

# Test Result Investigation

Team 8

Taebum Kim, Seonghoon Seo, Hyunseok Oh

## 1 Scheduling Test Result

In this section, it contains the result of *trial* test, whose test number is 6. At this test, we tested with 32 processes, and estimated the time interval of the lastly forked process takes. Each process do trial division of 499999991 for 500 times. For each test, we tested weight from 1 to 20. We tested both *equal weight* and *different weight*. The former test's all forked process has equal weights. However, the weight of forked processes of the latter test are all fixed to 10 (default weight), except for the lastly forked process. For each weight, we tested 10 times and took average value. It does not consider load balance because the result will not be changed much (all CPUS are almost balanced always).

### equal weight

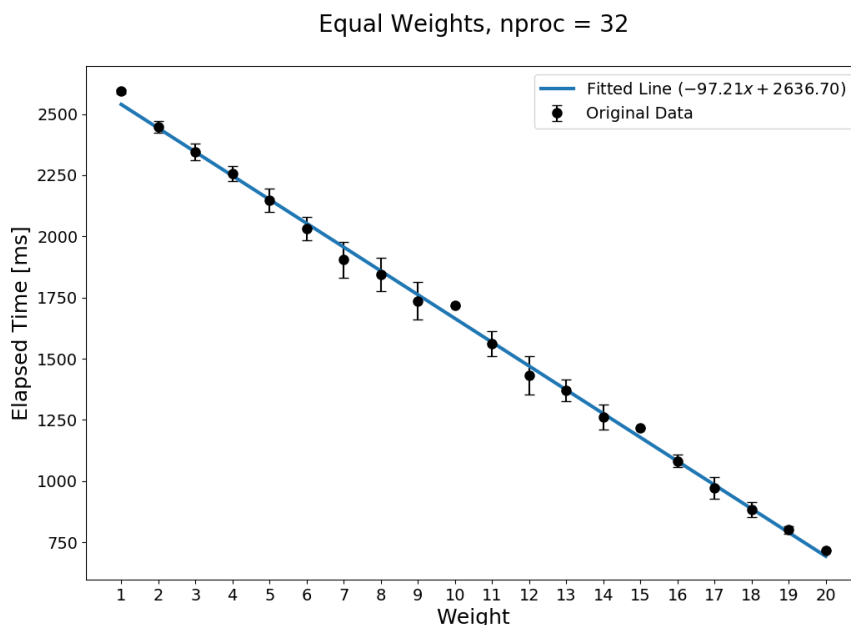


Figure 1: Test result of equal weight

Test result for the equal weight is shown as Fig 1. It shows that elapsed time decreases linearly (about 100 ms decrease per single weight increase). Elapsed time decreases since

the number of context switching decreases if each task has bigger weight. However, as the total required time is fixed for given task, each task can finish the job with fewer context switching.

