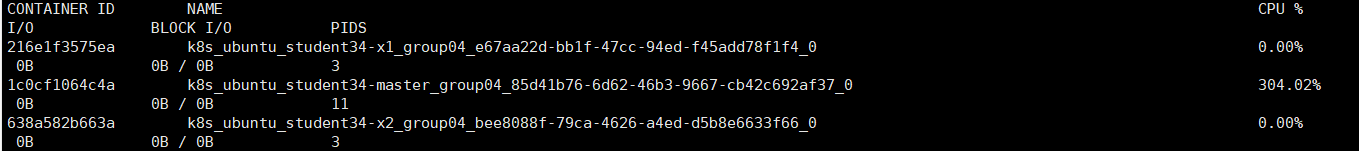
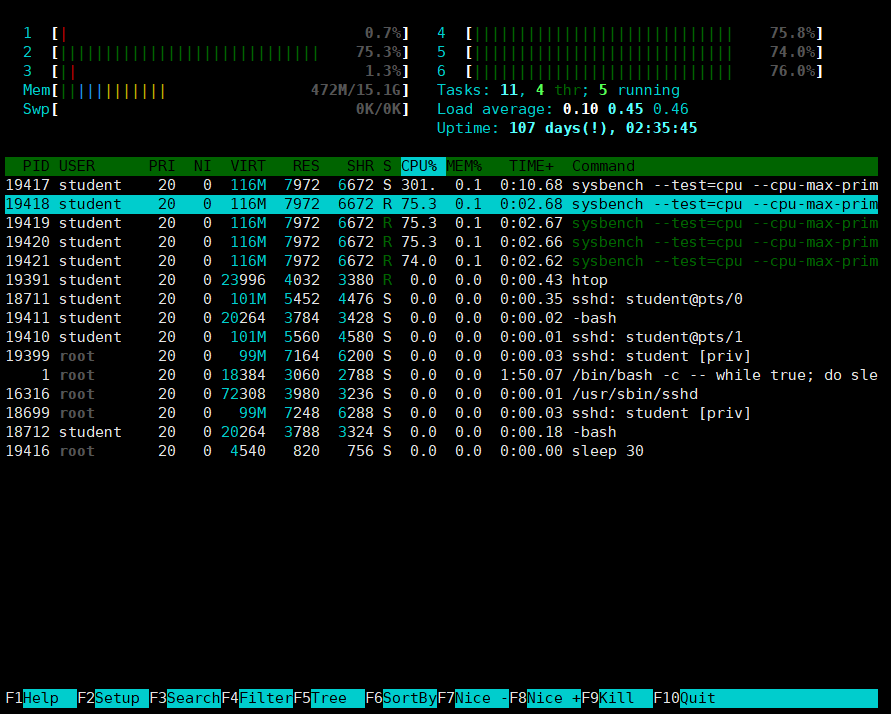


