**Answers file for studio09**

1. Haoxuan Sun, Haoran Qin, Brendan Yang

2.cpu memory pids

cpu memory pids

List of subdirectory:

root@rayraspberrypi:/sys/fs/cgroup/firstgroup# ls

cgroup.controllers cpu.max memory.min

cgroup.events cpu.stat memory.oom.group

cgroup.freeze cpu.weight memory.stat

cgroup.max.depth cpu.weight.nice memory.swap.current

cgroup.max.descendants memory.current memory.swap.events

cgroup.procs memory.events memory.swap.high

cgroup.stat memory.events.local memory.swap.max

cgroup.subtree\_control memory.high pids.current

cgroup.threads memory.low pids.events

cgroup.type memory.max pids.max

Contents of cpu.stat:

usage\_usec 0

user\_usec 0

system\_usec 0

nr\_periods 0

nr\_throttled 0

throttled\_usec 0

3.

The content of cpu.stat:

usage\_usec 7152077

user\_usec 7042504

system\_usec 109572

nr\_periods 0

nr\_throttled 0

throttled\_usec 0

Time result:

pi@rayraspberrypi:~/Desktop/cse522/CSE522-SP23/studio09 $ ./exec\_time

My PID: 8706

Press any number to continue...s

Generating matrices...

Multiplying matrices...

Multiplication done!

7.05user 0.08system 0:01.98elapsed 359%CPU (0avgtext+0avgdata 7348maxresident)k

0inputs+0outputs (0major+1562minor)pagefaults 0swaps

The user time is almost the same in these two ways. The system time measured by cgroup is slightly higher than that masured by time.

4.

CPU.stat results:

usage\_usec 36810077

user\_usec 36322856

system\_usec 487221

nr\_periods 48

nr\_throttled 37

throttled\_usec 65585142

The results from time for dense\_mm.

pi@rayraspberrypi:~/Desktop/cse522/CSE522-SP23/studio09 $ ./500exec

My PID: 18032

Press any number to continue...d

Generating matrices...

Multiplying matrices...

Multiplication done!

7.33user 0.06system 0:02.97elapsed 248%CPU (0avgtext+0avgdata 7336maxresident)k

0inputs+0outputs (0major+1561minor)pagefaults 0swaps

The elapsed time is higher than the previous question. The program took more time to execute the same program with higher CPU contention.

5.

(1)The cpu.weight is 100

(2)The value I set is 10000

(3)The time used for second program

usage\_usec 14347451

user\_usec 14149144

system\_usec 198306

nr\_periods 0

nr\_throttled 0

throttled\_usec 0

The user time usage and system time usage are about twice the values before.

cgroup will limit the CPU recourses for a program when CPU's in heavy thread.

6.

The MAX period values are 400000 800000

The bandwidth is 0.5

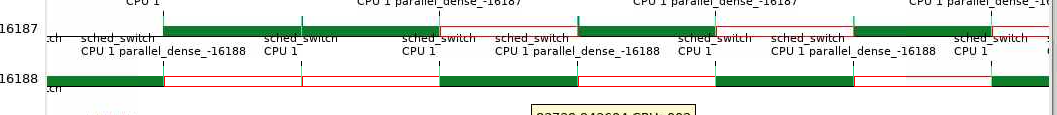
7.52user 0.11system 0:14.84elapsed 51%CPU (0avgtext+0avgdata 7332maxresident)k

