Low Rental Cost(LRC) scheduling algorithm

SID： 42055563

Student Name: Chenlin ZHU

Introduction:

The new algorithm should be improved based on the original algorithm (FF, BF, WF). Both BF and WF must traverse the entire partition to find their needs, which increases the waiting time and average turnaround time. Although FF reduces waiting time and average turnaround time, it reduces resource utilization (core utilization, memory utilization, and disk utilization). No matter which method has its own advantages and disadvantages, the new algorithm is improved from BF. The server we use will be selected based on the calculation of all resource utilization rates, instead of Just like BF, I only find the server that seems to be the most suitable. Later, I will use fitRate to represent the result of my algorithm calculation. The small the number, the higher the resource utilization rate

Problem definition:

The biggest problem with the current three existing algorithms is that the advantages of each algorithm are not enough to make up for its disadvantages, that is, when we feel that FF is more suitable, the negative effects brought by FF are unacceptable to us. Because FF does not need to traverse all partitions like BF and WF, it seems to save a lot of time, but because FF does not consider the computing efficiency of the server, this often leads to waste of resources. FF has too much randomness. If every server that appears first and available is just right, then the overall efficiency of FF will become extremely high, because this situation is equivalent to not traversing the entire partition version of BF, but we all know that this kind of luck is not good. We need a stable algorithm, not this kind of luck algorithm. The key word here is resource utilization and improving resource utilization will save costs.
Now let's clarify the purpose: to reduce Average turnaround time, is to reduce the time from waiting for entry to entry and the completion of the final job for each job; to reduce the average waiting time is to reduce the time each job waits before being executed; to reduce the total cost Sales; reduce the total number of servers to be used; improve the utilization of each server resource, including core utilization, memory utilization, disk utilization; increase time utilization, server available time plus job estimated run time divided by job estimated run time.

Algorithm description:

CoreUtilization: (ServerCoreNumber– JobCoreNumber)/ JobCoreNumber
MemoryUtilization: (ServerMemorySize– JobMemorySize)/ JobMemorySize
DiskUtilization: (ServerDiskSize – JobDiskSize)/JobDiskSize
TimeUtilization: (ServerAvailableTime + JobEstimatedRunTime)/JobEstimatedRunTime

Below I use an example to illustrate the calculation rules of my algorithm：

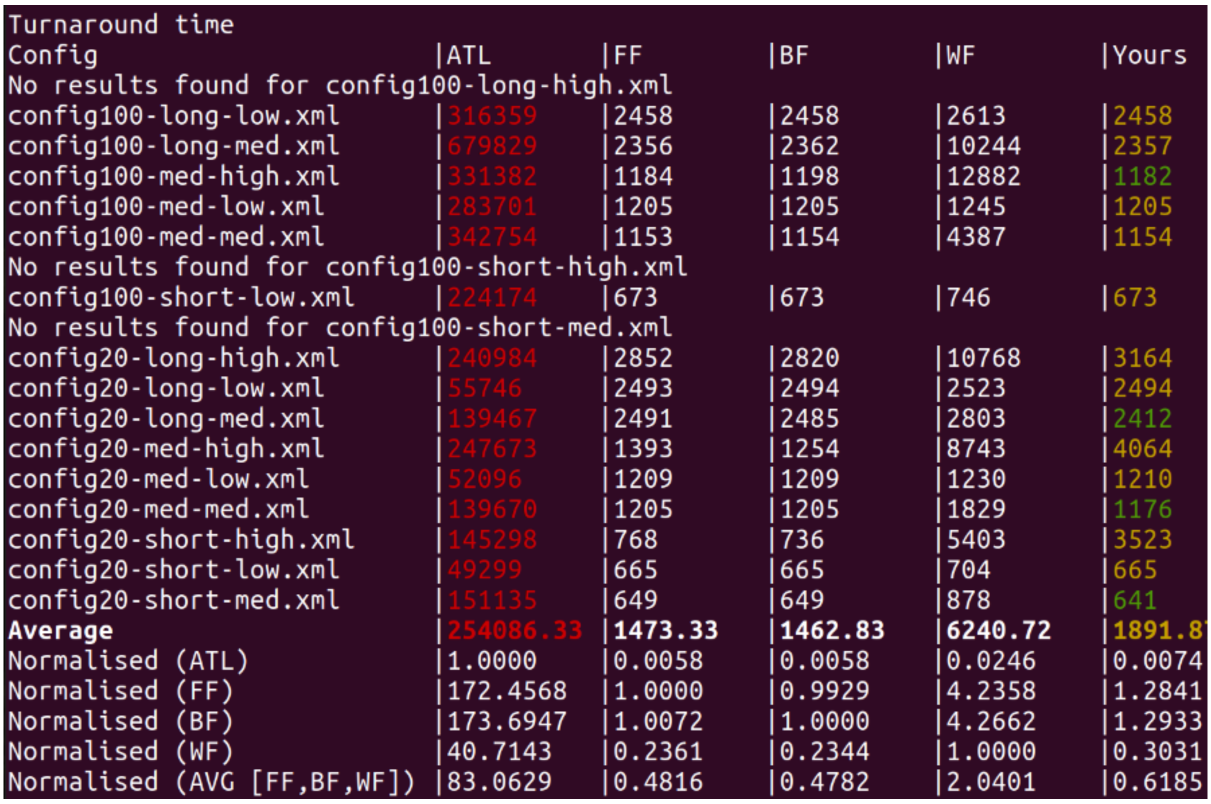
JOB:

