# Grid Simulator with Production Scheduling Algorithms

Miroslav Ruda[1] Hana Rudová[2]

[1]Institute of Computer Science
Masaryk University

[2]Faculty of Informatics
Masaryk University

Cracow, 2007

# Scheduling Algorithms



## Algorithms in Grid simulators

- SimGrid, GridSim, GSSIM , Alea
- development and testing of new algorithms
- comparison of algorithms

## Algorithms in production systems

- PBSPro, SGE, Maui, Moab
- in simulators: approximated with FIFO (with backfilling)
- hard to reimplement
  - many rules, features, bugs
  - closed source, algorithms not published

## Example



- Information about Excludus "Grid Optimizer"
  - uses innovative real-time scheduling algorithm
  - dynamic adaptive scheduling beyond traditional workload managers

# Scheduling Algorithms

## Algorithms in Grid simulators

- SimGrid, GridSim, GSSIM , Alea
- development and testing of new algorithms
- comparison of algorithms

## Algorithms in production systems

- PBSPro, SGE, Maui, Moab
- in simulators: approximated with FIFO (with backfilling)
- hard to reimplement
  - many rules, features, bugs
  - closed source, algorithms not published

## Example



- Information about Excludus "Grid Optimizer"
  - uses innovative real-time scheduling algorithm
  - dynamic adaptive scheduling beyond traditional workload managers

# Scheduling Algorithms

## Algorithms in Grid simulators

- SimGrid, GridSim, GSSIM , Alea
- development and testing of new algorithms
- comparison of algorithms

## Algorithms in production systems

- PBSPro, SGE, Maui, Moab
- in simulators: approximated with FIFO (with backfilling)
- hard to reimplement
  - many rules, features, bugs
  - closed source, algorithms not published

## Example

`www.excludus.com`

- Information about Excludus "Grid Optimizer"
  - uses innovative real-time scheduling algorithm
  - dynamic adaptive scheduling beyond traditional workload managers

# Simulator with production resource management system



## Experiments with PBSPro

- different setup of PBS scheduler
  - queues, priorities, backfilling, ...
- different setup of worker nodes
  - number of nodes per queue, ...
- modifications of PBS scheduler
- inclusion of virtual machines into PBS

Future: new scheduler

# Simulator with Virtual Machines

- Worker nodes represented by virtual machines
- Standard PBS Server and Scheduler
  - running on dedicated server
- Standard PBS Mom
  - running within each virtual machine
- Sleep jobs
  - no cpu/memory consumption

# Workloads

- Real workloads
- Czech Grid *META Centrum*
- Extracted from PBS accounting
- 2005-2007

- Jobs submitted with the same requirements
  - on worker nodes
  - to the same queues
  - with original owners, ...
- Time reduction
  - configurable reduce factor (600)
  - expected and real wall-clock time
  - job arrival time

# Vserver based virtual machines

- One kernel space - very lightweight
- Similar to
  - Linux chroot or BSD jail, with better protection
- Access limits
  - standard: filesystem
  - added: processes, network devices ...
- No hardware emulation, no paravirtualisation
  - no performance penalty
- Copy On Write filesystem
  - one RO root filesystem, with RW overlay filesystem
- System daemons
  - running only once in hosting environment

# Experimental Testbed

- Current workloads (year 2007)
  - January 4.700 jobs, March 14.000 jobs, Jan-March 70.000 jobs
- 150 Vserver domains
  - 16 core AMD machine . . . . can use more physical machines
  - represents 300 nodes . . . . . can be extended
- COW filesystem
  - 300 MB one system instalation
  - 12 GB used to represent 150 virtual machines
- Virtual machine: only PBS Mom + sshd
- Submission of all jobs
  - without any sleep takes less then 10 minutes
- Reduce factor 600
  - 1 month -> 1.5 hours, 1 year -> $\leq$ 1 day
  - reasonably small simulation overhead

# Evaluation Criteria

## Standard monitoring during simulation run

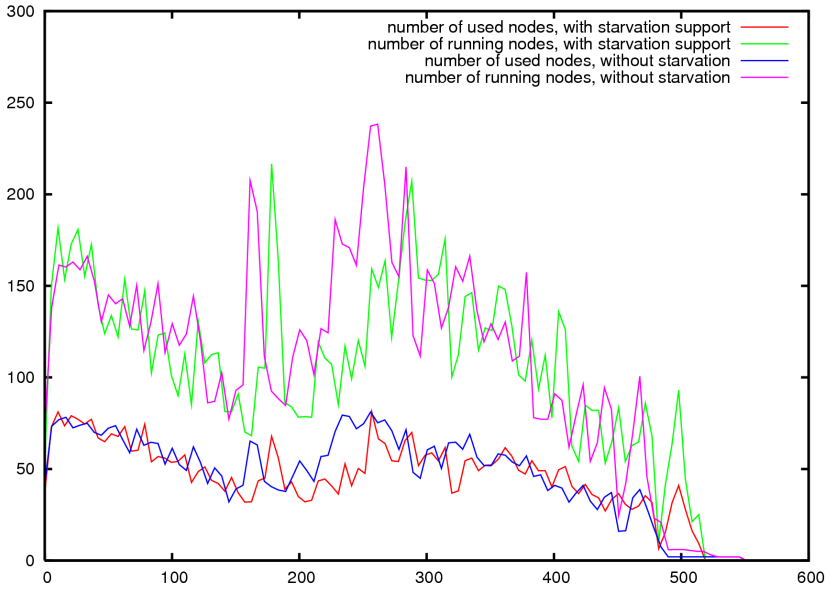
- number of running/waiting/done jobs
- number of used nodes

