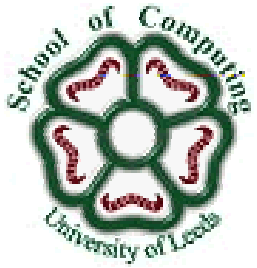
Adaptive Grid Resource Brokering

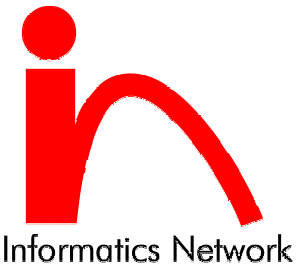
A. Othman, P Dew, K Djemame and I Gourlay

Informatics Research Lab.

School of Computing

University of Leeds



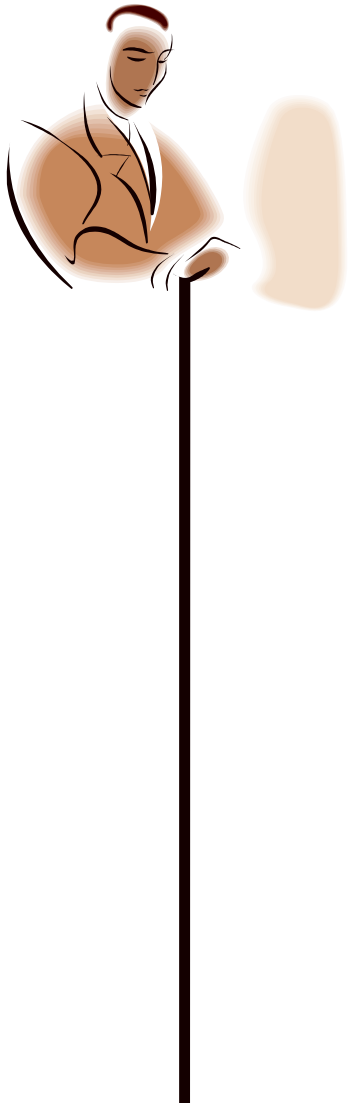


Introduction

- Grid Resource Brokering is defined as the process of making scheduling decisions involving resources over multiple administrative domains [Sch02]
- The need for Grid Resource Brokering

[Sch02] J. Schopf. *A General Architecture for Scheduling on the Grid*. Submitted to the Journal of Parallel and Distributed Computing, 2002

The Need for Adaptable Grid Resource Brokering



Adaptability

- Adaptability is crucial in the context of the Grid
- An adaptive application can change its architecture and behaviour depending on available resources, optimising itself to its dynamic environment [Fri94].

[Fri94] N. Friedman and K. Lieberherr. *Reuse of Adaptive Software through Opportunistic Parameterization*. Technical Report NU-CCS-94-17, Northeastern University, May 1994.

Adaptability in the Grid

- Adaptation can be implemented in an ad-hoc fashion by embedding adaptability in the applications code
  - Does not work well for global adaptation (e.g. multi-institutional virtual working environment)
  - Complicates both the application and adaptation code
  - Makes the reuse of adaptation strategies impossible

Proposed Solution

- A Reflective Technique is used to simplify adaptation in the Grid application
  - aims to separate concerns for functional and non-functional behaviour of a program
  - results in some code that is highly reusable and extensible [Bla99]

[Bla99] G. Blair, F. Costa and G. Coulson. *Experiments with Reflective Middleware.* In Proceedings of ECOOP'98 Workshop on Reflective Object Oriented Programming and Systems, Springer Verlag, 1998. Internal Report No MPG-98-11, Distributed Multimedia Research Group. Lancaster University. 1999.

