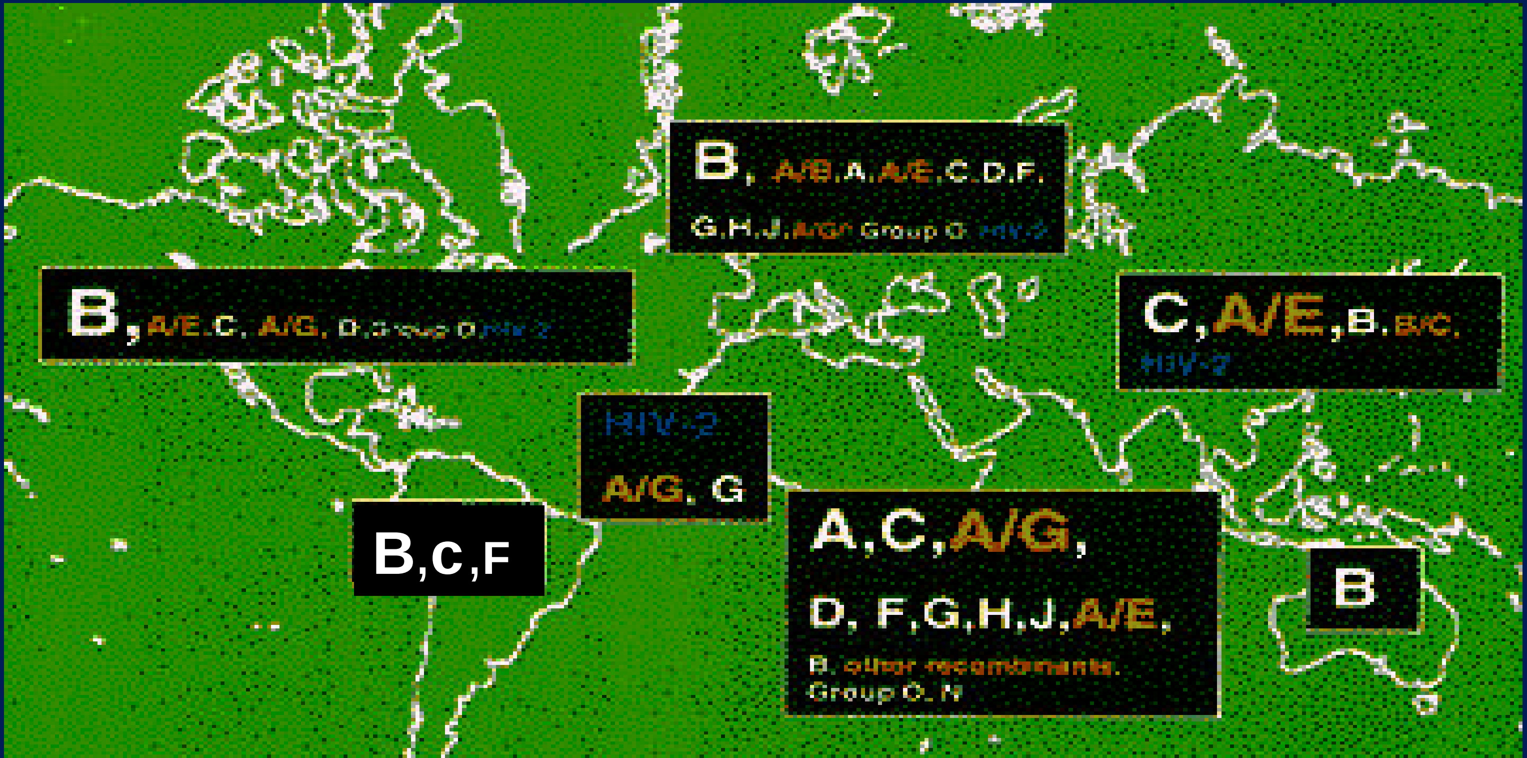
F 18% in Brazil

G

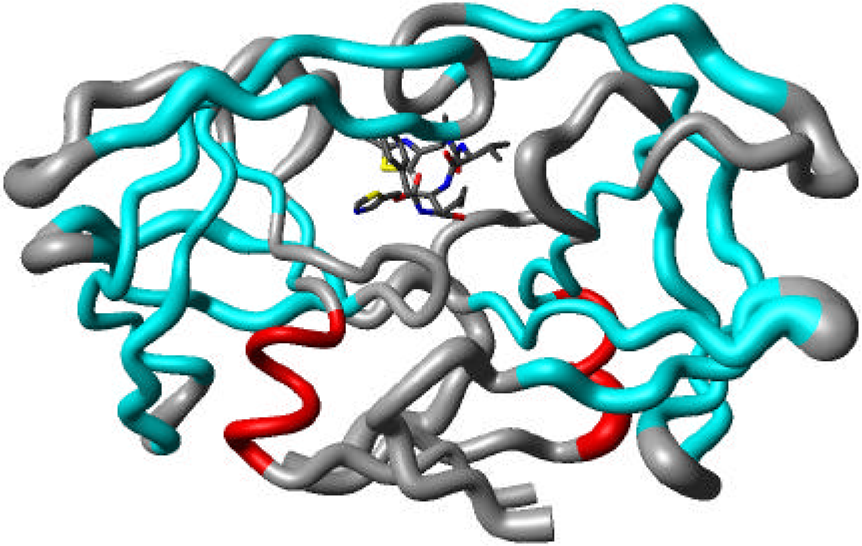
H

J

K

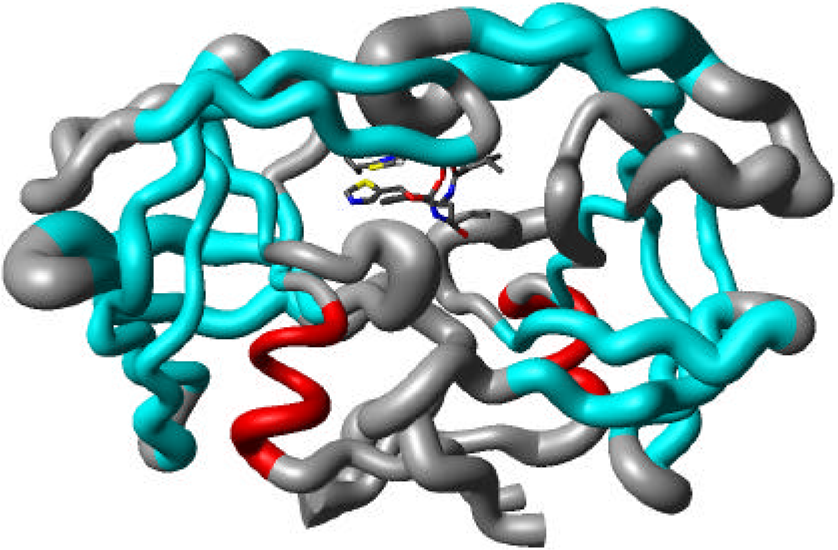


HIV protease + Ritonavir



Subtype B

Subtype F



The HIV Research Grid

- 55 machines in 6 administrative domains in the US and Brazil
  - The machines were accessed via User Agent, UA + Grid Machine Gateway, UA + ssh tunnel, and Grid Scripts
- Task = 3.3 MB input, 1 MB output, 4 to 33 minutes of dedicated execution
- Ran 60 tasks in 38 minutes
- Speed-up is 29.2 for 55 machines
  - Considering an 18.5-minute average machine

MyGrid Status

- MyGrid is open source and is available at http://www.ourgrid.org/mygrid
  - About 150 downloads
  - 2.0 version released two months ago
  - Base of the PAUÁ Grid, currently being deployed by HP Brazil
- Bag-of-tasks parallel applications can currently benefit from the Grid
  - However, firewalls, private IPs and the such make it much harder than we initially thought

