Stage 2: Design and Implementation of a New Scheduling Algorithm

Satyamurthy Bale

45527652

Introduction

The stage 2 of this project is an improvisation of stage 1. In stage 1, the jobs were assigned to the largest server available. The stage 2 of this project will focus on implementing a custom algorithm. This algorithm should work different from the baseline algorithms given in the tutorial class (BF, WF, FF). The goal of the custom implemented algorithm is to focus on minimizing either turnaround time, rental cost, or to maximize server resource utilization. The goal is to improve at least one metric at the cost of other metrics.

Problem Definition

The three ds-client algorithms supplied (First Fit [FF], Best-Fit [BF], Worst-Fit [WF]) are all appropriate, but their efficacy is restricted for distributed systems with more information from which scheduling judgments can be made. None of the metrics are constant in all the algorithms. The custom algorithm that I have created is made to reduce the total cost as compared to the other three baseline algorithms. The main problem I was facing was to improve the turnaround time, which initially was performing worse than the three algorithms while still outperforming ATL. After a lot of trial and errors I was able to improve the implementation.

Algorithm Description

The algorithm begins by requesting jobs from a set of servers that it has access to. Each job's internal job tracker is updated in real time with the latest information, ensuring that each job schedule is as efficient as possible. It next examines the list for servers that can complete the job request now or in the future.

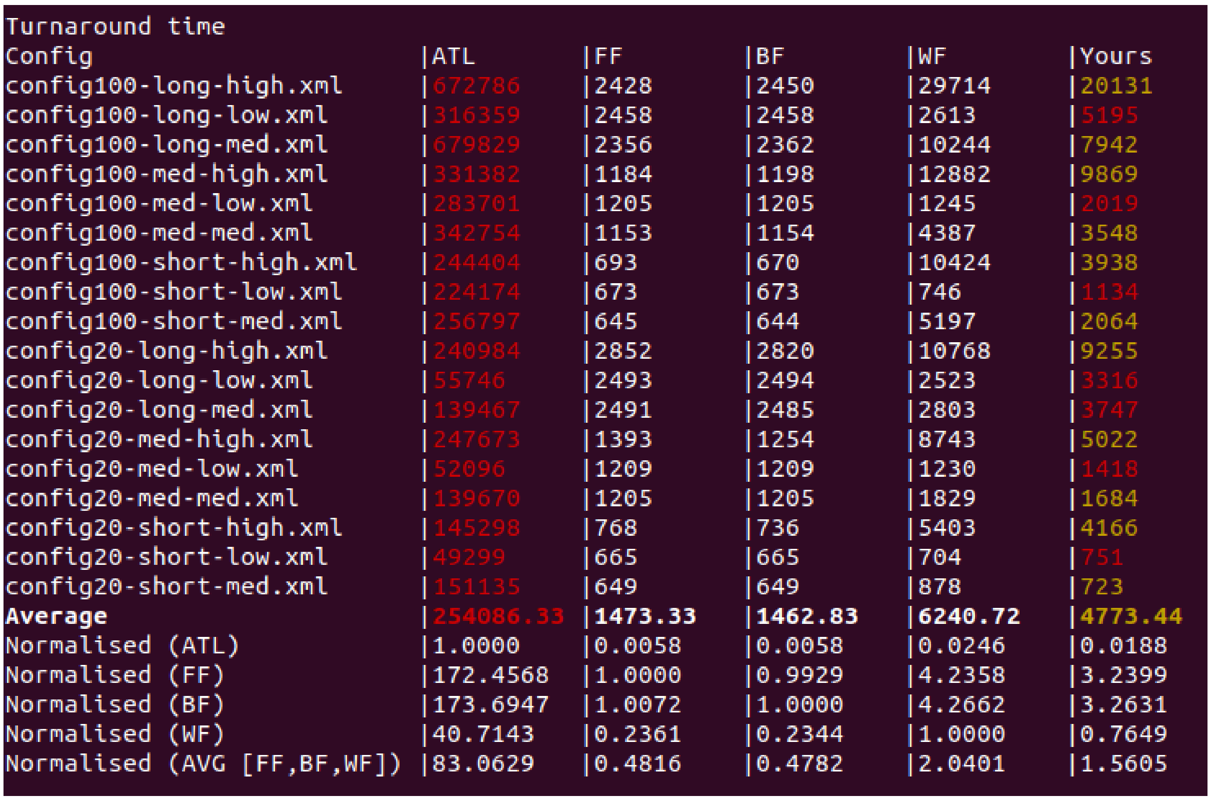
The algorithm is designed to find a server which will reduce rental cost. The algorithm takes Arraylist of server and job as an input and returns String. The algorithm includes a for loop which goes through all the servers in the list. The main thing to observe is that the cores required by server is greater than core required by job. The second step is check if the job start time is greater than job running time. To reduce the cost, it is better to choose a server which is idle or already active. In the case where we find no optimal server, the algorithm assigns the job to the first available server, which is same as first fit algorithm.

