Agent-Based Grid Load-Balancing

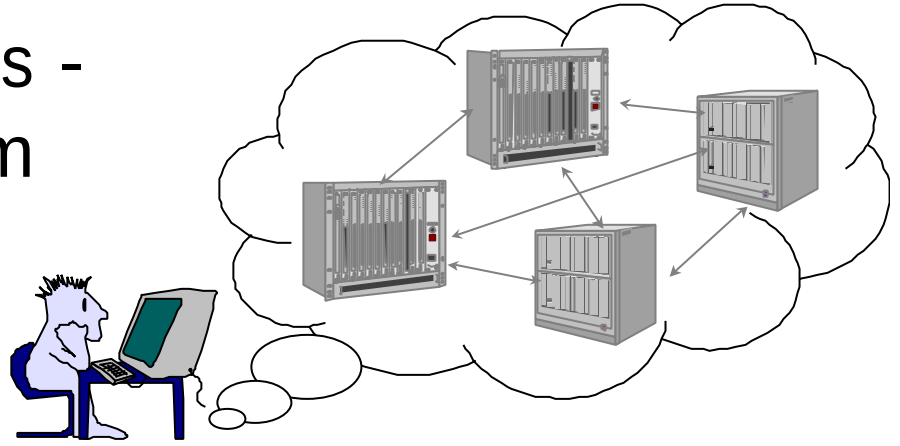
Daniel P. Spooner
University of Warwick, UK
Junwei Cao
NEC Europe Ltd., Germany

Outline

- Grid Computing & Middleware
- Agent-Based Methodology
- Distributed Service Discovery
- Grid Load-Balancing
- Experimental Results
- Conclusions & Future Work

Grid Computing

- Computational Grids - blueprint for a new computing infrastructure
- Data/Service/Community Grids - large-scale resource sharing in VOs
- Grid/Web Services - distributed system and application integration



Grid Middleware

- Resource Management & Scheduling
- Information Services (Monitoring & Discovery)
- Data Management and Access
- Application Programming Environments
- Security, Accounting, QoS ……

Condor, LSF, Ninf, Nimrod, ……

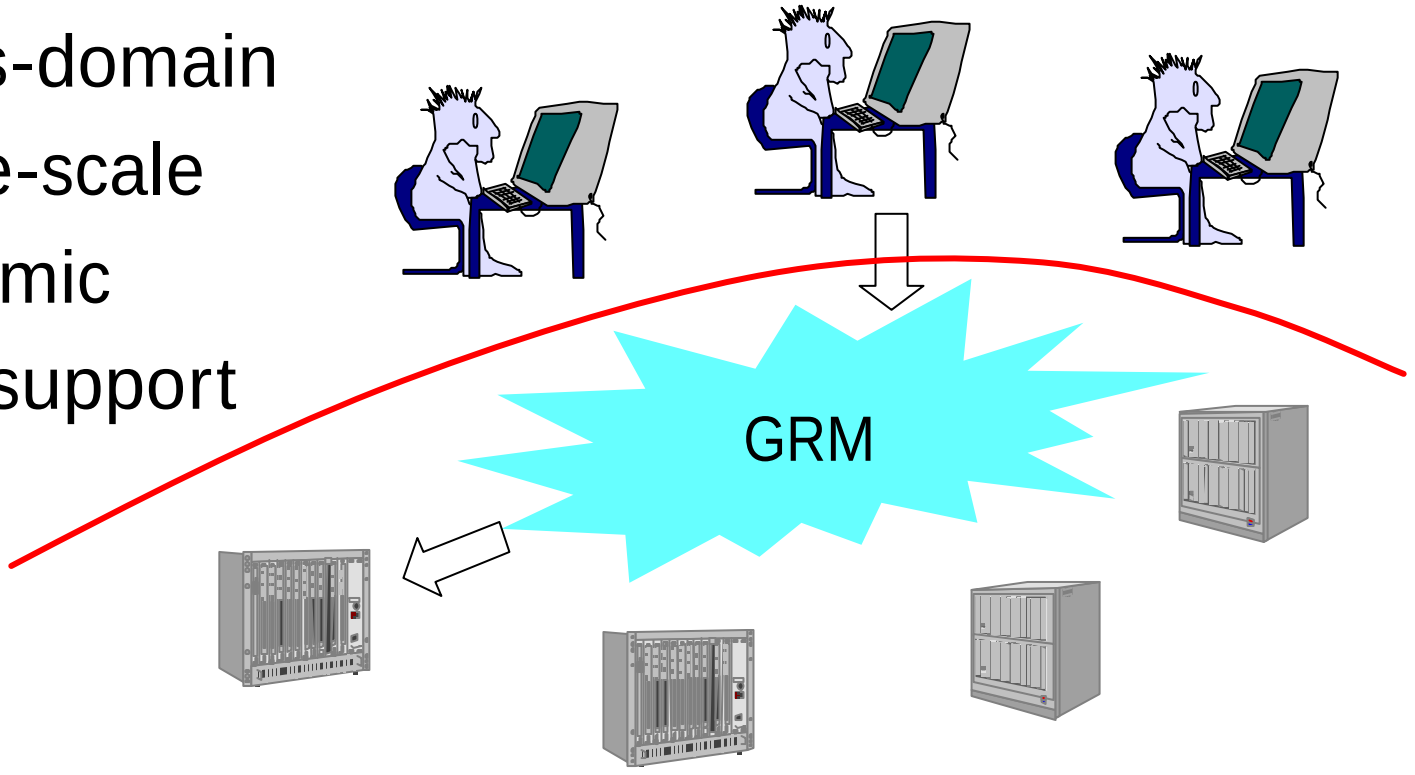
Globus, Legion, DPSS, ……

Java/Jini, CORBA, Web Services, ……

Grid Resource Management

Key challenges:

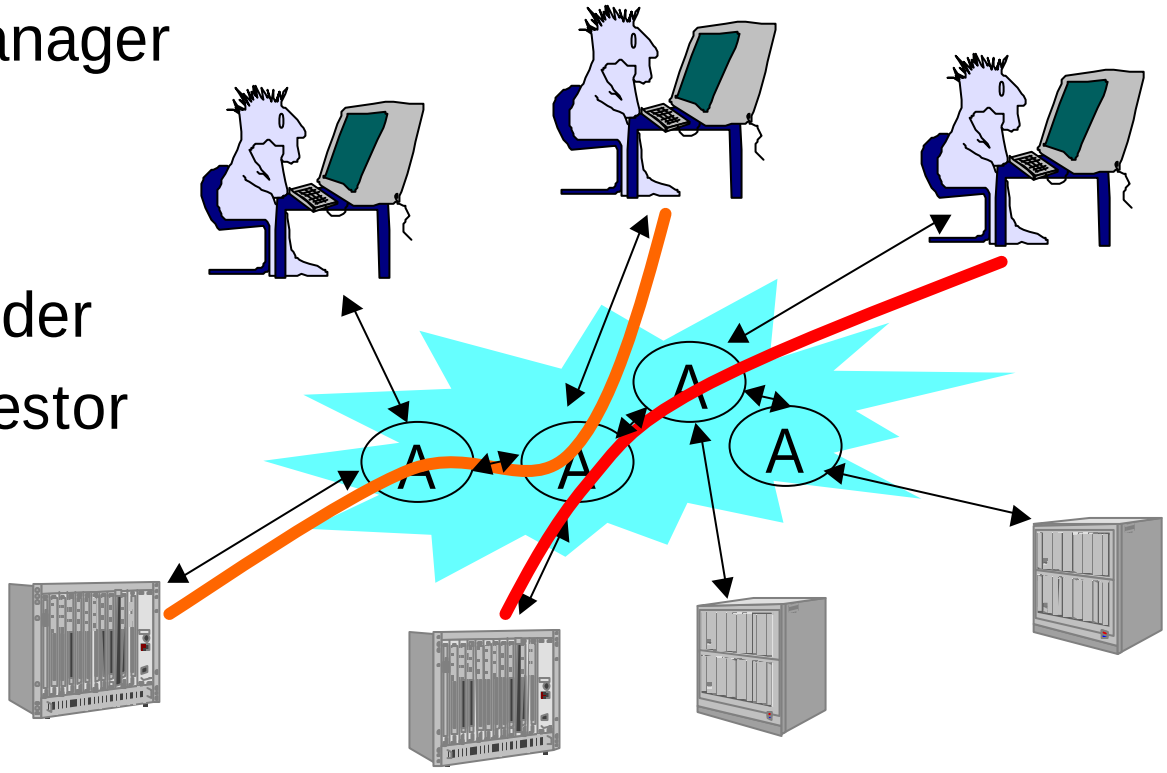
- Cross-domain
- Large-scale
- Dynamic
- QoS support



Agent-Based Methodology

An agent is:

- A local grid manager
- An user agent
- A broker
- A service provider
- A service requestor
- A matchmaker



Local Management

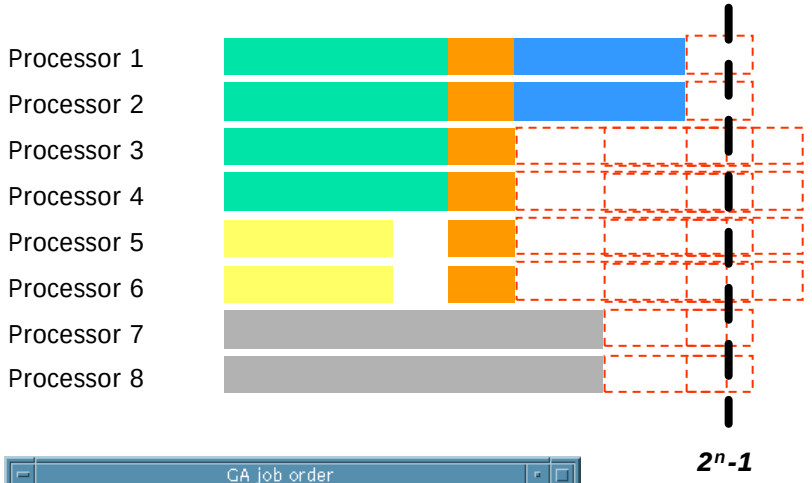
Performance Prediction

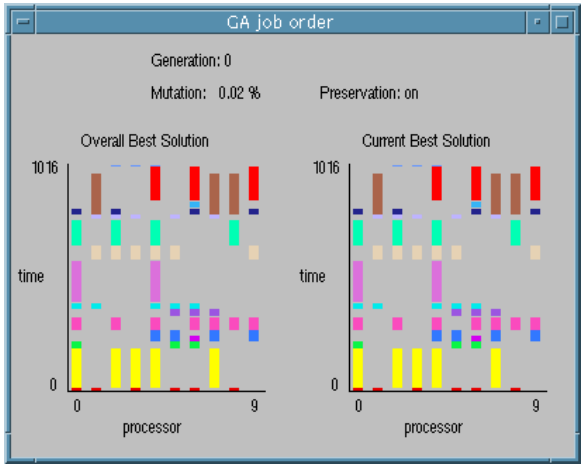
- Task models
- Hardware models

FIFO Algorithm

Genetic Algorithm

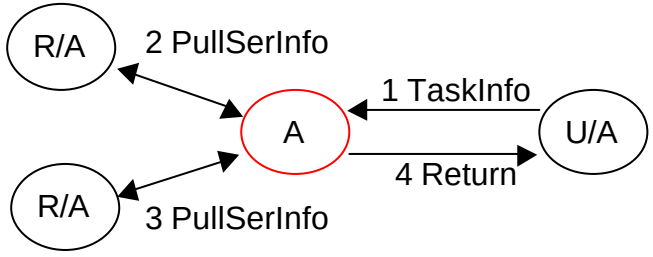
- Heuristic & Evolutionary
- Near-optimal on makespan, deadlines and idletime.