Demands of MyGrid Users

- More resources → OurGrid
  - People want to use more resources than they have access to
- Good “debugging” → MyGridDoctor
  - Abstractions are wonderful when they work, but when they fail... :-(
- More security → SWAN
  - Local resources
  - Use of grid machine as attack launchpad
- Richer programming model

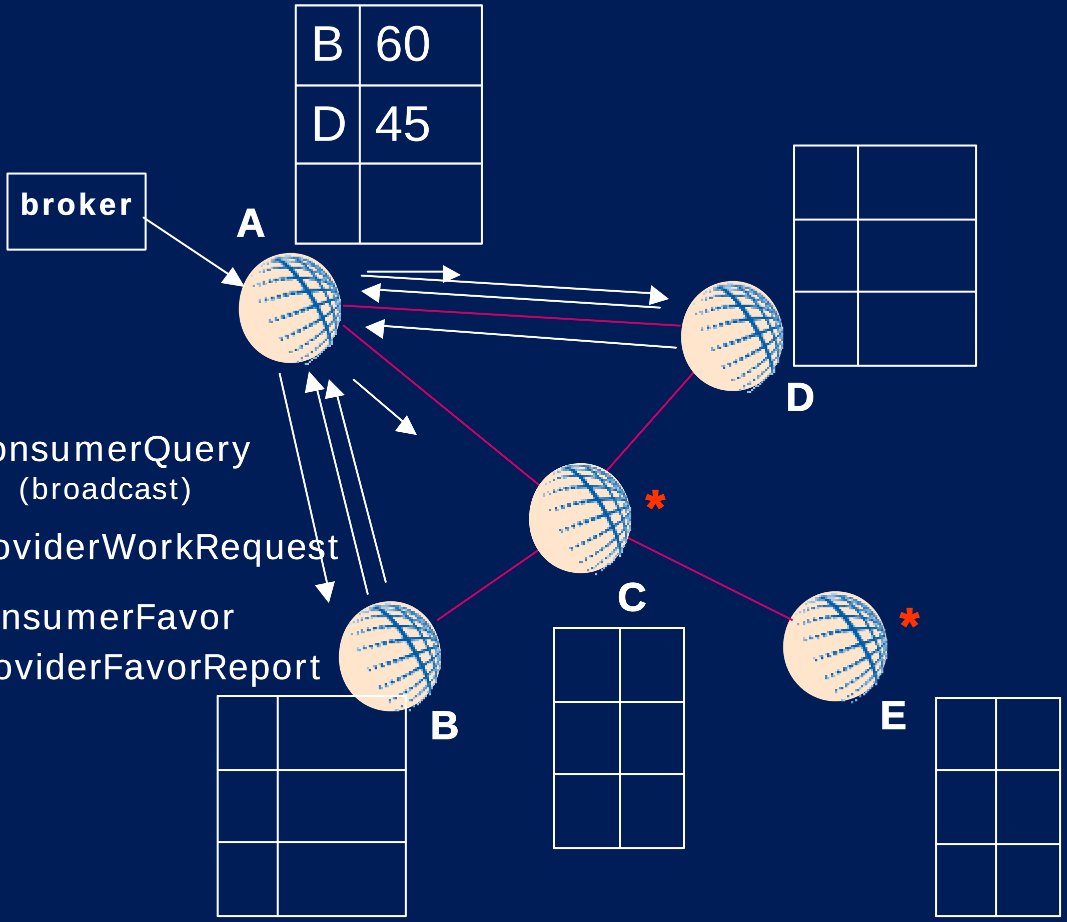
OurGrid: A Network of Favors

- Getting access to resources is out of scope of today’s grid solutions
  - Grid economy will solve this problem some day
- But BoT applications can use lots of resource now
- Let’s trade off generality for simplicity
  - There are at least 2 resource providers
  - Applications that shall use the resources need no QoS guarantees
- P2P resource sharing community
  - Network of Favors

Making people collaborate

- It’s important to encourage collaboration within OurGrid (i.e., resource sharing)
  - In file-sharing, most users free-ride
- OurGrid uses a P2P Reputation Scheme
  - All peers maintain a local balance for all known peers
  - Peers with greater balances have priority
  - The emergent behavior of the system is that by donating more, you get more resources
  - No additional infrastructure is needed

