Automatic Deployment for ASP Environments

Pushpinder Kaur CHOUHAN

24 May 2004
GRAAL Group Meeting

Outline

1. Introduction - PhD Topic
2. Automatic Deployment for Hierarchical NES
3. Automatic Middleware Deployment Planning on Clusters
4. Deadline Scheduling with Priority for CSS on the Grid
5. Future Work

Outline

1. Introduction - PhD Topic
2. Automatic Deployment for Hierarchical NES
3. Automatic Middleware Deployment Planning on Clusters
4. Deadline Scheduling with Priority for CSS on the Grid
5. Future Work

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective: Maximize throughput (ρ)
  - Constraints are like Deadline, Priority etc.
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (homo & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constraints are like Deadline, Priority etc.
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (homo & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (homo & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (homo & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (homo & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (home & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (home & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (home & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (home & hetro) are given

Automatic Deployment for Application Service Provider Environments

- Platform
  - Resources - homogeneous and heterogeneous
  - Connecting topoloy - some hierarchy (e.g.Tree, Start etc...)
- Scheduling
  - Objective Maximize throughput (ρ)
  - Constrants are like Deadline, Priority etc...
- Deployment
  - Objective - To propose best hierarchy (maximize throughput)
  - First: find bottleneck, when hierarchy is fix
  - Second: generate best hierarchy when number of nodes (home & hetro) are given

What is deployment

A deployment is the distribution of a common platform and middleware across many resources.

- Software deployment maps and distributes a collection of software components on a set of resources. Software deployment includes activities such as releasing, configuring, installing, updating, adapting, de-installing, and even de-releasing a software system.
- System deployment involves two steps, physical and logical. In physical deployment all hardware is assembled (network, CPU, power supply etc), whereas logical deployment is organizing and naming whole cluster nodes as master, slave, etc.

What is deployment

A deployment is the distribution of a common platform and middleware across many resources.

- Software deployment maps and distributes a collection of software components on a set of resources. Software deployment includes activities such as releasing, configuring, installing, updating, adapting, de-installing, and even de-releasing a software system.
- System deployment involves two steps, physical and logical. In physical deployment all hardware is assembled (network, CPU, power supply etc), whereas logical deployment is organizing and naming whole cluster nodes as master, slave, etc.

Try to answer

- How middleware services can be best mapped to the resource platform structure?
- How to carry out an adapted deployment on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?

Try to answer

- How middleware services can be best mapped to the resource platform structure?
- How to carry out an adapted deployment on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?

Try to answer

- How middleware services can be best mapped to the resource platform structure?
- How to carry out an adapted deployment on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?

Try to answer

- How middleware services can be best mapped to the resource platform structure?
- How to carry out an adapted deployment on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?

Try to answer

- How middleware services can be best mapped to the resource platform structure?
- How to carry out an adapted deployment on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?