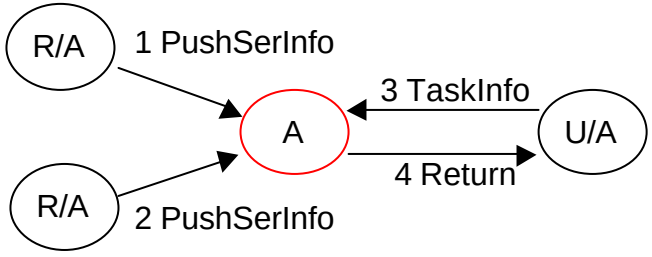


Service Discovery

- Pure data-pull
- No advertisement
- Full discovery
- Efficient when service change more quickly



- Pure data-push
- Full advertisement
- No discovery
- Efficient when requests arrive more frequently



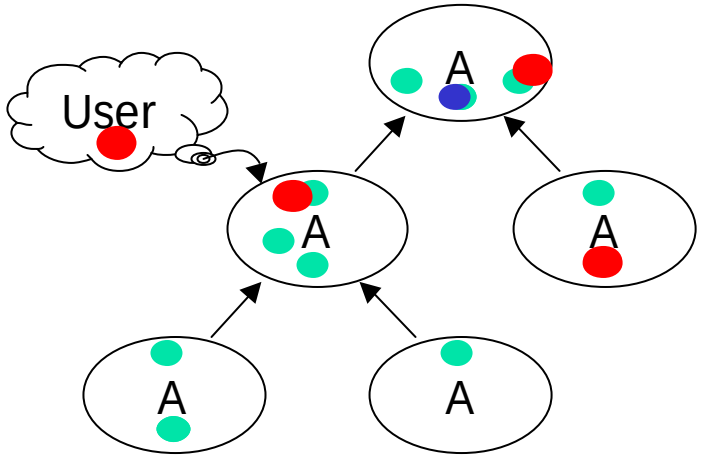
Centralised, not applicable for grid computing!

Optimisation Strategies

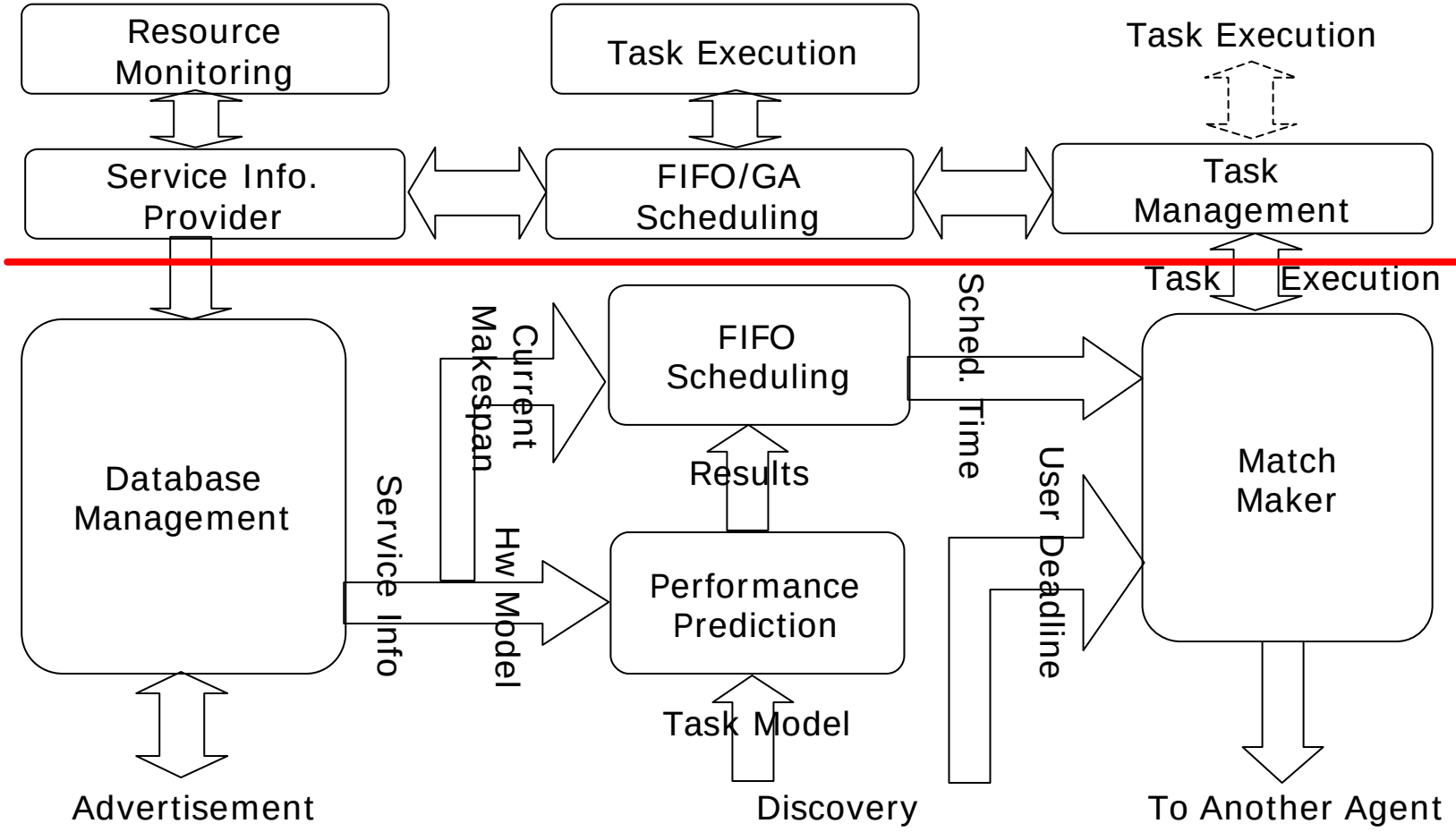
- Configurable data-pull or data-push
- Agent hierarchy
- Multi-step advertisement & multi-step discovery
- Efficient when frequencies of request arrivals and service changes are almost the same

Distributed!

balancing between advertisement and discovery!



