Optimization of Compute Costs in Hybrid Clouds with Full Rescheduling

Sophoan Sok*, Christian Plewnia†, Sansiri Tanachutiwat*, Horst Lichter†
* The Sirindorn International Thai-German Graduate School of Engineering
King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
Email: {sophoan.s-sse2018, sansiri.t}@tggs.kmutnb.ac.th
† Research Group Software Construction
RWTH Aachen University, Aachen, Germany
Email: {plewnia, lichter}@swc.rwth-aachen.de

***Abstract*—Hybrid clouds enable cost optimizations when handling fluctuating workloads. Assuming that the private cloud is cheaper than the pay-per-use public cloud when it is constantly used, a cost-optimal hybrid operation means that the private cloud handles the base workload and the public cloud the workload peaks. The tasks that cause the workload have to be scheduled such that the private cloud's usage is maximized. Ideally, the scheduler knows the future tasks and can plan for maximizing the private cloud's usage. However, obtaining such an accurate prediction of the future is not always possible.**

We propose an approach called *full rescheduling* that allows online schedulers to move already running tasks from the public to the private cloud in order to save costs. We further present an assessment method for studying the economic benefit of full rescheduling and apply it in an industry case study for a data processing application. The study's results show an economic benefit due to an increased usage of the private cloud instead of the public cloud. The method also provides a cost-optimal number of nodes for the private cloud.

***Index Terms*—full rescheduling, hybrid clouds, compute cost optimization, scheduling simulation, hybrid cloud cost model**

I. Introduction

Hybrid clouds combine local computing infrastructure, i.e. the private cloud, with computing resources of a public cloud provider [1]. One of the public cloud's benefits is that it grants the flexibility to obtain resources on-demand and pay only for the time the resources are used. In a hybrid cloud, this flexibility can be used to handle an over the course of the time changing workload. Here, the term workload refers to the amount of work caused by any tasks, such as services or data processing jobs, executed in the hybrid cloud. For example, services used by consumers in one geographic region might exhibit peak workload situations during evening hours when most consumers have leisure time. When the workload grows beyond the private cloud's capacity, the computing infrastructure can be supplemented by public cloud resources. Conversely, when the workload decreases, the public cloud resources can be released, and eventually, the private cloud might be able to server the workload on its own again.

Running tasks in a hybrid cloud requires deciding whether to assign them to private or public cloud resources. Assuming that the private cloud is paid for independent of whether its resources are used, a cost-optimal hybrid cloud operation must prefer using the private cloud resources whenever possible. Unfortunately, accurately estimating the future workload development is usually not a trivial endeavor. Without this knowledge about the future, it can happen that tasks are assigned to the public cloud while the private cloud is at capacity, only to observe shortly after that the workload for tasks in the private cloud decreases, thereby releasing resources, while the resources in the public cloud are still in use. This is a situation where costs occur that could be avoided if we either had an accurate estimate of the future and made an according scheduling decision or if we could freely move running tasks between public and private cloud.

Generally, having an accurate estimate of the future is preferable over moving running tasks due to the costs the later involves. Moving a running software process along with its state and possibly large amounts of data is not trivial and comes at a price. For example, the software must either be paused, if it supports a mechanism to freeze its state, or stopped; then, the state and data must be moved from a node in the public to the private cloud, which takes time and might create traffic costs; finally, the process must be started again on a node in the private cloud. These costs can be anything from negligible, e.g. for a stateless micro service, to infeasible, e.g. for a data processing job that works with massive amounts of data. However, the alternative of obtaining an accurate estimation of the future workload that allows an optimal scheduling in advance is often hard, if not impossible to obtain. Then, moving tasks might be the best available option.

In this paper, we investigate the concept of moving tasks already assigned to compute resources. Since it is a scheduler's job to assign tasks to compute resources, we suggest that schedulers should also be responsible for deciding when to move already running tasks from the public to the private cloud. We call schedulers that are able to do so capable of *full rescheduling*. As argued before, full rescheduling introduces overhead, and it is unclear whether the benefit of saving costs by using the public cloud is larger than the costs caused by the overhead. Consequently, we investigate in this paper how to find out whether full rescheduling is an economically beneficial approach. To this end, we study the following two

2020 IEEE International Conference on Smart Cloud (SmartCloud), pp 35-40
DOI 10.1109/SmartCloud49737.2020.00016

research questions:

1) Will the schedules created by a full rescheduling capable scheduler yield an economic benefit compared to the schedules created by a similar scheduler without the full rescheduling ability?
2) What is the optimal number of compute nodes in the private cloud of a hybrid cloud such that compute costs are minimized?