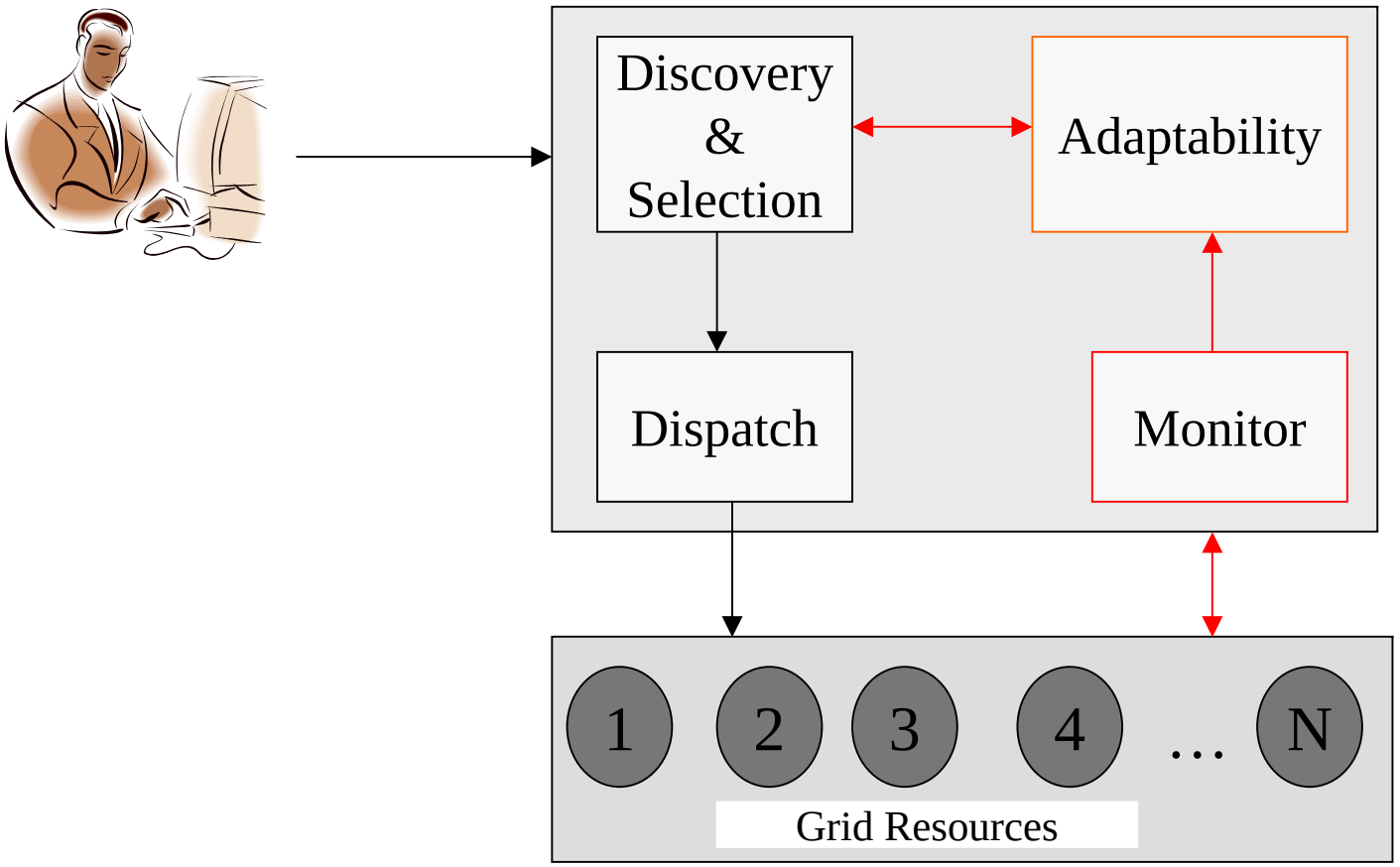
Adaptability through Reflection

- Isolate the user from the complexities of the system
  - No need to alter his/her code in order to achieve adaptability
- The benefits of using reflection [Bla98, Cou01]:
  - Flexibility to customize policies dynamically to suit run-time environment
  - High-level transparency to applications.

[[Bla98] G. Blair B. and G. Coulson M.. *The Case for Reflective Middleware*. Internal report number MPG-98-38, Distributed Multimedia Research Group. Lancaster University. 1998

[Cou01] G. Coulson**.** *What is Reflective Middleware?* IEEE Distributed System online.

