Group member netID: px6 ycc5

CS425 Distributed Systems MP4 Report: Distributed Machine Learning Inference System

Our design (including architecture and programming framework) for IDunno

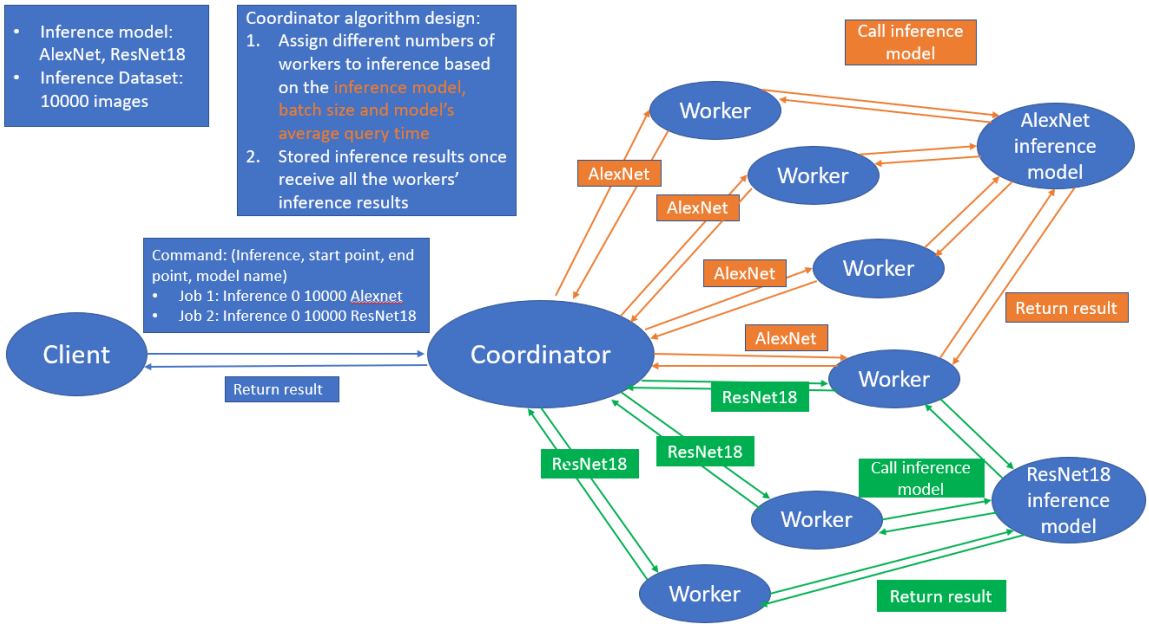


Fig 1. IDunno System Architecture

Datasets:

- The inference datasets contain 10000 images.

Model:

- AlexNet and ResNet18

Client:

- Client will send the inference queries to the coordinator, the queries format is " Inference, start_point, end_point, inference_model", which is like "Inference 0 9999 alexnet".
- After the client sends queries, the coordinator will automatically send the tasks to the worker based on the model batch size.

Coordinator:

- Coordinator will receive the queries from the client, and it will assign the queries task to different numbers of workers based on the inference model, the model's batch size(We fixed the AlexNet's batch size is 500, and ResNet18's batch size is 400) and the model past average query time.
- The coordinator will send a new task to workers once the previous task finishes. For example, when Coordinator receive the queries from client: " Inference 0 9999 alexnet", it would assign the first task to workers, and the command is like
- Once workers finish inference, the coordinator would receive the inference result from workers, and Coordinator would store all the inference results.

Worker:

- Workers will receive the job command from the coordinator, and it will according to the (start_point, end_point, inference_model) to inference.
- Once inference finishes, workers will return the result to the coordinator.

1) Fair-Time Inference:

1a) When a job is added to the cluster (1 to 2 jobs), what is the ratio of resources that IDunno decides on across jobs to meet fair-time inference?

ratio of resources[alexnet]/[resnet18]=average query time[alexnet]/[resnet18]

Group member netID: px6 ycc5

For example: average query time[alexnet]=6s,average query time[resnet18]=9s.
ratio of resources[alexnet]/[resnet18]=6/9=⅔
suppose there are 10 workers here,the master will give round(10*⅔) =7 machines the job of alexnet and give round(10*1/3) =3 machines the job of resnet18.

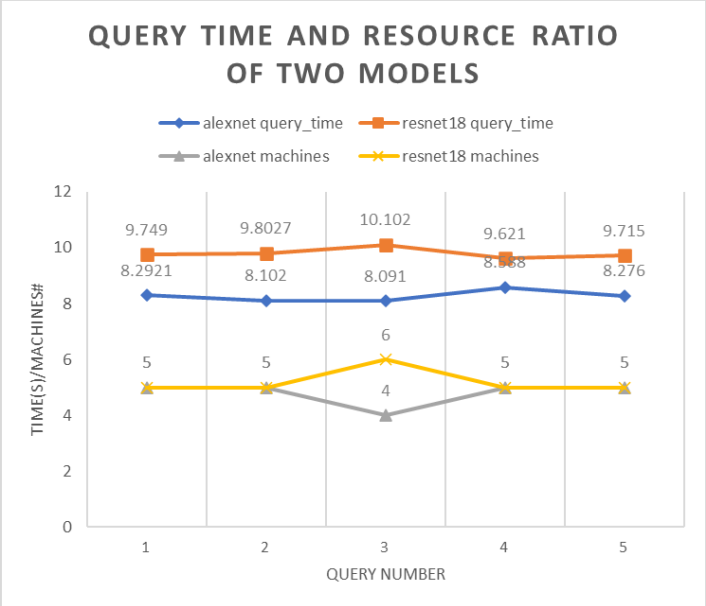


Fig 2. Query time and resource ratio of two models

1b) When a job is added to the cluster (1 to 2 jobs), how much time does the cluster take to start executing queries of the second job?

