Scheduling policies for Baadal - the IITD cloud

Nishant Kumar Devansh Dalal

August 17, 2014

Abstract

Baadal is the IIT Delhi academic cloud on which Virtual Machines (VMs) are run. Currently it is using the First Fit strategy for scheduling the VMs on the Physical Machines (PMs). Our current work focusses on improving the scheduling policies for Baadal. We also provide analysis of simulations we used in evaluating the different strategies.

1 Currently implemented Strategy

When a new request for a Virtual Machine comes in, it has associated with it the parameters - CPU (cores needed) and RAM. It is then placed using the First Fit strategy in the first host that meets the requirements of the new VM, after sorting the hosts in the order of total RAM.

When the user changes the level of a VM and restarts it, it is placed again in a new host following the same strategy as above. There is no migration policy being used currently, other than the fact that the admin can migrate a VM from one host to another.

2 Problem Statement Formulation

We modelled the situation as a 2-Dimensional Vector Bin Packing problem where each host server is represented as a vector with 2 parameters - the RAM and the CPU (cores).

The VM request coming from the user is a 2D vector with parameters of CPU and RAM required.

The problem consists of scheduling the VMs on the PMs based on 2 aspects - (a)online and (b) offline. In the online aspect the requests come one after another and there is no migration involved, while in the offline aspect no requests are coming and the VMs are migrated so as to shut down as many hosts as possible.

3 New Strategy

- It can be proven that online First Fit algorithm for 2-Dimentional vector bin packing is $O(d)$-competitive and will not use more than 2.7 times the optimal number of bins used by offline optimal packing [3].
- While scheduling a new VM by First Fit, the host list is sorted in decreasing order of their weighted sum of usedRAM and usedCPU. This is done in order to fit the new requests in highly filled hosts for compact packing.

- In our strategy we run a migration algorithm at regular intervals which tries to shut down the least utilized host(*victim*) if it can be done by migrating its VMs onto other hosts (*targets*).

- The victim selection algorithm tries to find the host with a minimum wieghted sum of usedCPU , usedRAM and average of resources used per VM on that host.

- The migration algorithm mentioned above considers the victim's VMs as level 1 VMs. It then tries to place the VMs on the hosts using the First Fit Decreasing algorithm (considering the weight function of the VMs as $0.9RAM + 0.1CPU$). If some VM (say A) fails to be placed on any host, it picks up another appropriate victim host (on which A can be placed by removing one or more smaller VMs) and the removed VMs are now considered as next level VMs. These are then merged with the original VM list and tried to be placed similarly. This strategy can be done uptil a few levels (we used 3 in our simulation). In case if a VM still fails to be placed on any host, the whole process is cancelled.

4 Analysis

- For comparing the various strategies we generated a list of requests for the simulation, where a request can be a request for a new VM , change of level of an existing VM or shutdown of a VM while maintaining that the average number of requests for new VMs is the same as the average number of requests for shutting down a VM.

- The policies used for scheduling are First Fit , Best Fit and Random Fit (without any migration) and the corresponding algorithms with migration (represented by the name of the corresponding algorithm and the suffix 2) and running after every 10 requests.

- In order to compare the above policies, a list of 1000 requests were generated and the various policies were run on this list of requests. A graph (as shown in fig (1)) was plotted of the average number of hosts used aganist the number of requests processed till then. As can be seen from the graphs First Fit with migration running outperforms other algorithms.

- Fig 3 and 4 depict the working of the migration algorithm which is called at regular intervals in the improved algorithm and tries to free a host on each call.

5 Comparison with Optimal Offline Algorithms

- For finding the optimal offline solutions , 2 linear programming algorithms were run - one optimal and one approximate.

