AN INTELLIGENT RESOURCE SELECTION SYSTEM BASED ON NEURAL NETWORK FOR OPTIMAL APPLICATION PERFORMANCE IN A GRID ENVIRNMENT

M. Castellano†, T. Coviello†§, G. Piscitelli†
†Dipartimento di Elettrotecnica ed Elettronica - Politecnico di Bari - ITALY
§Dipartimento Interateneo di Fisica & INFN di Bari - ITALY

Abstract

Grid computing is a large scale geographically distributed and heterogeneous system that provides a common platform for running different grid enabled applications. As each application has different characteristics and requirements, it is a difficult task to develop a scheduling strategy able to achieve optimal performance because application-specific and dynamic system status have to be taken into account. Moreover it may be possible to obtain optimal performance for multiple application simultaneously using a single scheduler. Hence in a lot of cases the application scheduling strategy is assigned to an expert application user who provides a ranking criterion for selecting the best computational element on a set of available resources. Such criteria are based on user perception of system capabilities and knowledge about the features and requirements of his application. In this paper an intelligent mechanism has been both implemented and evaluated to select the best computational resource in a grid environment from the application viewpoint. A neural network based system has been used to capture automatically the knowledge of a grid application expert user. The system scalability problem is also tackled and a preliminary solution based on sorting algorithm is discussed. The aim is to allow a common grid application user to benefit of this expertise.

INTRODUCTION

Nowadays several models to make computation are proposed and exploited. Among these supercomputers represent the classical approach for processing heavy computational jobs but at the same time they are both an expansive solution and unable to solve very huge computational scientific problems; enterprise-level distributed computing have limited cross-organizational capability, while peer-to-peer technology has limited scope and mechanisms. Finally web services and semantic web computational approaches are respectively not dynamic and focused on a particular problem. A novel technological solution to support collaborative work on wide area scale is nowadays known as the grid. A grid is a distributed collection of computational and storage resources which constitute an infrastructure for establishing, managing and evolving multi-organizational federations which are dynamic, autonomous and domain independent, known as Virtual Organizations (VO) [1]. Users within a particular VO have access to all or some of these resources, i.e. computing, data, instrumentations and services.

Recently several scientific communities are evaluating and testing the grid technology for resolving in a coordinate way novel and unexplored domain problems. The grid platform has being applied successfully in the high performance data mining services[2], in three-dimensional (3D) data acquisition in medical imaging applications [3], in biological sequence alignment[4] and in data-intensive high energy physics experiments currently being developed at CERN[5].

In large scale grid application a problem arises about the job execution performance[6]. In a lot of cases the application scheduling strategy is assigned to an expert application user who provides a criterion for selecting the best resource in order to improve the application performance. However it depends on the real availability of a user who plays the expert application role.

In this paper a solution is proposed to capture and describe in automatic way the knowledge of the grid application expert user. The aim is to provide an intelligent mechanism that leads to select the best grid resource for the job submission from the expert point of view and then to enable an unskilled grid application user to benefit of this expertise. Section 2 gives an overview on the basic grid architecture and a description of the selection process driven by the user. Section 3 describes the user selection model based on a feed-forward neural network algorithm. Finally the simulation and experimental results are shown.

A BASIC COMPUTATIONAL GRID REFERENCE ARCHITECTURE

Computational grids are created to serve different communities with widely varying characteristics and requirements. Therefore a general grid architecture is composed by fundamental elements as the basic service elements and the technological ones. In particular in high-performance applications the basic services are referred to computational and storage services, while the technological elements concern end systems, computer clusters, intranets and internets. Computing and storage physical resources are available respectively as Computing Element (CE) and

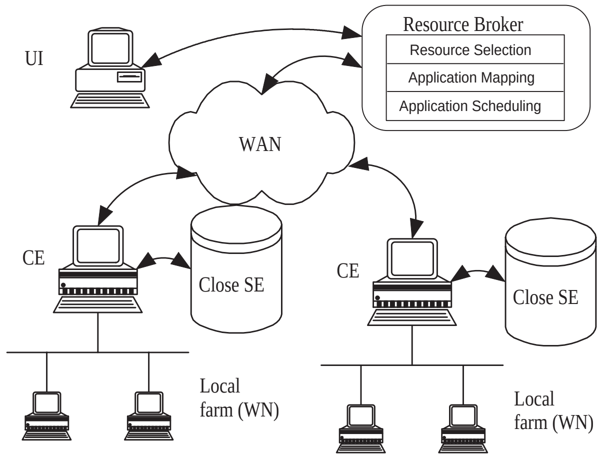

Figure 1: Grid architectural reference mode

Storage Element (SE) services (fig.1). Each CE delivers jobs to the Worker Nodes, which will perform the real work. The Computing Element provides an interface to the local batch queuing system. A Computing Element can manage one or more Worker Nodes [7].

