Imbalance Optimization in Scientific Workflows

Weiwei Chen
Information Science Institute
University of Southern California
Marina del Rey, CA, U.S.A.
wchen@isi.edu

Ewa Deelman
Information Science Institute
University of Southern California
Marina del Rey, CA, U.S.A.
deelman@isi.edu

Rizos Sakellariou
School of Computer Science
University of Manchester
Manchester, U.K.
rizos@cs.man.ac.uk

ABSTRACT
Scientific workflows are a means of defining and orchestrating large, complex, multi-stage computations that perform data analysis and/or simulation. Task clustering is a runtime optimization technique that merges multiple short workflow tasks into a single job such that the job execution overhead is reduced and the overall runtime performance of the workflow is significantly improved. However, current task clustering strategies fail to consider the imbalance problem of both task runtime and task dependency. In our work, we first investigate the different causes of runtime imbalance and dependency imbalance. We then introduce a series of metrics based on our prior work to measure the severity of runtime and dependency imbalance respectively. Finally, we study a wide range of real scientific workflows to generalize the relationship between these metrics and balancing methods.

Categories and Subject Descriptors
D.1.3 [**Concurrent Programming**]: Distributed Programming;

Keywords
Scientific workflow, data locality, overhead, balance

1. SUMMARY
In recent years, with the emergence of the fourth paradigm for science discovery, scientific workflows received a lot of attention among many science disciplines. Although the majority of the tasks within these workflow applications are often relatively small (from a few seconds to a few minutes in runtime), in aggregate, they represent a significant amount of computation and data. When executing these applications on a multi-system parallel environment, such as the Grid or the Cloud, significant system overheads may exist [2]. In order to minimize the impact of such overheads, task clustering [1] has been developed to merge small tasks into larger jobs so that the number of computational activities is reduced and their computational granularity is increased thereby reducing the (mostly scheduling related) system overheads. There are several challenges that are not yet addressed.

The first challenge is the imbalance measurement. Tasks may have diverse task runtimes and merging tasks into jobs that have diverse runtimes increases the difficulty of load balancing; this may lead to *runtime imbalance*. At the same time, merging non-dependent tasks that share no intermediate data between them seems safe at the first sight. However, any subsequent tasks that rely on the output data that their parent tasks produce may suffer a data locality problem since data may be distributed poorly and the data transfer time is increased; this may lead to *dependency imbalance*. None of the existing work has provided a solution to evaluate the severity of the two types of imbalance together. The first contribution we make is the classification of imbalance into two categories, runtime imbalance and dependency imbalance, and the specification of measurement metrics to evaluate their respective severity.

Existing workflow task clustering strategies have ignored or underestimated the imbalance problem a workflow may suffer. In [1][3], task clustering strategies use structural information of the workflows to develop algorithms to iterate through the nodes, for example, horizontal clustering that merges tasks at the same level of the workflow. However, such strategies typically ignore runtime information that may significantly influence overall runtime performance. As our second contribution, we first introduce the basic runtime-based clustering method and then two other balancing methods that can address the dependency imbalance problem. A trace-based simulation is performed on a series of widely used scientific workflows. In our experiments, we randomly selected 20% of workflow tasks and increased their runtime from 0% to 100% to simulate various runtime variations. For a selected astronomy application, we observed a speedup of up to 1.45 compared to a no clustering executions.

However, what makes this problem even more challenging is that the solutions are usually conflicting. For example, balancing runtime may aggravate the dependency imbalance problem, and vice versa. Dependency imbalance usually means that task clustering at one horizontal level forces the tasks at the next level (or even the next few levels) to have severe data locality problems. At the same time, runtime balancing methods are only concerned with the current level in a workflow. Thereby the third challenge is to provide a quantitative measurement of workflow characteristics as a criterion to select and balance these solutions. To achieve this goal, we studied a wide range of scientific workflows to establish a relationship between imbalance metrics and the balancing methods.

ICS'13, June 10–14, 2013, Eugene, Oregon, USA.

ACM 978-1-4503-2130-3/13/06.

2. REFERENCES
[1] Singh, G. et al. Workflow task clustering for best effort systems with Pegasus. *Mardi Gras Conference '08* (2008).

[2] Chen, W. and Deelman E. Workflow Overhead Analysis and Optimizations. The 6th Workshop on Workflows in Support of Large-Scale Science, Nov 2011, 11-20

[3] Chen, W. and Deelman, E. Integration of Workflow Partitioning and Resource Provisioning, *CCGrid 2012, Doctoral Symposium*, Ottawa, Canada, May 2012