In this paper, we first discuss related literature in section II and provide a more detailed description of full rescheduling in section III. In order to answer our research questions, we designed an assessment method that we present in section IV. The method uses a scheduling simulation in combination with a cost model. The simulation allows us to compare the schedules of a scheduler with and without the full rescheduling ability. The cost model is then used to evaluate the usage costs of the schedules produced by both schedulers. We further apply this method in an industry case study that we present in section IV. On the one hand, this case study illustrates our assessment method and, on the other hand, shows for one case whether full rescheduling yields an economic benefit. Finally, we conclude our paper in section VI.

II. Related Work

A number of research publications addressed cost-constrained scheduling when using a public cloud. Further, there is also previous research that considered rescheduling. Since both of these topics are related to our topic we present the respective pieces of literature here.

A. Cost-aware Schedulers

Followed by the introduction of cloud computing, execution costs are widely considered in scheduling problems since users are charged on a pay-per-use basis. To solve this scheduling problem in cloud environments, a self-adaptive global search-based optimization technique, Particle Swarm Optimization (PSO), is adopted by Pandey et al. to schedule workflow applications to cloud resources so that the execution costs are considered [2].

In a hybrid cloud environment, this problem becomes even more complicated as schedulers have to split the tasks among resources on both private and public cloud effectively. Bittencourt and Madeira proposed the Hybrid Cloud Optimized Cost (HCOC) scheduling algorithm with the goal of reducing the workflow makespan in order to meet the desired deadline while minimizing the costs [3]. Note that makespan refers to the timespan between the first tasks' start and the completion of all tasks. In the beginning, this algorithm generates a schedule where the tasks are assigned to the private cloud only. After that, a task is selected to be rescheduled to a resource from the public cloud if the workflow makespan does not meet the deadline. This is repeated until the makespan complies with the deadline.

To solve the cost and makespan optimized scheduling, Zhou et al. proposed two different approaches [4]. One is a single-objective workflow scheduling optimization technique, called Deadline-Constrained Cost Optimization for Hybrid Clouds (DCOH). Both approaches are based on genetic algorithms, and they are studied with the assumption that the execution time of tasks in the workflow is fixed.

B. Rescheduling

Cost-aware schedulers require an accurate prediction of execution information, such as a task's execution time, to determine the execution cost. However, in many cases, it is not possible to obtain accurate predictions due to changes of the execution environment. In order to deal with this uncertainty, rescheduling techniques have been proposed and employed as a part of some scheduling algorithms.

Sakellariou and Zhao proposed a selective rescheduling technique to perform the scheduling again on unexecuted tasks when there is any exceeding pre-limited delay between the actual and expected start time of a task [5]. The objective of their technique is to ensure the makespan of the workflow is optimized and the frequency of rescheduling attempts is minimized.

Rescheduling is also used directly as an essential step of some scheduling algorithms. In the previously mentioned HCOC scheduling algorithm [3], to optimize execution and maintain deadline, it repeats the step to reschedule the task, which is mapped to private cloud to a resource in the public cloud until the deadline is met (or the limited number of iteration is reached). Similarly, the Particle Critical Paths (PCP) algorithm [6] can reschedule the workflow execution without any further modification by reinvoking the planning procedure. With this rescheduling ability, the PCP is resilient to unexpected events, e.g. a task of a workflow does not finish in time, or computation is terminated because of a failing node. When such events happen, PCP can take corrective actions like rescheduling the subsequent tasks of the failed task to nodes with higher performance in order to speed up the remaining workflow and minimize the risk of missing the deadline.

III. Full Rescheduling

Since the methods described in the related work are not suitable when the task arrival time and execution time cannot be predicted accurately, we propose an alternative approach.

A *full rescheduling* strategy is an online scheduling strategy that considers already running tasks as well as queued tasks. The strategy's scheduler should be called repeatedly on every change of the environment in order to allow the scheduler to react to those changes and yield a new schedule.