- To compare our online algorithms with optimal offline algorithms ,a query list was generated in which each query was either for starting a new VM or for shutting down a VM. At appropriate intervals in the query list, the then

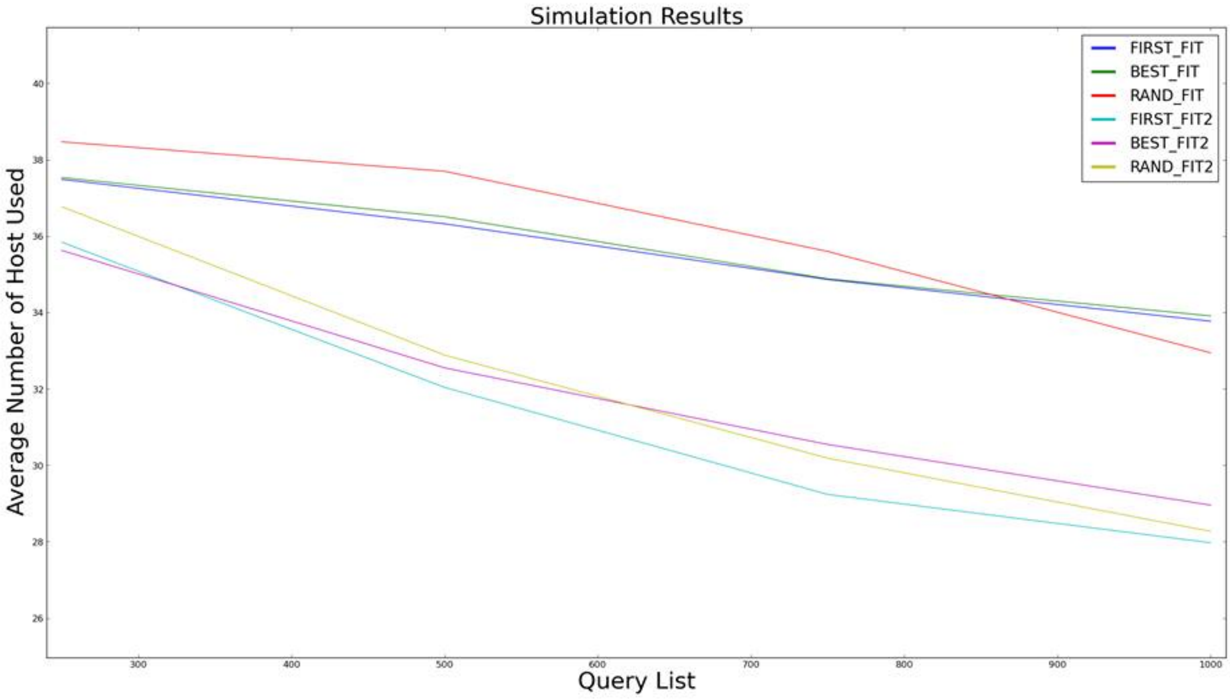

Figure 1: Comparison of various algorithms

situation of the hosts was give as input to the LP(Linear Programming) algorithm. Also, it was assumed that the number of hosts was not limited.

- Three such query lists were generated : one of 1000 queries , one of 5000 queries [both containing VM ram and cpu requests which are in powers of 2 and for which the hosts are each of 16 GB ram and 16 core cpu] and a float dataset [in which VM ram and cpu requests can be floating point numbers and hosts are each of 1GB ram and 1 core cpu].

- *For the first two datasets, our results exactly matched the optimal results*, the reason being that given any number of VMs on a host , there always exists a VM that will fill the host completely [since everything is in powers of 2]. Also, since we sort the hosts before placing a new VM, most of the hosts become completely full, making our strategy comparable to optimal solution.

- For the third float dataset, the results are as plotted in the figure (2). Also, for reference table (1) gives the number of hosts used by each algorithm. The reason for choosing small number of VMs in this dataset is that the LP algorithm becomes very slow for larger number of VMs.

