Running Thor over MyGrid

Walfredo Cirne

Universidade Federal de Campina Grande

Thor

- Thor is a Molecular Modelling package
- It simulates the molecular movement of atoms and molecules
- It determines the lower energy configuration of the system
- It was developed at UFRJ (Paulo Bisch´s team)

Fighting AIDS with Thor

- HIV virus mutates at a fast pace
- Regional variants are emerging
- We would like to better understand the drugs effect on the Brazilian HIV variant
- We need to run a Thor parameter sweep, varing the HIV variant and the inhibitor
  - Thor is used to simulate the inhibitor/protease interaction

Bag-of-Tasks Applications

- Parallel applications whose tasks are independent
  - 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…

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 Scope

- 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)

What is MyGrid?

- MyGrid is a framework to build and run BoT applications on user-defined grids
- The user provides:
  - A description of her Grid
  - A way to do remote execution and file transfer
  - “The application”
- MyGrid provides:
  - Grid abstractions
  - Scheduling

Simple MyGrid Example

```
# initial
mg-services mirror $PROC tarefa
mg-services put $PROC ENTRADA.$TASK $PLAYPEN

# grid
tarefa < ENTRADA.$TASK > SAÍDA

# final
mg-services get $PROC $PLAYPEN/SAÍDA
    resultados/SAÍDA.$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

[...]
```

Factoring with MyGrid

- Fatora n generates files tasks, init, gridi, and collect, and then invokes mg-addtask tasks
- tasks

```
task:
init= init
grid= grid1
final= collect
task:
init= init
grid= grid2
…
```