With a full rescheduling strategy, not only the tasks in the queue but also the currently running tasks are considered in scheduling decisions. Thereby, the scheduler is able to revise its previous scheduling decisions. The reason for this is to cope with unexpected events and react to changes in the information used in past scheduling decisions. However, the revised schedule may require to suspend those running tasks so they can be reallocated to different resources and continue the execution. Rescheduling tasks introduces an extra delay for data processing tasks that may lead to a longer overall

2020 IEEE International Conference on Smart Cloud (SmartCloud), pp 35-40
DOI 10.1109/SmartCloud49737.2020.00016

makespan. In case that business value is generated by tasks finishing as early as possible or before a deadline, rescheduling a task may also cause additional costs.

Whether or not full rescheduling yields a benefit also depends on the workflows that may be affected by reallocations. For example, if reallocations involve a costly transfer of huge amounts of data from the public to the private cloud or if the workflow cannot be resumed and must be restarted from the beginning, there may not be a benefit at all. Thus, there is a need for a method that helps to evaluate whether full rescheduling is beneficial for a specific scenario.

IV. Assessment Method for Full Rescheduling

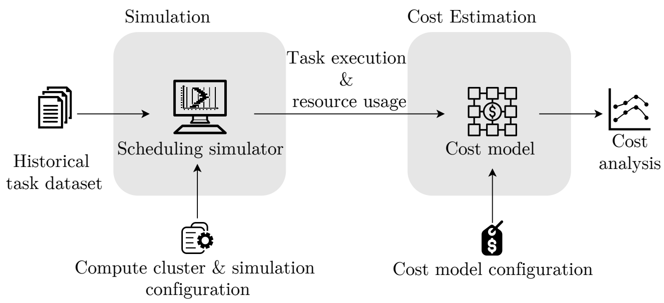


Fig. 1: Overview of the assessment method's process

To answer both our research questions, we designed an assessment method that allows us to compare full rescheduling enhanced strategies with existing scheduling strategies. This assessment method consists of two steps, as illustrated in figure 1. The first step of the assessment method is to simulate scheduling decisions based on a dataset of recorded tasks. This allows us to compare the generated schedules; then, we compare the costs that would occur if the schedules obtained during the simulation would be implemented, which allows us to answer both research questions.

We assume that any task can be reallocated by pausing it, moving its state to another node, and then continuing its execution there. Note that this means that data processing applications that achieve a progress over time must be able to save its state at any time or at least implement a checkpointing mechanism, i.e. saving it frequently. Otherwise, rescheduling such a data processing task would lead to an expensive restart which we think is unlikely to yield a cost-benefit in the end.

We published the source code for the scheduling simulation, the spreadsheet implementing the cost model computations, and the complete case study results on GitHub[1].

A. Simulation

We propose a scheduling simulation to compare the behavior and resource usage of one scheduling strategy with and one without the full rescheduling ability. For this, we feed the simulation with a dataset describing a history of tasks that contains each task's enqueue time and duration and a configuration of the hybrid cloud that should be simulated.

[1]https://github.com/swc-rwth/full-rescheduling-paper, git commit: 3465990

The simulation is run for each of the schedulers separately. In the simulation, the time of $t = 0$ corresponds to the time of the first task being enqueued. The scheduler is asked to assign it to a compute node in the hybrid cloud. With passing time, this is repeated for every task in the dataset, and each task requires the recorded time to complete its execution. Whenever a task is enqueued, the full rescheduling strategy can decide to reschedule tasks from the public to the private cloud. The simulation ends when all tasks finished.

As a result of the simulations for both schedulers, we get the generated schedules, i.e. the assignment of the tasks to the compute nodes over time. Thereby we can study the effects of applying full rescheduling for a historical task dataset and hybrid cloud setup. We can further repeat the simulation for different hybrid cloud setups.

We implemented such a simulation based on the scheduling simulator ALEA [7]. Note that our implementation is limited to setups with homogeneous nodes in the private and public cloud and exclusive usage of a compute node by a task.

B. Cost model

The schedules generated by the simulation can be used as input to a cost model to assess the resource usage costs that occur with and without full rescheduling. For this, we used the cost model proposed by Kashef and Altmann for hybrid cloud environments [8].