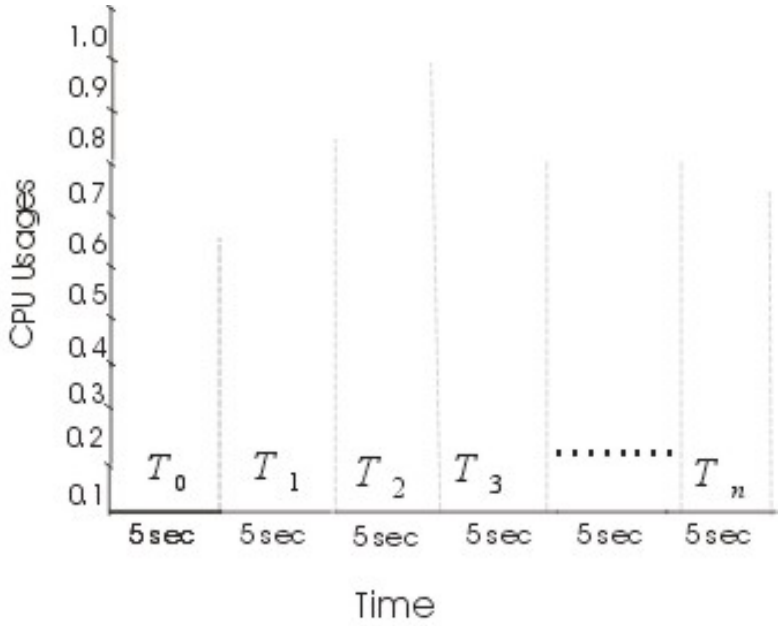
Adaptive Resource Broker



Prediction Modelling

$$T_{remaining} = \frac{T_{100\%} - \sum_{i=1}^{n-1} T_i F_i}{F_{estimate}}$$

$$F_{estimate} = \overline{F}_n = \frac{1}{n} \sum_{i=1}^{n} F_i$$



Run-Time Prediction (RTP)

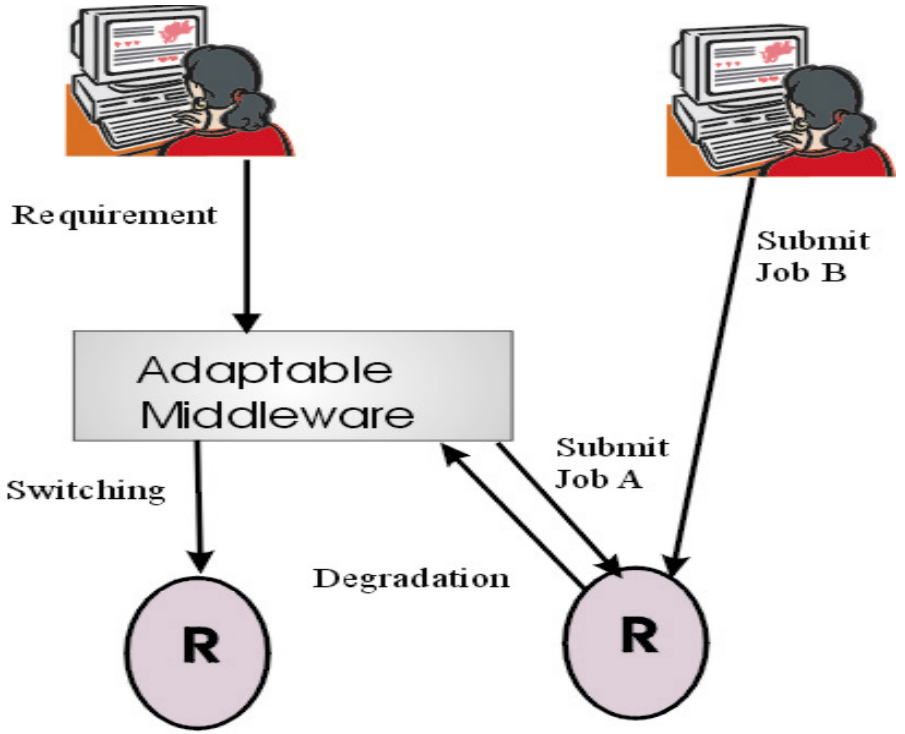
- The fractional CPU usage is measured periodically
- 1 representing 100% CPU usage and 0 corresponding to no CPU usage.
- If *n* samples are taken, then these can be used to predict the remaining time to completion of the job

Prediction Modelling

- Job *A* uses resource r through adaptive resource broker
- Job B accesses the same resource r

Since job *B* uses some portion of the resource

- Job *A* takes longer to finish
- The monitor passes information to the Adapter Manager, based on the RTP formula
- The Adapter Manager initiates migration



Experiments

- The experiments address the following questions:
  - When job requirements are not met, are jobs being successfully migrated?
  - Does this result in shorter job execution time, compared to the case when the adaptive middleware is not used?
- Run on Grid-test-bed consisting of 10 machines
  - Pentium IV processor (1.2 GHz) with 256 RAM
  - Linux 2.4
  - Globus 2.2.
  - Communication via a fast LAN Ethernet
- Do not address the issue of job migration times and their effect on the overall job execution time