Agent Implementation

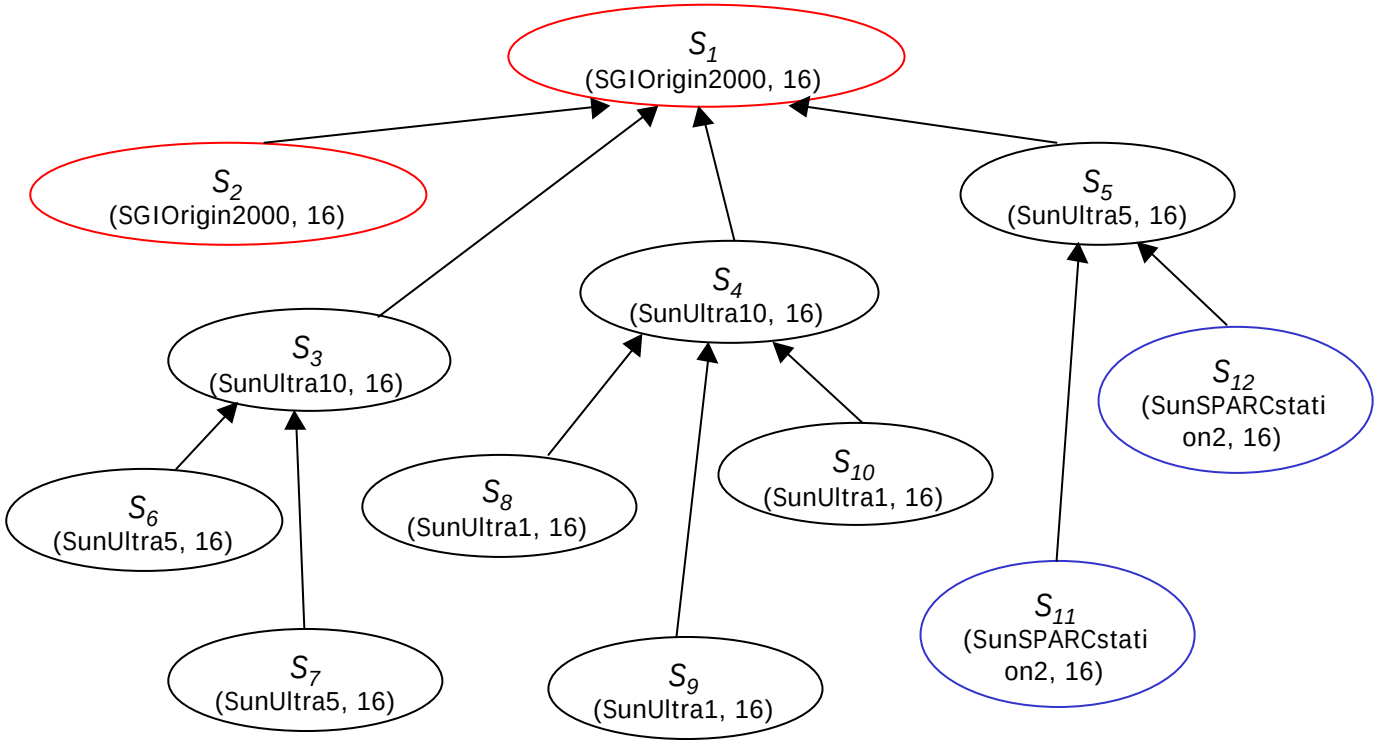


Load Balancing Metrics

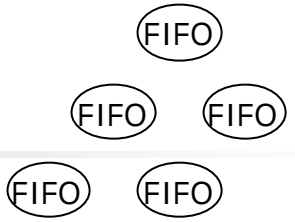
- Total makespan
- Average advance time of task execution completions (required deadline - actual task completion time)
- Average processor utilisation rate (busy time / total makespan)
- Load balancing level (1 - mean square deviation of processor utilisation rates / average processor utilisation rate)
- Total number of network packages for both advertisement and discovery