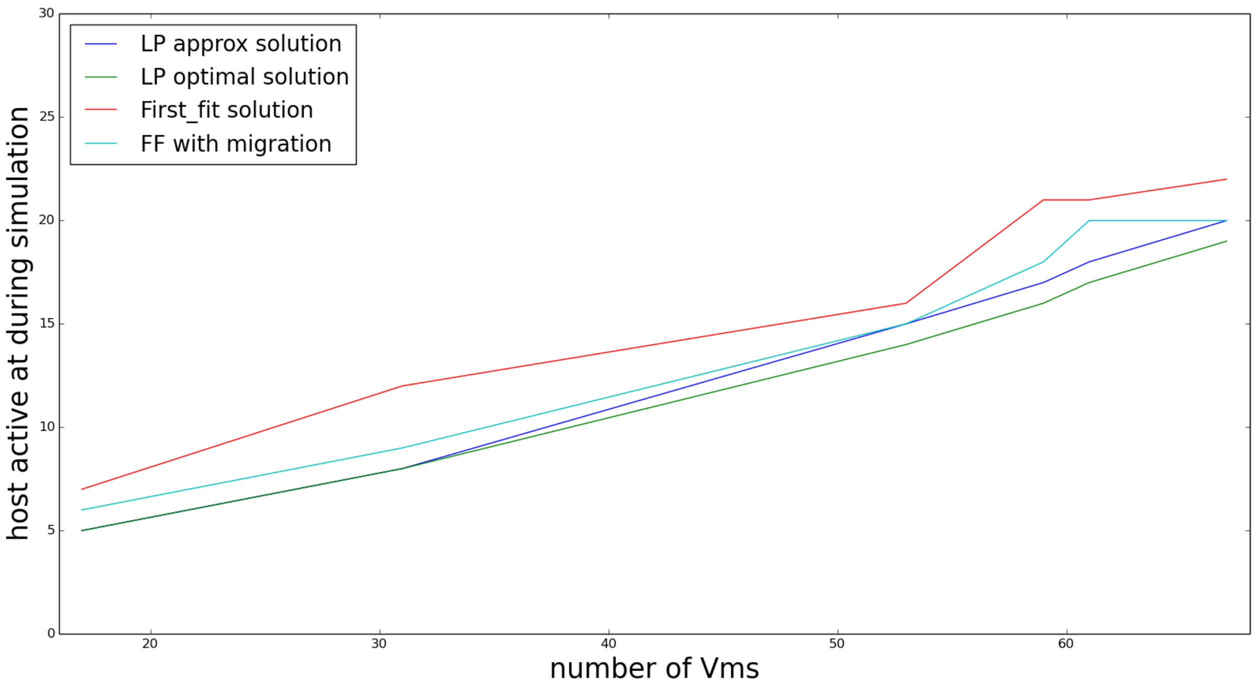

Figure 2: Comparison with optimal offline LP algorithm

Number of VMs	Optimal, approximate solution	Optimal solution	First Fit	First Fit with migration
17	5	5	7	6
31	8	8	12	9
53	15	14	16	15
59	17	16	21	18
59	16	15	19	17
61	18	17	21	20
67	20	19	22	20

Table 1: Comparison with optimal solution

6 Further Insights

- Even if the migration algorithm fails to migrate all the VMs of the victim host and shut down that host, we can still migrate as many VMs as possible. This may not shut down the victim host, but it will help in reducing fragmentation in the hosts.
- Migration can be used if First Fit fails to schedule a request so as to decrease the number of rejected requests during the simulation. This is

exemplified by the fact that the number of requests rejected for a list of 10,000 requests in the simulation were only 370 (when using migration for scheduling the VM in case the First Fit fails) as compared to 665 (without migration in case the First Fit fails).

- Once in a week or so, all the active VMs on all the PMs can be rescheduled using First Fit Decreasing algorithm so as to compactly fill the hosts. But there may be problems involved in migrating so many VMs.

7 Conclusion

From the comparison of the various algorithms and with the optimal solution, it can be concluded that since our VM requests are quatized [they are always in powers of 2] , First Fit algorithm with the migration policy included is a good approach for this specific problem.

References

[1] Samrajesh M.D. Al Shayeji M.H. An energy aware virtual machine placement algorithm. pages 242–246. IEEE, 2012.

[2] Anton Beloglazov and Rajkumar Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24:1397–1420, 2012.

[3] DS Johnson A.C.C. Yao MR Garey, RL Graham. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory*, pages 257–298, 1976.

[4] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder. Heuristics for vector bin packing. Microsoft Research, 2011.

[5] Subhash Suri. Bin packing algorithms. University of California.

[6] Seny Kamara Bruce Shepherd Yossi Azar, Ilan Cohen. Tight bounds for online vector bin packing. ACM , New York, 2013.

