Market-based Resource Allocation for Grid Computing: A Model and Simulation

by

Jacek Gomoluch and Michael Schroeder

Agents@City
Department of Computing
City University, London

Outline of the Presentation

- Motivation
- Simulation model: Actors in the system
- Simulation model: Interactions in the system
- Description of the resource allocation protocols
- Simulation parameters
- Simulation results
- Conclusions
- Real implementation
- Related work
- Future work

Motivation I

- There is a growing interest in coupling geographically distributed resources, resulting in **Computational Grids**
- One important aspect in such environments is **resource management**: how to **allocate computational resources**
- Over the past years, **economic approaches** to resource allocation have been developed
- These satisfy some **basic requirements** for the Grid:
  - they are naturally **decentralised** as decisions about whether to consume or provide resources are taken locally by the users and service providers, respectively
  - the use of **currency** provides **incentives** for service providers to participate and for users to behave responsibly

Motivation II

- A number of systems have been built which use **market-based policies** to allocate **computational resources.** However: **performance** of these policies (response time, timeliness of delivery, etc.) has not been sufficiently studied for **Grid settings**.
- The performance of a policy will depend on many factors, including
  - **demand and supply**
  - **communication delays: bandwidth and latency**
  - **resource heterogeneity**
  - **variation of sizes and priorities of tasks**

Agents@City - Department of Computing

Motivation III

- We examine the performance of market-based resource allocation protocols by **simulations** and by **experiments** in a real distributed system
- **Expected benefits:**
  - results will allow choice of the most suitable resource allocation protocol for a given scenario
  - the market-based protocols which are examined can be deployed in an open, economic environment - provided the required security measures are in place
- In this paper we evaluate two market-based protocols (and a conventional one) via **discrete-event simulation** for various scenarios

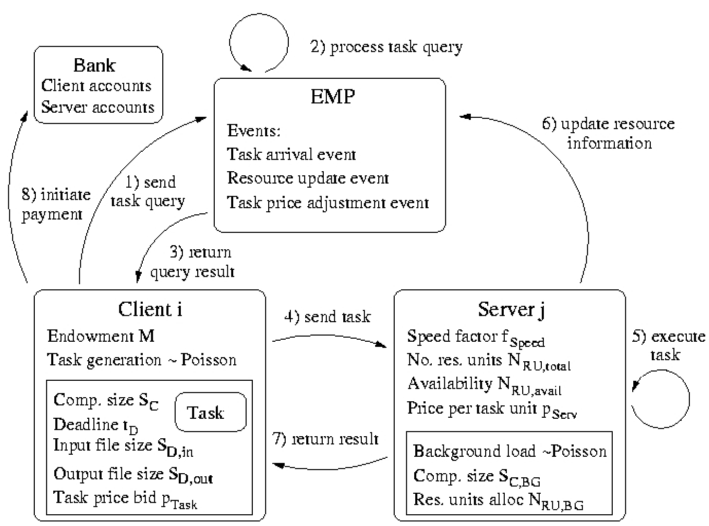
Motivation IV

- Simulation enables us to model **almost any type and number of resources**, parameters determining message delays, task creation, speed of servers, etc. can easily be adjusted, resulting in a rich parameter space
- Our simulation model is backed by **well-established probability distributions** and it is currently being verified by real experiments
- We study the following **resource allocation protocols**:
  - Continuous Double Auction Protocol (CDA)
  - Proportional Share Protocol (PSP)
  - Round Robin Protocol (RR)

Assumptions

- As our objective is to assess the **performance** of the resource allocation policies, we can make the following simplifying assumptions:
  - the actors in the system are **benevolent** and do not attempt to cheat. Hence, **security issues** can be **neglected**
  - we consider the market as a **tool** for the efficient allocation of resources, therefore **accounting** and **payment procedures** can be **neglected**