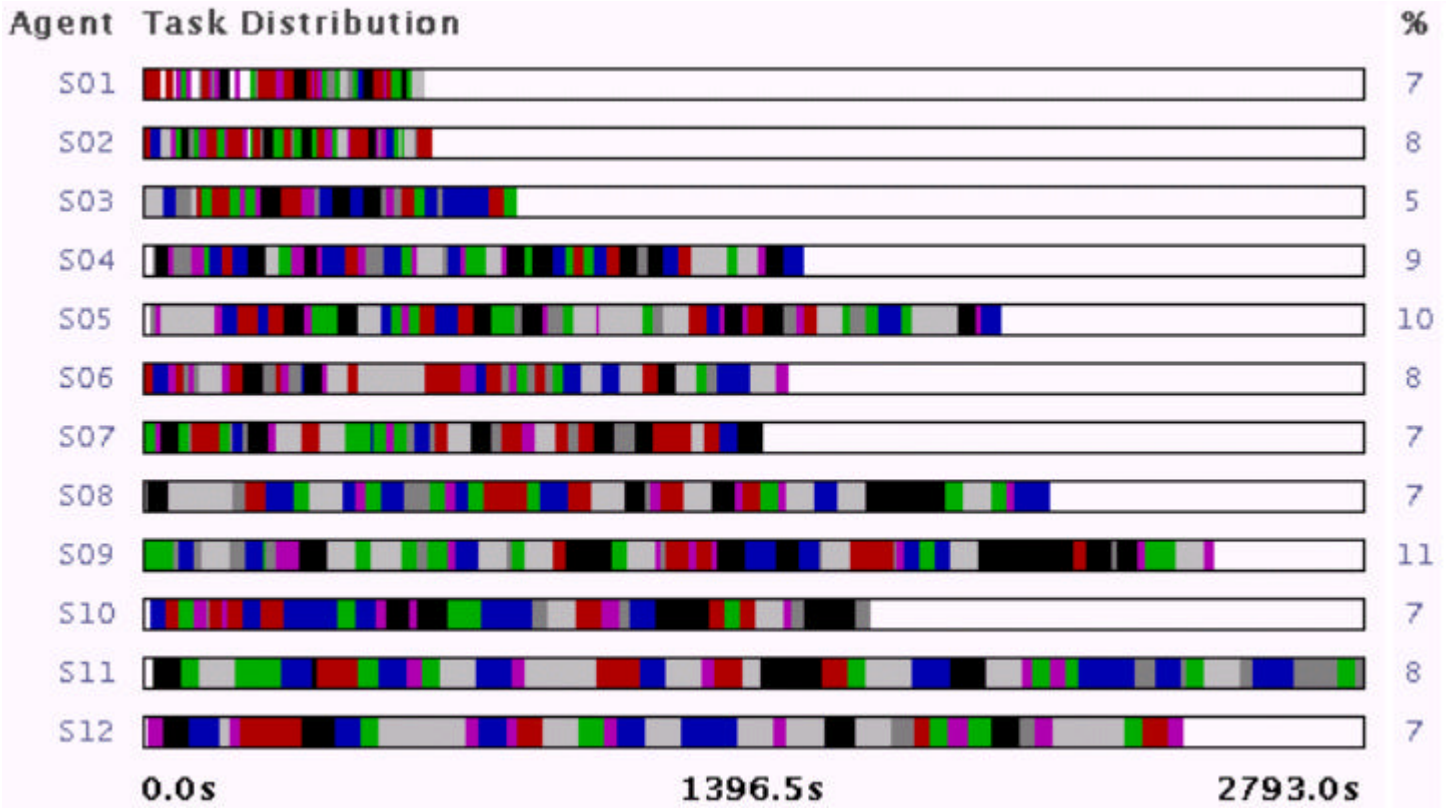
Experiment Design

Tasks:
sweep3d
fft
improc
closure
jacobi
memsort
cpi

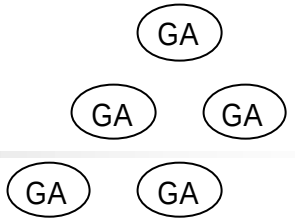


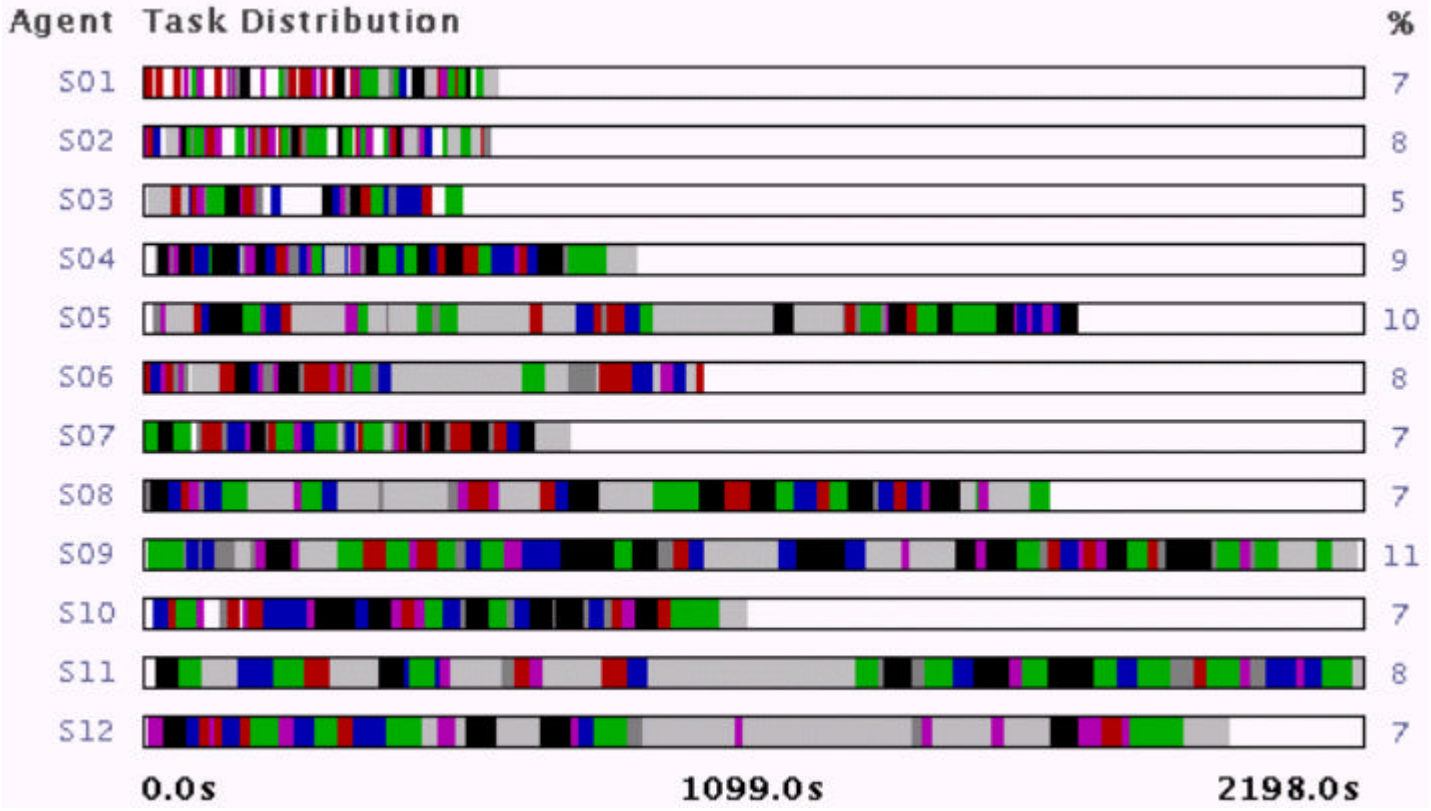
Experiment 1



