# Can Intelligent Optimisation Techniques Improve Computing Job Scheduling In A Grid Environment? Review, Problem and Proposal

**Wei Huang[1],** Tim French[1,2], Carsten Maple[1], Nik Bessis[1]

[1] Department of Computing and Information Systems, University of Bedfordshire, Park Square, Luton, LU1 3JU, UK (Corresponding author: wei.huang@beds.ac.uk )

[2] Applied Semiotics with Informatics Laboratory, Informatics Research Group, School of Systems Engineering, University of Reading, Whitenights, RG6 6AH, UK

## Abstract

In the existing Grid scheduling literature, the reported methods and strategies are mostly related to high-level schedulers such as global schedulers, external schedulers, data schedulers, and cluster schedulers. Although a number of these have previously considered job scheduling, thus far only relatively simple queue-based policies such as First In First Out (FIFO) have been considered for local job scheduling within Grid contexts. Our initial research shows that it is worth investigating the potential impact on the performance of the Grid when intelligent optimisation techniques are applied to local scheduling policies. The research problem is defined, and a basic research methodology with a detailed roadmap is presented. This paper forms a proposal with the intention of exchanging ideas and seeking potential collaborators.

## 1. Introduction

In recent years, there has been increasingly interest in using network-based resources for large scale data-intensive computation problems. These problems, usually found in a number of disciplines such as high-energy physics, astronomy and bioinformatics, involve loosely coupled computing jobs and geographically distributed resources e.g. supercomputing powers and large datasets. Within this context, the Grid computing paradigm originated as a new infrastructure to address these problems and is now an established technology for large scale resource sharing and distributed integration within both science and industry setting [1]. As one of the most important future trends, the importance of Grid is widely recognised and Grid-related projects are heavily funded world wide, e.g. US Globus, UK e-Science, and EU FP6.

Effective computation and data scheduling is rapidly becoming one of the main challenges in Grid computing, and is seen as being vital for its success. Different strategies have been proposed for effective job and data scheduling for such complex systems [2-5]. Ranganatham and Foster in [2] proposed a generic Grid scheduling architecture in which the scheduling logic is encapsulated in three modules: External Scheduler (ES), Local Scheduler (LS) and Dataset Scheduler (DS). Each user submits jobs to an external scheduler that decides to which remote site these jobs are allocated. The local scheduler at a site decides how to schedule all pending jobs based on available resources. The dataset scheduler at each site keeps track of the popularity of each dataset requested and it will replicate popular datasets to remote sites depending on a number of strategies. A total of twenty different ES and DS algorithm combinations were proposed and evaluated. However, in order to simplify the study, FIFO (first in first out) was used as a local scheduling policy. Yarmolenko et al. [6] also indicated that traditionally the enactment of jobs on parallel computing resources have been based on queue-based scheduling systems, namely "run the job when it gets to the head of the queue".

This paper reports upon the state-of-the-art literature review on the methodologies used for Grid scheduling, particularly those for local scheduling policies in a Grid environment. The rest of the paper is structured as follows. Section 2 reviews related work in Grid scheduling. In Section 3, the problem of local scheduling policies is identified and the research objectives are presented. A basic research methodology containing a detailed research roadmap is given in Section 4 and Section 5 concludes the paper.

## 2. Related Work Review

A variety of factors need to be considered for the effective scheduling of resources in Grid environments, e.g. resource utilization, response time, global and local allocation policies and scalability. Ranganatham and Foster [2] indicate that effective scheduling in a Grid system is complex. The large amount of input data needed for each computing job would suggest that job scheduling strategies should ideally take data locality into account. Considering the large number of users and resources that a Grid system supports, decentralized strategies may perform better than centralized strategies. Thus, an important aspect for Grid scheduling is to allow local scheduling policies. Yarmolenko et al. [6] indicate that it was only recently that more flexible approaches were used and Service Level Agreements (SLAs) could effectively be used for job farming. In their paper, they presented their work on an UK e-Science project that investigated the effects of SLA based approaches for computing job scheduling. A coordinator based architecture and evaluation of different policies for the negotiation of SLAs between the coordinator and the resources were considered. The work showed that the use of information available in the SLAs agreed by the User might fundamentally alter the behaviour of the coordinator and hence the performance of the overall system. However, these effects need to be understood in more detail before new designs can take advantage of the possibilities offered by SLAs.