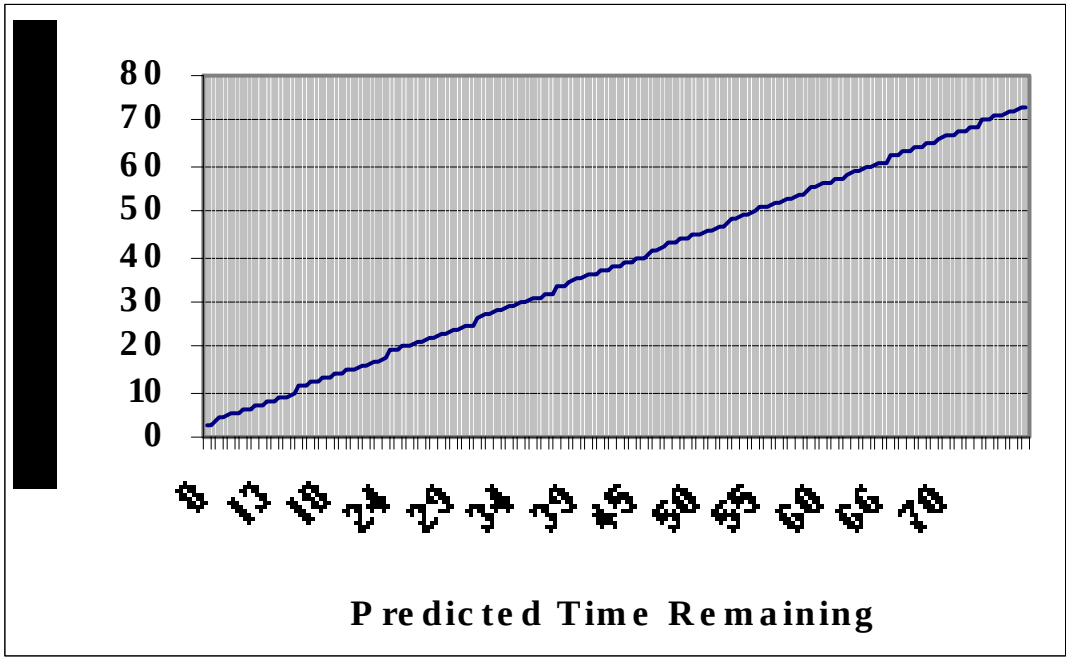
Experiments cont.

- CPU usage periodically measured
- Experiments run with the assumption that the user knows how long the job would take to run with 100%

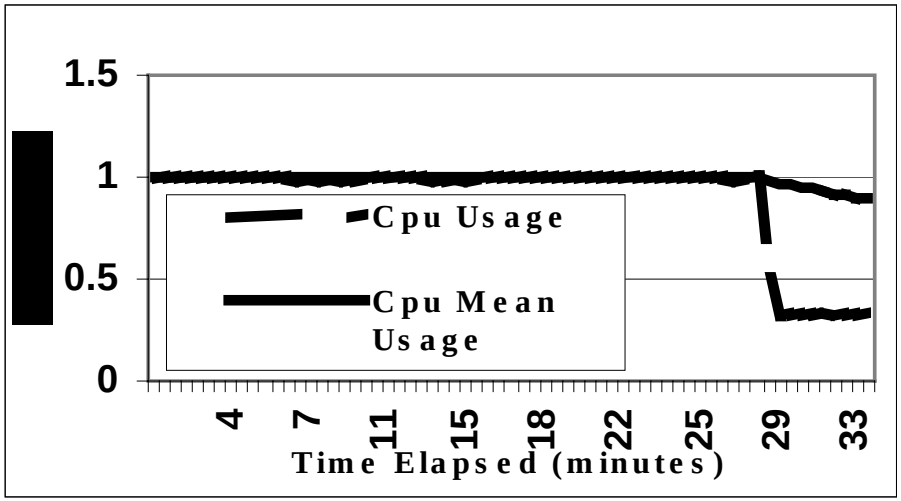
Experiments cont.

