

How should you Structure your Hierarchical Scheduler?

Pushpinder Kaur CHOUHAN, Holly DAIL,
Eddy CARON, and Frédéric VIVIEN

HPDC - The 15th IEEE International Symposium on High
Performance Distributed Computing

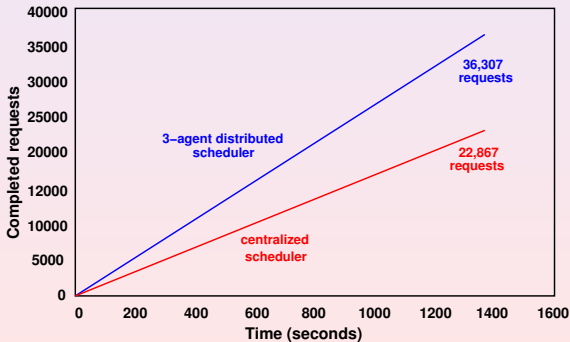
What is Deployment

A **deployment** is the mapping 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.

Problem Statement

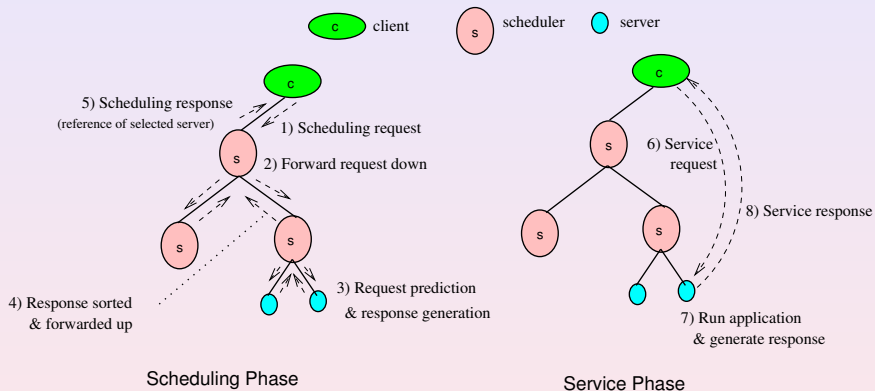
- How to carry out an adapted deployment of middleware services on a cluster with hundreds of nodes?
- Which resources should be used?
- How many resources should be used?
- Should the fastest and best-connected resource be used for middleware or as a computational resource?



Find an optimal deployment of agents and servers onto a set of resources.

- **optimal** deployment is the deployment that provides the maximum throughput.
- ρ is the throughput of the platform calculated as the completed requests per second.

Platform deployment architecture and execution phases



Lemma

The completed request throughput ρ of a deployment is given by the minimum of the scheduling request throughput ρ_{sched} and the service request throughput $\rho_{service}$.

$$\rho = \min(\rho_{sched}, \rho_{service})$$

- ρ_{sched} the scheduling throughput in requests per second, is defined as the rate at which requests are processed by the scheduling phase.
- $\rho_{service}$ the service throughput in requests per second, is defined as the rate at which requests finish the service response phase.

Lemma

The scheduling throughput ρ_{sched} is limited by the throughput of the agent with the highest degree.

- Scheduling throughput is controlled by slowest agent
- Slowest agent is the one with highest degree

Lemma

The service request throughput $\rho_{service}$ increases as the number of servers included in a deployment increases.

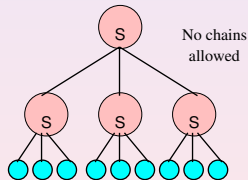
Service requests are only sent to a single server.

Complete Spanning D-ary tree

A **complete d-ary tree** is a tree in which every level, except possibly the deepest, is completely filled. All internal nodes except one have a degree, or number of children, equal to d ; the remaining internal node is at depth $n - 1$ and may have any degree from 1 to d .

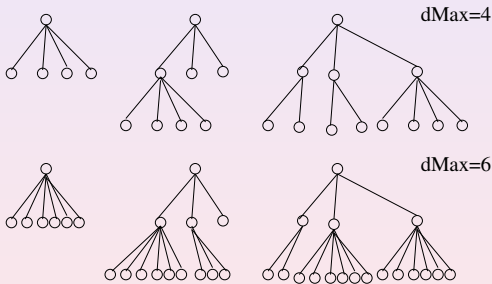
A **spanning tree** is a connected, acyclic subgraph containing all the vertices of a graph.