The system component for both managing grid resources and executing applications is the workload management system (WMS) [8, 9, 10]. The Resource Broker (RB) is the core component of the Workload Management System which plays the role of a super-scheduler. Its main activities are resource selection, mapping and scheduling. Selection task refers to the process of selecting proper resources which are available and fulfill application requirements. Mapping task has the responsibility both to assign an application task to compute resources and to distribute data. Finally scheduling task deals with the allocation of computation over-time. The component, which allows users to interact with the grid system, is the User Interface (UI). UI permits the submission, the status monitoring and the output result retrieval of jobs. User can request to the system a list of computational resources which fulfill the application requirements. Finally he can express a certain ranking criterion choosing the best resource for his jobs.

USER RESOURCE SELECTION MODEL

Often scheduling process is performed by an expert grid user who chooses the best computational resource according to both job type and grid status. Grid status can be described by the status of each computing element. In a lot of cases resource selection is performed on heuristic base and therefore it is not described in a formal way. Although a formal description is not available, a study on a suitable set of resource selection examples can give rise to a formal generalization.

Artificial neural networks are the natural candidates for this task, because they are able to describe in sub-symbolic fashion the knowledge acquired by example observation.

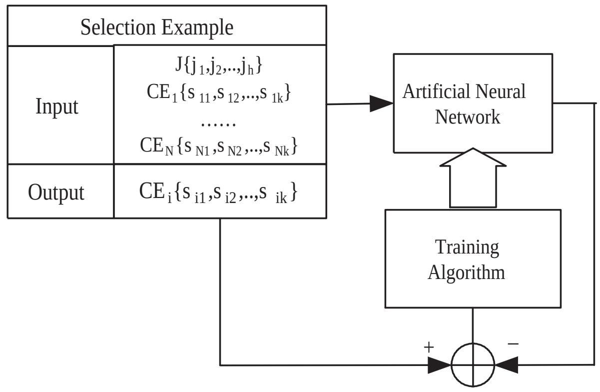

Figure 2: User behavior model

Neural networks are computational models based on processing elements which are connected together such to make a scheme similar to a human nervous system. They are parallel and distributed structures for information processing, composed by elaboration elements interconnected by unidirectional channels. Each channel has an associated numeric weight according to matured experience during the learning phase.

The learning phase is carried out feeding the neural network with a set of user resource selection examples. A specific algorithm is used to modify synaptic matrix for minimizing the neural network error which results from comparison of user and neural network selection. Fig. 2 shows as examples are built. Both job description and computing element status are described respectively by J$\{j_1, j_2, j_h\}$ and CE$_i$ $\{s_{i1}, s_{i2}, ..., s_{ik}\}$ vectors. The job description and the grid status are the input data; the selected computing element is the expected output.

For studying a user resource selection model based on neural network mechanism, a simulation has been done. As an example, the user behavior has been modeled using the Minimum Completion Time algorithm [11]. It requires four input parameters, shown in tab. 1. The first three parameters describe the computing element status. The job description is represented by the computational weight parameter. In the simulation a feed forward neural network with one hidden layer is used, trained by supervised modality [12].

Table 1: MCT Parameters

Parameter	**Unit of measurement**
CE average SpecInt95	[Specint95]
Queuing mean time per job	[s]
Number of jobs running on the resource	[]
Job Computational Weight	[Specint95*s]

In our simulation scenario we considered a computational grid composed by 5 computing elements. Therefore