## Analysis of accounting data

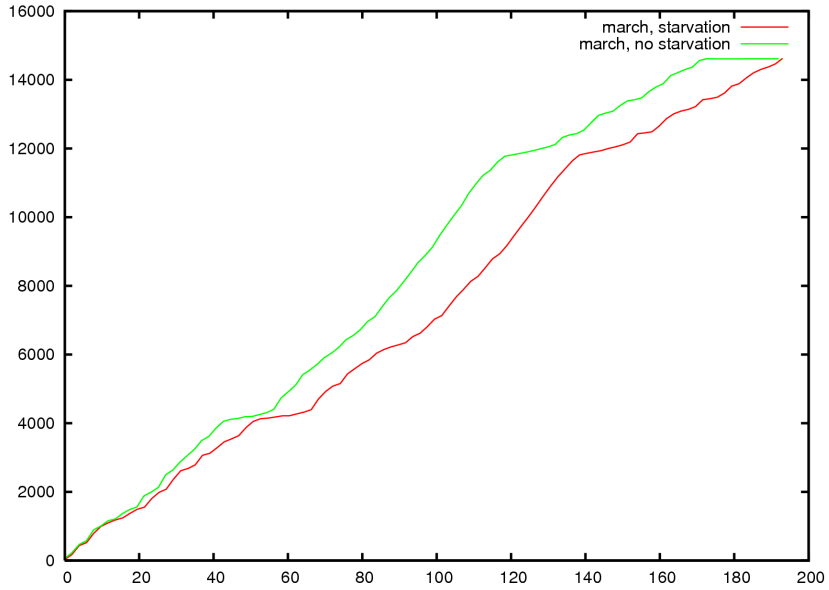
- Weighted Response Time (WRT)
- Weighted SlowDown Time (WSD)
- Weighted Wait Time (WWT)
- metrics per user, queue
- also structured by number of nodes used by job

# Experimental Results for March Workload






Number of running jobs and number of used worker nodes. First simulation with "starvation support", second without.

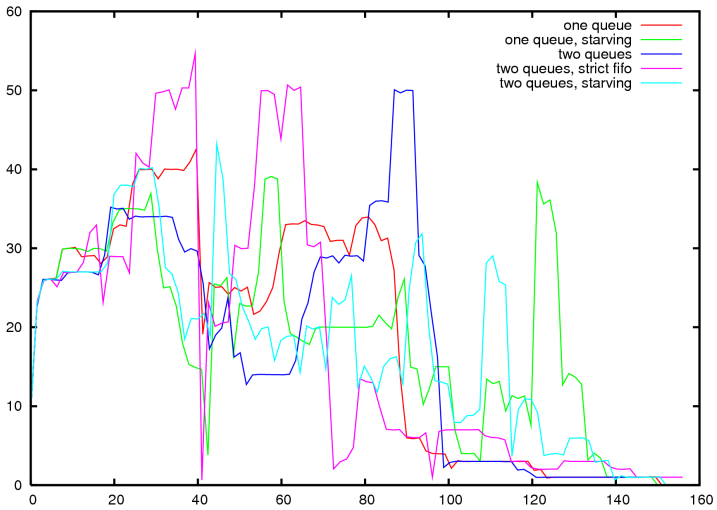



Number of finished jobs.

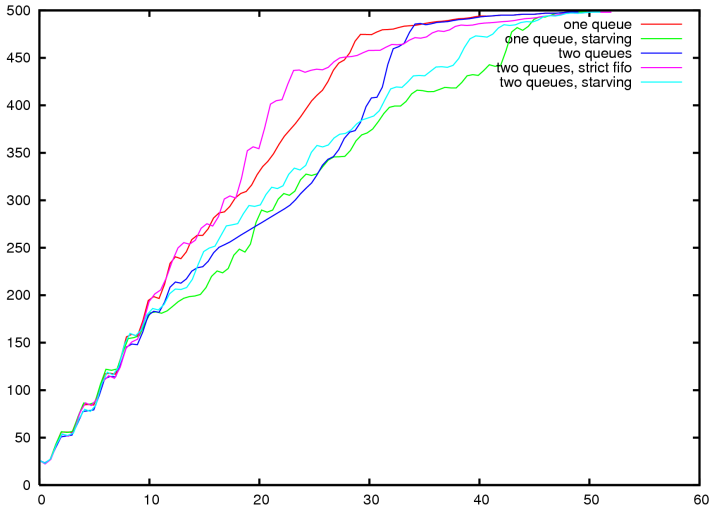
# Experimental Results for Parallel Jobs I.



