**Perfect job scheduler Algorithm**

**Group 26**

**Rui Yan 44486375**

# Introduction

Stage 3 is to design and implement a new scheduling algorithm to "optimize" one or more goals. It balances more indicators and arranges more work effectively. A good algorithm is to make the application better so stage 3 work optimization stage 2 algorithm. I will call it Perfect Pit (PF) Algorithm. Algorithm optimization is to optimize the performance of the algorithm, such as time complexity, space complexity, correctness, and robustness. With the advent of the era of big data, the number of data to be processed by the algorithm is also growing and the scene of processing problems is changing. To enhance the ability of the algorithm, it is necessary to optimize the algorithm. Algorithm optimization is generally to optimize the algorithm structure and convergence.

# Problem definition

Load balancing is based on the existing network structure, which provides a cheap, effective and transparent way to expand the bandwidth of network equipment and servers, increase throughput, strengthen the capacity of network data processing, and improve the flexibility and availability of the network.

The meaning of load balance is to allocate to multiple operation units for execution, such as web server, FTP server, enterprise key application server and other key task servers, so as to complete the work together

The existing load balancing algorithms are mainly divided into static and dynamic ones. The static load balancing algorithm allocates tasks with a fixed probability without considering the state information of the server, such as rotation algorithm, weighted rotation algorithm, etc.; the dynamic load balancing algorithm determines the task allocation with the real-time load state information of the server, such as the minimum connection method, weighted minimum connection method, etc.

# Algorithm Description

## Definition

Load balance means to balance the load (work task) and allocate it to multiple operation units for operation, such as FTP server, web server, enterprise core application server and other main task servers, so as to complete the work tasks together.

Load balancing is built on the original network structure, which provides a transparent, cheap and effective way to expand the bandwidth of servers and network equipment, strengthen the network data processing capacity, increase throughput, improve the availability and flexibility of the network.

The mainly is three classes:

1. Polling method

The polling method is to distribute the user's request to the server in turn, just like counting one by one, and allocating in turn. This algorithm is relatively simple, it has the advantage of absolute equilibrium, but it is because of absolute equilibrium that it must pay a great price, for example, it cannot guarantee the rationality of the assignment task and assign tasks according to the server's affordability.

2. Random method

The random method is to randomly select a server to distribute tasks. It guarantees that the decentralization of requests reaches a balanced purpose. At the same time, it is stateless and does not need to maintain the last selected state. However, as the number of tasks increases, its effects tend to have some of the disadvantages of the polling algorithm after polling.

3. Minimum connection method

The minimum connection method distributes tasks to the nodes with the smallest number of connections currently, so it is a dynamic load balancing algorithm. After a node receives a task, the number of connections will increase by 1. When the node fails, the node weight is set to 0, and no task is assigned to the node.

**3.2 Algorithm example:**

FF is an algorithm of scheduling server tasks according to priority. The disadvantage of FF is that it is the first one to obtain according to the priority of resources. Such an algorithm cannot guarantee the optimal utilization of system and server resources, BF algorithm is based on the strategy of optimizing resource utilization, but only two factors are considered, so this algorithm is not a comprehensive resource optimization algorithm, so we need to find more optimized algorithm to improve server utilization. In this way, I optimize algorithm by combining FF and BF.

