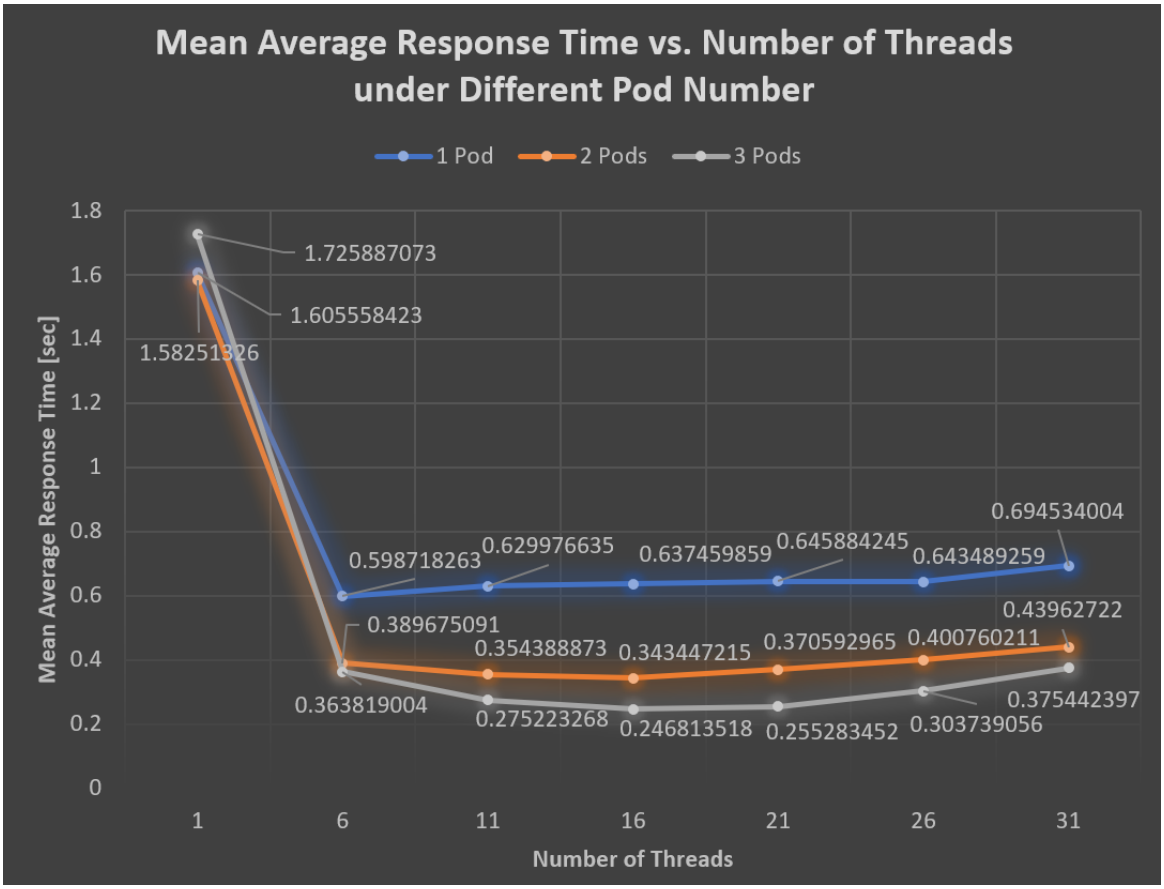
FIT5225 Cloud Computing Assignment 1

Student: Xiaohui Ding (31291252)

Tutor: Mohammad Goudarzi / Mahbuba Afrin [Thu 4 pm - 6 pm (Tutorial 6)]

-



pod=1	1	6	11	16	21	26	31
1	1.632326053	0.601726674	0.633654449	0.659432	0.641447723	0.660154583	0.704109769
2	1.582131555	0.608364291	0.622493999	0.62645739	0.640551306	0.622532373	0.676520072
3	1.602217661	0.586063825	0.633781455	0.62649019	0.655653706	0.647780821	0.702972172
mean	1.605558423	0.598718263	0.629976635	0.63745986	0.645884245	0.643489259	0.694534004
pod=2	1	6	11	16	21	26	31
1	1.601535261	0.389088294	0.35998893	0.34450082	0.386903934	0.451617897	0.422973586
2	1.56035094	0.392221384	0.358648229	0.36114323	0.333459347	0.347630072	0.459261885
3	1.585653579	0.387715597	0.344529461	0.3246976	0.391415613	0.403032664	0.436646188
mean	1.58251326	0.389675091	0.354388873	0.34344722	0.370592965	0.400760211	0.43962722
pod=3	1	6	11	16	21	26	31
1	1.730003243	0.352094483	0.285669917	0.25213861	0.240168869	0.285777591	0.351352828
2	1.586426029	0.362674868	0.267491244	0.2494206	0.268552665	0.312523961	0.383543504
3	1.861231947	0.376687663	0.272508644	0.23888134	0.257128822	0.312915616	0.39143086
mean	1.725887073	0.363819004	0.275223268	0.24681352	0.255283452	0.303739056	0.375442397
pod\thread	1	6	11	16	21	26	31
1	1.605558423	0.598718263	0.629976635	0.63745986	0.645884245	0.643489259	0.694534004
2	1.58251326	0.389675091	0.354388873	0.34344722	0.370592965	0.400760211	0.43962722
3	1.725887073	0.363819004	0.275223268	0.24681352	0.255283452	0.303739056	0.375442397

Analysis & Justification [Word Count: 406]

1. Same pod number, different thread number

For $thread = 1$, it always has the slowest average response time under the same pod number. This is because the tasks will be handled one by one.

For $thread = 31$, it always has the second slowest time. Although it can handle multiple tasks at the same time, the time cost of on frequent context switch will slow down the total time. It will also fill the storages and crush the program. To avoid this and get the experiment result, I added a swapfile to my VM to extend the virtual storage.

For the rest of threads, $thread = 6$ is the fastest of $pod = 1$, and $thread = 16$ is the fastest of $pod = 2,3$. The overall trending is a downward parabola. But comparing to the difference between $thread = 1$ and $thread > 1$, the difference among $thread > 1$ is much smaller.

2. Same thread number, different pod number

For $thread = 1$, because it will always deal with 1 thread at the same time, there is no difference of various pod number.

For $thread > 1$, the response times are all shortened. The more pods we have, the shorter response times will be. This is because there will be more pods working on handling the multiple tasks.

3. Other Observation

Actually, in an ideal condition, with reasonable number of I/O operations, if there is only a single core, the influence of thread number should be neglectable, and the response times should be similar.

In my experiments, since my code has not been optimized and has lots of I/O operations, having more threads will still help to fasten these operations. This might be the reason why the times for $thread > 1$ are still shorter than $thread = 1$ under $pod = 1$.

My overall response times are also a bit long, indicating that I can optimize the code and Dockerfile. I have to use VPN to run the experiments due to my local network policy, and VPN sometimes can be unstable. So, this is also a factor to consider.

I also notice that in my pod logs, although the load balancing seems alright, there's still some improvement can be done. Maybe I will try to replace NodePort type of Service by LoadBalancer in the future. But this time, I'll leave it what it is.

Still, lots of things to learn!