This cost model considers 20 cost factors divided into the six groups electricity, hardware, software, labor, business premises, and service. Some of the cost factors describe variable costs that depend upon the resource usage provided by the simulation. For example, costs for electricity and public cloud resources depend on how much we use the private and public cloud, respectively. Other factors describe fixed costs independent of resource usage. Note that assigning values to the cost factors is not trivial. Possible sources are experiences gained by monitoring past resource usage and its costs, as well as researching current market price developments for hardware, software, and labor. With values for the cost factors and the simulation results, the cost model allows us to calculate the costs for the compute resource usage for the timespan described by the task dataset.

V. Case Study

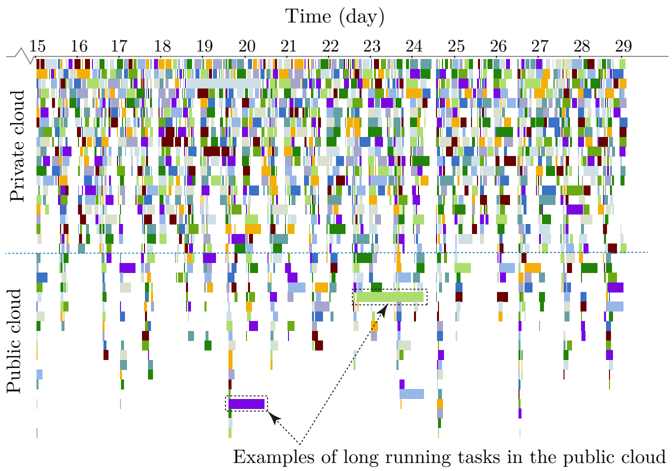
We applied the assessment method for a task dataset of a real-world data processing application. This data was extracted from log files and initially described 7,883 tasks over a period of the 29 days of February 2020. Of these tasks, 33 tasks were removed due to missing information. The remaining 7,850 tasks amount for total execution time of about 468 days and 23 hours. As schedulers, we used the First Come First Serve with Cloud Awareness (FCFS) and a full rescheduling enhanced version of the same (FCFS-FR). Since the FCFS is cloud-aware, it can prioritize the usage of the private cloud over the public cloud. We assumed that the task's processes are capable of resuming their work and that each task reallocation will take 5 minutes, which is an estimate including the time

2020 IEEE International Conference on Smart Cloud (SmartCloud), pp 35-40
DOI 10.1109/SmartCloud49737.2020.00016

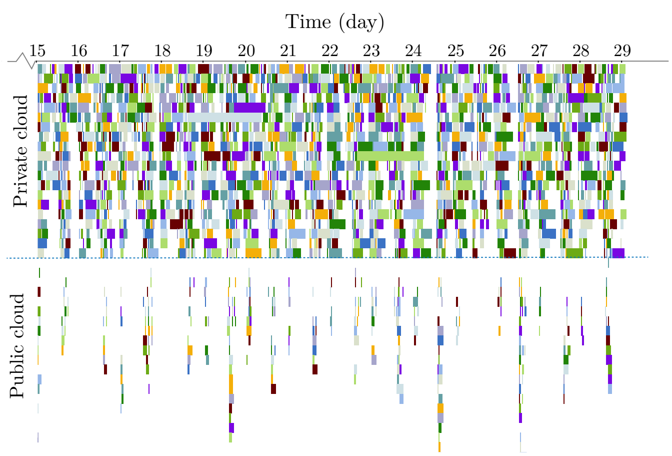
for transferring necessary data as well as starting the task's process. The simulation was repeated for multiple hybrid cloud setups with the number of private cloud nodes ranging from 10 to 45 nodes.

For the cost model factors, we conducted market research in June 2020. We considered the private cloud to be operated in a data center in Germany. We chose the Amazon Web Services (AWS) as the public cloud provider and retrieved the costs for the public cloud resources from AWS for Frankfurt region, Germany.