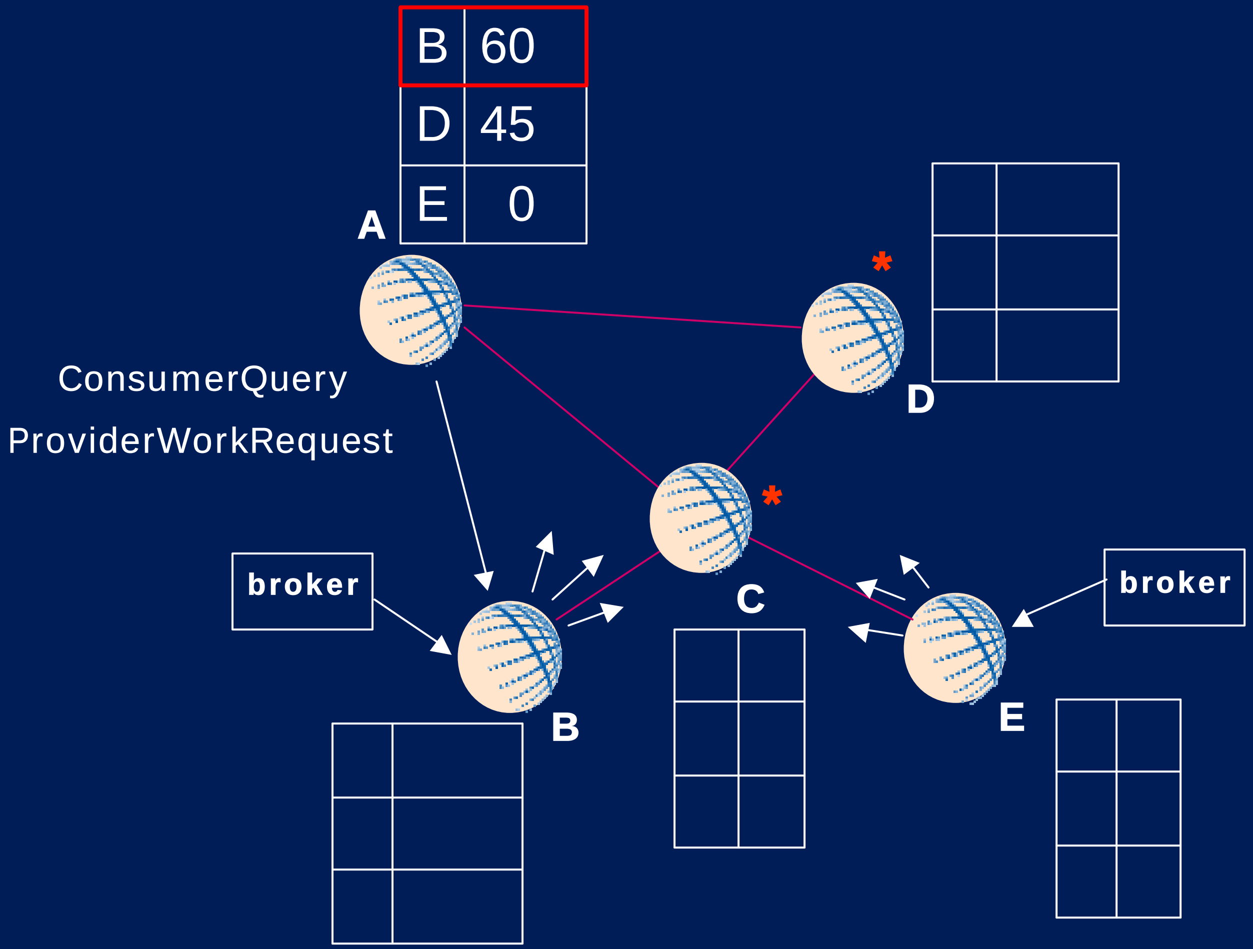
OurGrid resource sharing [1]