Simulation Model I



Agents@City - Department of Computing

Simulation Model II

- Clients
  - generate tasks whose arrivals are modelled by a **Poisson process** (which is suitable for describing user session arrivals on the Internet, and also the arrival of workload in supercomputing centres)
- Tasks
  - **independent** tasks / "embarrassingly parallel" applications
  - Parameters: **computation size** [MFLOPS*sec], size of input / output data [bytes], **weight** (priority), **deadline**, **price bid** [per processing time, MFLOPS*sec]
  - **no preemption** (i.e. migration) of tasks that have started

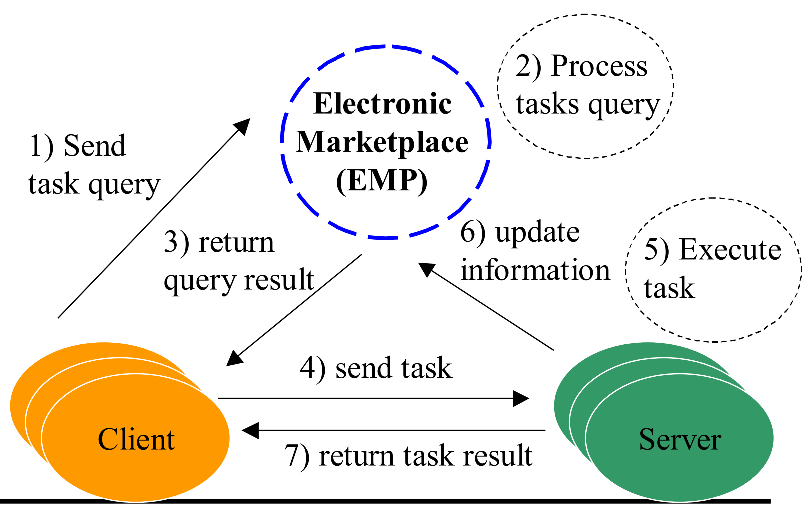
Simulation Model III

- Servers (computational resources)
  - Parameters: **speed**, number of **processing units** (either CPU power of a whole processor or a fraction of it)
- Resource scheduling policies at the Servers
  - 1) allocate all available resource units of the Server
  - 2) proportional sharing of the Server resource
- Background load (e.g. processes of local users)
  - tasks executing on the resources which are **not under the control of the marketplace**, Poisson arrival process
  - occupy one or more processing units of a resource
  - is given **priority**, may even suspend executing tasks

Simulation Model IV

- **Electronic Marketplace (EMP)**
  - provides facilities for the Servers to **advertise** resources and for the Clients to **search** for a suitable offer
  - **parameters** that are published include: resource speed, number of available processing units, resource price
- **Communication Model**
  - actors are distributed over the Internet, therefore communication delays need to be considered
  - two parameters: latency and bandwidth
  - simplifying assumption in this paper: just latency
  - lognormal distribution (supported by empirical data)

Interactions in the System



Agents@City - Department of Computing

Continuous Double Auction (CDA)

- Working principle:
  - when a task arrives at the EMP it searches all available resource offers and returns the '**best**' match
  - **if demand>supply**: tasks are **prioritised** according to their price bids
  - tasks may **adjust** their **prices** at **periodic intervals**
  - Servers may adjust their prices at any time
- Reasons for examining CDA:
  - the studied scenario requires a **double** auction, i.e. a many-to-many protocol
  - the auction has to be **continuous** (not periodic)

Proportional Share Protocol (PSP)

- Working principle:
  - tasks are sent to the resource where they will execute fastest - this depends on the tasks already executing there
  - resources are split: the resource shares allocated to tasks are **proportional** to their price bids
- Reasons for examining PSP:
  - similar policies have been deployed for scheduling tasks in computational clusters
  - PSP could be superior to CDA in certain situations, e.g. for heterogeneous resources

Round Robin (RR)

- Working principle:
  - an arriving task is allocated to the **next available** resource
  - **not market-based:** no pricing used, hence there are no priorities
- Reasons for examining RR:
  - simplicity and thus little effort for processing queries
  - for certain situations its performance will be as good as that of more complex protocols

