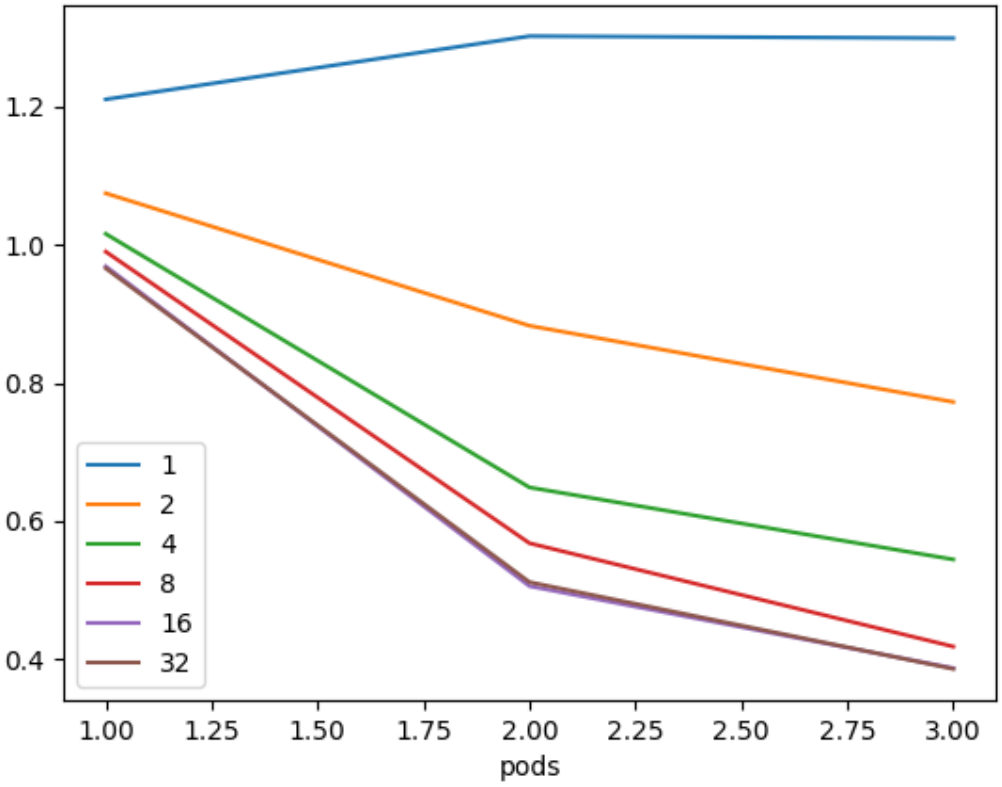
Name: Kshitij Pandey
ID: 30764505
Login: kpan0021
Tutor: Shashikant Ilager

Observation:



Output Plot

pods/threads	1	2	4	8	16	32
1	1.210754054	1.07466154	1.015911985	0.990086434	0.96883608	0.965916321
2	1.302374195	0.882770702	0.648522336	0.567662774	0.50558925	0.511282086
3	1.299420508	0.772326633	0.544183234	0.417940905	0.386723367	0.385585282

The final data after testing

After running 4 tests for each thread vs pod count and calculating the average of all the values by dividing it by the number of images which is 128, I finally got this plot. Following observations have been made after testing:

1. For thread 1 and pod 1 the average response time has been higher since the whole load is on a single pod. We can see some improvements for 2 threads vs 1 pod since

the request is now being divided into 2 threads. This improves as we keep increasing the number of threads from 2 to 4 and from 4 to 8 but the response time does not improve much after the number of threads reach 16 and 32.

2. We see the same trend for 1 thread vs 2 pods because even though we have increased the number of pods the request being sent is still in one thread but as soon as we increase the request thread from 1 to 2 and so on we see major improvements in response time from the n thread vs 1 pod because now the request is being sent simultaneously over multiple threads and now there are 2 pods to handle those requests.
3. For 3 pods the initial request with 1 thread still has a high response time since a single thread is sending all the 128 images but we see a drastic improvement as soon we start increasing the number of threads. Now with multiple request threads and 3 pods the load is getting balanced properly and the response time has improved significantly.
4. Still the average response time is almost the same when we reach 16 or 32 as number of threads. For threads 16 and 32 vs 2 and 3 pods we see that the response time is almost the same for each combination of thread vs pod.
5. This can be observed in the plot since the last 2 lines have almost merged with each other since the difference between their response time is almost equal.