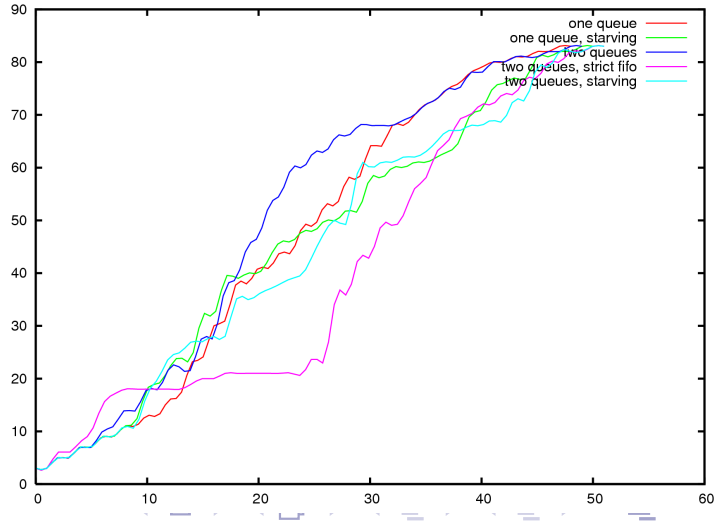
## number of running jobs




## number of finished jobs




## number of finished parallel jobs



# Experimental Results for Parallel Jobs II.



## Wait time based on number of nodes used by job

| Nodes | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 166 | 326 | 261 | 99 | 272 |
| 2 | 40 | 44 | 25 | 244 | 28 |
| 4 | 308 | 337 | 229 | 522 | 352 |
| 8 | 339 | 513 | 252 | 425 | 584 |
| 12 | 434 | 580 | 393 | 615 | 853 |
| 16 | 617 | 343 | 524 | 694 | 355 |
| 28 | 817 | 361 | 758 | 884 | 430 |
| 32 | 820 | 676 | 761 | 857 | 747 |
| 40 | 1052 | 937 | 1020 | 808 | 1048 |

A one queue
B one queue, starvation support
C two queues
D two queues, strict fifo
E two queues, starvation

# Jobs with Preemption



- Motivation: better support of parallel jobs or priorities
- Two virtual machines running on physical machine
  - first machine: standard jobs
  - second machine: privileged/parallel jobs
- Magrathea allows
  - several VMs running on a single computer
  - jobs submitted directly to VMs
- When job is started in privileged domain, Magrathea
  - suspends job in standard domain (if needed)
  - almost all cpu/memory resources are given to privileged domain (but standard is still running)
- Support of simulator
  - Magrathea installed on simulated machines too
  - sleep jobs must respect preemption

# Conclusion & Future Work

## New Grid simulator

- Inclusion of production resource management system
- New experiments: PBSPro (and other algorithms)
- Novel proposal with virtual machines
- New experiments: scheduling with Magrathea

## Future work

- Study: limits of the simulator
  - efficient scheduling algorithms needed
  - cannot use actual load on machines
  - monitoring issues
- New scheduler

# Standard submit script in simulator



```
#!/bin/bash
reduce=600 #reduce factor

sleep $(($SIMSLEEP/$reduce)) #gap in workload

sudo $SIMUSER qsub -q $SIMQUEUE
      #the same node requirements
      -l nodes=$SIMNODESL
      -l walltime=$(($SIMREQL/$reduce)) «EOF

      #sleep instead of real job
      sleep $(($SIMWALL/$reduce))
      EOF
```

# Preemption in simulator



```
reduce=600
sleep $(($SIMSLEEP/$reduce))
sudo $SIMUSER qsub -q sim$SIMQUEUE
      -l nodes=$SIMNODESL
      -l walltime=$(($SIMREQL/$reduce))«EOF

  sleeptime = $(($SIMWALL/$reduce))
  while ($sleeptime >0) do
    sleep $sleeptime
    #check long how job has been preempted
    sleeptime=`magrathea-preempted-time`;
  done
  EOF
```

# Weighted Response Time, SlowDown, and Wait Time



$$SA_j = reqResources_j \times (endTime_j - startTime_j)$$

$$TotalSA = \sum_{j \in Jobs} SA_j$$

$$SD = \frac{(endTime_j - submitTime_j)}{runtime_j}$$

$$WRT = \frac{\sum_{j \in Jobs}(SA_j(endTime_j - submitTime_j))}{TotalSA}$$

$$WSD = \frac{\sum_{j \in Jobs} SA_j \times SD_j}{TotalSA}$$

$$WWT = \frac{\sum_{j \in Jobs} SA_j \times (startTime_j - submitTime_j)}{TotalSA}$$