Review of “RALBA: A Computation-aware Load Balancing Scheduler for Cloud Computing - Altaf Hussain, Muhammad Aleem, Abid Khan, Muhammad Azhar Iqbal, Muhammad Arshad Islam”

By Justinian-Ionut Petcu

In this paper review, I will present the work of Altaf Hussain, Muhammad Aleem, Abid Khan, Muhammad Azhar Iqbal, and Muhammad Arshad Islam on load balancing paradigms, more exactly on a Resource-Aware Load Balancing Algorithm (RALBA), an algorithm that ensures a balanced distribution of workload based on the resources of the corresponding VMs in the cloud.

Cloud computing is a very known business paradigm. The aspect most relevant to this review is the guaranteed on-demand availability of computing power over the Internet to be able to solve High-Performance Computing problems. Load balancers are used to distribute the work over a set of virtual machines.

The paper argues that the static scheduling strategies (preferred over dynamic strategies due to the low overhead in scheduling and outperforming them over the final results) usually produce poor resource utilization. This has serious repercussions on the environment and also affects costs overall. According to a review[i], 1.6 million tons of additional CO2 emissions are released only from the idle computing resources in data centers. Also, idle resources cost an additional 19 billion dollars in terms of resources and energy. To overcome this, scheduling mechanisms were proposed to achieve better load balance. A brief description of the most common scheduling techniques, their positive aspects, and their respective drawbacks in regards to load balancing and resource optimization will be discussed next.

Random selection (RS) allocates jobs without taking into account the current load of the virtual machines. It has a simple implementation and low overhead but may cause low resource utilization and load imbalance.

Round-Robin passes tasks to VMs in sequential order. When the last VM has received a task, the cycle is restarted with a new round. It is very similar to RS, as VMs with smaller capabilities could receive bigger tasks, and it may cause load imbalance.

Minimum Completion Time (MCT) scans every VM for each candidate task to determine the most suitable machine to complete the task. It may cause low resource utilization as faster resources will be overloaded.

Min-Min/Max-min are based on MCT but also take into account the minimum earliest finish time. They may penalize larger/smaller jobs and cause load imbalance.

Next, the RALBA algorithm and overview will be discussed. In the study, Cloudsim, a framework for modeling and simulation of cloud computing infrastructure, is being used to evaluate the performance of the scheduling algorithm. A cloud data center is formed from a collection of physical machines at the physical layer. The resources of these machines are shared using the virtualization concept thus creating the virtual instance layer. For a computation-aware scheduler a Cloud resource manager that can track the status of all the VMs in the virtual layer is needed. This resource manager should be able to create and terminate VMs when required, while also providing RALBA information about status and computing capabilities.

RALBA is made out of two sub-schedulers: Fill and Spill. Fill scheduler selects the VM j with the largest computing share for all the VMs available and determines the largest task from the remaining possible tasks and assigns it to VM j. The computing share of VM j is modified after the allocation. Fill scheduler will repeat this process until there does not exist any VM that can take any more tasks from the set of remaining possible tasks or the set of jobs to be scheduled becomes empty. After the fill scheduler, RALBA switches to the Spill scheduler to finish allocating the remaining jobs. Spill selects the most demanding job and matches it to the VM that produces the earliest finish time for that job. After the allocation, the expected completion time of the VM is updated. The Spill process is repeated until there are no more tasks to be assigned.

Regarding the complexity of RALBA, with M being the number of VMs in the cloud center and N being the total number of jobs to be scheduled, the worst case for the Fill scheduler where it finds the most powerful VM using M comparisons and determines the most extensive task after a maximum of N comparisons for each machine would be O(M^2N). For the Spill scheduler where it selects the most demanding job and matching it the VM with the lowest EFT the complexity would be O(Mn), where n is the remaining jobs after the Fill scheduler has finished execution. Taking into account that in real-world scenarios M is significantly lower than N, RALBA produces better execution time than Min-Min/Max-Min/RASA, TASA, and others.

RALBA was tested against other algorithms on two sets of workloads, a synthetic workload, and a Google-like realistic workload. The results are analyzed based on the following factors: average execution time, Average Resource Utilization Ratio (ARUR), and average throughput. *Each experiment is performed 5 times and the analysis is conducted on average values*. Next, the average comparison between RALBA and RR, RS, MCT, Min-Min, Max-Min, TASA, and Sufferage algorithms will be presented. RALBA has outperformed the other algorithms and the following table show only the margin that RALBA has over them in the synthetic workload and the Google-like workload.

	TASA	Sufferage	Max-Min	RASA	Min-Min	MCT	RS	RR
Makespan	5.5%	6.9%	243.9%	246.4%	46.1%	41.9%		
ARUR	10.9%	9.8%	103.3%	117.6%	121.6%	65%	624%	541.2%
Throughput	6.2%	7.5%	151.6%	160.1%	47.2%	42.1%	2047%	1812%

Table 1. Average values for the synthetic workload

	TASA	Sufferage	Max-Min	RASA	Min-Min	MCT	RS	RR
Makespan	3.8%	1.8%	47.5%	24.5%	28.1%	18.6%		
ARUR	16.9%	7.4%	15.9%	13.9%	78.8%	30.1%	550.6%	541.7%
Throughput	5.6%	3.1%	20.1%	8.4%	33.8%	21.9%	1125%	1785%

Table 2. Average values for the Google-like realistic workload

In conclusion, the authors have shown that inefficient resource utilization can cause reduced makespan and low throughput and offered a solution that was compared to the current state-of-the-art scheduling heuristics. The results have shown that RALBA has attained better execution time and throughput. The exact algorithm is also available in the study and the authors plan to further enhance this technique.

[i] P. P. Ray, ―The Green Grid SAGA -- A Green Initiative to Data Centers: A Review,‖ Indian J. Comput. Sci. Eng., vol. 1, no. 4, pp. 333–339.