JOBN	submitTime	jobID	estRuntime	core	memory	disk
	101	1	380	2	900	2500

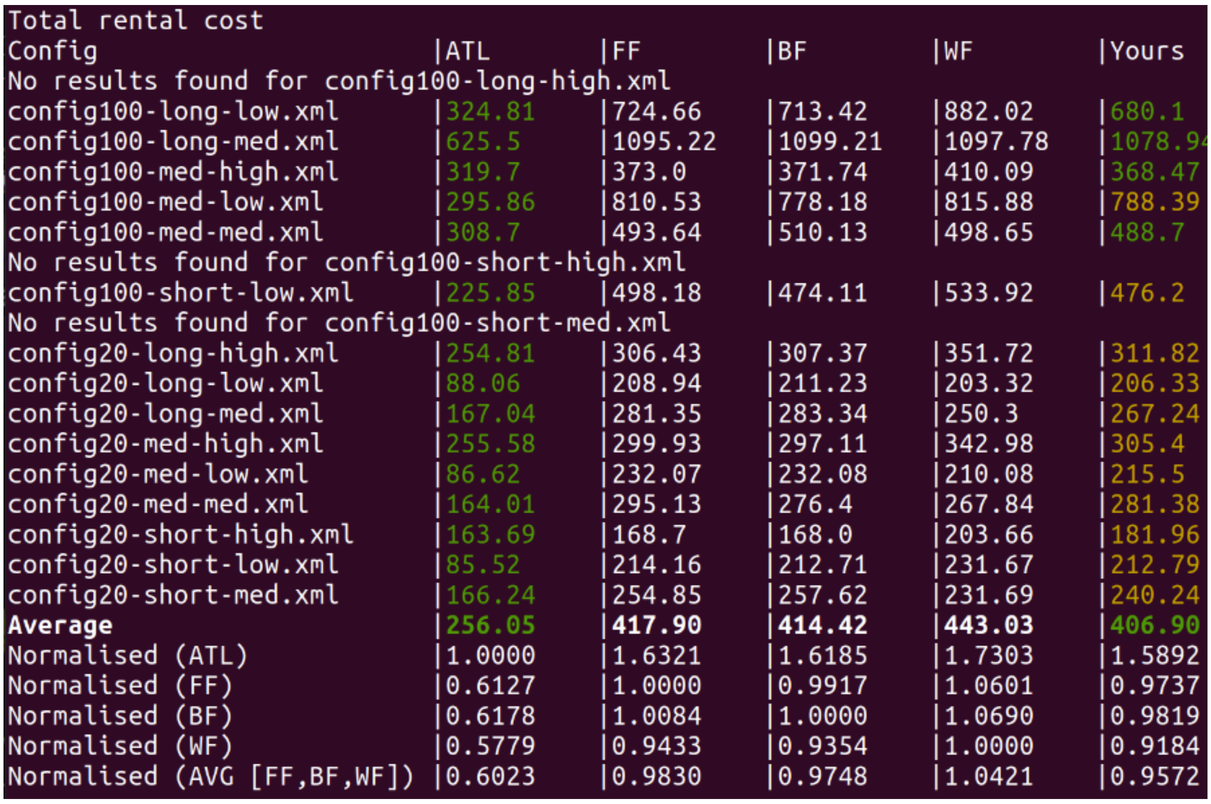
Server：

Small	0	Inactive	-1	2	8000	64000
Medium	0	Inactive	-1	4	16000	128000

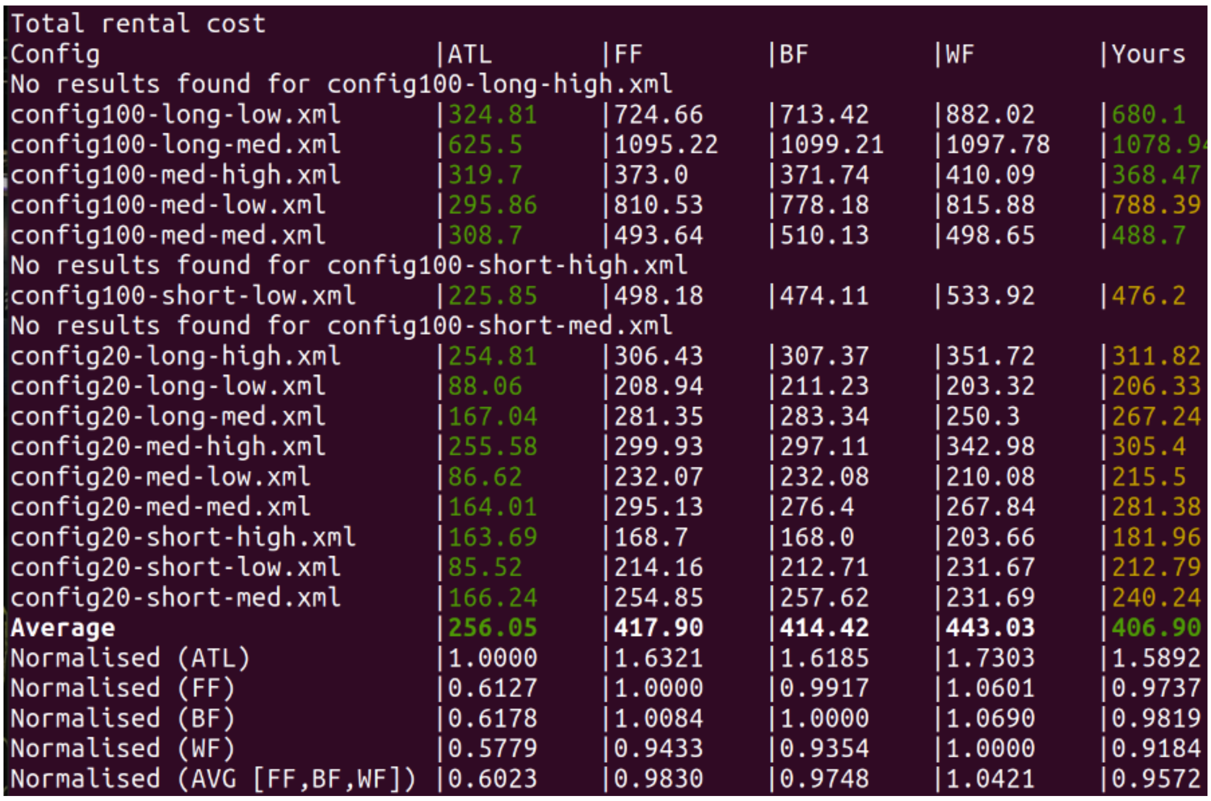
fitRate= CoreUtilization+ MemoryUtilization+ DiskUtilization+ TimeUtilization
We will calculate the fitRate for these two servers separately:
Small Server:
CoreUtilization:2/2=1
MemoryUtilization:(8000-900)/900=7.9
DiskUtilization:(64000-2500)/2500=24.6
TimeUtilization:(-1+380)/380=1
fitRate=34.5
Medium Server:
CoreUtilization:4/2=2
MemoryUtilization:(16000-900)/900=16.7
DiskUtilization:(128000-2500)/2500=50.2
TimeUtilization:(-1+380)/380=1
fitRate=69.9
Small Server’s fitRate is smaller than the medium server, so the job should be assigned to small server.
First, I take two different servers, small and medium, just to calculate the difference of fitRate more clearly, so that the difference can be seen more clearly. In fact, the more advantage of this algorithm is that when the various values of the servers are similar, how do we decide which server to assign to. When two almost identical servers appear, choose a server with a lower fitRate to reduce server total rental cost.
Implementation details:



```
Turnaround time
Config                       |ATL        |FF         |BF         |WF         |Yours
No results found for config100-long-high.xml
config100-long-low.xml       |316359     |2458       |2458       |2613       |2458
config100-long-med.xml       |679829     |2356       |2362       |10244      |2357
config100-med-high.xml       |331382     |1184       |1198       |12882      |1182
