Chapter III
Assisting Efficient Job Planning and Scheduling in the Grid

Enis Afgan
University of Alabama at Birmingham, USA

Purushotham Bangalore
University of Alabama at Birmingham, USA

ABSTRACT

Grid computing has emerged as the next generation computing platform. Because of the resource heterogeneity that exists in the grid environment, user jobs experience variable performance. Grid job scheduling, or selection of appropriate mappings between resources and the application, with the goal of leveraging available capacity and imposed requirements is thus at the heart of successful grid utilization. Grid job scheduling can be viewed as a function of resource heterogeneity, resource and application availability, and application options. This chapter presents work that incorporates all of these factors to provision and present individual users with alternative job options in terms of cost and time tradeoffs. Inherently, this leads to more effective scheduling policies. To support these aims, a framework is introduced with a novel scheduling methodology that introduces new user-scheduler interaction levels and a new layer of scheduling that includes application parameter selection and parameter value optimization.

INTRODUCTION

Over the past few years, grid computing (Foster and Kesselman, 1999) has gained popularity as the emerging architecture for next-generation high performance distributed computing. Goal of providing ubiquitous access to distributed, High Performance Computing (HPC) resources that are shared between multiple organizations through “virtualization” and “aggregation”, is only as efficient as its overall perception by the end users though. In order to realize this goal, grid
Assisting Efficient Job Planning and Scheduling in the Grid

middleware provides a standard set of services for authentication, authorization, resource allocation and management, job submission and monitoring, as well as data transfer and management (Berman et al., 2003), thus providing necessary abstractions from individual resources. Because grid provides access to heterogeneous pool of resources, grid job scheduling, or selection of appropriate mappings between resources and application, is at the heart of grid success. However, this scheduling process is perplexed with complexities due to resource availability, application and resource dependencies (Berman, 1998), as well as user goals and requirements. In order for grid to gain wide spread appeal and make efficient use of available resources, grid scheduling needs to be raised to the level where it can accommodate for such aspirations. User experience has to be tailored to support the individual user and provide them with the options they need and desire.

Access to grid resources is typically handled through a job submission interface (e.g., a web-based portal (Gannon et al., 2003), command line interface) where the user is requested to supply job options and parameters. Options requested depend on the scheduling engine employed behind the submission interface and can range from as few as application name and job input files to as many as an individual application supports (e.g., requirements for number of processors employed, amount of memory needed, speed of data transfer). If the latter grid job submission interface is perceived from a perspective of a typical user, such as an applied scientist, all the available resources and choices might appear equivalent and inherent differences would not be recognized, while the selection of which resource to run the job on might be random or based on previous experiences. In addition, once a routine has been established, even though the resource availability, input data, algorithms, or even the applications change, the user may always use the same resource and/or options. The resulting observation is that grid experience does not meet user’s expectations while leading to underutilized and inefficient use of resources (in terms of both, cost and time).

Grid job scheduling can be viewed as a function of resource heterogeneity, resource and application availability, and application options. In this paper, a grid scheduling framework is presented that advances grid job scheduling to include all of above stated parameters. Through incorporation of these factors, the goal of this work is to provision and present users with concrete and detailed options regarding their jobs, as well as provide more effective and efficient scheduling policies. To support these aims, the framework introduces a novel scheduling methodology that introduces new user-scheduler interaction levels and a new layer of scheduling that includes application parameter selection and parameter value optimization. To accommodate for the overall framework goals, individual components within the proposed framework will allow specification of application requirements, enabling initial application registration on the grid, an application profiling system developed and customized for data analysis tools, data analysis tools geared toward job parameter optimization, and resource-job cost normalization environment enabling comparisons between resource cost provider formats. These tools act as support mechanisms for the framework as a whole to assist users with planning and scheduling their job execution on the grid.

Aim of this work is to realize cost-to-application-runtime tradeoffs from user’s perspective. This is accomplished by providing job scheduling alternatives, thus providing support for situation-oriented user requirements and considerations. The act of resource selection and scheduling is the central point in all aspects of the grid and, as the grid evolves, will be a key to account management and cost tracking. Providing users with easily understood and applicable, yet targeted job alternatives is thus a key requirement for the grid to see wide overall acceptance.