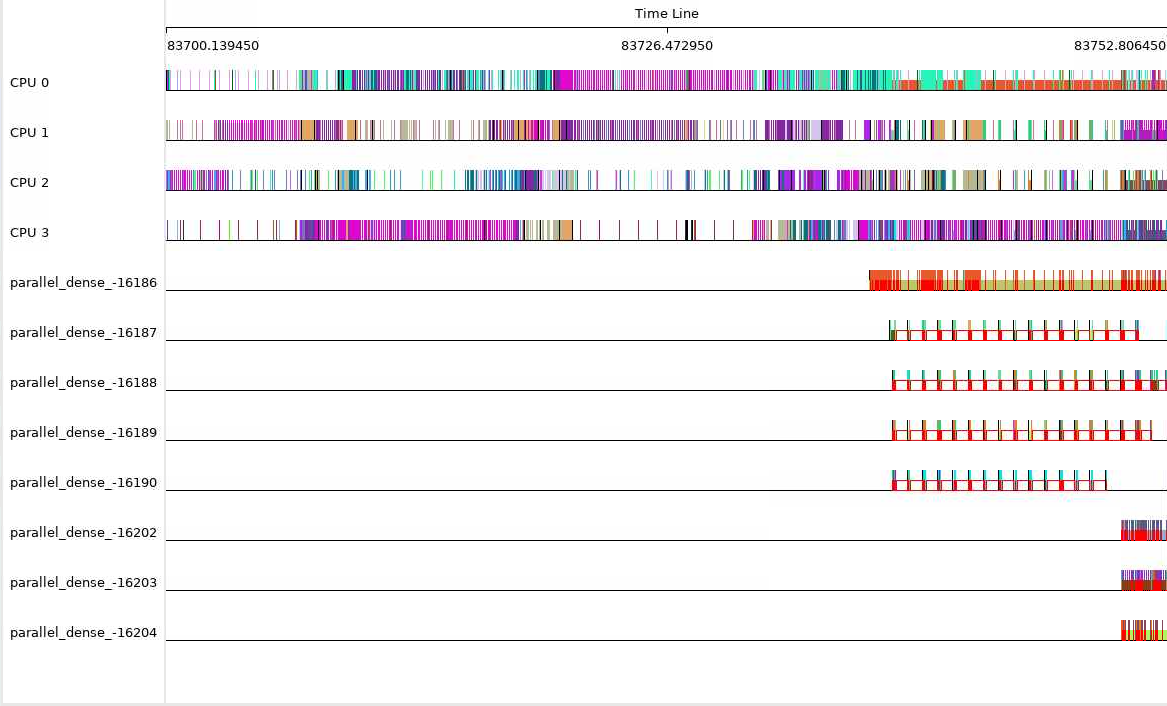
0inputs+0outputs (0major+1560minor)pagefaults 0swaps

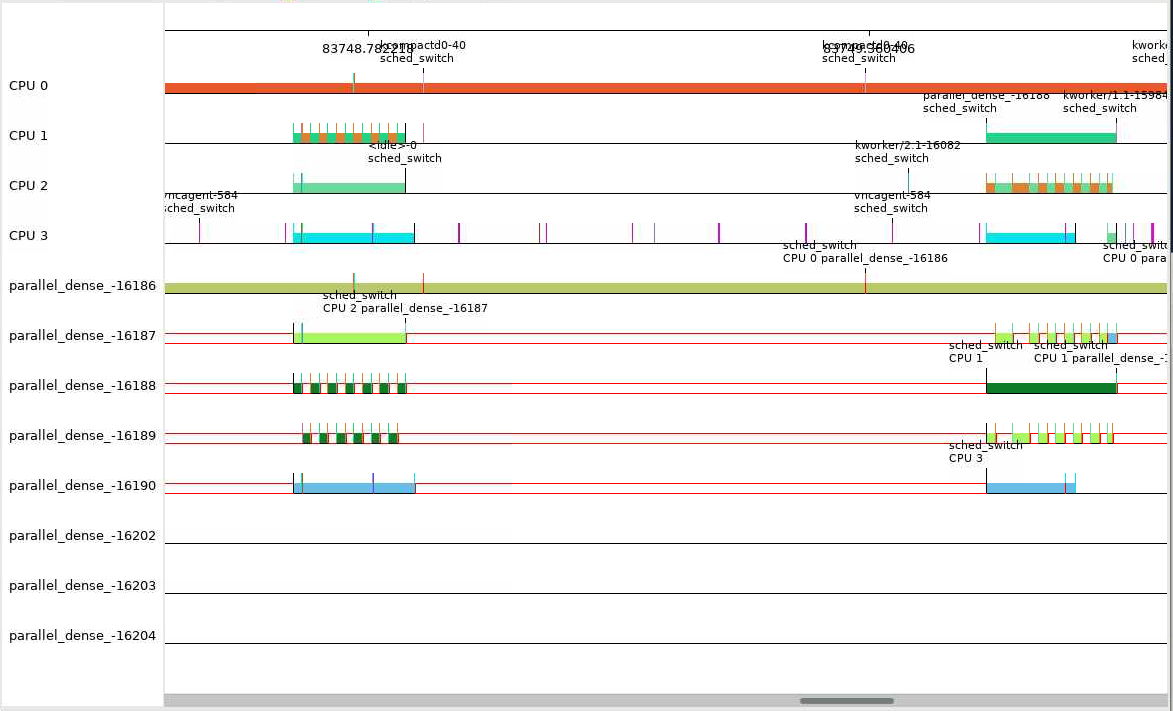
The user time and system time are almost the same as in question3, however, the elasped time is about 7 times long due to the bandwidth constraint.