config100-med-low.xml        |283701     |1205       |1205       |1245       |1205
config100-med-med.xml        |342754     |1153       |1154       |4387       |1154
No results found for config100-short-high.xml
config100-short-low.xml      |224174     |673        |673        |746        |673
No results found for config100-short-med.xml
config20-long-high.xml       |240984     |2852       |2820       |10768      |3164
config20-long-low.xml        |55746      |2493       |2494       |2523       |2494
config20-long-med.xml        |139467     |2491       |2485       |2803       |2412
config20-med-high.xml        |247673     |1393       |1254       |8743       |4064
config20-med-low.xml         |52096      |1209       |1209       |1230       |1210
config20-med-med.xml         |139670     |1205       |1205       |1829       |1176
config20-short-high.xml      |145298     |768        |736        |5403       |3523
config20-short-low.xml       |49299      |665        |665        |704        |665
config20-short-med.xml       |151135     |649        |649        |878        |641
Average                      |254086.33  |1473.33    |1462.83    |6240.72    |1891.8
Normalised (ATL)             |1.0000     |0.0058     |0.0058     |0.0246     |0.0074
Normalised (FF)              |172.4568   |1.0000     |0.9929     |4.2358     |1.2841
Normalised (BF)              |173.6947   |1.0072     |1.0000     |4.2662     |1.2933
Normalised (WF)              |40.7143    |0.2361     |0.2344     |1.0000     |0.3031
Normalised (AVG [FF,BF,WF])  |83.0629    |0.4816     |0.4782     |2.0401     |0.6185
```