**Q1**

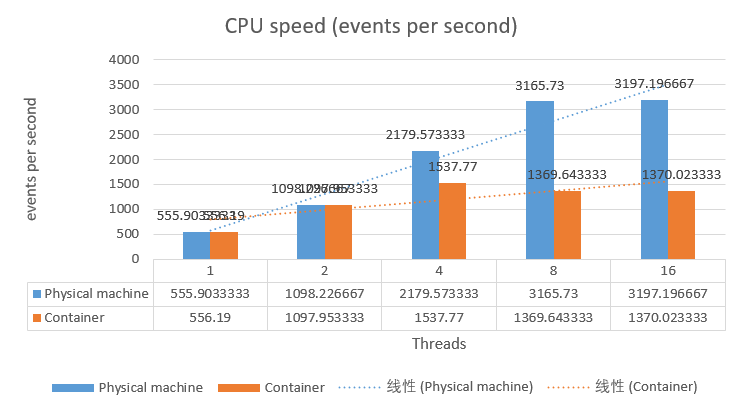
Docker stats

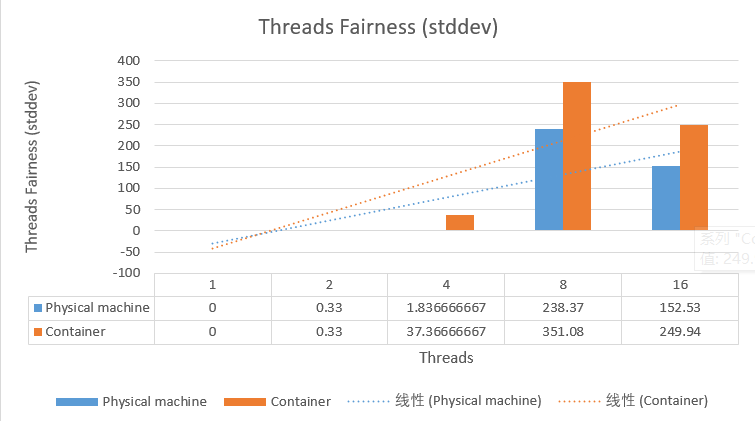


Htop



Comparison images





we can see from the first graph that as the number of nodes increases, the CPU speed of the physical machine and containers continues to increase, reaching a bottleneck when the number of nodes is 8 and 4. In addition, when the number of nodes is 2 and 4, the CPU speeds of the physical machine and the container are almost the same, but as the number of nodes increases, the CPU speed gradually widens the gap. When the number of nodes is 16 it has exceeded the two of the container CPU speed.

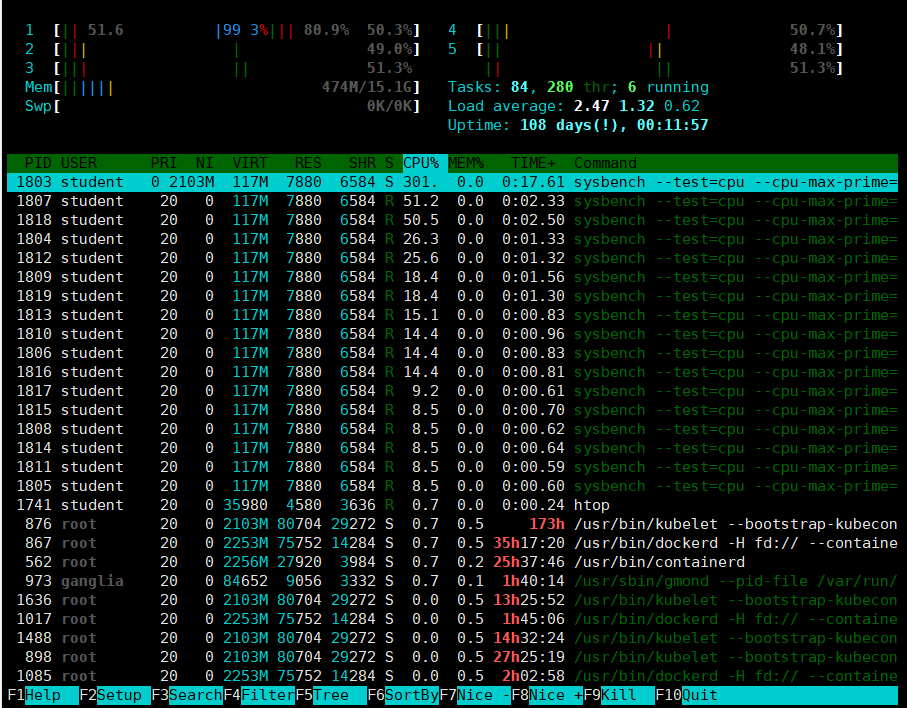
Through the analysis of the second picture, we can draw the following conclusions. 1. As the number of nodes increases, the volatility of the number of runs gradually increases. 2. When the number of nodes is 1 and 2, the volatility of the physical machine is similar to that of the container, but when the number of nodes is greater than or equal to 4, the volatility of the container is greater than that of the physical machine. 3. When the number of nodes is 8, the volatility reaches a high value, and as the number of nodes continues to increase, the volatility tends to decrease.

**Q2（需要大家再考虑一下）**

No. The CPU speed increased not linearly.

The smallest value of N that achieved the highest CPU speed on studentXX-master is 4, on studentXX is 8.

Run the **sysbench --test=cpu --cpu-max-prime=20000 --num-threads=16 run** command in container, we can see the CPU of physical machine is not used totally, and there is a upper usage limit 300% for one container.



Run the **sysbench --test=cpu --cpu-max-prime=20000 --num-threads=16 run** command in physical machine, we can see the CPU of physical machine is totally used, and there is no usage limit for it.