Simulation: Scenarios

- **Identical tasks**
  - independent tasks with the **same priority**
  - all tasks have the same price bids
- **Weighted tasks**
  - independent tasks with **different priorities/weights**
  - uniform distribution of task weights [0;10]
  - task price bids are **proportional** to the task weights

Simulation: Performance Metric

- The objective is to provide the best possible service to the users of the resources
  - in our simulations we use the mean **completion time** of the independent tasks as performance metric
  - for **tasks with different priorities** we use the **weighted completion time** (task priority * its completion time)
- Other relevant performance metrics which are currently being studied - though not in this paper - include:
  - **makespan**, i.e. completion time of a larger application which consists of several smaller tasks
  - **'value delivered to the user'** by timely execution of the tasks

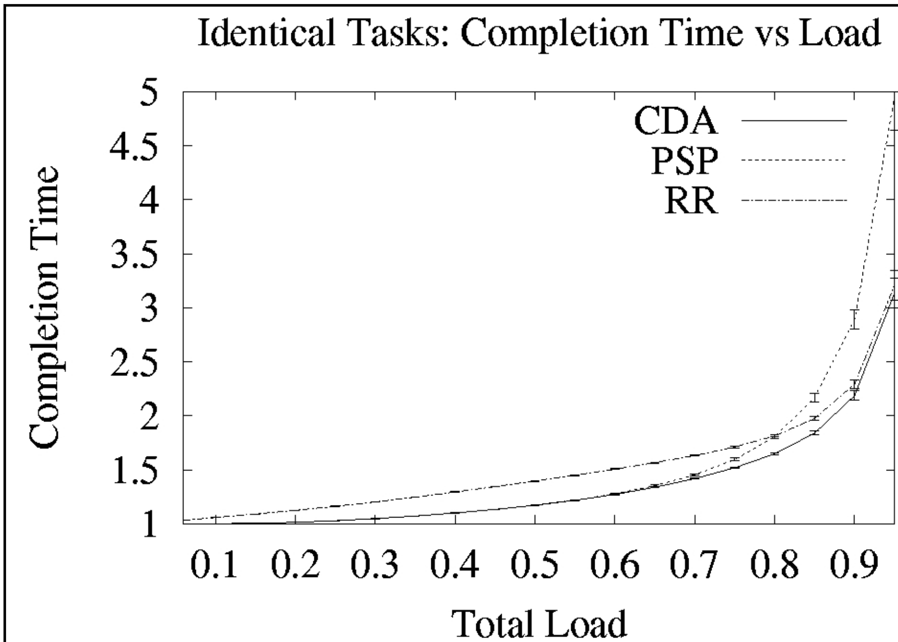
Simulation: Default Parameters

- Total duration of the experiments [time units]: 1300
- Initial period without measurements [time units]: 100
- Final margin without measurements [time units]: 200
- Repetitions of the experiment: 40 (**95% confidence interval shown**)
- Number of Clients: 1; number of Servers: **10**
- Tasks: Size: **10.0**, price: 100.0, no deadlines, data sizes neglected
- Server resources: Number of processing units: **10**, speed: **1.0**
- Background tasks: size: **10.0**; **1** processing unit allocated at a time
- Average load [% of system capacity]: **40% arriving tasks** + **40% background load**. Results in **80% utilisation**
- Communication delays: zero
- No adjustment of the task price bids