Outline

1 Introduction - PhD Topic

2 Automatic Deployment for Hierarchical NES

3 Automatic Middleware Deployment Planning on Clusters

4 Deadline Scheduling with Priority for CSS on the Grid

5 Future Work

Automatic Deployment for Hierarchical NES

Find bottleneck, when hierarchy is fix

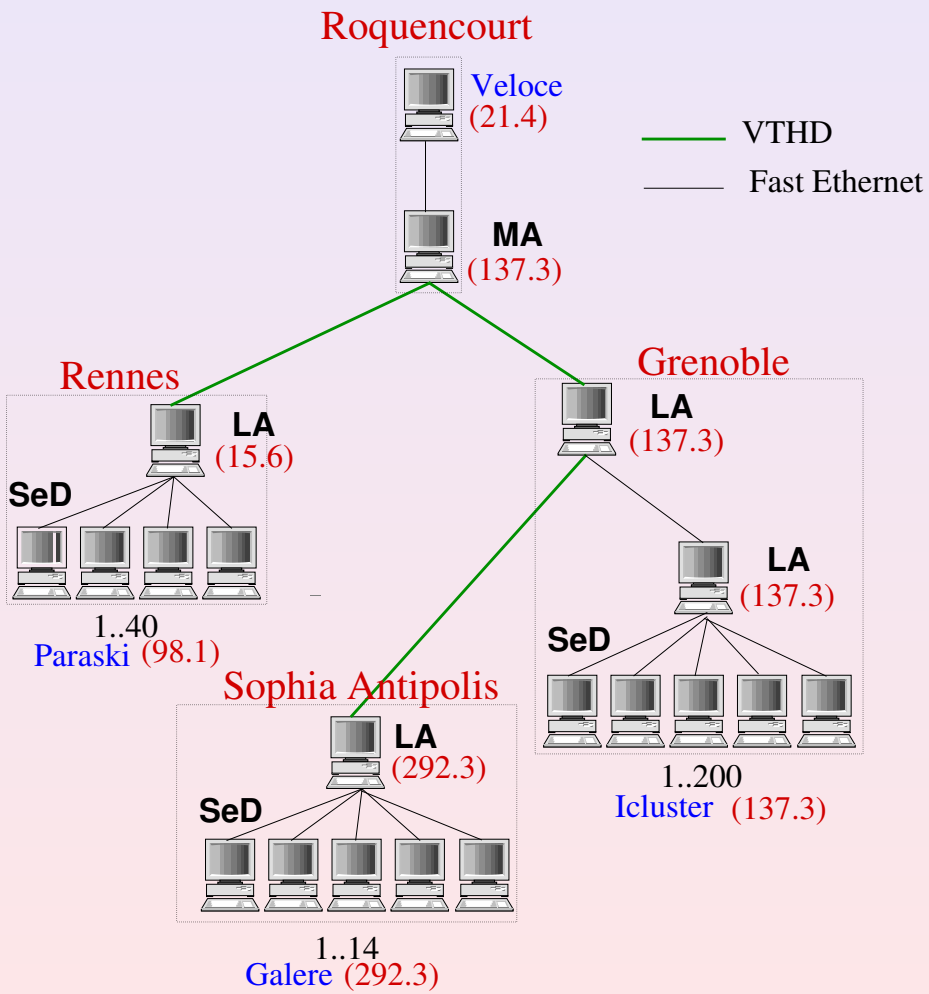
1. Calculate the throughput of each node
2. Find the bottlneck
3. Remove the bottleneck, if possible

Ref: Eddy Caron, Pushpinder Kaur Chouhan and Arnaud Legrand.Automatic Deployment for Hierarchical Network Enabled Server. The 13^{th} Heterogeneous Computing Workshop (HCW 2004), April 2004.

