FUTUREWEI
Technologies

Four Challenges of MindSpore

2022-11-23
OpenMindSpore Project
Silicon Valley System Software Lab

Contents

Overview

- MindSpore experienced four challenges in the research and production based on the inputs from BAO Chong at the open community meeting on 2022-11-23.

Goals

- Understand this challenges and digest the pain points
- Provide analytical thoughts and solutions
  - Suggestion input for MindSpore architecture design
  - Improve dynamic, training accuracy, reliability and quick recovery on failures
- Plan for next year project based on needs and our strengths.

Challenge 1: Pains in Auto Parallelism (AP) and AutoDiff (1)

Community can run forward network which can be split into several sub-graphs to run. But they found two implementation issues:

Issue 1: Difficult to combine dynamic splitting with AP.

- AP perceive the **global** network topology. We ***cannot*** make AP to perceive the small domain of the network. Instead, AP will construct a big domain. This will cause conflicts. Therefore, all subgraphs can only be run on a single card now.
- The community are seeking the solution to define a small network domain for the AP.

Challenge 1: Pains in Auto Parallelism (AP) and AutoDiff (2)

Issue 2. MS graph splitting approach introduces the complexity of Auto Diff.

- MS splits the graph using Python, which is easy to implement. The reading cost is also low.
- We first split subgraphs, and then sequentially connect those subgraphs according to the backward logic. Then it compute the differentiations.
- We cannot see a whole graph after we construct a backward graph.
- Pathways splits the graph after computing differentiations. It is implemented in C++, and the implementation cost is high.

Challenge 2: High Lose Precision on Ascend Chips

Reproducing SOTA model in MS, found an issue below:

- On Ascend cards, the lose precision of MS is higher than that of PyTorch (on GPU).
  - The model team has no detail analysis yet.
  - SU Teng said that he met similar issue on big models. Su's discovery: some operators (ops) were originally sequential but modified after the optimizing for Ascend chips. Those ops were updated with parallelism. It looks reasonable, however it introduced different lose precision.
  - It is very difficult to compare with the PyTorch model to confirm the correctness of our MS model.
  - This also causes difficulties to reuse the hyperparameter in the paper.
- On GPU, the lose precision of MS is close to that of PyTorch
  - Even the ops are implemented differently but the lose is still better than the Ascend's version.

Challenge 3: Introduce Dynamic Computing

- Pathways can dynamically add new layers based on the current training output results. How can MindSpore? Opportunity?
  - The big model team finds that it is useful to have such feature. Dynamically add new layers based on intermediate training results and initialize the weights.
  - It is very difficult to do static analysis for auto parallelism on such a dynamic graph.

Challenge 4: Long Disaster/failure Recovery Time

- Happened frequently on the large-scale distributed training for big models.
- **Causes**: hardware failure/exception, sometime don’t know the reason
- **Actions**: interrupt current training and recover to the previous training
- **Scenarios**:
  - Recovery of trainings takes tens of minutes for the graph re-compilation and HCCL reconstruction due to HW changes.
  - *Recovery of big models*: can spend up to 2 hours for the pangu model
  - *Recovery of car BU trainings*: takes tens of minutes to load/store TBs data (although the model is not so big)
- **Consequence**: user experience for customers is bad.