A **complete spanning d-ary tree (CSD tree)** is a tree that is both a complete d-ary tree and a spanning tree.



$dMax$ set is the set of all trees for which maximum degree is equal to $dMax$.

- Examples : 3 trees from dMax set 4 and dMax set 6.



Theorem


The optimal throughput ρ of any deployment with maximum degree $dMax$ is obtained with a CSD tree.

- By Lemma1 $\rho = \min(\rho_{sched}, \rho_{service})$
- By lemma2 ρ_{sched} is limited by agent with maximum degree
- By Lemma3 $\rho_{service}$ increases with $|S|$

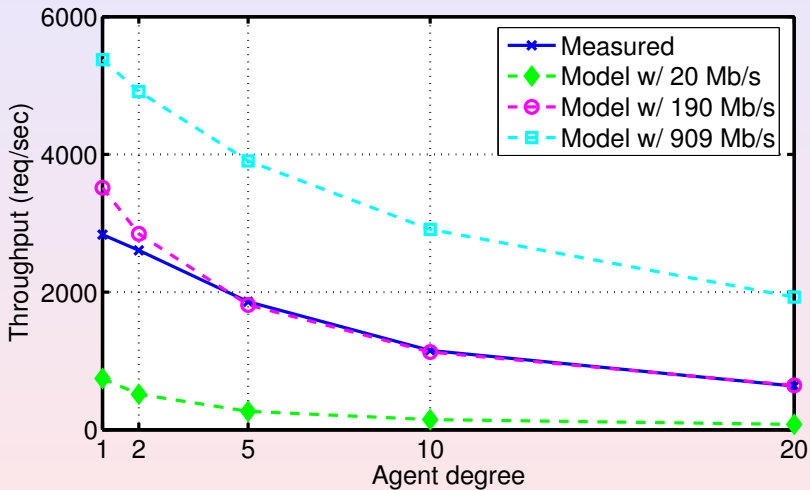
Corollary

The complete spanning d -ary tree with degree $d \in [1, |\mathbb{V}| - 1]$ that maximizes the minimum of the scheduling request and service request throughputs is an optimal deployment.

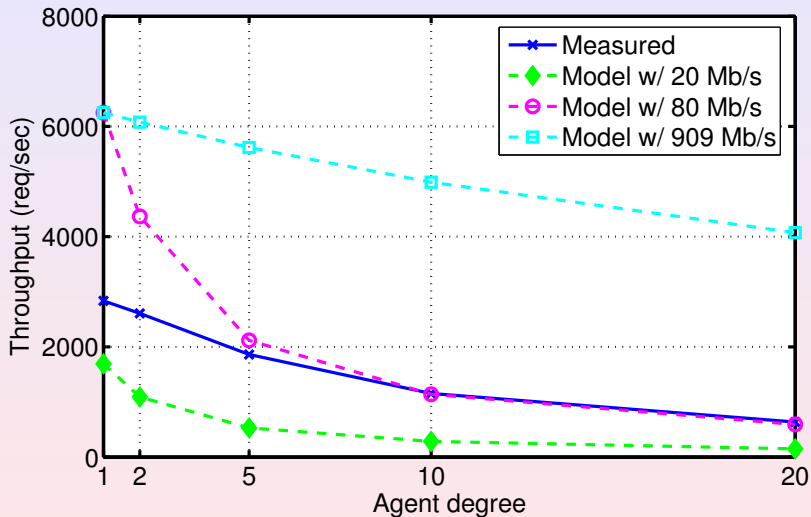
- Test all possible degrees $d \in [1, |\mathbb{V}| - 1]$
- Select MAX $\min(\rho_{sched}, \rho_{service})$

- **Software:** GoDIET is used to deploy , an hierarchical Problem Solving Environment.
(<http://graal.ens-lyon.fr/DIET>)
- **Job types:** DGEMM, a simple matrix multiplication (BLAS package).
- **Workload:** steady-state load with 1 - 200 client scripts (each script launches requests serially)
- **Resources:** dual AMD Opteron 246 processors @ 2GHz, each with cache size of 1024KB, 2GB of main memory and a 1Gb/s Ethernet
 - **Lyon** cluster - 55 nodes
 - **Sophia** cluster - 140 nodes

