

July 31st 2014

Dear Editor and Reviewers,

We would like to thank you for your thorough review and the useful comments and suggestions formulated about our paper “Imbalance Optimization and Task Clustering in Scientific Workflows” submitted to Future Generation Computer Systems.

The remainder of this letter contains our answers to your reviews.

We thank you again for your valuable review, and we remain available for any further information you may need.

Sincerely yours,

Weiwei Chen, Rafael Ferreira da Silva, Ewa Deelman, Rizos Sakellariou

Reviewer 1

Reviewer comment:

There happen some minor language errors from time to time, but nothing serious.

Answer: We revised the manuscript to improve the text.

Reviewer comment:

There are some places where article requires reorganization: e.g. description of table 2 and the text of paragraph are one-sentence short and almost identical giving the impression that the table has two identical signatures one at the top and on the bottom.

Answer: We improved the descriptions of the tables and the paragraph to avoid this problem of identical signatures.

Reviewer comment:

There is a gap in all proposed algorithms (No. 1,2,3). Authors introduce two input parameters that control algorithms: First, C - maximum allowed number of tasks per job and second R - maximum allowed number of jobs per horizontal level of the directed acyclic graph of the workflow. In the case of mismatched values of C and R (e.g. too small values of both parameters in relation to number of tasks at some level of the workflow DAG) algorithm would miss some tasks. For comparison, algorithms cited from literature (4,5,6) do not have this problem.

Answer: We fixed the description of the three algorithms.. In practice, either C or R will be set and we get the other parameter by following $C \times R =$ the total number of tasks in a level (as shown in Algorithm 1). Usually we would set the R to be one or two times of the number of available VMs and our prior work[16] has evaluated the performance with different graunlarity. We perform task clustering in a level by level approach and thus the pair of C and R are calculated within the level.

Reviewer 2

Reviewer comment:

The authors say that : “In this paper, we focus on novel quantitative metrics that are able to demonstrate the imbalance problem in scientific workflows.” Load imbalance in parallel execution is unavoidable in any real situation, so showing it happens in scientific workflows is somehow expected.

Answer: The sentence was rewritten to highlight the contribution of our work with respect to the literature: *“In this work, we introduce and use quantitative imbalance metrics to perform task clustering in scientific workflows.”*

Reviewer comment:

The paper focus on scientific workflows to state what are the main imbalance issues during parallel scientific workflow execution. However, it is not clear what are the load balancing differences between scientific workflows and other data flow parallel scenarios like map/reduce or database operations.

Answer: Actually, the focus of this work is not to show the main imbalance issues during parallel scientific workflow execution. Instead, we focus on how to use imbalance metrics to assist task clustering in a way that the task groups are more balanced (e.g. runtime or data dependency), and thus system overheads such as queuing time and data transfer are reduced. We changed the title of the manuscript to “Using Imbalance Metrics to Optimize Task Clustering in Scientific Workflow Executions” to better reflect the focus of our work. Also, we changed some sentences in the introduction to highlight our main focus and contributions.

Reviewer comment:

According to the Introduction Section, “The quantitative metrics and balancing methods were introduced and evaluated in [16] on five workflows.” However, it was difficult to identify the “novel quantitative metrics” as the paper’s contribution along the paper.

Answer: This manuscript is an extension of our previous paper published in a conference [16]. The term “novel” in this context means that these metrics were not used in any work before ours. We removed the term to avoid misunderstandings. The main contributions of this manuscript are summarized in the introduction section: *In this manuscript, we complement this previous work by studying (i) the performance gain of using our balancing methods over a baseline execution on a larger set of workflows; (ii) the performance gain over two additional task clustering methods from the literature; (iii) the performance*

impact of the variation of the average data size and number of resources; and (iv) the performance impact of combining our balancing methods with vertical clustering.