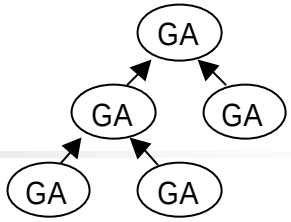


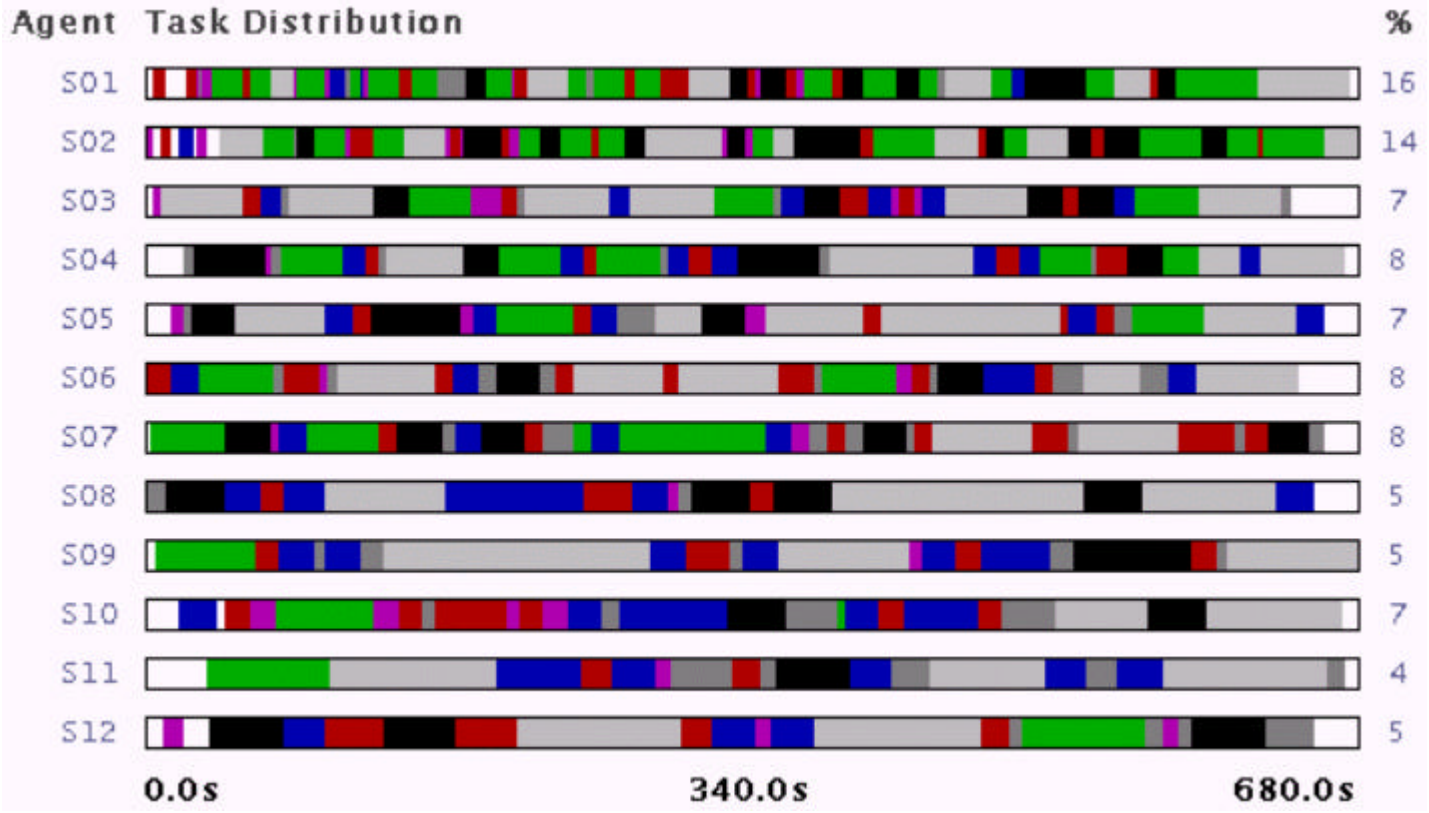
Experiment 2



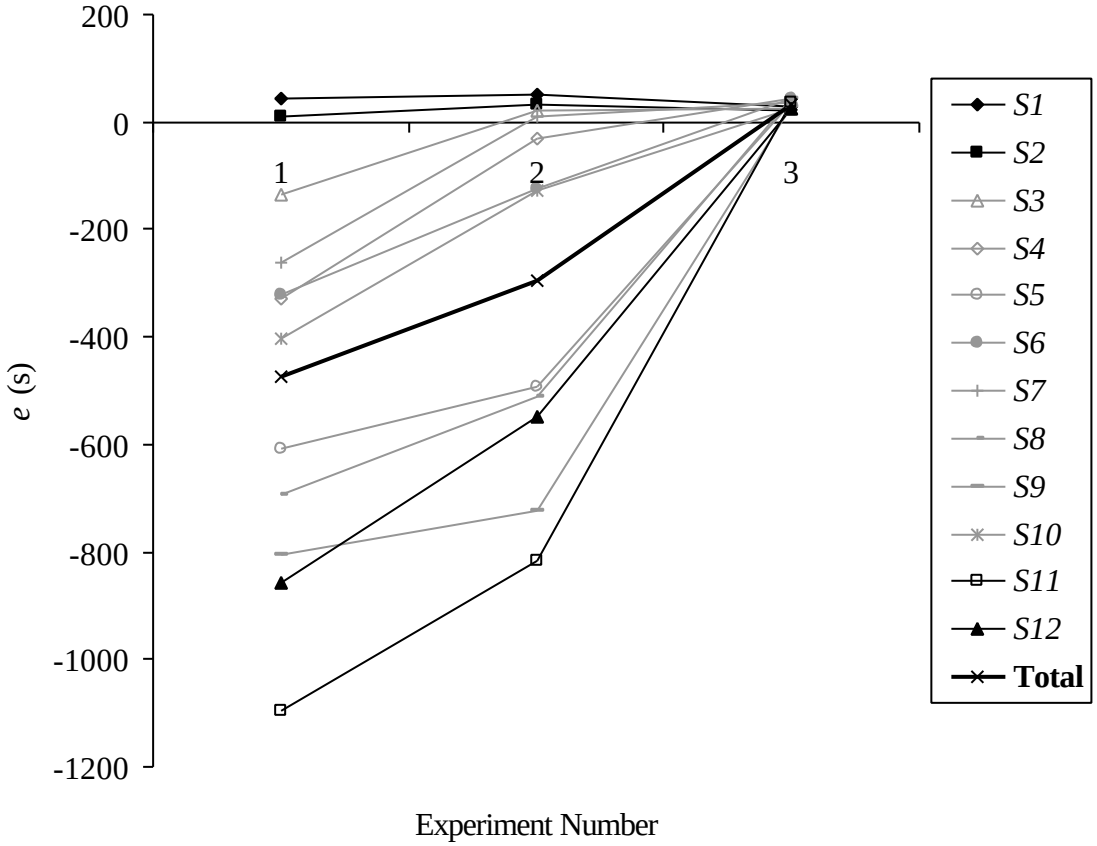


Experiment 3



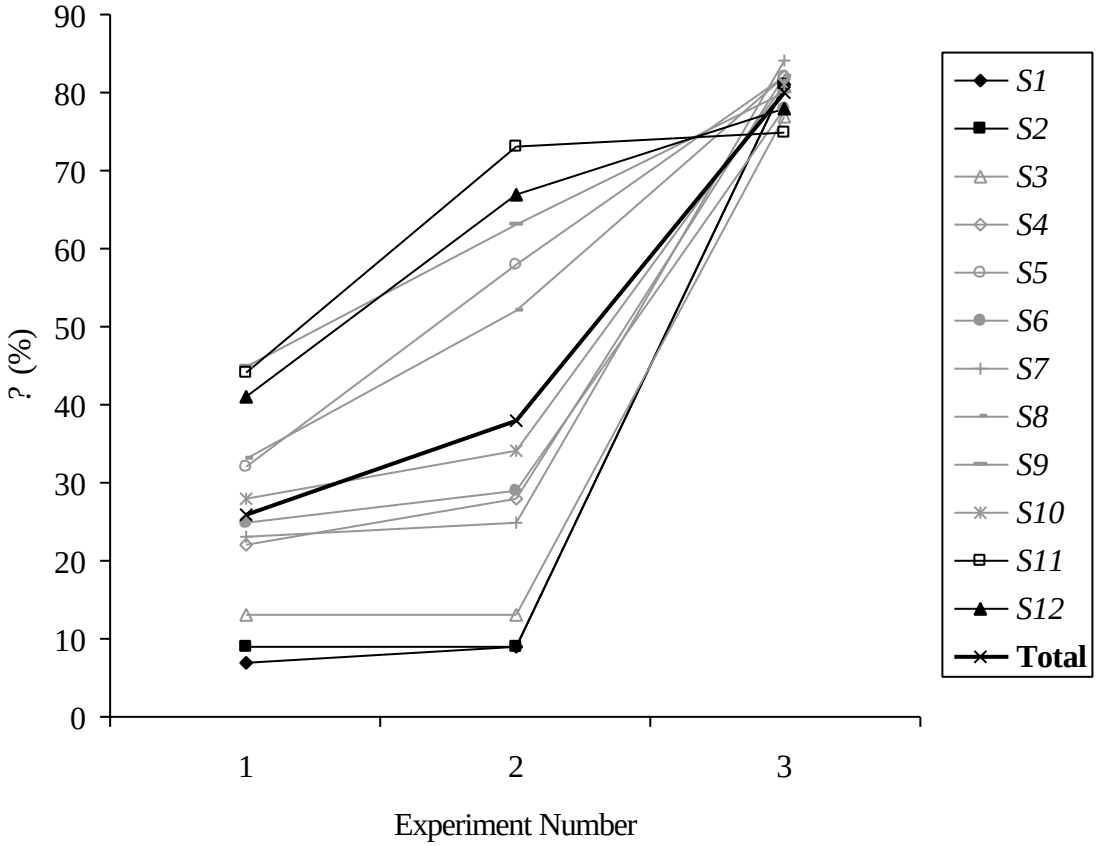