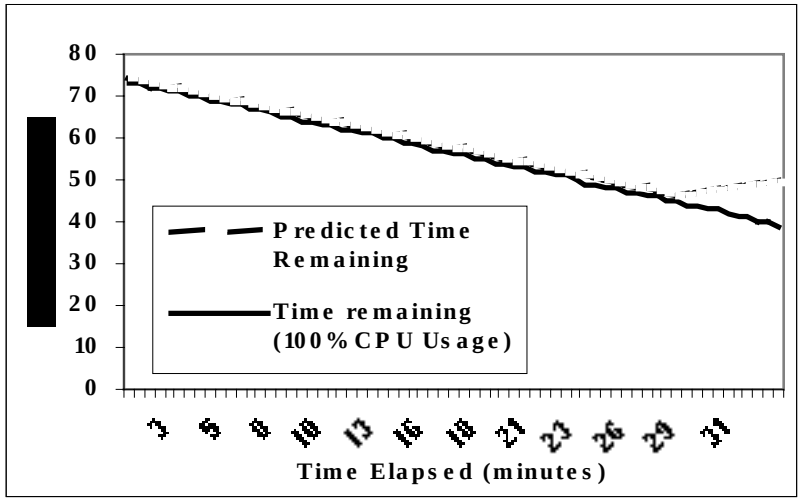
- Experiment 1
  - Job is executed from start to finish
  - Remaining job time predicted by the RTP formula during the course of the computation is compared to the actual execution time
- Experiment 2
  - Job is executed and during execution, other jobs are submitted to the same resource
  - Migration is initiated
  - Compared to case where adaptation is not used.

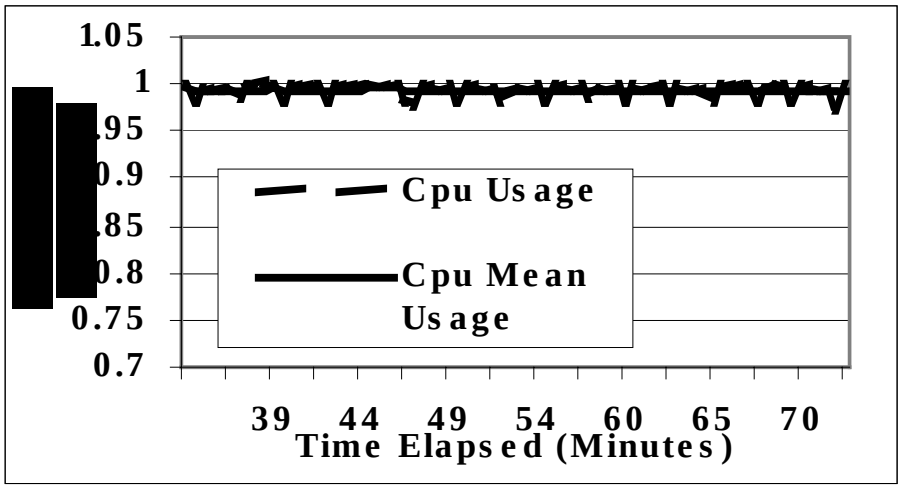
Results of Experiment 1

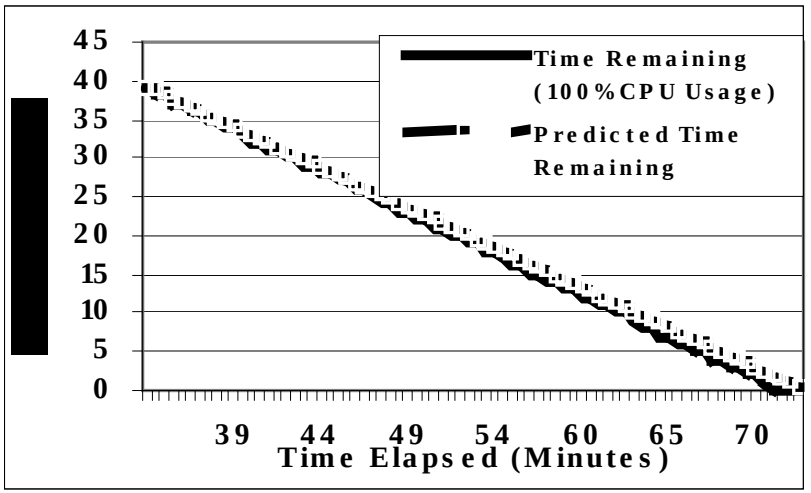


Results of Experiment 2









Conclusion

- Reflective technique enables application configuration and adaptation based on resource characteristics and user preferences
- Reflective middleware permits run-time mechanisms to automatically decide when and how to adapt the application in reaction to changes in resource conditions
- Future work
  - Run the same experiment in a wide area network, e.g.. White Rose Grid
  - Developing an interactive job, where the user can change the attributes of the job during run time