Implementation

In the attempt to keep the code minimal and easy to understand, I used the same repository as project 1. The project 1 was implemented in a manner where everything was made easy to use by making different methods. The Server and Job class contains all the required information that is necessary while implementing. The Server class also functions as a semi-nested class for the Servers class, which is its major purpose. This class holds a list of all servers that use the Server class, as well as any functions that iterate over all servers, decreasing the number of iterators used in the main code. A function for getsCapable was added which gets the list of capable servers for assigning a job. A method for converting String files to Integer was created for Server and Job. Lastly, the custom algorithm was created and after a lot of trial and error the metric to improve was decided. All the print statements were commented out to make the algorithm run quicker in the demo test.

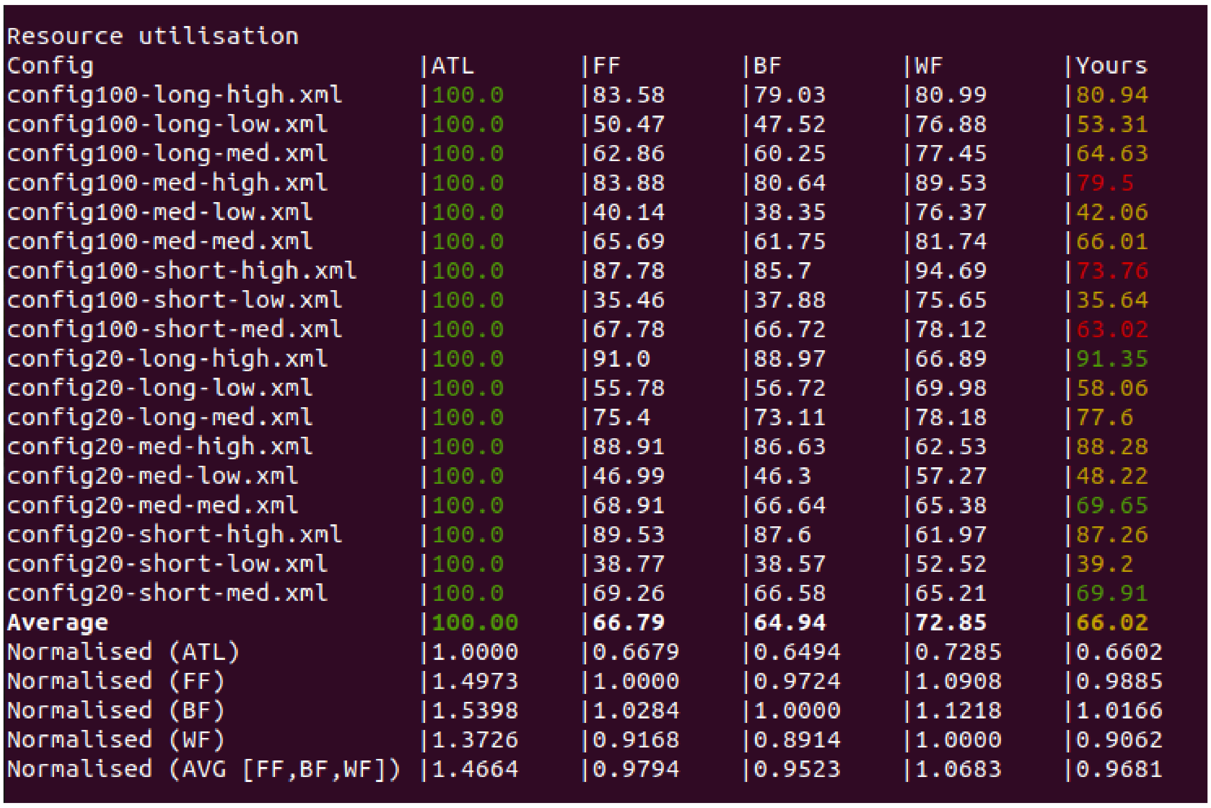
Evaluation

Turn around time



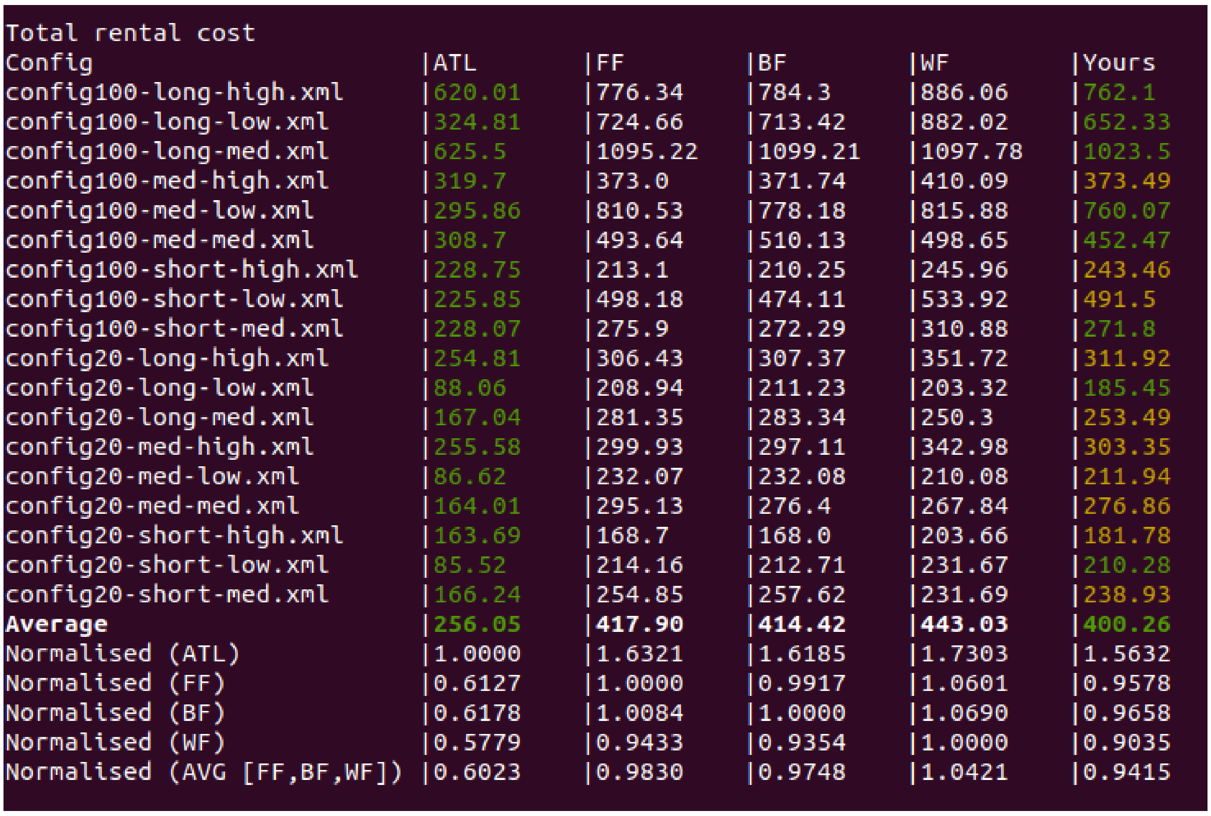
```
Turnaround time
Config                      |ATL       |FF        |BF        |WF        |Yours
config100-long-high.xml     |672786    |2428      |2450      |29714     |20131
config100-long-low.xml      |316359    |2458      |2458      |2613      |5195
config100-long-med.xml      |679829    |2356      |2362      |10244     |7942
config100-med-high.xml      |331382    |1184      |1198      |12882     |9869