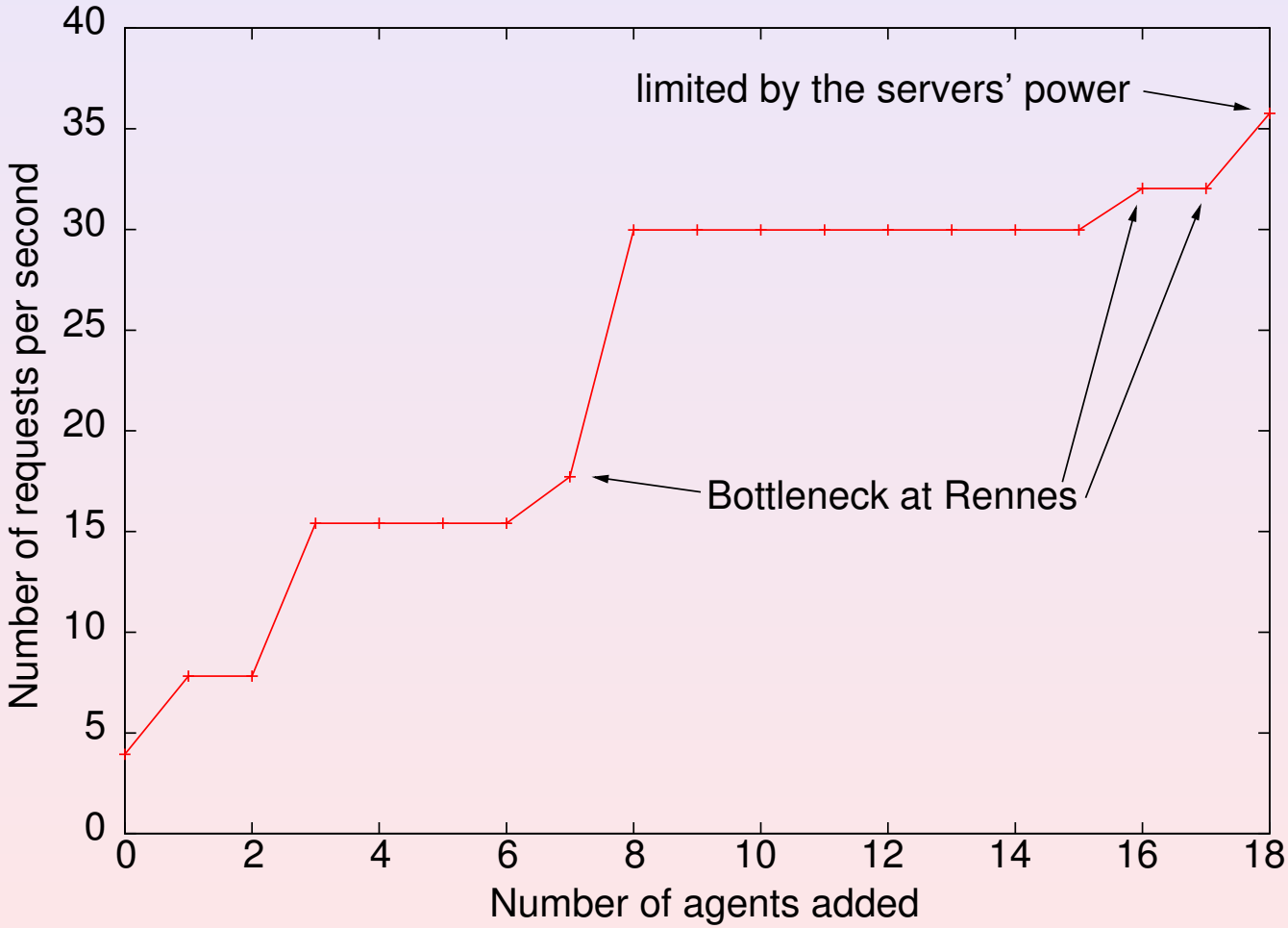
Experimental Platform - VTHD Network

- High speed network (2.5Gb/s) between INRIA research centers and several other research institutes.

Testbed


Roquencourt
Veloce
(21.4)
VTHD
Fast Ethernet
MA
(137.3)
Rennes
LA
(15.6)
SeD
1..40
Paraski (98.1)
Grenoble
LA
(137.3)
LA
(137.3)
SeD
1..200
Icluster (137.3)
Sophia Antipolis
LA
(292.3)
SeD
1..14
Galere (292.3)

Throughput of the platform



Outline

1 Introduction - PhD Topic

2 Automatic Deployment for Hierarchical NES

3 Automatic Middleware Deployment Planning on Clusters

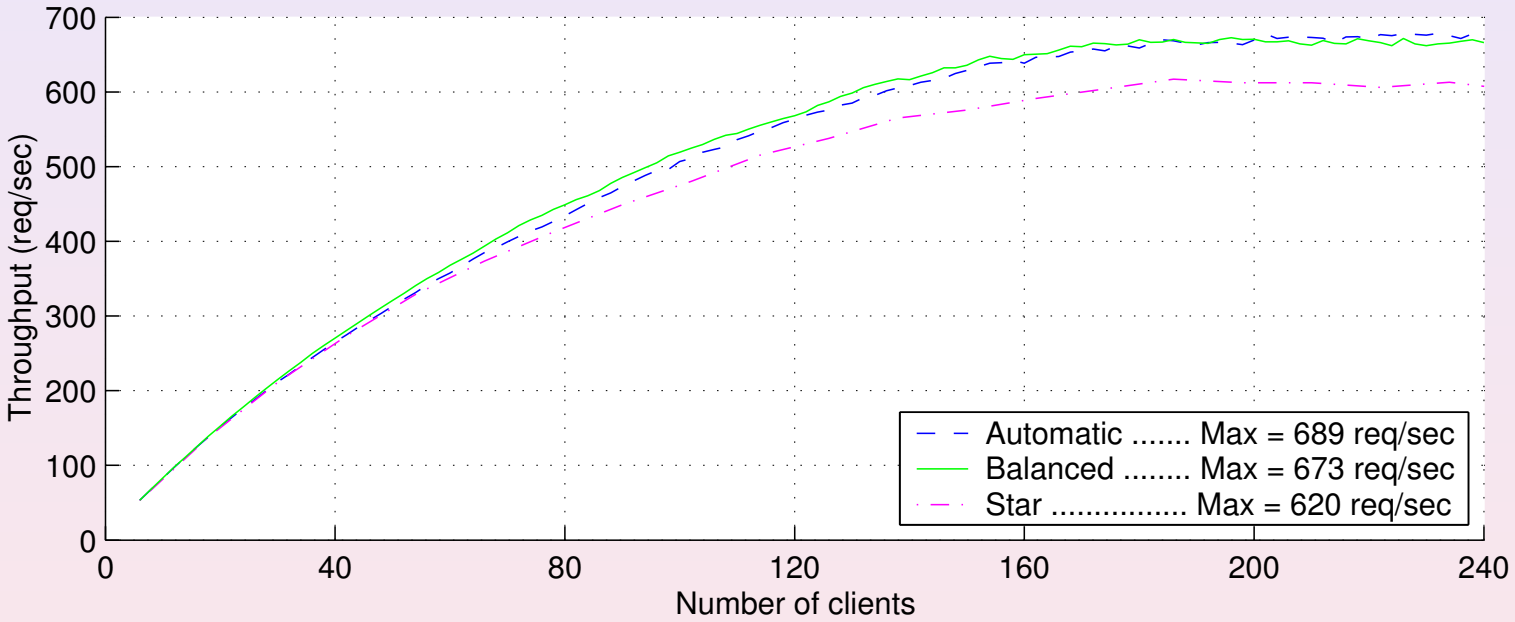
4 Deadline Scheduling with Priority for CSS on the Grid