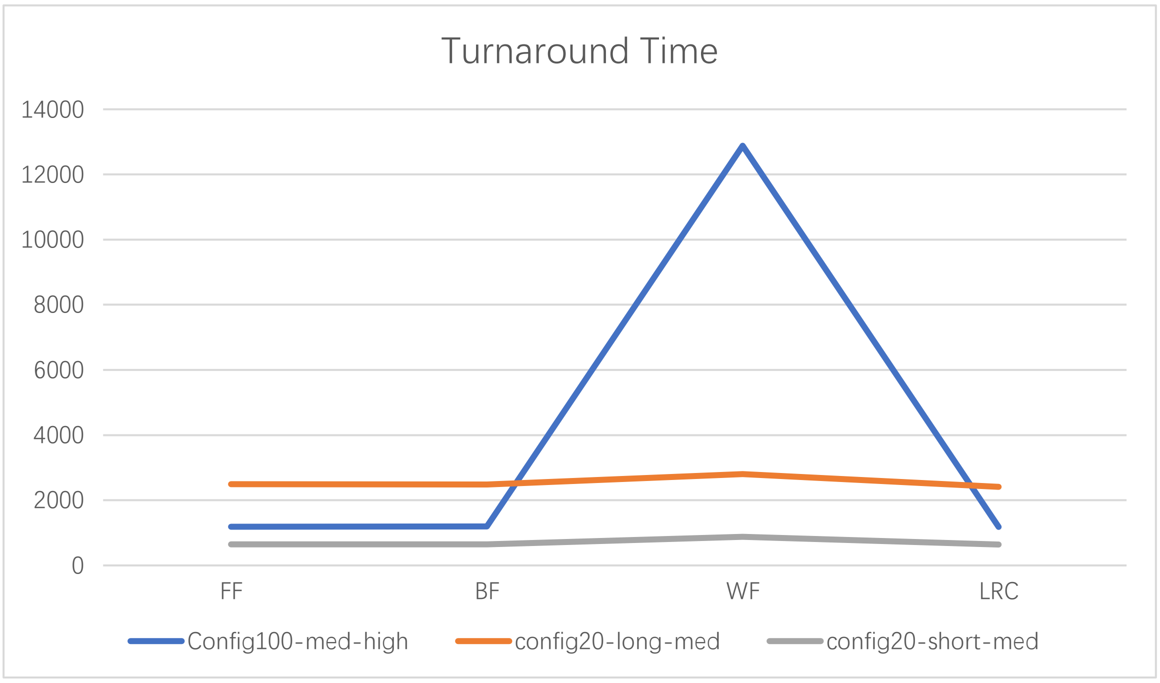


```
Total rental cost
Config                       |ATL        |FF         |BF         |WF         |Yours
No results found for config100-long-high.xml
config100-long-low.xml       |324.81     |724.66     |713.42     |882.02     |680.1
config100-long-med.xml       |625.5      |1095.22    |1099.21    |1097.78    |1078.9
config100-med-high.xml       |319.7      |373.0      |371.74     |410.09     |368.47
config100-med-low.xml        |295.86     |810.53     |778.18     |815.88     |788.39
config100-med-med.xml        |308.7      |493.64     |510.13     |498.65     |488.7
No results found for config100-short-high.xml
config100-short-low.xml      |225.85     |498.18     |474.11     |533.92     |476.2
No results found for config100-short-med.xml
config20-long-high.xml       |254.81     |306.43     |307.37     |351.72     |311.82
config20-long-low.xml        |88.06      |208.94     |211.23     |203.32     |206.33
config20-long-med.xml        |167.04     |281.35     |283.34     |250.3      |267.24
config20-med-high.xml        |255.58     |299.93     |297.11     |342.98     |305.4
config20-med-low.xml         |86.62      |232.07     |232.08     |210.08     |215.5
config20-med-med.xml         |164.01     |295.13     |276.4      |267.84     |281.38
config20-short-high.xml      |163.69     |168.7      |168.0      |203.66     |181.96
config20-short-low.xml       |85.52      |214.16     |212.71     |231.67     |212.79
config20-short-med.xml       |166.24     |254.85     |257.62     |231.69     |240.24
Average                      |256.05     |417.90     |414.42     |443.03     |406.90
Normalised (ATL)             |1.0000     |1.6321     |1.6185     |1.7303     |1.5892
Normalised (FF)              |0.6127     |1.0000     |0.9917     |1.0601     |0.9737
Normalised (BF)              |0.6178     |1.0084     |1.0000     |1.0690     |0.9819
Normalised (WF)              |0.5779     |0.9433     |0.9354     |1.0000     |0.9184
Normalised (AVG [FF,BF,WF])  |0.6023     |0.9830     |0.9748     |1.0421     |0.9572
```