B	60
D	45

* = no idle resources now

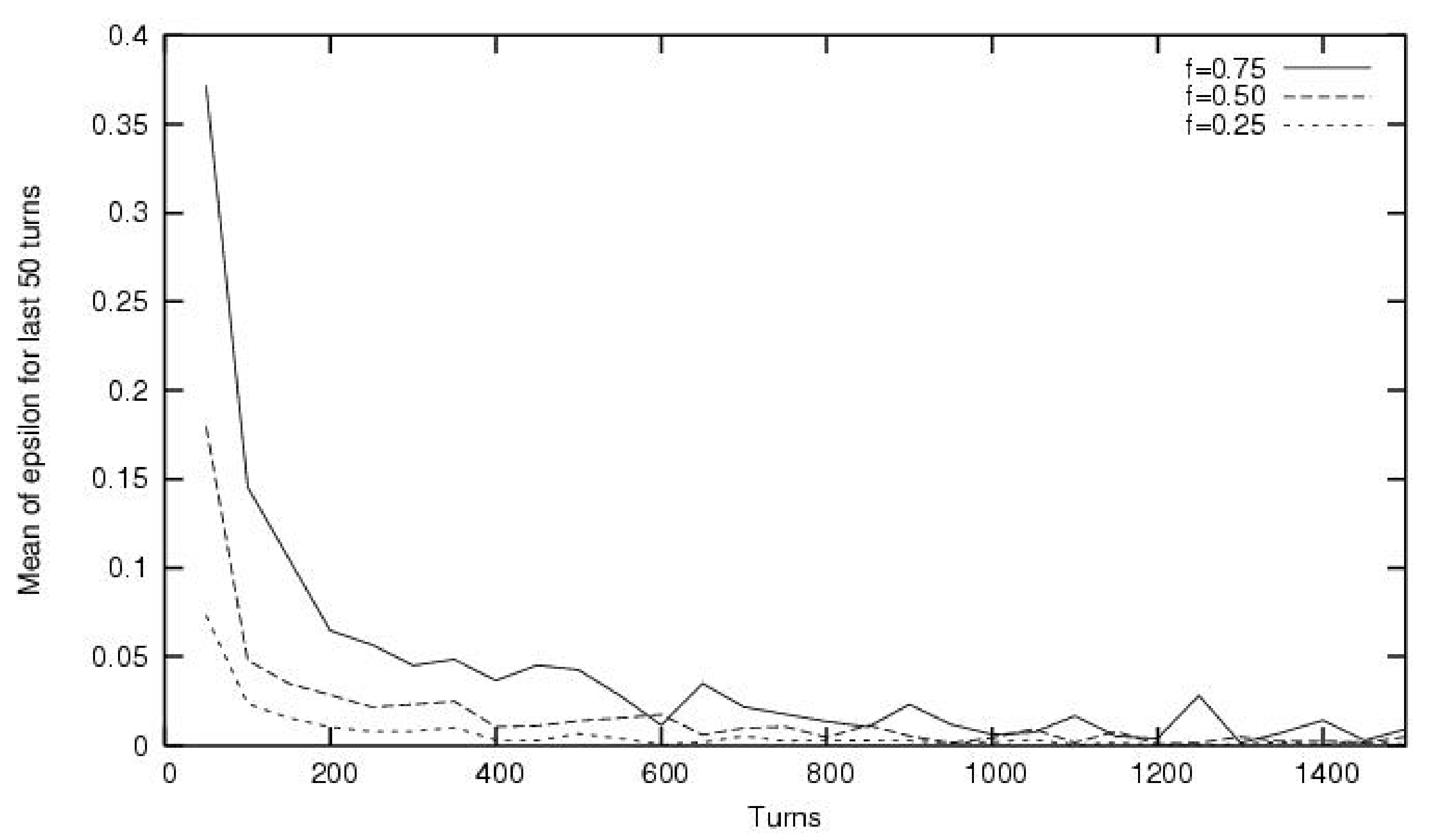
OurGrid resource sharing [2]



