Summary

Alpa is a framework that automates model training through generation of execution plans that take care of data, operator and pipeline parellelism. It achieves this through a hierarchical model that splits optimizations into intra-operator and inter-operator methods.

Intra-Operator Parallelism

Refers to partitioning of tensors along their dimensions to execute them at different portions of the time. For splitting and merging of the operator, communication is required.

Inter-Operator Parallelism

Refers to assiging different operators of the graph to execute on each graph.

Alpa first optimises the model w.r.t inter-operator parallelisation latecy through slicing the model and device cluster into stages and device meshes and their respective mapping onto the high-speed device meshes. For intra-operator parallelization, the cost of executing a stage is minimized on a given device mesh. The intra-op pass is repeatedly queried for each mesh created by the inter-op pass.

Strengths

- Considers both intra-operator and inter-operator parallelism, hierarchical nature ensures that best possible optimization occurs for a wide variety of models
- Facilitates deployment over a large range of servers

Weaknesses

- Focused on throughput, not latency, hence cannot be used for inference of large models such as GPT4
- Does not model communication latency between different stages and assumes its small
- Does not optimize for best possible overlap between computation and communication

Possible Improvements

- Modelling communication along with computation allows Alpa to be more useful in cases where large models are trained, overlapping compute with communication, reducing latency