Factoring with MyGrid

- init

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

- grid1

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

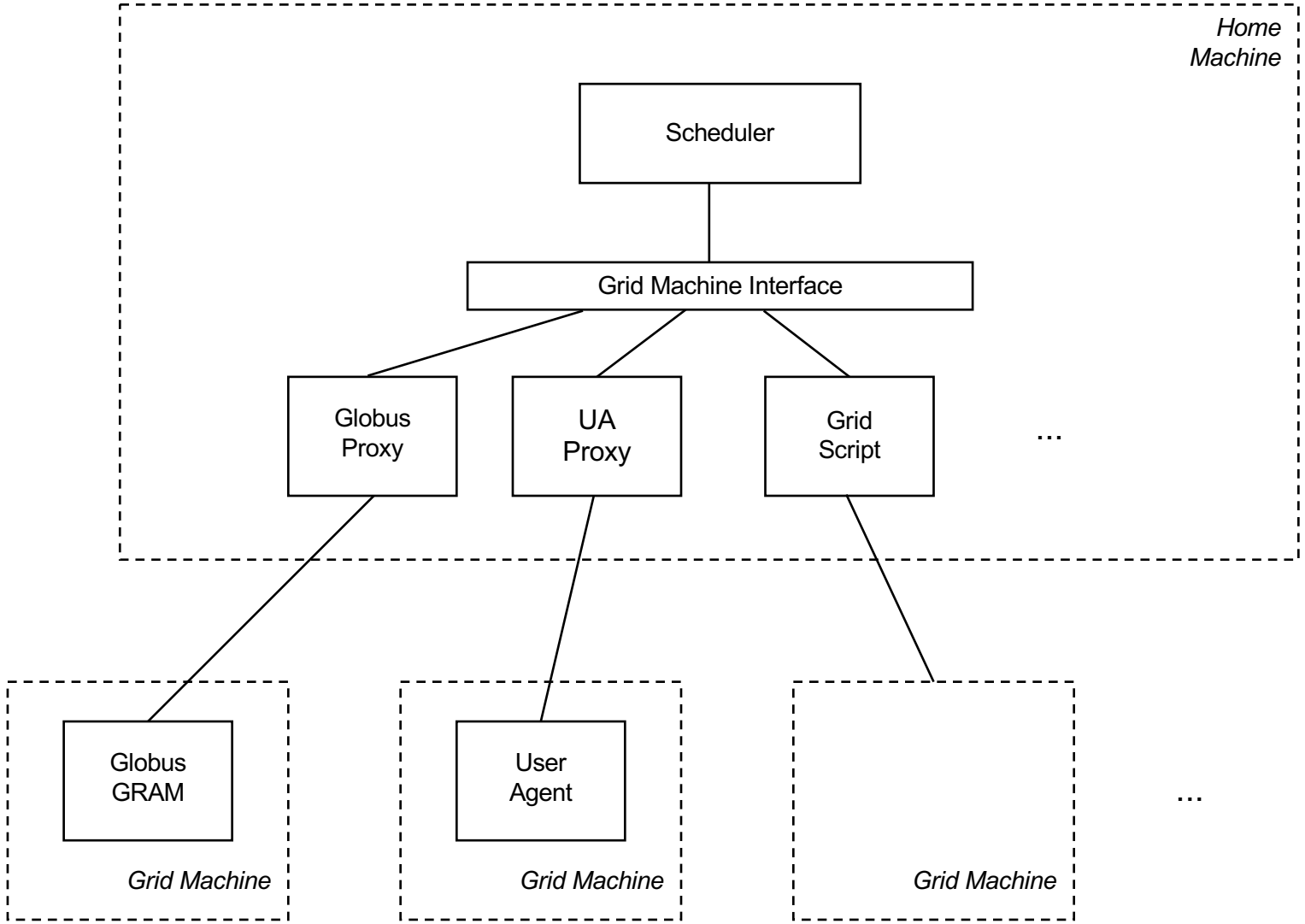
- grid2

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

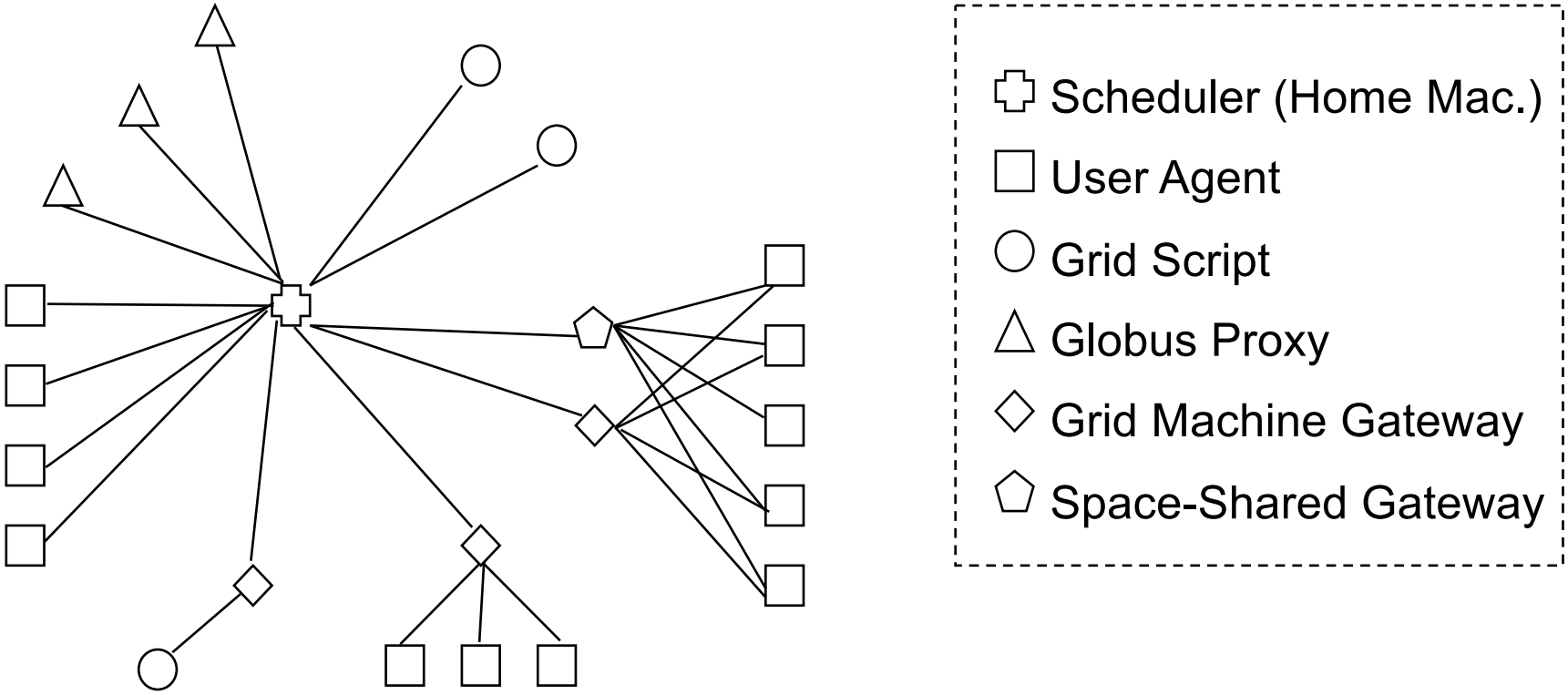
- collect

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

Making MyGrid Encompassing