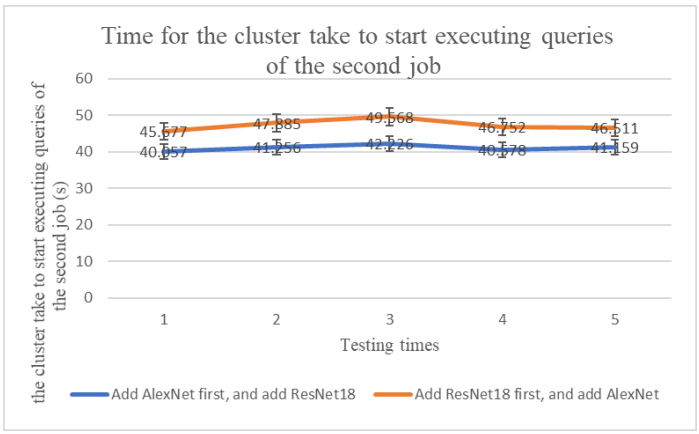


Fig 3. Time for the cluster take to start executing queries of the second job

When executing the AlexNet model(job) first, and adding the ResNet18 model(job), the cluster will take 40 to 42 seconds to start the second job. When executing the ResNet18 model(job) first, and adding the AlexNet model(job), the cluster will take 45 to 49 seconds to start the second job.

2) After failure of one (non-coordinator) VM, how long does the cluster take to resume "normal" operation (for inference case)?

time to recover=mp2 period+resend all the working inference command of failed machine.
3s+n*send time

Group member netID: px6 ycc5

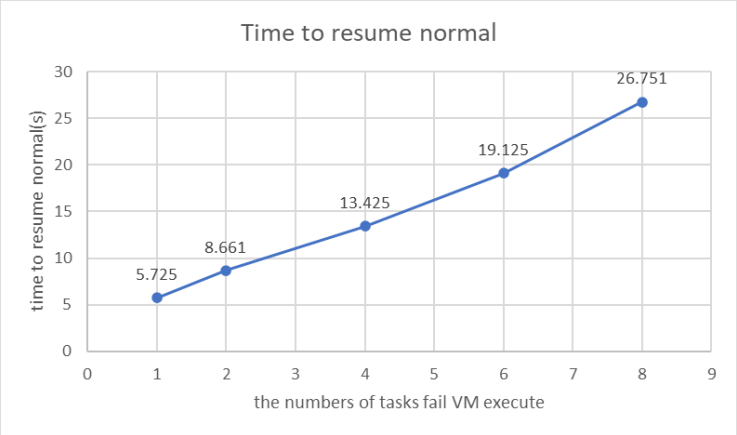


Fig 4. Time to resume normal

As the fig 4 shows, when the failed VM holds more tasks, the cluster will take more time to recover.

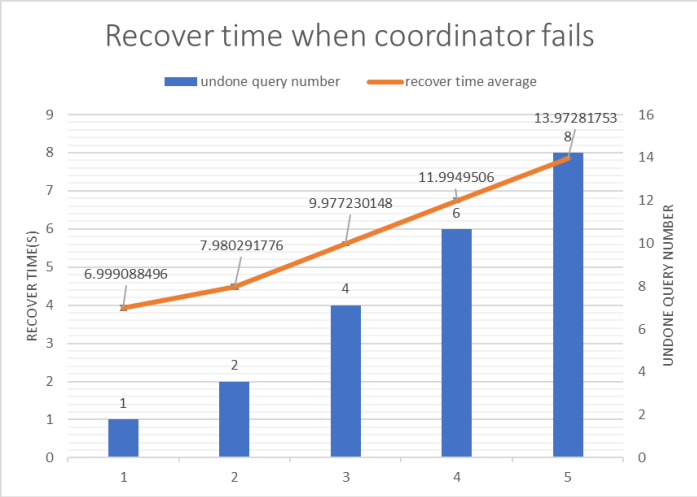
3) After failure of the Coordinator, how long does the cluster take to resume "normal" operation (for inference case)? Unless otherwise mentioned all experiments are with multiple jobs in the cluster.

Theoretically, if we analyze the time it will take to resume normal operation after failure of coordinator,

time to recover=mp2 membership detection period+time to recover all metadata of coordinator + time to resend all the working inference command of failed machine.

t_r=t_detect+t_coordinator+n*send time

Therefore, we can see that the more query is left working on the master, the more time it will take to recover.



As we can see from the measurements, it indeed has a trend of increasing recovery time when coordinator fails , as the number of undone query increases.