### different weight

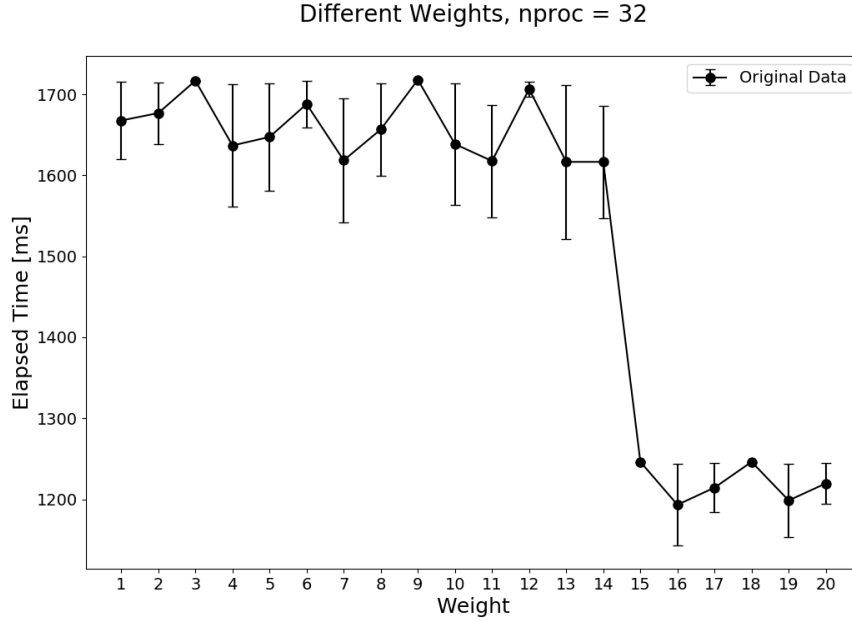


Figure 2: Test result of different weight

Test result for the different weight is shown as Fig 2. It seems quite interesting, since it looks like a step function (sharp decrease after weight 14). We thought this could be occur since all other tasks' weight are fixed, which means that the period of the target task takes cpu is fixed. In other words, the period of target task requeuing might be fixed. However, that task can finish job earlier if it has more time slice (no redundant requeuing). We can infer that the number of requeuing decrease when weight becomes 15 (enough to finish the task early).

## 2 Load Balance Test Result

In this section, it contains the result of *balance* test, whose test number is 8.

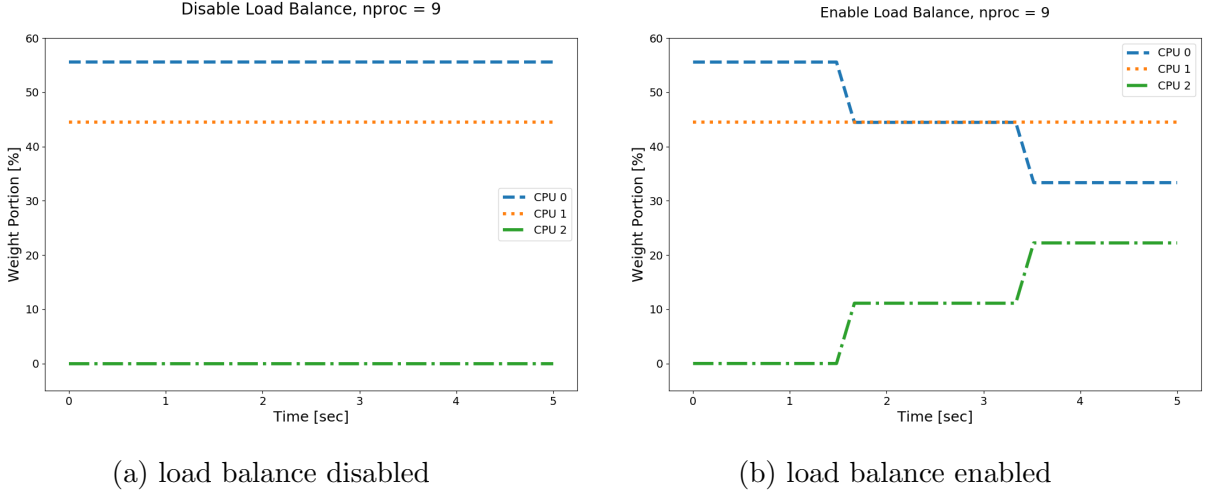


Figure 3: Simple load balancing test result

To see that load balance works well, we modified some functionality. For *select\_task\_rq\_wrr*, which decides CPU a task should be assigned to, we changed it to return 0 and 1 alternately. Thus, the forked process cannot be assigned to CPU 2. We forked 9 child processes and monitored the *weight portion* of each CPU. Weight portion of *i*-th CPU is defined as

$$\text{weight portion}[i] = \frac{w[i]}{\sum_{j=0}^{CPUS} w[j]} \times 100$$

, where  $w[i]$  is weight of *i*-th CPU's wrr queue.

The result is shown as Fig 3. Fig 3a is the test without load balance, and it keeps original weight portion forever. However, Fig 3b is the test with load balance. About 1.5, and 3.5 seconds (we can check that load balancing period is 2000 ms), load balancing occurred and we can see weight portion changes.