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



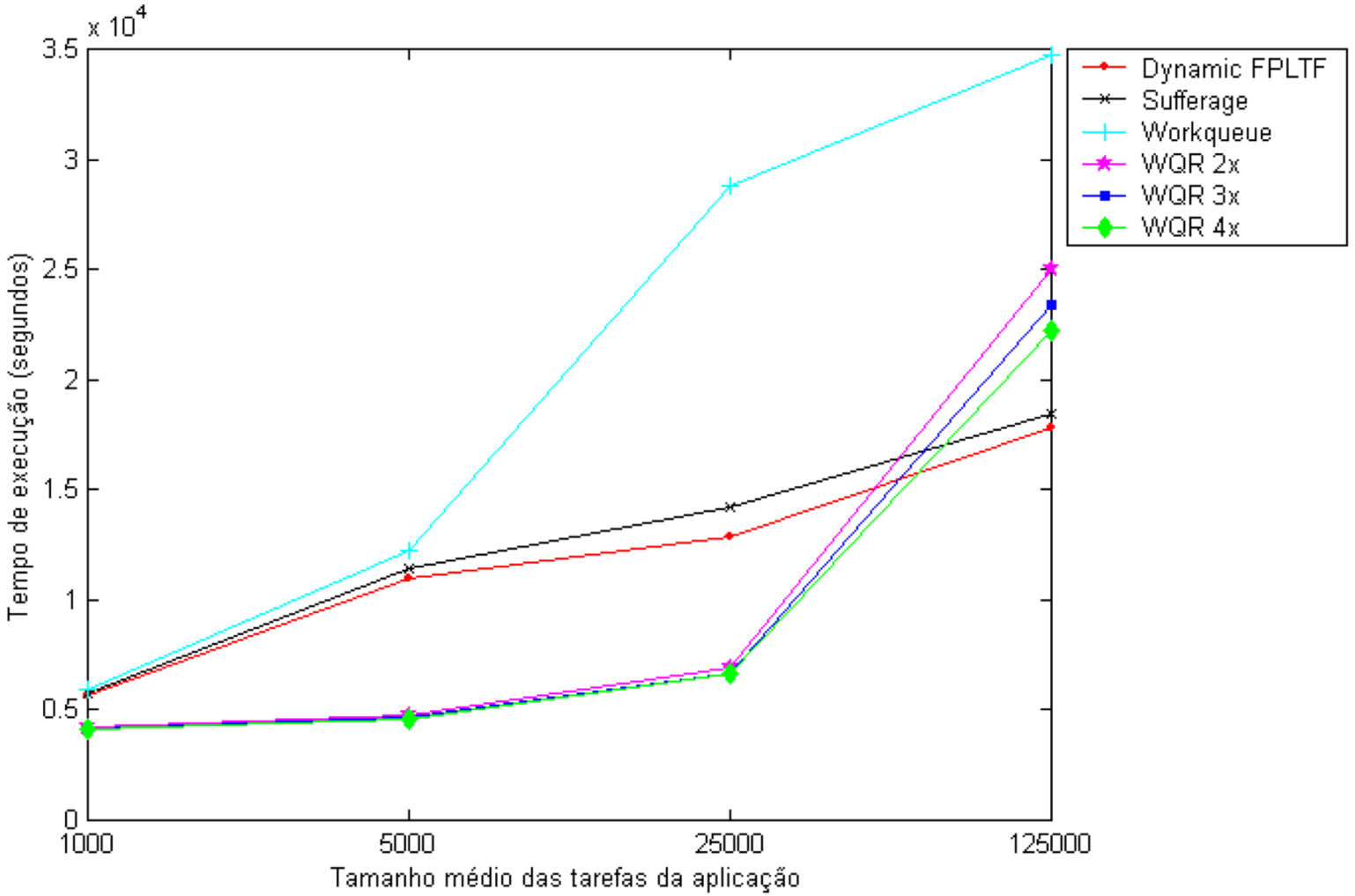
The Scheduling Challenge

- 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 Grid Machine Interface would have to be richer, and thus harder to implement
- Moreover, getting application information makes the system harder to use and less simple
  - The user would have to provide task size estimates