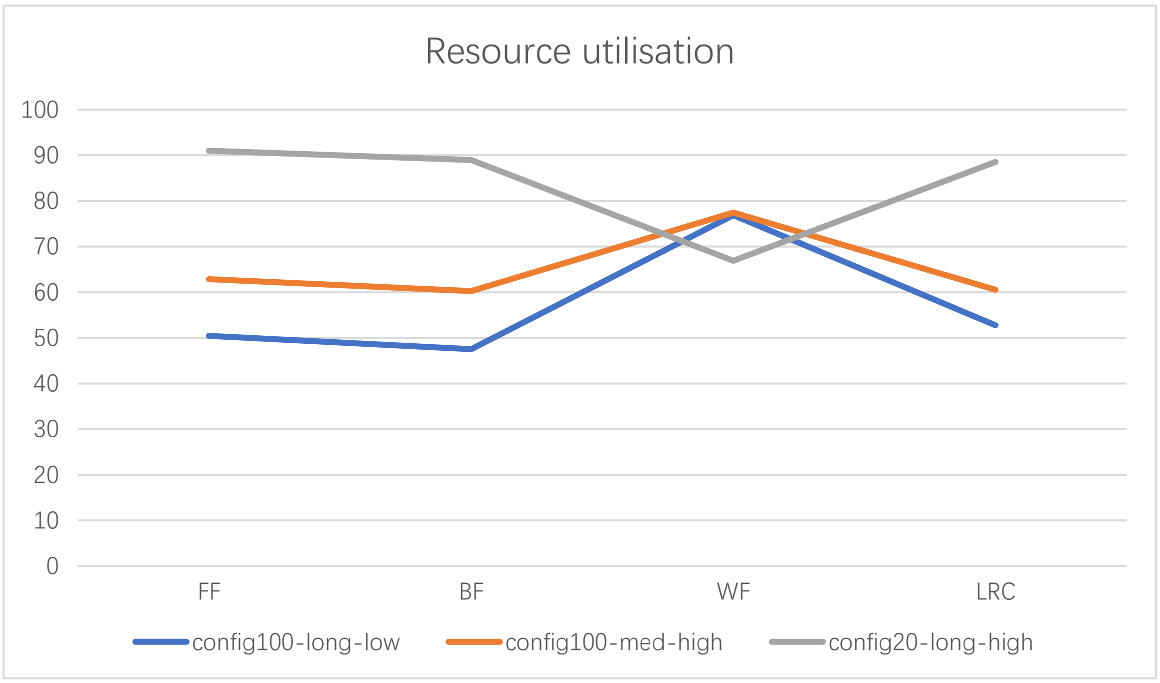


```
Total rental cost
Config                     |ATL       |FF        |BF        |WF        |Yours
No results found for config100-long-high.xml
config100-long-low.xml     |324.81    |724.66    |713.42    |882.02    |680.1
config100-long-med.xml     |625.5     |1095.22   |1099.21   |1097.78   |1078.9
config100-med-high.xml     |319.7     |373.0     |371.74    |410.09    |368.47
config100-med-low.xml      |295.86    |810.53    |778.18    |815.88    |788.39
config100-med-med.xml      |308.7     |493.64    |510.13    |498.65    |488.7
No results found for config100-short-high.xml
config100-short-low.xml    |225.85    |498.18    |474.11    |533.92    |476.2
No results found for config100-short-med.xml
config20-long-high.xml     |254.81    |306.43    |307.37    |351.72    |311.82
config20-long-low.xml      |88.06     |208.94    |211.23    |203.32    |206.33
config20-long-med.xml      |167.04    |281.35    |283.34    |250.3     |267.24
config20-med-high.xml      |255.58    |299.93    |297.11    |342.98    |305.4
config20-med-low.xml       |86.62     |232.07    |232.08    |210.08    |215.5
config20-med-med.xml       |164.01    |295.13    |276.4     |267.84    |281.38
config20-short-high.xml    |163.69    |168.7     |168.0     |203.66    |181.96
config20-short-low.xml     |85.52     |214.16    |212.71    |231.67    |212.79
config20-short-med.xml     |166.24    |254.85    |257.62    |231.69    |240.24
Average                    |256.05    |417.90    |414.42    |443.03    |406.90
Normalised (ATL)           |1.0000    |1.6321    |1.6185    |1.7303    |1.5892
Normalised (FF)            |0.6127    |1.0000    |0.9917    |1.0601    |0.9737
Normalised (BF)            |0.6178    |1.0084    |1.0000    |1.0690    |0.9819
Normalised (WF)            |0.5779    |0.9433    |0.9354    |1.0000    |0.9184
Normalised (AVG [FF,BF,WF])|0.6023    |0.9830    |0.9748    |1.0421    |0.9572
```