config100-med-low.xml       |283701    |1205      |1205      |1245      |2019
config100-med-med.xml       |342754    |1153      |1154      |4387      |3548
config100-short-high.xml    |244404    |693       |670       |10424     |3938
config100-short-low.xml     |224174    |673       |673       |746       |1134
config100-short-med.xml     |256797    |645       |644       |5197      |2064
config20-long-high.xml      |240984    |2852      |2820      |10768     |9255
config20-long-low.xml       |55746     |2493      |2494      |2523      |3316
config20-long-med.xml       |139467    |2491      |2485      |2803      |3747
config20-med-high.xml       |247673    |1393      |1254      |8743      |5022
config20-med-low.xml        |52096     |1209      |1209      |1230      |1418
config20-med-med.xml        |139670    |1205      |1205      |1829      |1684
config20-short-high.xml     |145298    |768       |736       |5403      |4166
config20-short-low.xml      |49299     |665       |665       |704       |751
config20-short-med.xml      |151135    |649       |649       |878       |723
Average                     |254086.33 |1473.33   |1462.83   |6240.72   |4773.44
Normalised (ATL)            |1.0000    |0.0058    |0.0058    |0.0246    |0.0188
Normalised (FF)             |172.4568  |1.0000    |0.9929    |4.2358    |3.2399
Normalised (BF)             |173.6947  |1.0072    |1.0000    |4.2662    |3.2631
Normalised (WF)             |40.7143   |0.2361    |0.2344    |1.0000    |0.7649
Normalised (AVG [FF,BF,WF]) |83.0629   |0.4816    |0.4782    |2.0401    |1.5605
```