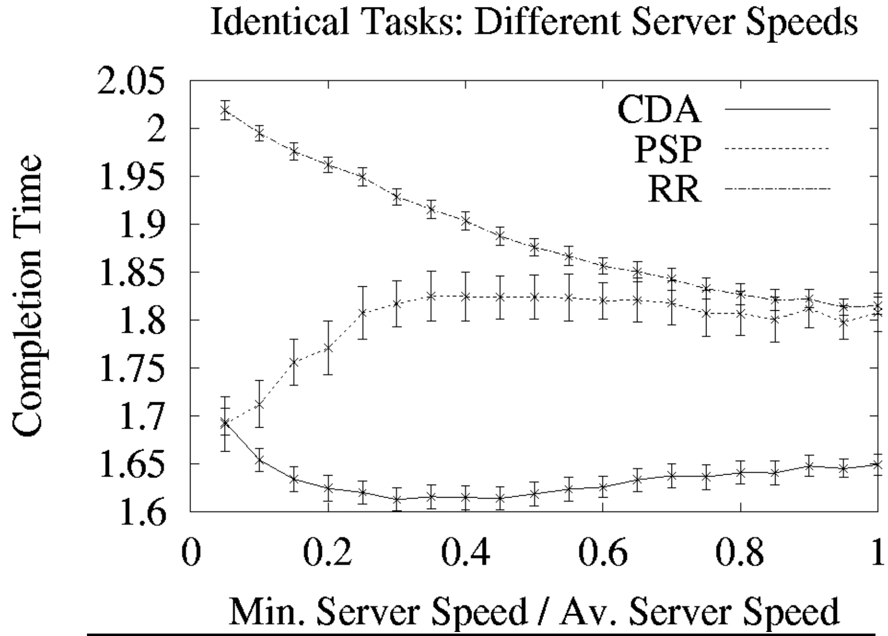
Agents@City - Department of Computing

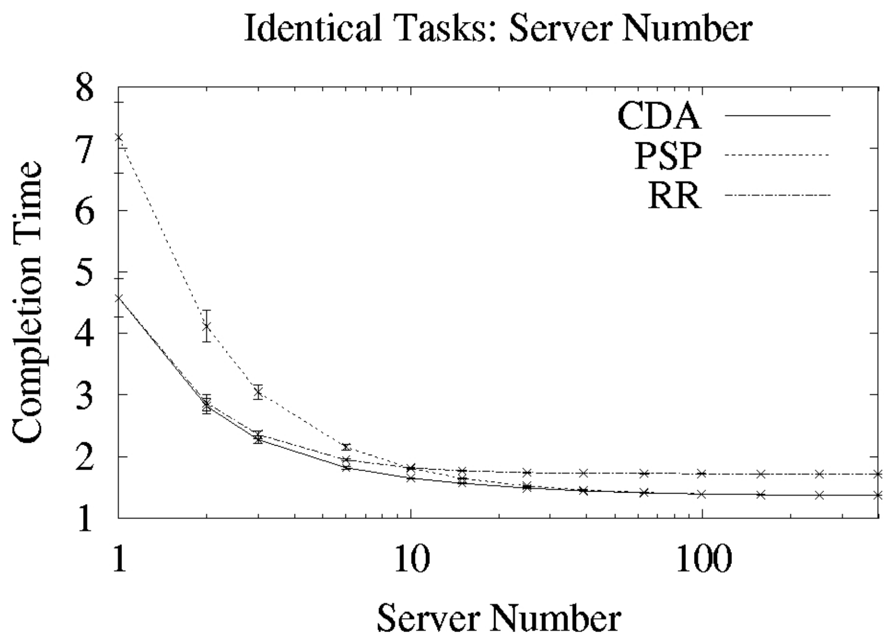
Simulation: Variation of Parameters

- Overall load in the system
  - average load as % of the overall capacity in the system
- Number of Servers
  - number of Servers is low in a cluster, but high in a Grid
- Heterogeneity of resources
  - different speed of computational resources
  - resources are likely to be similar in a cluster, but heterogeneous in a Grid
- Communication delays
  - communication delays are low in a cluster, high in a Grid