Task Execution



Both GA and agents contribute towards the improvement in task executions.

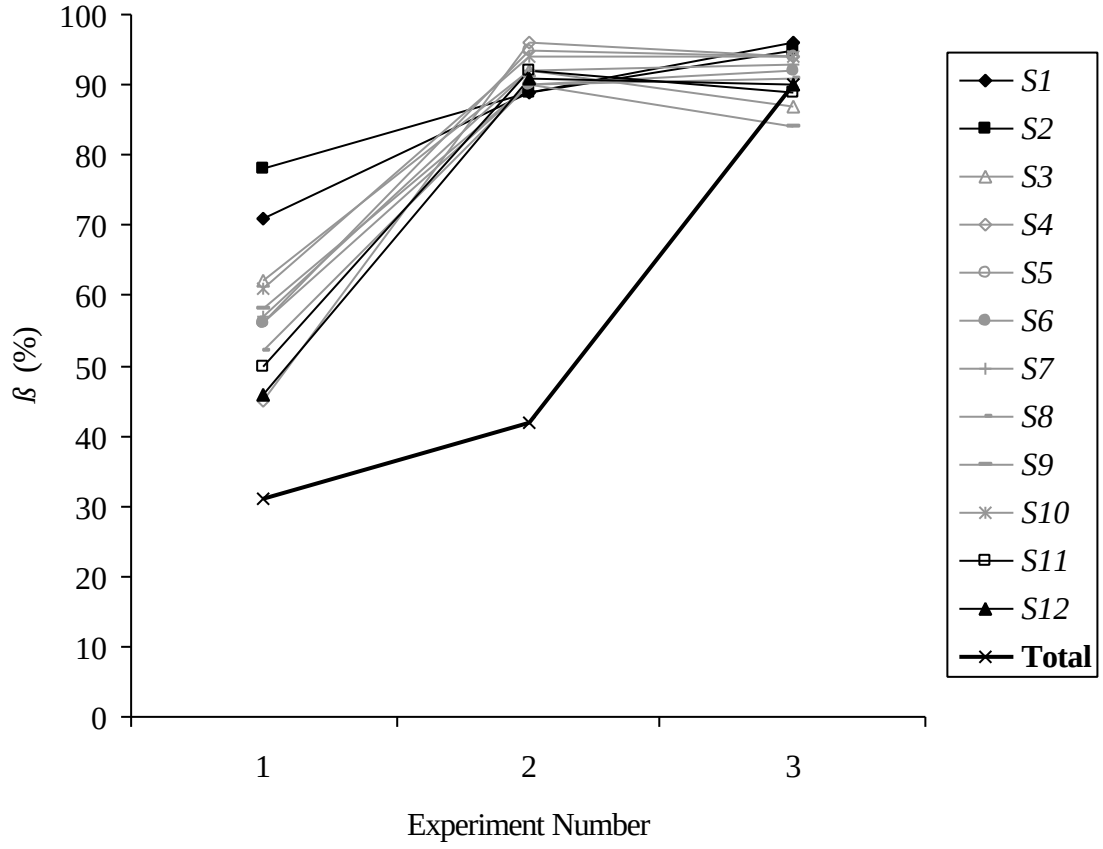
Resource Utilisation



Less powerful S11 & S12 benefit mainly from the GA.