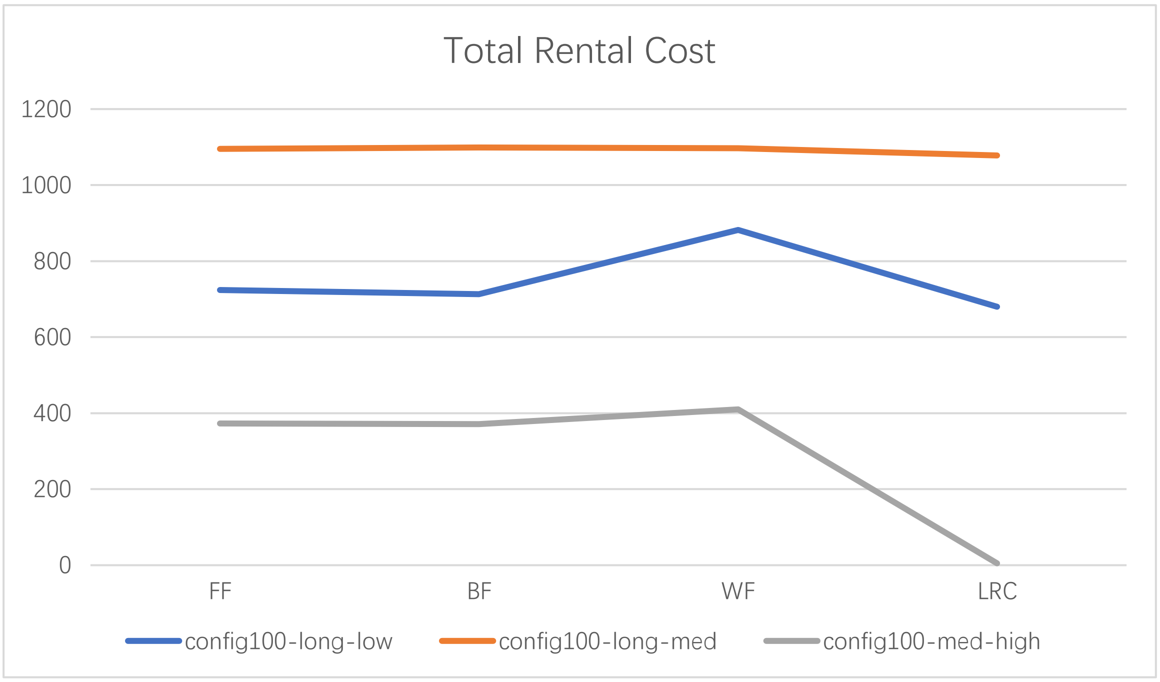
Evaluation:



The advantage of LRC algorithm in turnaround time is not obvious, the value is slightly higher than FF and BF, but far lower than WF



The four algorithms of this item are very close, but LRC also has no advantage, the value is slightly lower than the other three.



The original idea of LRC is to reduce the overall cost while improving resource utilization through this algorithm. Although LRC has no advantage in the resource utilization calculated by the system, when calculating the total rental cost, LRC It is the lowest cost. Because the LRC algorithm is more efficient in medium and large servers, and the hourly price of medium and large servers is more expensive, so the total rental cost will be lower.

Conclusion:

Now, the purpose of LRC is to reduce Total Rental Cost and improve resource utilization by optimizing the algorithm. However, Turnaround Time and Resource Utilisation have no advantages compared with the other three items. Through the final numerical comparison, it is found that the neglect of small and tiny servers has led to a decrease in overall resource utilization. Because the calculation method when designing LRC leads to the small and tiny values being very close, the advantages of LRC may not be realized. If it is modified, I think I will separately propose a set of algorithms suitable for small servers for LRC, to reduce the Total Rental Cost and increase the Resource Utilization.

References:

S,LING,” Four dynamic memory allocation algorithms in Java: BF + NF + WF + FF”,21-11-2020.[Online]. . Available: https://javamana.com/2020/11/20201121205038461f.html. [Accessed 30/05/2021]

My gitbub: https://github.com/louisflyaway/Comp3100Stage2