7.

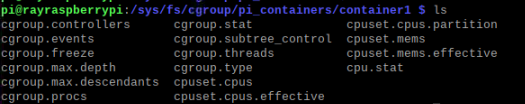
As the graphs show below, the dense\_mm program is scheduled periodically. Since we limited the demse\_mm program to run a max time of 400000us for every period of 800000us, the program will run for 0.4s and then stop 0.4s. In addition, the elapsed time is about five times the originally program.



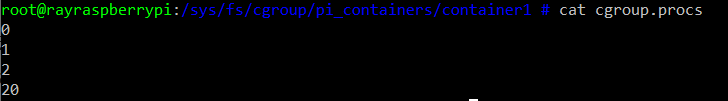




8.

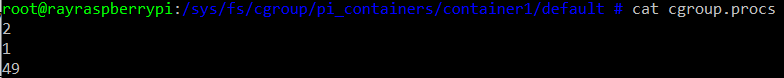




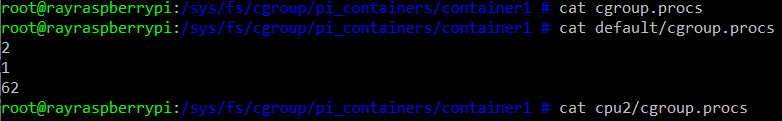


The procs in container1 and container1/default:





After removing the root PID to default, the contents in “cgroup.procs”, “default/cgroup.procs”, and “cpu2/cgroup.procs” are:

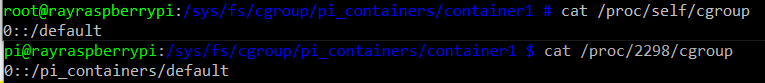


The contents in "/sys/fs/cgroup/pi\_containers/container1/default/cgroup.procs” is:



The difference of the PID values from the same file is because one is in container, and the other is not. The PID values in container have corresponding values in global views.

The contents in “/proc/self/cgroup” printed from inside and outside of container:

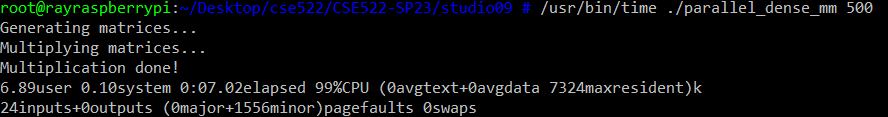


The PID for the container root is set to the default of “pi\_container”, therefore the cgroup contains the container inside of it.

9. The content of “cpu2/cgroup.procs” is:



Run dense\_mm inside the container:



The parallelism is truly restricted, because the elapsed time is limited to about four times the time as before due to CPU core usage down from 4 to 1.