Resource utilization



```
Resource utilisation
Config                      |ATL       |FF        |BF        |WF        |Yours
config100-long-high.xml     |100.0     |83.58     |79.03     |80.99     |80.94
config100-long-low.xml      |100.0     |50.47     |47.52     |76.88     |53.31
config100-long-med.xml      |100.0     |62.86     |60.25     |77.45     |64.63
config100-med-high.xml      |100.0     |83.88     |80.64     |89.53     |79.5
config100-med-low.xml       |100.0     |40.14     |38.35     |76.37     |42.06
config100-med-med.xml       |100.0     |65.69     |61.75     |81.74     |66.01
config100-short-high.xml    |100.0     |87.78     |85.7      |94.69     |73.76
config100-short-low.xml     |100.0     |35.46     |37.88     |75.65     |35.64
config100-short-med.xml     |100.0     |67.78     |66.72     |78.12     |63.02
config20-long-high.xml      |100.0     |91.0      |88.97     |66.89     |91.35
config20-long-low.xml       |100.0     |55.78     |56.72     |69.98     |58.06
config20-long-med.xml       |100.0     |75.4      |73.11     |78.18     |77.6
config20-med-high.xml       |100.0     |88.91     |86.63     |62.53     |88.28
config20-med-low.xml        |100.0     |46.99     |46.3      |57.27     |48.22
config20-med-med.xml        |100.0     |68.91     |66.64     |65.38     |69.65
config20-short-high.xml     |100.0     |89.53     |87.6      |61.97     |87.26
config20-short-low.xml      |100.0     |38.77     |38.57     |52.52     |39.2
config20-short-med.xml      |100.0     |69.26     |66.58     |65.21     |69.91
Average                     |100.00    |66.79     |64.94     |72.85     |66.02
Normalised (ATL)            |1.0000    |0.6679    |0.6494    |0.7285    |0.6602
Normalised (FF)             |1.4973    |1.0000    |0.9724    |1.0908    |0.9885
Normalised (BF)             |1.5398    |1.0284    |1.0000    |1.1218    |1.0166
Normalised (WF)             |1.3726    |0.9168    |0.8914    |1.0000    |0.9062
Normalised (AVG [FF,BF,WF]) |1.4664    |0.9794    |0.9523    |1.0683    |0.9681
```

