Parameterized specification, configuration and execution of data-intensive scientific workflows

Vijay S. Kumar · Tahsin Kurc · Varun Ratnakar · Jihie Kim · Gaurang Mehta · Karan Vahi · Yoonju Lee Nelson · P. Sadayappan · Ewa Deelman · Yolanda Gil · Mary Hall · Joel Saltz

Received: 7 November 2009 / Accepted: 16 March 2010 / Published online: 10 April 2010 © Springer Science+Business Media, LLC 2010

Abstract Data analysis processes in scientific applications can be expressed as coarse-grain workflows of complex data processing operations with data flow dependencies between them. Performance optimization of these workflows can be viewed as a search for a set of optimal values in a multidimensional parameter space consisting of input performance parameters to the applications that are known to affect their execution times. While some performance parameters such as grouping of workflow components and their mapping to machines do not affect the accuracy of the analysis, others may dictate trading the output quality of individual components (and of the whole workflow) for performance. This paper describes an integrated framework which is capable of supporting performance optimizations along multiple such parameters. Using two real-world applications in the spatial, multidimensional data analysis domain, we present an experimental evaluation of the proposed framework.

Keywords Scientific workflow · Performance parameters · Semantic representations · Grid · Application QoS

1 Introduction

Advances in digital sensor technology and the complex numerical models of physical processes in many scientific domains are bringing about the acquisition of enormous volumes of data. For example, a dataset of high resolution image data obtained from digital microscopes or large scale sky telescopes can easily reach hundreds of Gigabytes, even multiple Terabytes in size. These large data volumes are transformed into meaningful information via data analysis processes. Analysis processes in scientific applications are expressed in the form of workflows or networks of interdependent components, where each component corresponds to an application-specific data processing operation. Image datasets, for instance, are analyzed by applying workflows consisting of filtering, data correction, segmentation, and classification steps. Due to the data and compute intensive nature of scientific data analysis applications, scalable solutions are required to achieve desirable performance. Software systems supporting the analysis of large scientific datasets implement several optimization mechanisms to reduce execution times. First, workflow management systems take advantage of distributed computing resources in the Grid. The Grid environment provides computation and storage resources; however, these resources are often located at disparate sites managed within different security and administrative domains. Workflow systems support execution of workflow components at different sites (Grid nodes) and reliable, efficient staging of data across the Grid nodes. A Grid node may itself be a large cluster system or a potentially

1DMetrix array microscopes can scan a slide at 20x+ resolution in less than a minute. The Large Synoptic Survey Telescope will be able to capture a 3.2 Gigapixel image every 6 seconds, when it is activated.
heterogeneous and dynamic collection of machines. Second, for each portion of a workflow mapped to such clusters, they enable the fine-grain mapping and scheduling of tasks onto such machines.

The performance of workflows is greatly affected by certain parameters to the application that direct the amount of work to be performed on a node or the volume of data to be processed at a time. The optimal values of such parameters can be highly dependent on the execution context. Therefore, performance optimization for workflows can be viewed as a search for a set of optimal values in a multidimensional parameter space, given a particular execution context. Workflow-level performance parameters include grouping of data processing components comprising the workflow into ‘meta-components’, distribution of components across sites and machines within a site, and the number of instances of a component to be executed. These parameters impact computation, I/O, and communication overheads, and as a result, the total execution time. Another means of improving performance is by adjusting component-level performance parameters in a workflow. An example of such a parameter is the data chunk size in applications which analyze spatial, multidimensional datasets. Another example is the version of the algorithm employed by a component to process the data.

We classify workflow-level and component-level performance parameters into two categories:

(i) Accuracy-preserving parameters (such as data chunk size) can affect the performance of an operation without affecting the quality of the analysis output, and (ii) Accuracy-trading parameters can trade the quality of the output for performance gains, and vice-versa. An example of an accuracy-trading parameter is the ‘resolution’ at which image data are processed. A classification algorithm might process low-resolution images quickly, but its classification accuracy would likely be lower compared to that for higher resolution images. When optimizations involve accuracy-performance trade-offs, users may supplement their queries with application-level quality-of-service (QoS) requirements that place constraints on the accuracy of the analysis [17]. For instance, when images in a dataset are processed at different resolutions to speed up the classification process, the user may request that a certain minimum accuracy threshold be achieved.

In this paper, we describe the design and implementation of a framework that can support application execution in a distributed environment and enable performance optimization via manipulation of accuracy-preserving and/or accuracy-trading parameters. We present an instance of our framework that integrates multiple subsystems at different levels:

- Wings [12] to facilitate high-level, semantic descriptions of application workflows
- Pegasus [11], Condor [23], and DataCutter [2] to support scalable workflow execution across multiple institutions and on distributed clusters within an institution
- ECO [5] to enable compiler optimizations for fine-grain computations executing on specific resources

Application developers and end-users can use our framework to provide high-level, semantic descriptions of application structure and data characteristics. As our initial focus is on addressing performance requirements in spatial, multidimensional data analysis applications, we have developed extensions to core ontologies in Wings to be able to describe spatial datasets and also to enable automatic composition and validation of the corresponding workflows. Once a workflow has been specified, users can adjust workflow-level and component-level parameters based on their QoS requirements to enable performance optimizations during execution. As part of this effort, we have also extended Condor’s default job-scheduling mechanism to support performance optimizations stemming from accuracy-performance related trade-offs. We show how our framework supports parameter-based optimizations for real biomedical image analysis workflows using two cluster systems located at two different departments at the Ohio State University.

2 Related work

Workflow management systems for the Grid and Services Oriented Architectures, such as Taverna [22], Kepler [19] and Pegasus [11] seek to minimize the makespan by manipulating workflow-level parameters such as grouping and mapping of a workflow’s components. Our framework extends such support by providing the combined use of task- and data-parallelism and data streaming within each component and across multiple components in a workflow to fully exploit the capabilities of Grid sites that are high-end cluster systems. Glatard et al. [13] describe the combined use of data parallelism, services parallelism and job grouping for data-intensive application service-based workflows. Our work is in the context of task-based workflows where execution plans are developed based on abstract workflow descriptions. We also address performance improvements by adjusting domain-specific component-level parameters.

The Common Component Architecture (CCA) forum\(^2\) addresses domain-specific parameters for components and the efficient coupling of parallel scientific components. They seek to support performance improvements through the use of external tunability interfaces [4, 21]. The Active Harmony system [8, 9] is an automatic parameter tuning system that permits on-line rewriting of parameter values at run-time.