B	60
D	45
E	0

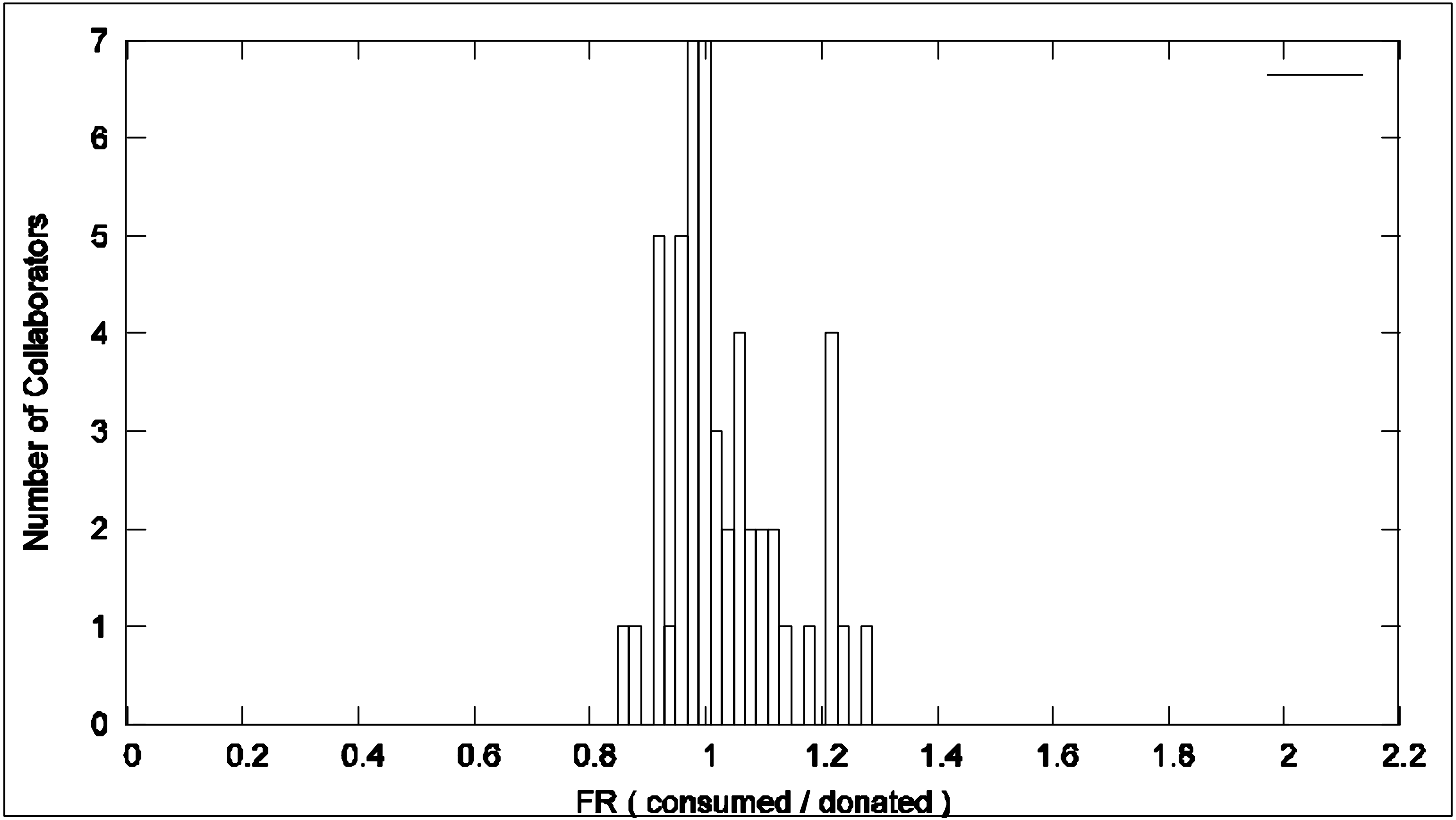
* = no idle resources now

Free-rider consumption



- Epsilon is the fraction of resources consumed by free-riders

Equity among collaborators



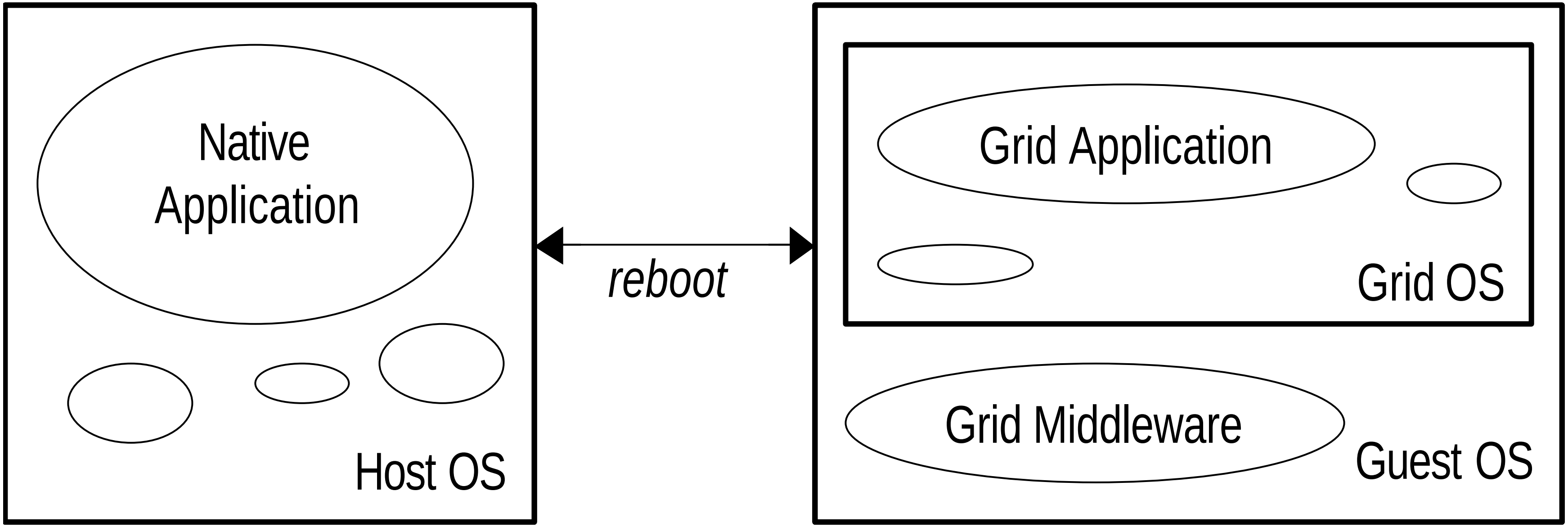
Sandboxing for BoT applications

- In the OurGrid Community, a peer runs unknown code that comes from the Grid
- This an obvious security concern
  - Threat to local data and resources
  - Use of machine as drone to attack others
- We leverage from the fact BoT applications communicate only to receive input and send output to run the guest application in a very tight sandbox, with no network access