Rental Cost



```
Total rental cost
Config                      |ATL       |FF        |BF        |WF        |Yours
config100-long-high.xml     |620.01    |776.34    |784.3     |886.06    |762.1
config100-long-low.xml      |324.81    |724.66    |713.42    |882.02    |652.33
config100-long-med.xml      |625.5     |1095.22   |1099.21   |1097.78   |1023.5
config100-med-high.xml      |319.7     |373.0     |371.74    |410.09    |373.49
config100-med-low.xml       |295.86    |810.53    |778.18    |815.88    |760.07
config100-med-med.xml       |308.7     |493.64    |510.13    |498.65    |452.47
config100-short-high.xml    |228.75    |213.1     |210.25    |245.96    |243.46
config100-short-low.xml     |225.85    |498.18    |474.11    |533.92    |491.5
config100-short-med.xml     |228.07    |275.9     |272.29    |310.88    |271.8
config20-long-high.xml      |254.81    |306.43    |307.37    |351.72    |311.92
config20-long-low.xml       |88.06     |208.94    |211.23    |203.32    |185.45
config20-long-med.xml       |167.04    |281.35    |283.34    |250.3     |253.49
config20-med-high.xml       |255.58    |299.93    |297.11    |342.98    |303.35
config20-med-low.xml        |86.62     |232.07    |232.08    |210.08    |211.94
config20-med-med.xml        |164.01    |295.13    |276.4     |267.84    |276.86
config20-short-high.xml     |163.69    |168.7     |168.0     |203.66    |181.78
config20-short-low.xml      |85.52     |214.16    |212.71    |231.67    |210.28
config20-short-med.xml      |166.24    |254.85    |257.62    |231.69    |238.93
Average                     |256.05    |417.90    |414.42    |443.03    |400.26
Normalised (ATL)            |1.0000    |1.6321    |1.6185    |1.7303    |1.5632
Normalised (FF)             |0.6127    |1.0000    |0.9917    |1.0601    |0.9578
Normalised (BF)             |0.6178    |1.0084    |1.0000    |1.0690    |0.9658
Normalised (WF)             |0.5779    |0.9433    |0.9354    |1.0000    |0.9035
Normalised (AVG [FF,BF,WF]) |0.6023    |0.9830    |0.9748    |1.0421    |0.9415
```

The goal of getting low rental cost was achieved. However, there is a huge increase in turnaround time. Through observation, it is safe to say that this algorithm is better than worse-fit in all cases.

Conclusion

In conclusion, this algorithm successfully reduces the total cost as compared to the baseline algorithms. While the difference was not by any significant amount for rental cost, the algorithm performed better than worse fit in all the metrics. The algorithm performed decently for this project; however the test is done using only 18 files. A result performed with a huge load of files may produce a different result.

Reference

https://github.com/satyabale/comp3100ass2