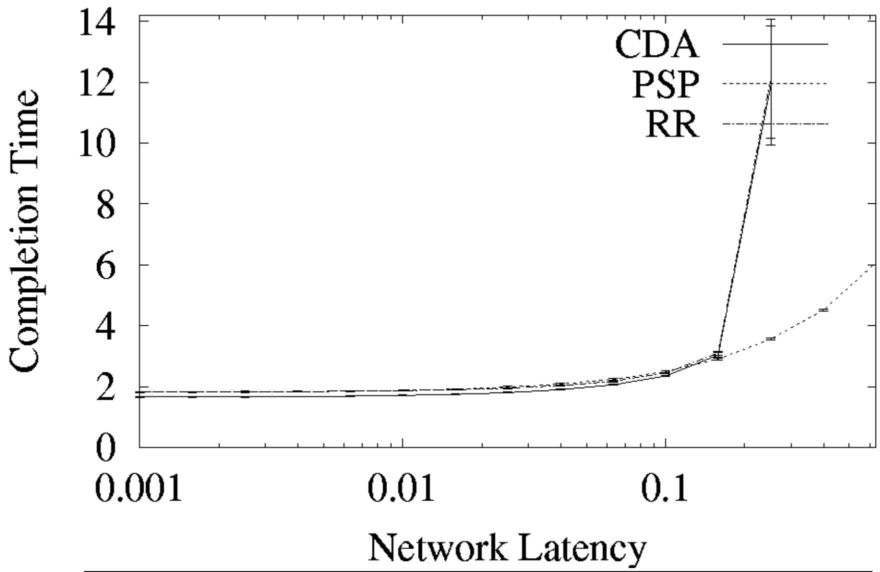
Agents@City - Department of Computing





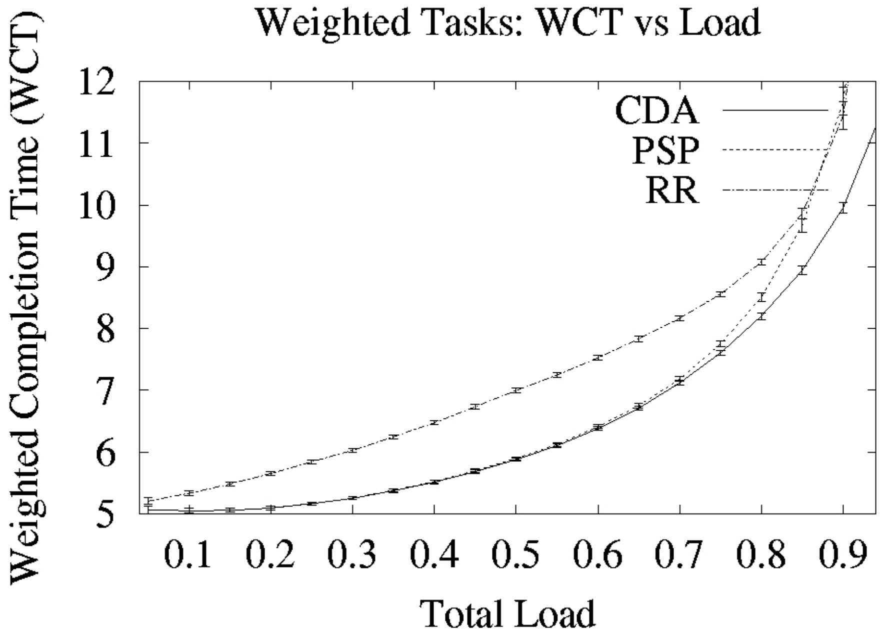
Agents@City - Department of Computing

Identical Tasks: Network Latency

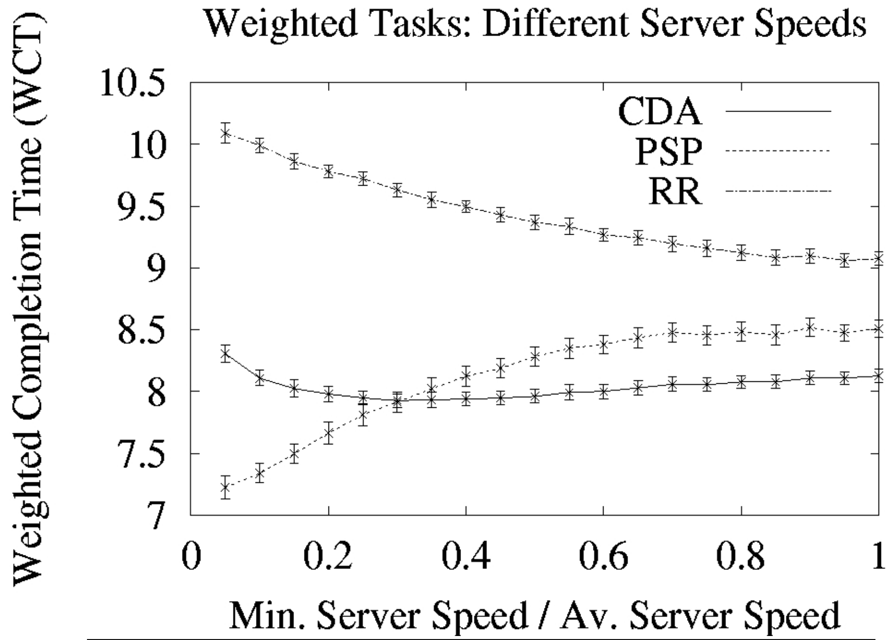


Agents@City - Department of Computing

Weighted Tasks: WCT vs Load



Agents@City - Department of Computing



Conclusions

- **Homogeneous resources (e.g. in a Linux cluster)**
  - Continuous Double Auctions (CDA) perform best
  - however: for low load Round-Robin(RR) and Proportional Share (PSP) will perform equally well
- **Resources of different quality (as in a Grid)**
  - Round-Robin will be worse than the two market-based protocols (due to their greedy strategy and prioritisation of tasks with higher weights)
  - CDA performs best in most cases, but PSP may outperform it for high resource heterogeneity or high communication delays
  - PSP degrades for high load

Real Implementation

- **Verification** of the simulations in a real environment
- Evaluation of market-based protocols for applications from **Bioinformatics** (two types: 'embarrassingly parallel' applications; applications with task dependencies)
  – PSIMAP: determining protein interactions
- We developed a Java-based Framework. Features include:
  – **JADE** as middleware
  – supports parallel execution of Java code on remote resources
  – **measures performance** of remote resources [MFLOPS]
  – **multiple** communication protocols (HTTP, RM
  – ships code (Java classes) over **any communication protocol**

Related Work

- Theory: Ferguson, Wellman, Sandholm
- Systems: Different resource allocation strategies
  – **State-based** vs **model-based**:Are the allocations based on the current snapshot of the system, which is expensive to obtain, or according to a model, which predicts the system state and which may be inaccu-rate ?