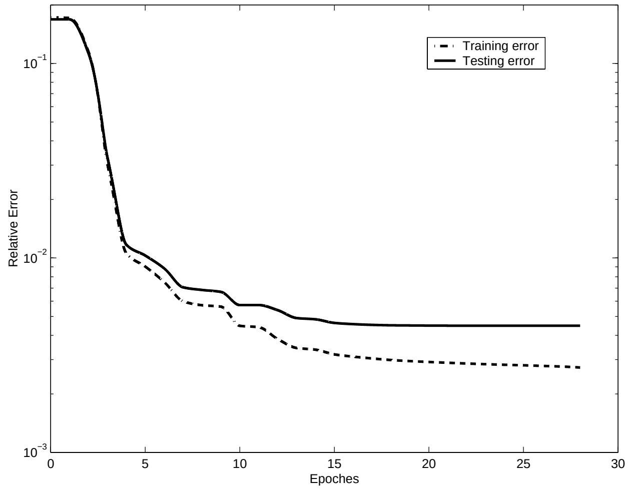

Figure 3: Training and testing neural network results

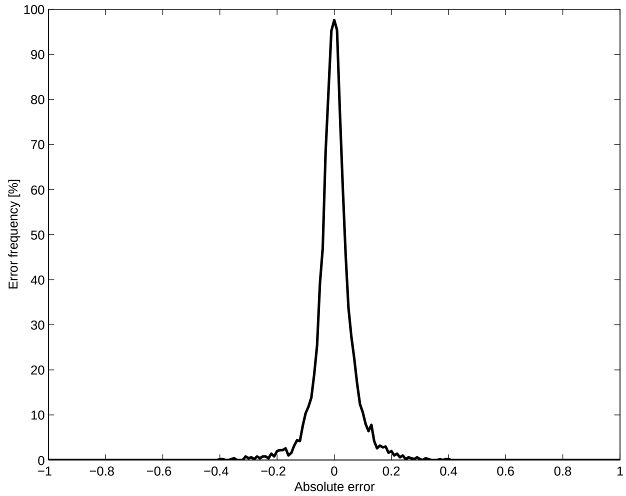

Figure 4: System performance

the input layer of the neural network contains 16 nodes. The output layer of neural network is composed by a single node whose output depicts the selected computing element according to the simple indicator as expressed in equ. 1:

$$O = \frac{\|CE_{selected}\|_2}{N_{Parameters} + 1} \tag{1}$$

where $N_{Parameters}$ represents the component number of CE status vector. For the neural network training and testing phase two sets, each composed by 5000 examples, have been used. The training phase has been stopped at 28th epoch when the network over-training has been observed (fig. 3). After training phase, a validation of neural network has been performed using a sample of 10000 job submissions.

The system performance has been measured comparing the neural network selection against the expert user selection. The efficiency of the system has been evaluated considering the error frequency, obtained as the difference between the norm of the computing element selected by the user and the norm of the computing element selected by the neural network based system (fig. 4).
Previous solution is little scalable when the number of computing elements becomes larger. So we are investigating an alternative solution based on a sorting algorithm. We assume that the grid expert user choices the best grid resource, selecting the first of a ordered list of available computing elements. This ordering can be carried out through a bubble sort algorithm. The basis of this algorithm is to repeatedly iterate through the list comparing every adjacent pair of elements and swapping them if they are not in the correct relation. At the same manner we can use the neural network system to compare adjacent pair of computing elements through the list of available resources. We can stop at the first iteration because we are interested in the resource on top of the list. The advantage of this solution is that the system become more scalable if the number of computing elements increases.

CONCLUSIONS

In this paper a skilled user-model based on a neural network algorithm has been implemented and evaluated. The study has been done to develop an intelligent and automatic mechanism for making available to the large grid communities the expertise of a skilled grid user in system resource selection. Presented work is evolving in searching a more scalable solution based on a sorting algorithm.

REFERENCES

[1] I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications.

[2] S.Orlando,P.Palmerini, R.Perogo,F.Silvestri: Scheduling High Performance Data Mining tasks on a data grid Environment. Proceedings of Europar (2000)

[3] J. Gregor, M. Paulus, G. Smith, J. Wall: Applying Grid Computing to 3D Medical Imaging Problems.

[4] Asim YarKhan, Jack J.Dongarra: Biological Sequence Alignment on The Computional Grid Using The Grads Framework. Kluwer Academic Publishers (2003)

[5] L.Robertson: LHC Computing Grid Project. http://lhcgrid.web.cern.ch/LHCgrid/presentations (2001)

[6] Dail, H. Obertelli, G. Berman, F. Wolski, R. Grimshaw: Application-aware scheduling of a magnetohydrodynamics application in the Legion metasystem. Heterogeneous Computing Workshop (2000)

[7] A.Ghiselli:DataGrid Prototype 1. http://web.datagrid.cnr.it/pls/portal30/docs/3180.pdf

[8] DataGrid-01-Ten-0118-1_1: WP1-WMS software administrator and user guide. Data Grid public document

[9] http://www.eu-datagrid.org/

[10] DataGrid-D1.2: http://server11.infn.it/workload-grid/docs/DataGrid-01-D1.2-0112-0-3.pdf: WP1: Workload Management (September 2001).

[11] M. Maheswaran, S. Ali, H.J. Siegel, D. Hensgen, and R.F. Freund: Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems. Journal of Parallel and Distributed Computing (November 1999) n. 59 pagg. 107-131.

[12] Martin T. Hagan, Howard B. Demuth,Mark Beale: Neural Network Design. PWS Publishing Co, Boston, MA, USA (1997).