In another recent UK e-Science project, Palmer and Mitrani [7] considered optimising the tree structure of large-scale Grids containing many processors. They argued that a “flat” structure, where only one single level master node (i.e. a global scheduler) controlled all processors and decided where incoming jobs should be executed, was not always efficient as the master node could become easily overloaded when demand was high. A tree structure involving a hierarchy of master nodes (i.e. global and local schedulers) could control subsets of processors so as to avoid bottleneck problems but might introduce additional processing and transfer delays. A simple heuristic approach was adopted in their paper for the dynamic reconfiguration of the tree structure as load changed. It was shown through numerical experiment that, for a given set of parameters and job distribution policy, there was an optimal tree structure that minimized the overall average response time. Kubicek et al [8] also proposed an architecture that allowed the dynamic reconfiguration of servers to process incoming jobs by switching servers between conceptual pools. A number of heuristic policies have been used to make optimal switching decisions, and a prototype system was developed to demonstrate these concepts.

Thomas et al. [9] investigated the performance effects of a delay in propagating information concerning computing node failure. In Grid computing environments, scheduling will generally be performed by global or external schedulers remotely from computing nodes. However, if a node has failed it is in no position to communicate its state and mechanisms should be applied to let the schedulers know this. The authors indicate that none of the current studies adequately deal with the consequence of failure at the resource level and very few services have been constructed with a fault tolerant perspective.

Cawood et al. [10] developed two fully Globus-enabled Grid scheduling tools, TOG (Transfer-queue Over Globus) and JOSH (JOb Scheduling Hierarchically), based on the Grid Engine and the Globus Toolkit (GT). Grid Engine is an open source distributed resource management system for computing resources within an organisation and the Globus Toolkit is an API for connecting distributed resources among different organizations. TOG has the potential to integrate Grid Engine V5.3 and Globus Toolkit V2.2 to allow access to remote resources, and JOSH used the Golbus Toolkit V3’s Managed Job Service (MJS) to run the job submission and termination scripts on behalf of the client user. However, the authors also indicate that due to the performance and robustness concern of the GT V3.0, a number of organisations have reluctantly decided not to consider JOSH for deployment. JOSH take-up might be further affected by the switch to WS-RF in GT 4.0.

## 3. Problem Definition

In the existing Grid scheduling literature, the reported methods and strategies are mostly related to high-level schedulers such as global schedulers, external schedulers, data schedulers, and cluster schedulers. Although a number of these have previously considered job scheduling, thus far only relatively simple queue-based policies such as First In First Out (FIFO) [1, 6] have been considered for local job scheduling within Grid contexts. An interesting question arises based on the current state-of-the-art Grid literature: what happens if more intelligent techniques are adopted for local scheduling within a Grid environment? Furthermore, will these techniques help to improve the efficiency of job scheduling and resources management for the whole Grid? For example, due to the relatively slower data transfer rate and frequent data transfer latency between geographically distributed Grid services, local scheduling policies need to consider factors such as jobs' priority, temporal constraints, jobs' waiting policies, datasets limitation, finite capacity of storage for job queues and data, and resource failure. Resource trust expectations also need to be considered and confidence levels can potentially become one of the criteria used to schedule and/or reschedule resources [11-12]. Under these circumstances, a queue-based scheduling policy is simply not good enough and an intelligent method will be more efficient.

This research aims to investigate the impact on the performance of a Grid system when intelligent optimisation techniques are adopted by local schedulers within a generic Grid scheduling architecture. After the completion of the planned research, it will become clear which local scheduling policies and their combination are efficient so that they can improve the performance of the whole Grid computing.

The specific research objectives are namely:

1. To identify a variety of factors that local schedulers need to consider within a general Grid scheduling architecture;
2. To develop intelligent scheduling models and algorithms for computation and data scheduling within a Grid system;
3. To develop a simulator that can be used to evaluate the performance of a Grid system adopting one or more intelligent techniques for scheduling;
4. To demonstrate the efficiency of the developed models and methodologies through simulated Grid environments.

These research objectives are important to the future study of the Grid. The Grid paradigm originated as a new computing infrastructure for data-intensive computing problems and is now an established technology for the sharing of large scale geographically distributed resources. Achieving these objectives will make the resource allocation more efficient within a Grid.