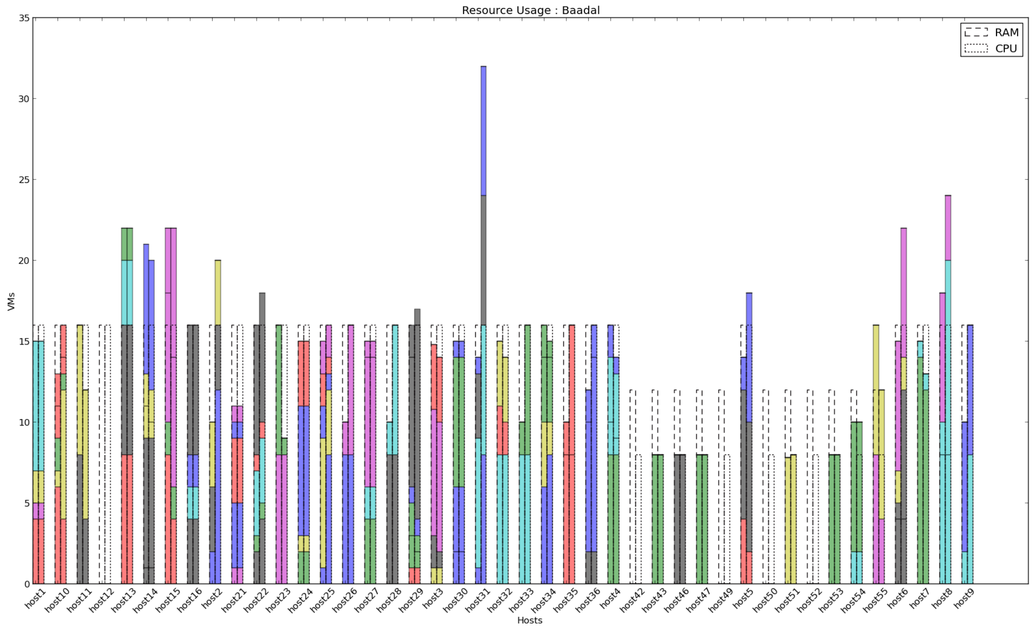

Figure 3: Migration Algorithm : Initial State of VMs and Hosts

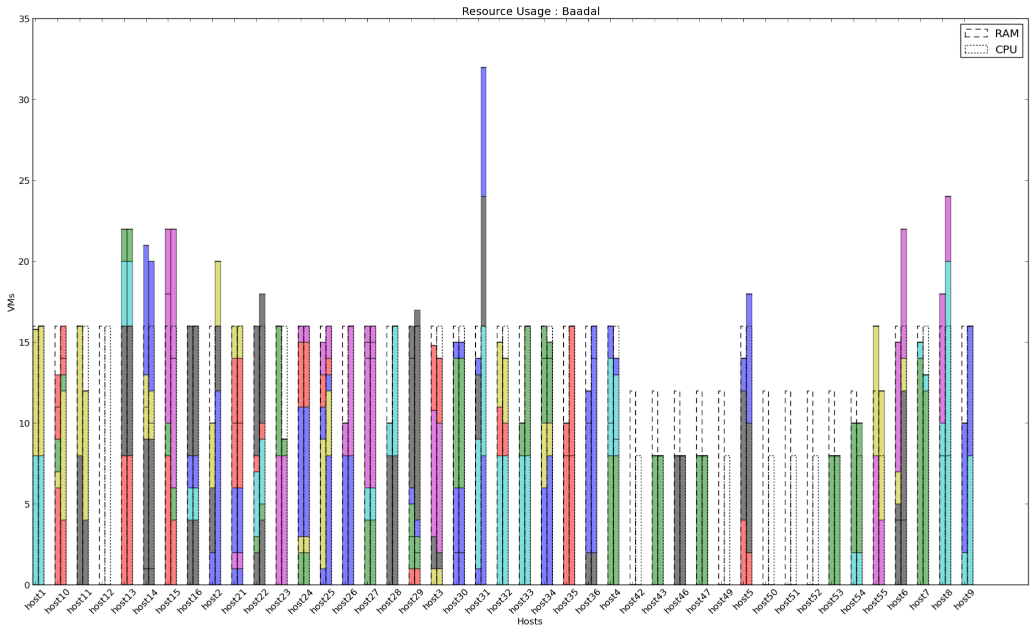

Figure 4: Migration Algorithm : Host 51 shut down and its VMs migrated to other hosts