5 Future Work

Hierarchicy construction (Homogeneous resources)

1: Calculate MSPA (Maximum Servers Per Agent)
2: Calculate $MAPA_l$ (Maximum Agents Per Agent)
3: Calculate required nodes $\sum_{k=0}^{l} MAPA^k + MAPA_l \times MSPA$
4: **if** $(n\_req_{less}) > (n\_req_{more})$ **then**
5: call Make_Hierarchy
6: call Node Removal Algorithm
7: **else**
8: call Make_Hierarchy
9: call Node Addition Algorithm
10: **end if**

Comparison of automatically-generated hierarchy



Ref: Eddy Caron, Pushpinder Kaur Chouhan and Holly Dial.Automatic Middleware Deployment planning for clusters RR 2005-26

Outline

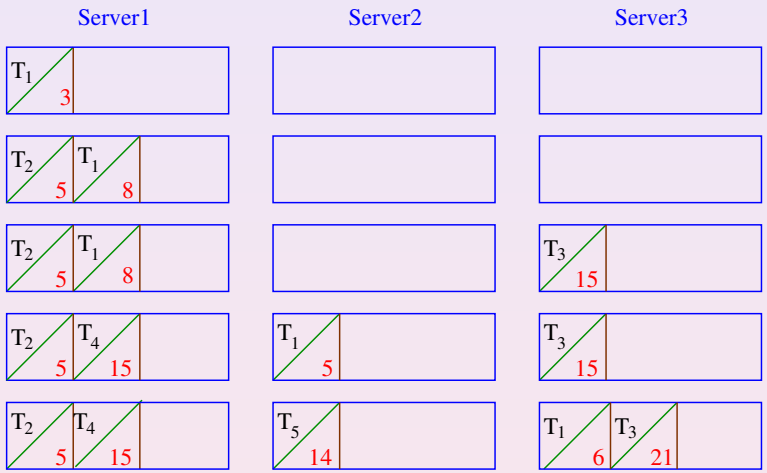
1. Introduction - PhD Topic
2. Automatic Deployment for Hierarchical NES
3. Automatic Middleware Deployment Planning on Clusters
4. Deadline Scheduling with Priority for CSS on the Grid
5. Future Work

Priority and Deadline Mechanism

1: **repeat**
2: **for all** server S_i **do**
3: **if** can_do(S_i, T_a) **then**
4: $T_{aS_i} = \frac{W_{send}}{F_{Bd}} + \frac{W_{recv}}{P_{recv}} + \frac{W_{aS_i}}{F_{S_i}}$
5: **end if**
6: **if** $T_{aS_i} < TD_a$ **then**
7: count_fallback_tasks$(T_a, T_{aS_i}, TP_a, TD_a)$
8: **if** $TF_{aS_i} < TD_a$ **then**
9: best_server(S_i,best_server_name)
10: **end if**
11: **end if**
12: **end for**
13: task_submit(best_server_name,task_name)
14: Re-submission(task_name)