  – **Static** vs **preemptive**: Are tasks assigned to a host once and then stay there, or can they migrate if it turns out at a later stage that it is advantageous to leave the machine ?

Related Work: Systems

state-based & static	state-based & preemptive
• SPAWN (Waldspurger) • Dynasty (Backschat) • POPCORN, MAJIC (Nisan) • G-Commerce (Wolski)	• Traveler (Wims/Xu) • D'Agents (Bredin) • System by Keren/Barak
model-based & static	**model-based & preemptive**
• Challenger (Chavez) • Nimrod-G (Buyya)	• Work by Harchol-Balter

Market-based Systems

Agents@City - Department of Computing

Related Work: Performance evaluation of economic policies

- **POPCORN (Nisan et al.)**
  - comparison of several auction protocols (Vickrey, Double,…) for allocation of independent tasks in a Grid
  - metrics: price stability, social efficiency
- **G-Commerce (Wolski et al.)**
  - commodity market vs auctions for a Grid setting
  - metrics: price stability, resource utilisation
- **Work by Ferguson et al.**
  - Comparison of several decentralised auction protocols to conventional policies; independent tasks in a cluster
  - metrics: task completion time

Agents@City - Department of Computing

Future Work

- **Other scenarios**
  - performance metrics other than completion time; e.g. deadline misses, 'value delivered to the user'
  - scenario: tasks with dependencies
- **Improved resource allocation policies**
  - examine preemptive vs non-preemptive policies; communication costs
- **Real Implementation**
  - experiments to verify simulation results
  - scenario tasks with dependencies

? ? ? ? ? ?

Thanks for your attention !

Any questions ?

? ? ? ? ? ? ? ?

Agents@City - Department of Computing