Reviewer comment:

It seems that in this paper the authors evaluate previous results on different scenarios and made some comparisons with task based clustering techniques from the literature, and finally combined the balancing methods with a vertical clustering technique. One of the difficulties is generalizing the results. There is a rich set of simulation results, but combining them within the load imbalance problem is difficult. What are the heuristics that should be followed? What should be done when designing a scientific workflow load balancing algorithm? The contribution presented in this paper is fragmented, which makes it difficult to take advantage of the results from the simulated experiments.

Answer: We added a decision tree to the end of the evaluation section that combines all the balancing methods proposed in this work into a single heuristic. We believe that this decision tree may drive the development of police engines to interactively select a balancing method.

Reviewer comment:

One of the new results is exploring vertical clustering, but as the authors mention, using VC is always a good idea in the presence of pipelines, but it is limited by the properties of the workflow definition (synchronization).

Answer: The added value in using vertical clustering is to ensure that pipelines are prioritized (VC-*prior*) and not broken by horizontal clustering techniques. This results can be seen, for instance, in Fig. 27 (former Fig. 26).

Reviewer comment:

In the Related Work Section, it seems that the main difference between related work and the paper’s contribution is taking data dependency into account. This is an important issue in scientific workflows, but there are several other aspects in a data-flow execution, which is typical in scientific workflows, in addition to data locality. For example, the techniques only consider imbalance on task’s execution time. However, data skew is also critical in data-flows like scientific workflows. Different input data partitioning has not been evaluated. Even if the workflow input data is uniformly distributed among horizontal clusters or vertical clusters, that does not guarantee uniform execution time along the data-flow running on

the workflow tasks. Predicting these data skew situations depends on previous knowledge of data characteristics. Differences on data convergence, similarity and other data properties may lead to poor load balance, all very complex to predict before the execution, even through sampling. The experiments only evaluate differences in data size, but not in data value distribution, like data skew. Please show experimental results with better data, such as in the presence of data skew.

Answer: We agree that data skew is also important, but it is not the only source of imbalance. Our experimental dataset is based on real workflow execution traces, and the data distribution is not uniform. It is possible that data skew may arise when synthetic workflows are generated.

Reviewer comment:

The number of single core VMs = 20 is disappointing in a load imbalance evaluation, since balancing becomes more complex as more cores or VMs are added. The motivation of the paper if for large scale executions. The explanation of using 20 single core VMs (account limitations in Amazon) is not convincing. The results of Fig 22 make it evident that as more VMs are added, the gains decrease. Considering the decrease rate in performance gains from 2 to 20 VMs, shown in Fig 22, it seems that no gain would be obtained after 40 VMs, which is still small scale. Considering all VM homogeneous is also unrealistic, even in cloud providers that guarantee a specific performance. Since the experiments are done by simulation, using more VM should not be a problem. Please show experimental results with better data, such as at larger scale.

Answer: We agree that 20 virtual machines do not represent large-scale execution. We used this amount of VMs for the sake of simplicity. We conducted a new experiment (Fig. 24) where we vary the number of VMs between 200 and 1800 and show that the results are proportional equivalent to the ones obtained with 20 VMs. This result confirms that the same imbalance properties are present in the case of 20 VMs.

Reviewer comment:

The results are all shown as performance gains over the baseline. The balancing “accuracy” was not evaluated. For example, idleness in processing nodes is typically measured when evaluating load balancing techniques. It is important to know the precision of the imbalance prediction, since there are so many execution situations that may incur in imbalance (CPU performance, IO operations, data skew like non-uniform selectivity or similarity in specific values). For example, some load balancing systems (e.g. Falkon) show limitations

of load balancing under failures or when facing unpredictable task execution times. What is the overhead for imbalance prediction ? It seems that the algorithms already have a task performance value distribution before the execution and this sampling/prediction time was not considered while comparing with baseline.

Answer: As stated in a previous answer, the focus of this work is not to address the main imbalance issues during workflow execution, but the development of imbalance metrics to improve the performance of task clustering. We consider that the evaluation of the imbalance prediction is out of the scope of this work.