August 2021

A METHOD TO AUTOMATICALLY MANAGE AI TRAINING TASK ON HYBRID COMPUTING PLATFORM

HP INC

A Method to Automatically Manage AI training Task on Hybrid Computing Platform

Abstract

Developing AI model is becoming more and more complex. Researchers need to manage heterogeneous computing resources to run their training tasks. We propose a system to manage all the heterogeneous computing resources and schedule AI training tasks. This system can greatly reduce researchers' effort on managing resources and training tasks. With this system researchers can focus on developing AI models with lower cost and higher efficiency.

This disclosure relates to a distributed AI training task dispatch system to solve this problem. The system includes:

1. Computing resource management
2. Training task monitoring and scheduling
3. Quota and cost management

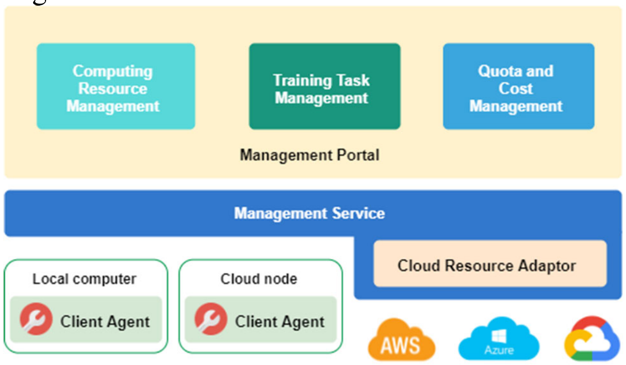

Fig. 1 System Components

Fig 2 describes the overall system architecture.

1. A client agent will be installed on every computing node to gather info, deploy training environment, run training tasks, collect task running status and output.
2. Management service communicates with client agents and exposes operational APIs to management portal.
3. AI researchers use management portal to manage their resources and training tasks.

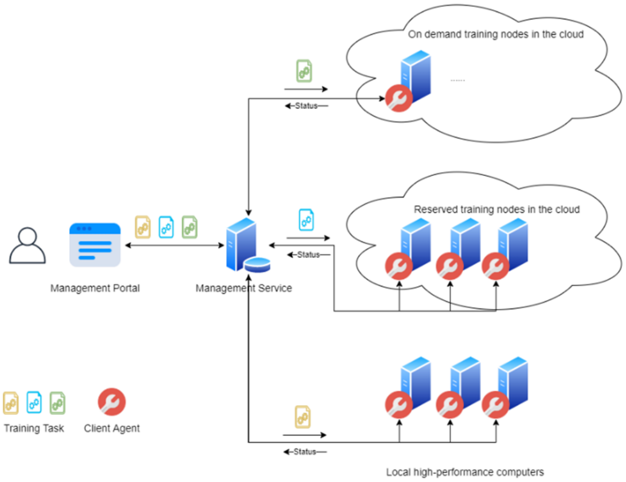
On demand training nodes in the cloud
......
Status
Reserved training nodes in the cloud
Status
Management Portal
Management Service
Training Task
Client Agent
Status
Local high-performance computers

Fig. 2 System Architecture

Computing Resource Management

The system manages three types of computing resources:

1. Local high-performance computers owned by AI research team
2. Reserved training nodes in the cloud.
3. On demand training nodes in the cloud.

For local high-performance computers and reserved training nodes in the cloud, the system maintains a resource table, which records the computing power (TFLOPS) and GPU memory of each node.

• Tracking AI Training Task Status

The system uses two methods to estimate the remaining training time:

1. Dynamically collect the prediction error of AI models at runtime. Then use curve-fitting algorithm to find a decline fitting function which can describe how the error decreases over time. Then use this fitting function to estimate when the prediction error will meet the target. We use polynomial curve-fitting and multiple regression curve-fitting algorithm to find the fitting function.
2. Collect historical training data, which include but not limited to, type of AI training tasks (e.g., computer vision, nature language processing, etc.), type of AI model (e.g., CNN, RNN, GAN, …), model size, hyper parameters (e.g., learning rate), prediction error. Then use these historical data to train an MLP to predict how long the training result will converge. Fig. 3 illustrates a proposed architecture of the MLP.

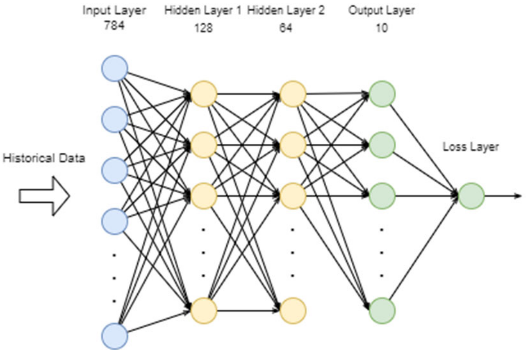

Fig. 3 MLP model

The system also monitors the running status (running, idle, error) of each node. For the running tasks, the system dynamically shows the estimation of the remaining running time. .For on-demand training nodes in the cloud, the system automatically allocates training nodes in the cloud and sets up the AI training environment needed by the task.