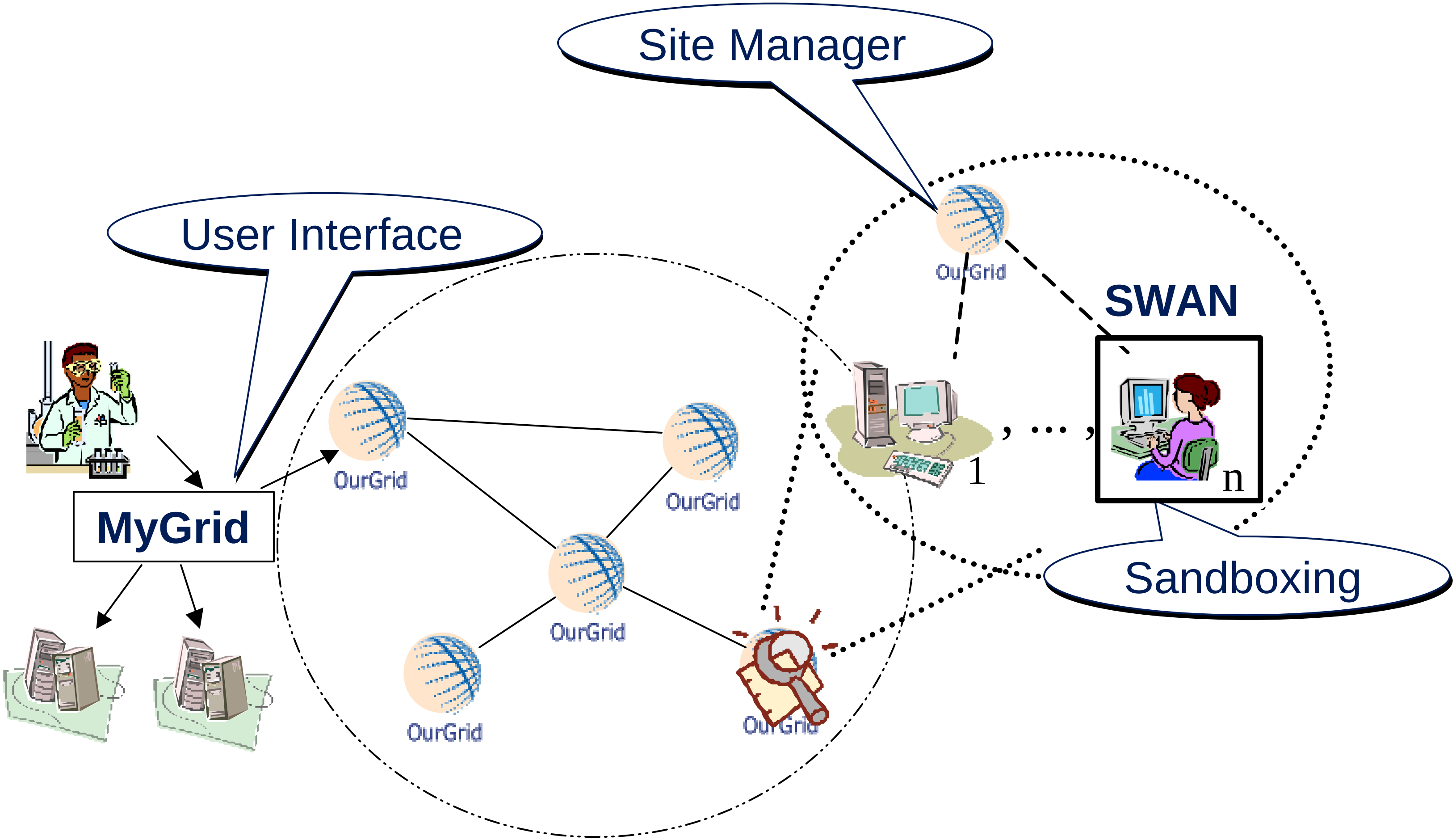
Adding a second line of defense

- We also reboot to add a second layer of protection to the user data and resources
- This has the extra advantage of enabling us to use an OS different from that chosen by the user
  - That is, even if the user prefers Windows, we can still have Linux
- Booting back to the user OS can be done fast by using hibernation

SWAN architecture



OurGrid overall architecture



Collaboration/Interest on OurGrid

- HP Brazil R&D
- HP Labs Bristol
- HP Partners
  - LNCC, UniSantos, UniFor, Instituto Atlântico
  - CESAR/UFPE, Instituto Eldorado, IPT, AMR
  - PUCRS, UniSinos, UFRGS, USP
- Others
  - UCSD, UnB, UFBA, UCS, UniCap, UFPB, UFAL ...

Questions?

OurGrid

Thank you!
Merci!
Danke!
Grazie!
Gracias!
Obrigado!

More at www.ourgrid.org