A. Results



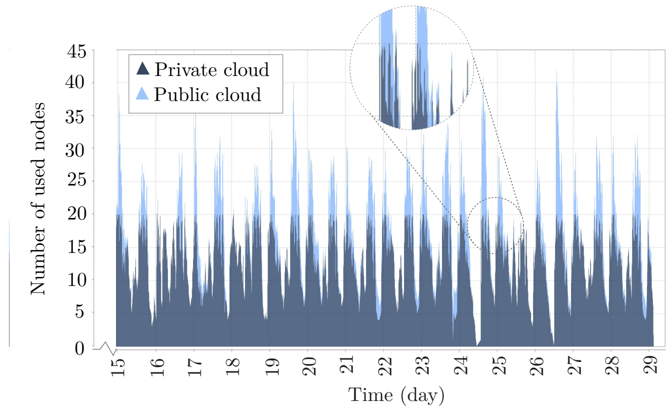
(a) Using FCFS



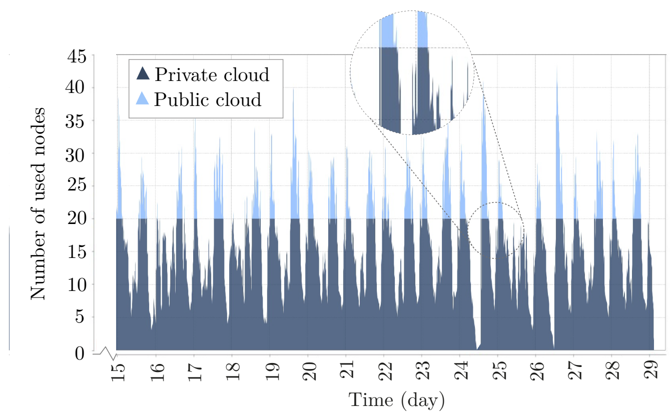
(b) Using FCFS-FR

Fig. 2: Schedule diagrams showing the allocation of tasks (colored rectangles) to nodes (rows) for the scenario with 20 compute nodes in the private cloud

The schedule diagram in figure 2a is the result of task execution from scheduling simulation by using FCFS in the scenario of 20 compute resources available in the private cloud. As indicated by the annotations in the figure, there are some long-running tasks scheduled to run in the public cloud, which has a higher usage cost than the private cloud. On the other hand, the schedule diagram 2b is the result of the task execution using FCFS-FR in the same scenario. Compared to the result from FCFS, the public cloud usage is lower. Without full rescheduling, we can observe some long-running tasks in the public cloud. This is no longer the case with full rescheduling since any long-running task is rescheduled to the private cloud as soon as there are private cloud resources available.



(a) Using FCFS



(b) Using FCFS-FR

Fig. 3: Compute resource usages for the scenario with 20 nodes available in the private cloud

Figure 3 illustrates the compute resource usage in the same scenario of 20 compute resources available in the private cloud. In the case of using FCFS, the number of allocated nodes in the private cloud fluctuates all the time – even during load peak periods. In contrast, when using FCFS-FR, all compute nodes in the private cloud are continuously fully utilized during peak periods.

The line graph in figure 4 illustrates the usage time of compute resources in the public cloud in different scenarios that there are the numbers of available compute resources ranging from 10 to 45 nodes in the private cloud. Overall, the usage time of compute resources in the public cloud from both the FCFS and the FCFS-FR declines gradually when the number of available compute resources in the private cloud increases. Also, the compute resources in the public cloud are

2020 IEEE International Conference on Smart Cloud (SmartCloud), pp 35-40
DOI 10.1109/SmartCloud49737.2020.00016

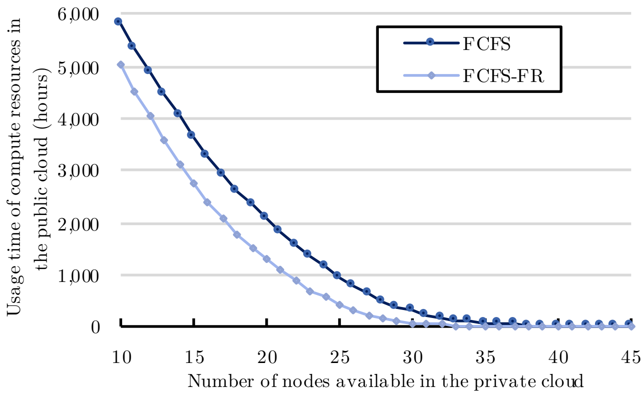


Fig. 4: Estimated usage of public cloud nodes for FCFS and FCFS-FR for all scenarios over a time period of 29 days