Training task monitoring and scheduling

• Rule-Based Task Management

Based on the computing resource management, the system helps AI researchers to manage their training tasks.

Typically, an AI training task includes:

1. Training framework
2. AI model
3. Hyperparameters
4. Training data
5. Need distributed training or not
6. Minimum GPU memory required

For AI researchers, they usually have two additional requirements for the training task:

1. The training task is time-sensitive and it should be processed as quickly as possible.
2. The training task is cost-sensitive and the cost should be as low as possible.

For the training data, there are some conditions which limit the resource type that the training task can run on. For a given training task, there are three situations:

1. The task can only run on the local computers.

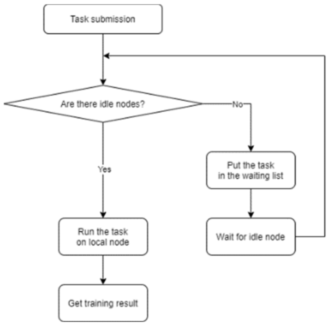

Fig. 4 Process flow diagram for situation 1

2. The task can only run on the cloud nodes.

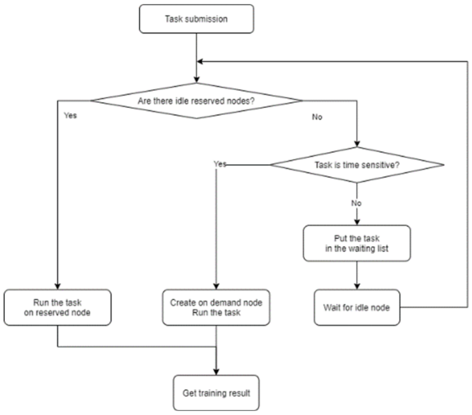
Task submission
Are there idle reserved nodes?
Yes
No
Yes
Task is time sensitive?
No
Put the task in the waiting list
Run the task on reserved node
Create on demand node Run the task
Wait for idle node
Get training result

Fig. 5 Process flow diagram for situation 2

3. The task can run on both local computers and cloud nodes.

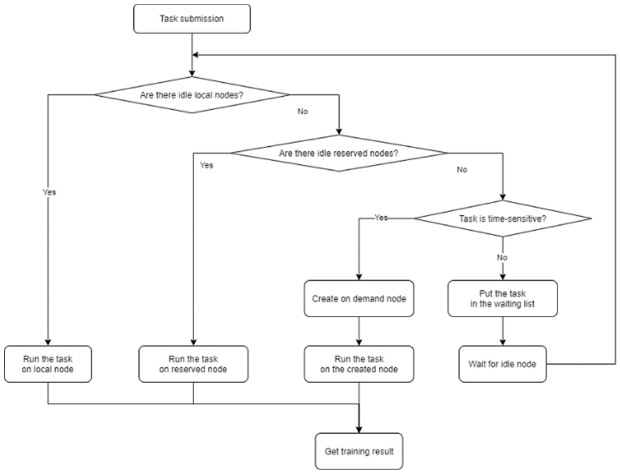
Task submission
Are there idle local nodes?
No
Are there idle reserved nodes?
Yes
No
Yes
Yes
Task is time-sensitive?
No
Create on demand node
Put the task in the waiting list
Run the task on local node
Run the task on reserved node
Run the task on the created node
Wait for idle node
Get training result

Fig. 6 Process flow diagram for situation 3

- **Dynamic Task Schedule**

In below situation, a training task runs into trouble and needs to be rescheduled to obtain more computing resources.

1. The training task meets out of memory exception.
2. The estimated remained time is significantly longer than researcher's expectation.

In these cases, the system will take the snapshot of the running task and try to allocate more computing resources. If success, the system will continue the task with more computing resources. Otherwise, the system will notify the researcher.

When a local / reserved training node completes a task, if there are no more tasks in the waiting list, the system will try to find a task running on an on-demand node which can also run on the idle node. If there is such a task and the estimated remaining running time is long enough (more than 1 hour for example), the system will take the snapshot of the task and try to reschedule it to local / reserved training node.

Quota and cost management

The system tracks the following metrics:

1. Total use rate of local / reserved training nodes.
2. Total on demand node create count, running time, estimated cost.
3. Local / reserved training time used by each researcher.
4. On demand node create count, running time, estimated cost by each researcher.

Disclosed by Yiling Yin, Zi-Jiang Yang, Sheng Cao, Xi He, Qi-Feng Tang, Jian Gao and Zhoujian Zhang, HP Inc.