More powerful S1 & S2 benefit mainly from agents.

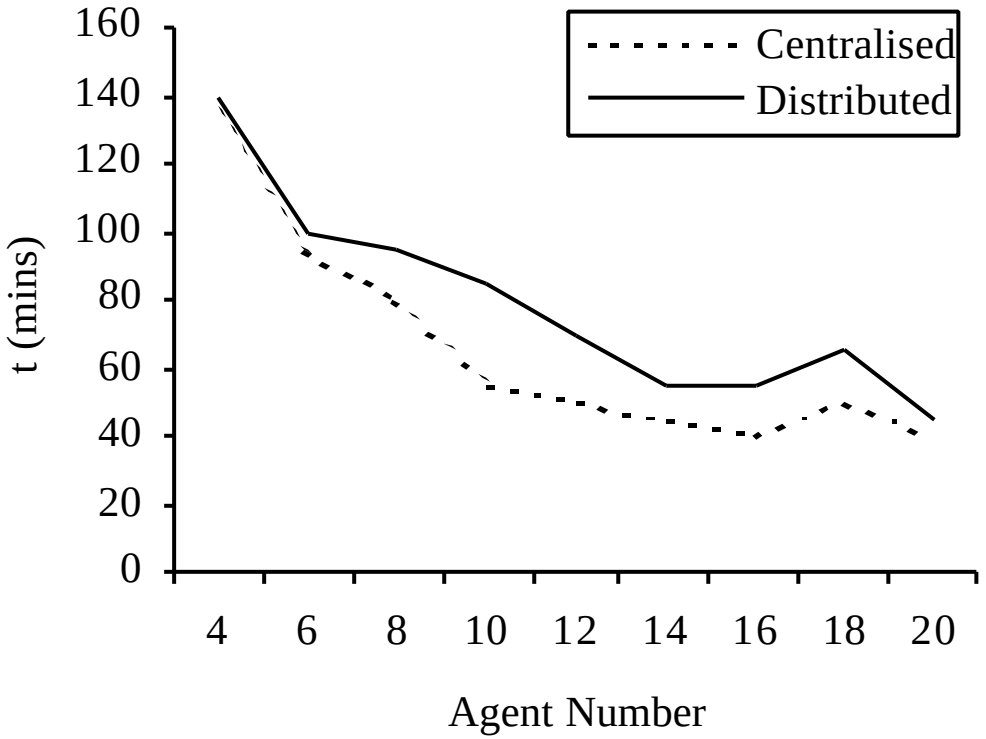
Load Balancing



The GA contributes more to local grid load balancing.

Agents contribute more to global grid load balancing.

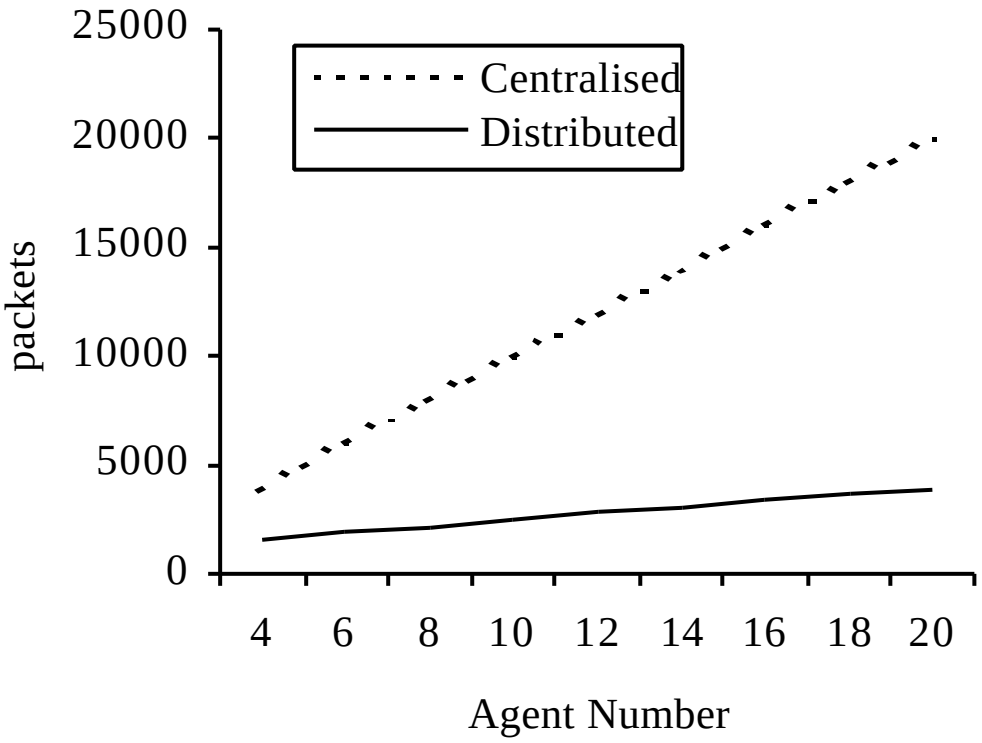
Total Makespan



The centralised pure data-pull can always achieve the best results

Distributed agents with the hierarchical model can also achieve reasonably good results

Network Package



The network overhead for the pure data-pull strategy to achieve better results is very high.

Distributed agent-based service advertisement and discovery can scale well.

Conclusions

- An multi-agent paradigm provides a clear high-level abstraction of grid resource management system.
- Distributed service advertisement and discovery strategies can be used to improve agent performance.
- Agent-based framework is scalable, flexible, and extensible for further enhancements.