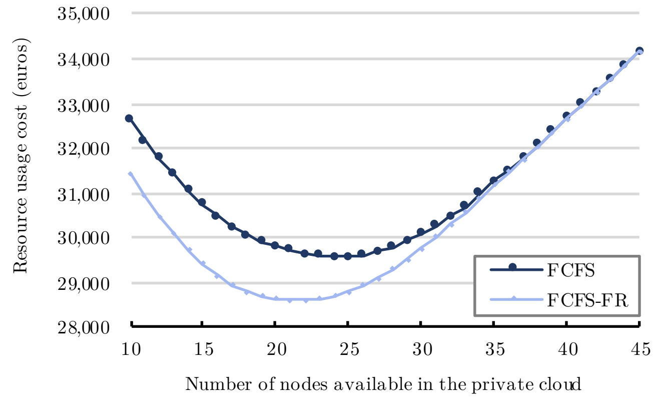


Fig. 5: Estimated compute costs for FCFS and FCFS-FR for all scenarios over a time period of 29 days

not used at all in the case of 45 nodes of compute resources available in the private cloud. If we compare the usage time of compute resources in the public cloud in case of using the FCFS-FR to the FCFS, there is a reduction of that usage time in all scenarios.

Private cloud nodes	Rescheduled tasks	Extra time due to rescheduling	Extra time relative to total runtime
10	2,903	10 days 03:13:51	2.16%
15	3,246	11 days 13:53:32	2.47%
20	2,856	9 days 22:54:43	2.12%
25	1,802	6 days 06:40:27	1.34%
30	524	1 days 23:20:21	0.42%
35	110	13:07:49	0.12%
40	20	01:39:53	0.01%
45	0	00:00:00	0.00%

TABLE I: Number of rescheduled tasks and extra time caused by the rescheduling for eight of the evaluation scenarios

As shown in table I, the number of rescheduled tasks is around 3,000 when there are 10 to 20 nodes in the private cloud. This number declines rapidly as the number of available nodes in the private cloud increases. The maximum of the extra time introduced to the total execution time due to the task reallocation is just over 11 days and a half in case there are 15 nodes available in the private cloud. However, this extra time is amounted to only 2.47% compared to all tasks' total execution time from the data source. Also, only around 2% of total execution time increases in half of all evaluation scenarios among all the simulation environment setups. In contrast, the increased rate of total execution time is less than 0.5% in the other half.

Figure 5 shows the estimated resource usage cost by using FCFS and FCFS-FR in varying simulation environment setups in which there are available nodes from 10 to 45 in the private cloud. Using the FCFS-FR, the estimated total resource usage cost shows a similar trend as the case of using the FCFS. However, compared with the case of using FCFS, for setups with 10 to 25 nodes in the private cloud, the total cost is lowered by around 1,200 euros in the studied time interval of February 2020. This difference in resource usage cost decreases and zero when there are 45 nodes available in the private cloud. Based on these results, the scenario that produces the minimum cost is 25 and 21, respectively, when using FCFS and FCFS-FR.

B. Discussion

By comparing the task execution from the scheduling simulation when using the FCFS and FCFS-FR illustrated in figure 2, there are some apparent differences in the task execution schedules. Also, the compute resource usage graphs in figure 3 shows that the compute resources in the private cloud are hardly fully utilized most of the time when using the FCFS, even during the peak workload periods. However, when using the FCFS-FR, the compute resources in the public cloud are used only when there are no available compute resources in the private cloud.

The most noticeable drawback of full rescheduling is the extra task execution time caused by the reallocation of running tasks. This extra time was in no scenario more than 2.47% of the total execution time of all tasks; this maximum occurs for the scenario with 15 nodes as shown in table I.

In our case study, the extra task execution time caused by the rescheduling does not lead to an increase of the makespan for all the 7,850 tasks in the dataset for February 2020. We think that the reason for this is the seasonal workload distribution of our case study's data processing application. Every day there are short peak workload periods during which running tasks are rescheduled. While the rescheduling may cause an accumulation of delays in this period, the delays are compensated later when the tasks can be processed on otherwise idle private cloud resources. We argue that effects like the impact on the makespan should be studied before considering implementing full rescheduling in order to make an informed decision whether the drawback of full rescheduling is acceptable. This can be studied using the assessment method we proposed.