Priority and Deadline Mechanism

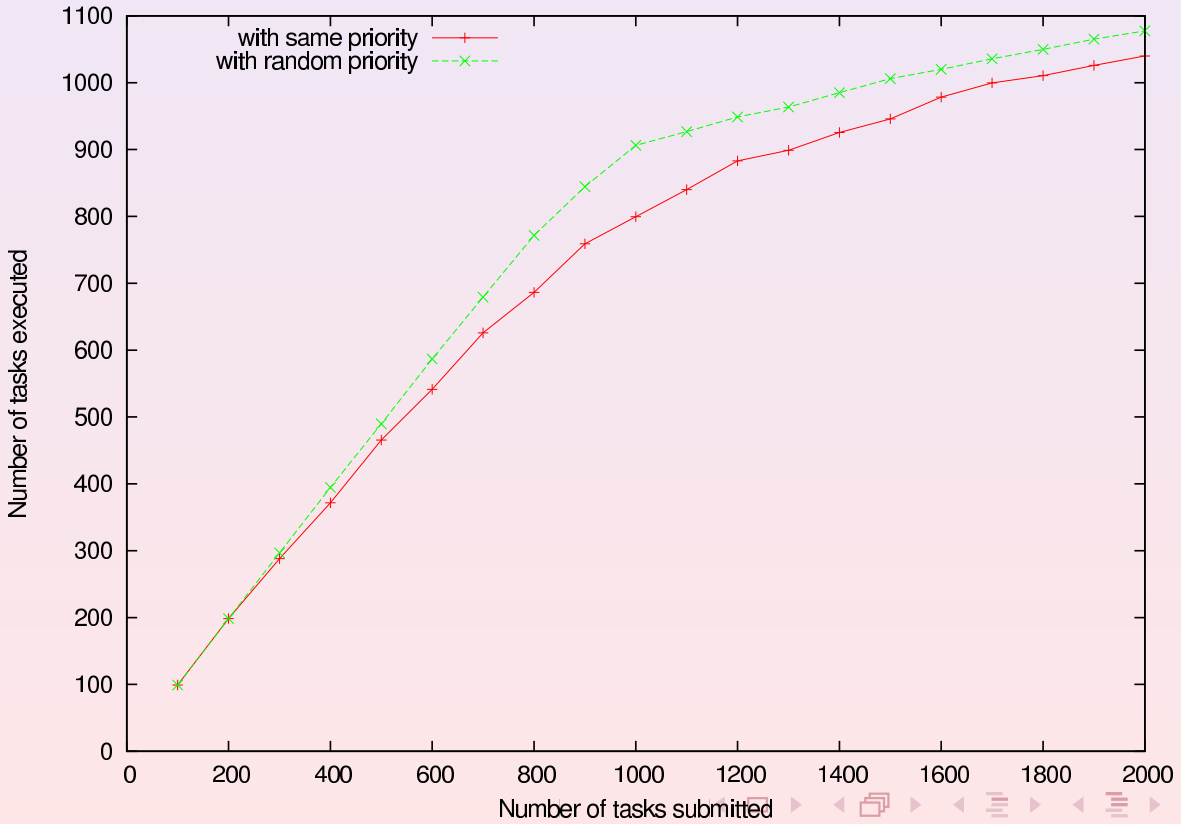
			Exec. time on server		
Task	Priority	Deadline	S_1	S_2	S_3
1	3	15	3	5	6
2	5	10	5	12	9
3	2	30	11	20	15
4	4	20	10	np	17
5	5	15	12	14	np



Ref:Eddy Caron, Pushpinder Kaur Chouhan and Frederic Desprez. Deadline scheduling with priority for client-server systems on the Grid. Grid Computing 2004. IEEE International Conference On Grid Computing. Super Computing 2004, October 2004.

Priority based tasks are executed without fallback mechanism

Testbed 100 servers, priority range 1-10, deadline $= 5 \times T_{aS_i}$



Outline

1 Introduction - PhD Topic

2 Automatic Deployment for Hierarchical NES

3 Automatic Middleware Deployment Planning on Clusters

4 Deadline Scheduling with Priority for CSS on the Grid

5 Future Work

Future work

- Goal is to develop, deployment planning and re-deployment algorithms, for middleware on heterogeneous clusters and Grids
- Scheduling (Grid, Clusters . . .)