## 4. Proposed Research Methodology

The project will follow a systematic approach as follows:

1. State-of-the-art review. Several factors are assumed to be considered for Grid computing e.g. jobs' priority, temporal constraints, jobs' waiting policies, datasets and storage limitation. The first step of the project will be aimed at verifying these assumptions and matching them to current state-of-the-art.
2. Pilot scenarios development. Key characteristics will be identified for the Grid global and local scheduling and a number of pilot scenarios will be developed for assessing the impacts of local policies on the performance of different strategies of external scheduler (ES) and dataset scheduler (DS).
3. Grid system modelling. The Grid system will be modelled as a network of computing sites distributed geographically, each comprising a number of processors and a limited amount of storage. It is assumed that a number of users, each one associated with a particular site, submit jobs. Each job requires specific computation and dataset resources to be available, which may not reside on the site where the job is submitted. The general Grid scheduling architecture proposed in [2] will be adopted.
4. Modelling of scheduling problems. Novel constraint-based scheduling models for local schedulers within a Grid system will be proposed. The problem can be considered as constraint satisfaction problems (CSPs) where a set of jobs, a set of processors and a limited amount of data storage are given; each job requires a number of processors and an amount of storage for a certain time, satisfying a set of constraints such as jobs' priority, time-bound constraints etc, and the objective is to maximize the efficiency of a computing site i.e. to schedule as many jobs as possible within a deadline.
5. Development of scheduling

algorithms. A number of intelligent optimisation techniques including constraint programming (CP) will be considered to solve scheduling problems with a large number of constraints quickly. Specific scheduling algorithms and approaches for local schedulers within a Grid environment will be developed.

6. Simulator development. The project plans to evaluate the performance of developed models and methodologies by simulation that can allow the test of a wide range of scenarios. Currently, a number of Grid simulators, such as ChicagoSim, GridSim, SimGrid, and OptorSim, are available. A specific discrete event simulator for the project will be developed on top of one of available Grid simulators.
7. Verification programme. The developed models and methodologies will be integrated within the general Grid scheduling architecture [2] and demonstrated through a simulator-based Grid environment. The verification will be based on integrating different combinations of scheduling strategies of external scheduler (ES) and dataset scheduler (DS) with the developed scheduling approaches of local scheduler (LS) within a common environment. The effect on the performance of different ES and DS scheduling algorithms by developed models and methodologies will be investigated.

## 5. Conclusions and Future Work

This paper has reviewed state-of-the-art literature in Grid scheduling and has identified some potential problems in local scheduling policies within Grid systems. The initial literature research has established that it is worth investigating the impact of scheduling Grid computing by using intelligent optimisation techniques within local scheduling policies so as to deal with more complex constraints in job scheduling and resource management. A basic research methodology with a detailed roadmap for future research is proposed. Constraint programming techniques will be used in this investigation and other intelligent optimisation techniques will also be considered. The viabilities of the proposed scheduling models and algorithms will be evaluated through simulated Grid environments.

## References:


[1] I. Foster, C. Kesselman, 1999. The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann, San Mateo.

[2] K. Ranganathan and I. Foster, 2003. Simulation studies of computation and data scheduling algorithms for data grids. Journal of Grid Computing, 1(1), 53-62.

[3] M. Caramia, S. Giordani and A. Iovanella, 2004. Grid scheduling by on-line rectangle packing. Networks, 44(2), 106-119.

[4] Y. Gao, H. Rong and J. Huang, 2005. Adaptive grid job scheduling with genetic algorithms. Future Generation Computer Systems, 21, 151-161.

[5] C. Weng and X. Lu, 2005. Heuristic scheduling for bag-of-tasks application in combination with QoS in the computational grid. Future Generation Computer Systems, 21, 271-280.

[6] V. Yarmolenko, R. Sakellariou, D. Ouelhadj, J. M. Garibaldi, 2005. SLA Based Job Scheduling: A Case Study on Policies for Negotiation with Resources. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2005.

[7] J. Palmer, I. Mitrani, 2005. Optimal Tree Structures for Large-Scale Grids. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[8] C. Kubicek, M. Fisher, P. McKee, R. Smith, 2004. Dynamic Allocation of Servers to Jobs in a Grid Hosting Environment. Proceedings of the UK e-Science All Hands Meetings, 2004.

[9] N. Thomas, J. Bradley, W. Knottenbelt, 2004. Performance of A Semi Blind Service Scheduler. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[10] G. Cawood, T. Seed, R. Abrol, T. Sloan, 2004. TGO & JOSH: Grid Scheduling with Grid Engine & Globus. Proceedings of the UK e-Science All Hands Meetings, Nottingham, 2004.

[11] S. Ramchurn, D. Huynh and N. Jennings, 2004. Trust in multi-agent systems. The Knowledge Engineering Review, Vol. 19:1, 1-25.

[12] T. French and W. Huang, (2005). Grid enabled collaborative computing: Is trust the hardest issue to address?. Proceedings of the 11th Chinese Automation and Computing Society Conference in the UK, pp131-136, ISBN 0 9533890 8 1, Sheffield, England, 10 September 2005.