In order to answer the first research question, i.e. whether full rescheduling yields an economic benefit, we obtained the estimated compute costs when applying FCFS as well

2020 IEEE International Conference on Smart Cloud (SmartCloud), pp 35-40
DOI 10.1109/SmartCloud49737.2020.00016

as FCFS-FR for the number of private cloud nodes ranging from 10 to 45. As shown in figure 5, in half of the scenarios the resource usage cost would have been lowered by around 1,200 euros for the time period of February 2020 if we had applied full rescheduling. Although the cost-saving effect declines when the number of private cloud nodes rises, it is still positive. The only case where there is no cost-saving effect, i.e. the compute costs are the same for FCFS and FCFS-FR, is for the scenario with 45 nodes in the private cloud; in this case there is no cost-saving potential as no public cloud resources are used. Thus, our case study shows that full rescheduling can yield an economic benefit. Note that this analysis does not consider the one-time costs for implementing full rescheduling, i.e. being able to reallocate running tasks.

As for the second research question, i.e. what the optimal number of compute nodes in the private cloud is, the proposed assessment method can be used to find a hybrid cloud setup with minimal compute costs. This is done by comparing the costs for a series of scenarios with different numbers of nodes in the private cloud. When planning a cost-optimized private cloud setup, the number of nodes have to chosen based on the scenario where the costs are minimal. Note that this only works as long as the workload behavior can be expected to be similar to what the used dataset described. Assuming that this is the case for our case study, then the best hybrid cloud setup would be a private cloud with 21 nodes supplemented by the on-demand compute resources in the public cloud.

VI. Conclusion and future Work

In this paper, we introduced the concept of full rescheduling for online scheduling strategies in hybrid clouds. We described an assessment method that allows to evaluate the cost-saving potential of full rescheduling without needing to implement it. We applied the method in a case study for a data processing application and showed that full rescheduling can yield an economic benefit. Further, we described how the assessment method can be used to find a number of nodes for the private cloud that minimizes the hybrid cloud's compute cost.

Further research is required to explore the practical value of full rescheduling. First, the assessment should be extended to support more kinds of scenarios. The currently limiting factors are the assumptions of having homogeneous compute nodes in the hybrid cloud, the exclusive usage of one compute node by a task, and that the simulation only supports setting one rescheduling delay for all tasks.

Second, the technical challenges of reallocating running tasks from one compute node to another need to be studied. This reallocation requires that the tasks are designed in a way that they can be paused, their state can be saved, and that they then can be continued. Further, a design for the management of the reallocation is required, i.e. notifying a task to pause, transferring its state, and triggering its start using its transferred state. This research could result in design recommendations or even a software framework that lowers the initial one-time effort for implementing full rescheduling.

Finally, it would be interesting to repeat the study presented in this paper for other cases and for other scheduling strategies than the first come first serve with cloud awareness strategy in order to learn more about when full rescheduling can yield an economic benefit.

References

[1] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.

[2] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. Perth, Australia: IEEE, 2010, pp. 400–407. [Online]. Available: http://ieeexplore.ieee.org/document/5474725/

[3] L. F. Bittencourt and E. R. M. Madeira, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds," *Journal of Internet Services and Applications*, vol. 2, no. 3, pp. 207–227, Dec. 2011. [Online]. Available: http://link.springer.com/10.1007/s13174-011-0032-0

[4] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, p. 101631, Nov. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1383762119302954

[5] R. Sakellariou and H. Zhao, "A Low-Cost Rescheduling Policy for Efficient Mapping of Workflows on Grid Systems," *Scientific Programming*, vol. 12, no. 4, pp. 253–262, 2004. [Online]. Available: http://www.hindawi.com/journals/sp/2004/930924/abs/

[6] S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1400–1414, Aug. 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6104035/

[7] D. Klusáček, M. Soysal, and F. Suter, "Alea – Complex Job Scheduling Simulator," in *Parallel Processing and Applied Mathematics*, R. Wyrzykowski, E. Deelman, J. Dongarra, and K. Karczewski, Eds. Cham: Springer International Publishing, 2020, vol. 12044, pp. 217–229, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-030-43222-5_19

[8] M. M. Kashef and J. Altmann, "A Cost Model for Hybrid Clouds," in *Economics of Grids, Clouds, Systems, and Services*, K. Vanmechelen, J. Altmann, and O. F. Rana, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7150, pp. 46–60, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-28675-9_4