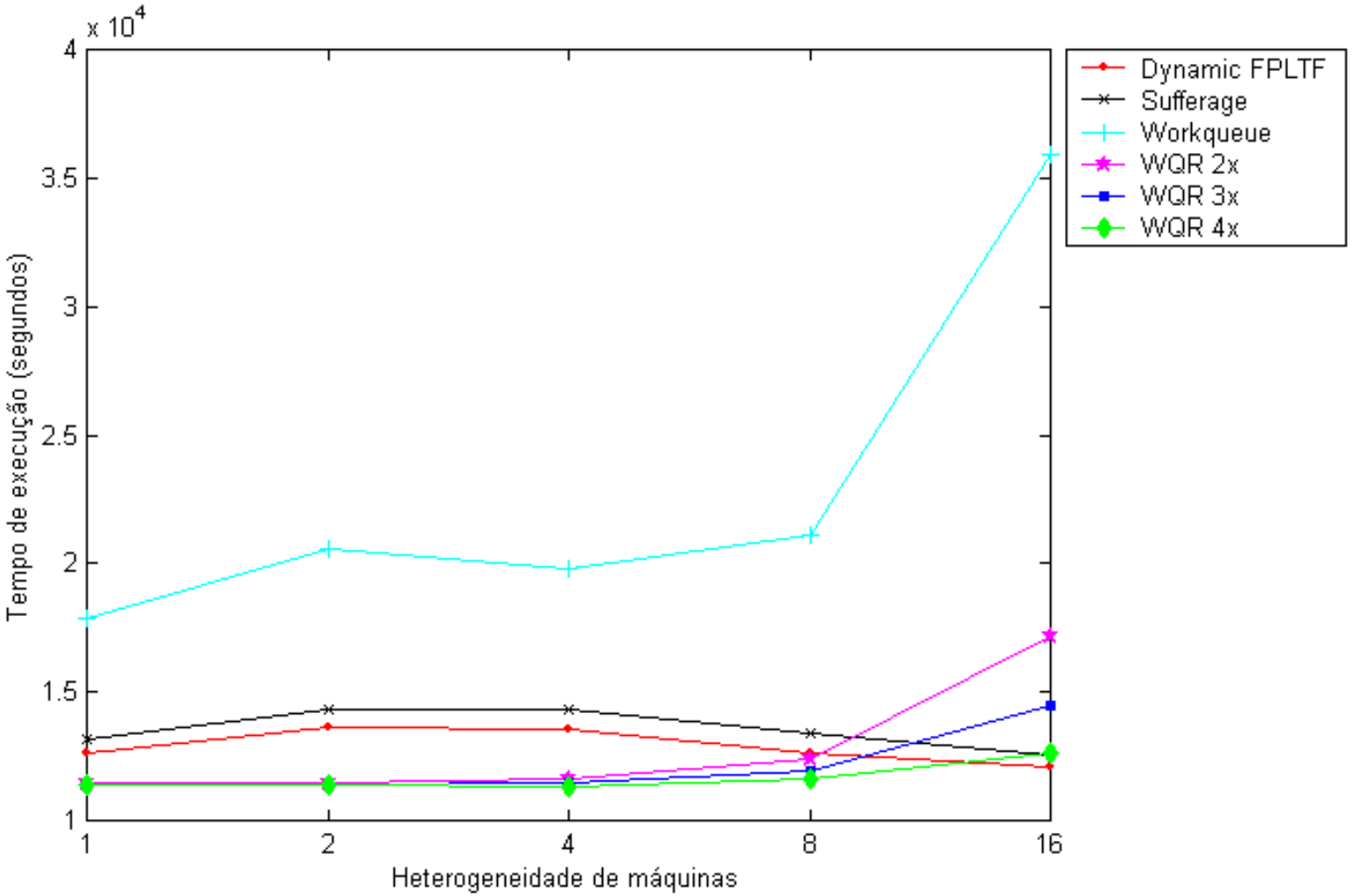
Scheduling With No Information

- 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

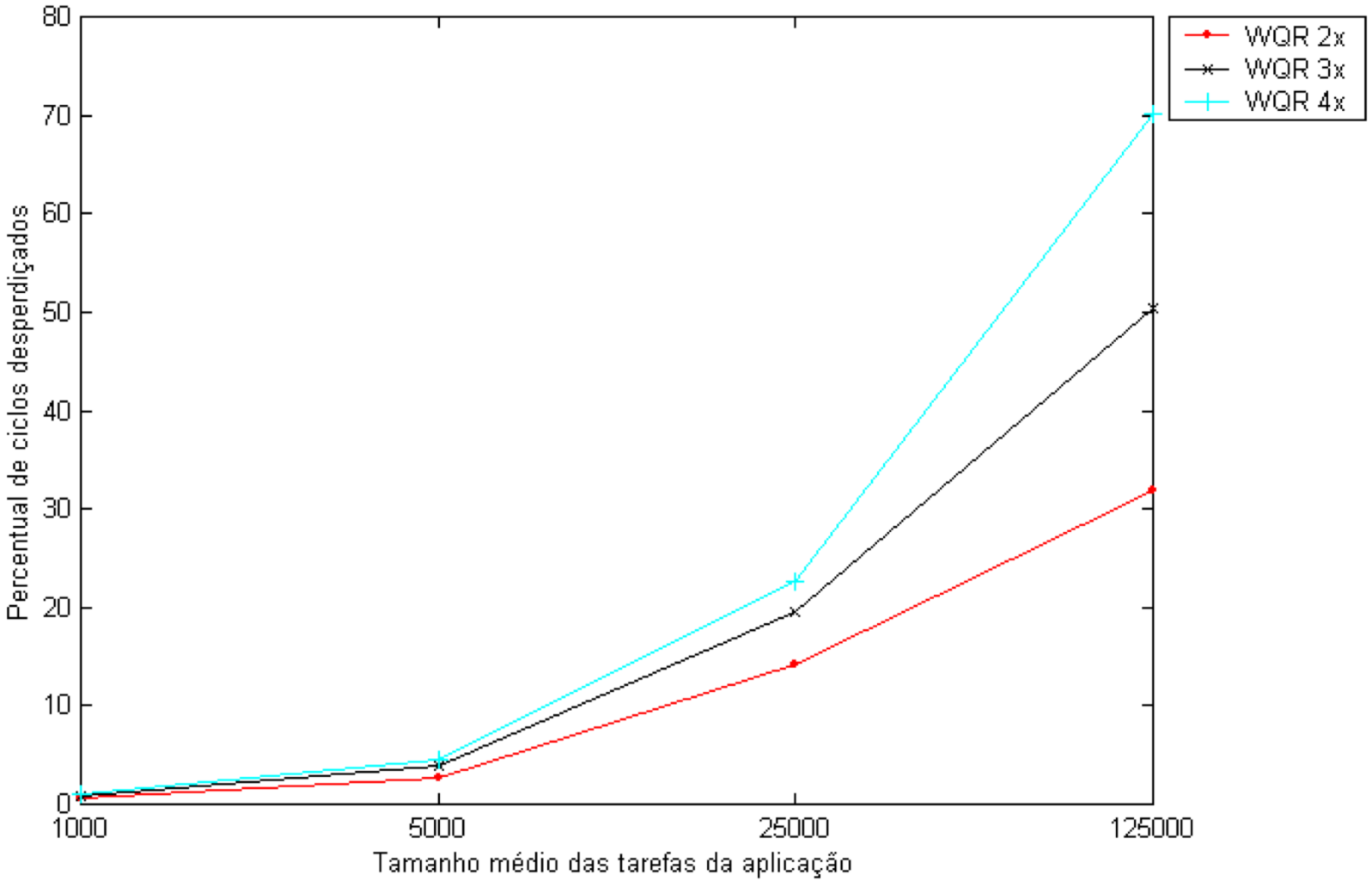
Application Granularity



Grid Heterogeneity



Application Granularity



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

Fighting AIDS

- 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, 200 KB 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

Conclusions

- Bag-of-tasks parallel applications can currently benefit from the Grid
- Running grid applications at the user-level is a viable strategy
  - However, firewalls, private IPs and the such make it much harder than we initially thought
  - Is “upperware” the way to go for new middleware development?

Future Work

- Make MyGrid OGSA-complaint
- Create OurGrid, a community grid for resource sharing
- Extend the scheduler for data intensive applications
  - Such a scheduler should try to minimize data movement