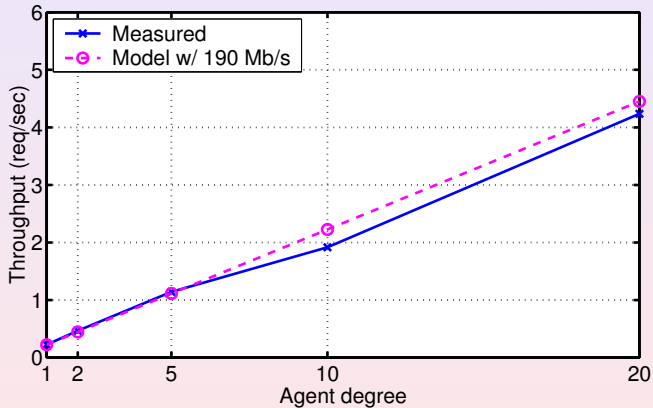
Throughput validation - Serial Model (DGEMM 10)



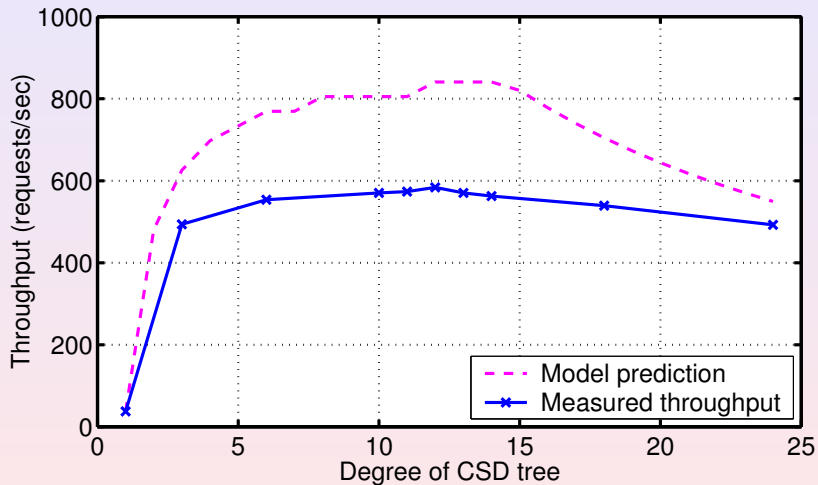
Throughput validation - Parallel Model (DGEMM 10)



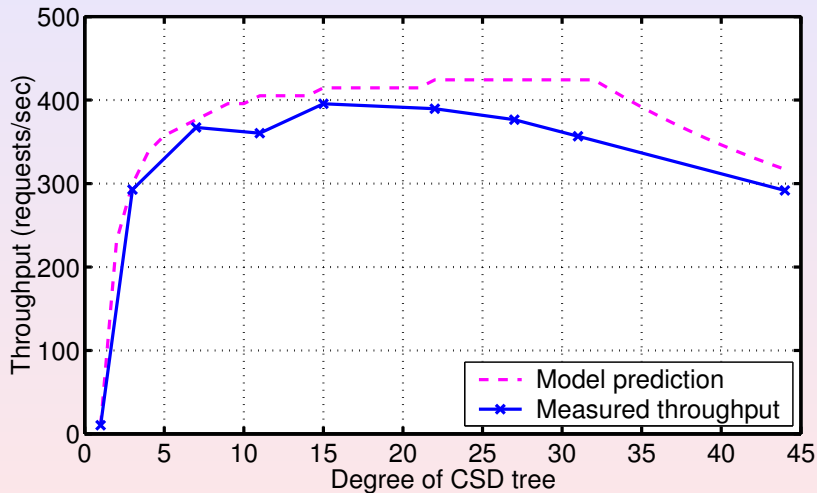
Throughput validation - DGEMM 1000, bandwidth 190Mb/s



Deployment selection validation - DGEMM 200, 25 Nodes



Deployment selection validation - DGEMM 310, 45 Nodes



Summary Table

DGEMM Size	Nodes	Best	Selected	Model	Star	Tri-ary
10	21	1	1	100.0%	22.4%	50.5%
100	25	2	2	100.0%	84.4%	84.6%
200	45	3	8	86.1%	40.0%	100.0%
310	45	15	22	98.5%	73.8%	74.0%
1000	21	20	20	100.0%	100.0%	65.3%

- Conclusion
 - Determines how many nodes should be used
 - Designs the hierarchical organization
 - Proved an optimal deployment is a CSD tree
 - Deployment prediction is easy, fast and scalable
 - Experiments validated the model
- Future work
 - Develop re-deployment approaches
 - Dynamically adapt the deployment to workload levels
 - Develop deployment planning and re-deployment algorithms for middleware on Grids