# Implementation details

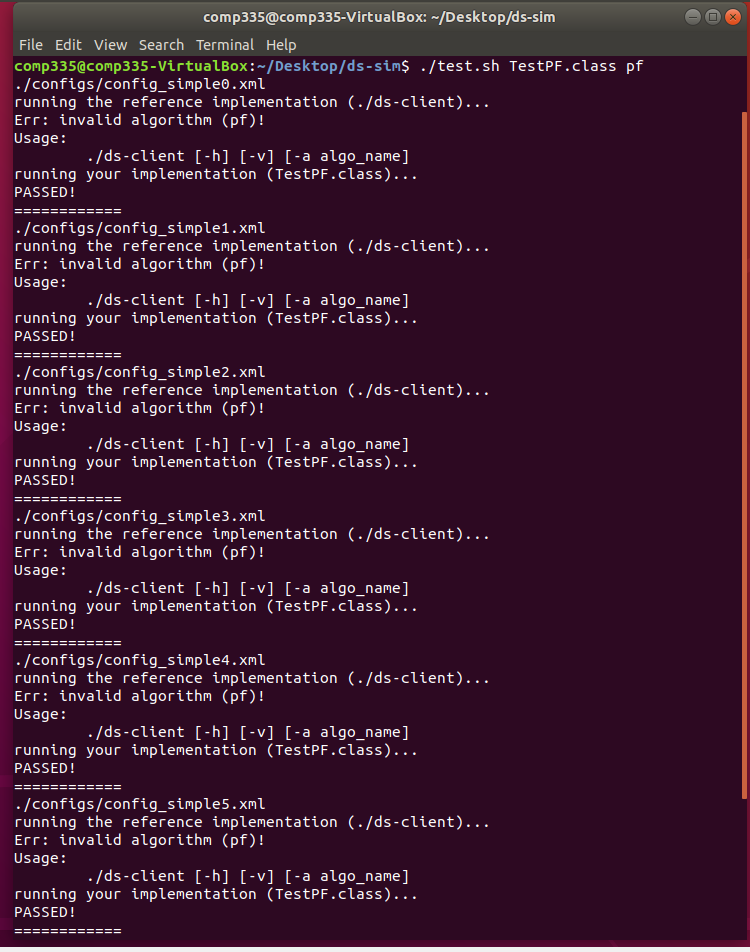
The algorithm optimization part is below. The logic is that firstly select an available server for job by FF algorithm, then find a way to balance the multiple factors optimized algorithm like BF, so it is a better algorithm.

PFClient.java

private void cntj(Job job) throws Exception {  
 String send, recv;  
 Boolean find = false;  
 int bestFit = Integer.*MAX\_VALUE*;  
 int minAvail = Integer.*MAX\_VALUE*;  
 int wstFit = Integer.*MAX\_VALUE*;  
 int bestMemoryFit = Integer.*MAX\_VALUE*;  
 Server select = null;  
 Server selectW = null;  
  
 for (Type type : this.system.getTypeList()) {  
 for (Server server : type.getServers()) {  
 if (!server.getType().equals(type))  
 continue;  
 send = server.getCNTJ();  
 this.sendMsg(send);  
 recv = this.recvMsg();  
 //if(job.getID()==19)  
 //System.out.printf("server: %s, avail:%b, fit:%d\n", server.showAll(), server.available(job), server.fit(job));  
 int state = server.getState();  
 int fit = server.fit(job);  
 int fitMemory = server.fitMemory(job);  
 int fitDisk = server.fitDisk(job);  
 if (server.available(job)) {  
 if (!find) {  
 select = server;  
 find = true;  
 }  
 int availTime = server.getAvailTime();  
 if ((fit < bestFit) && (availTime < minAvail) &&  
 (fitMemory < bestMemoryFit)) {  
 bestFit = fit;  
 minAvail = availTime;  
 bestMemoryFit = fitMemory;  
 select = server;  
 find = true;  
 }  
 //} else if ((state % 2 == 1) && type.available(job)) {  
 }  
 }  
 }  
  
 if (find) {  
 send = String.*format*("SCHD %d %s", job.getID(), select.show());  
 this.sendMsg(send);  
 this.recvMsg("OK");  
 } else {  
 send = String.*format*("SCHD %d %s", job.getID(), selectW.show());  
 this.sendMsg(send);  
 this.recvMsg("OK");  
 }  
 }  
}

# Evaluation

**The test case:**



# Summary

There is no best in the world of algorithms, only better, for the resource scheduling problem is the same. According to the actual situation, the optimization algorithm is also a process of gradual improvement, which cannot be modified without theoretical basis.

1. **User Guide**
2. make
3. ./test.sh TestPF.class pf