

Scaling Distributed Machine Learning with the Parameter Server

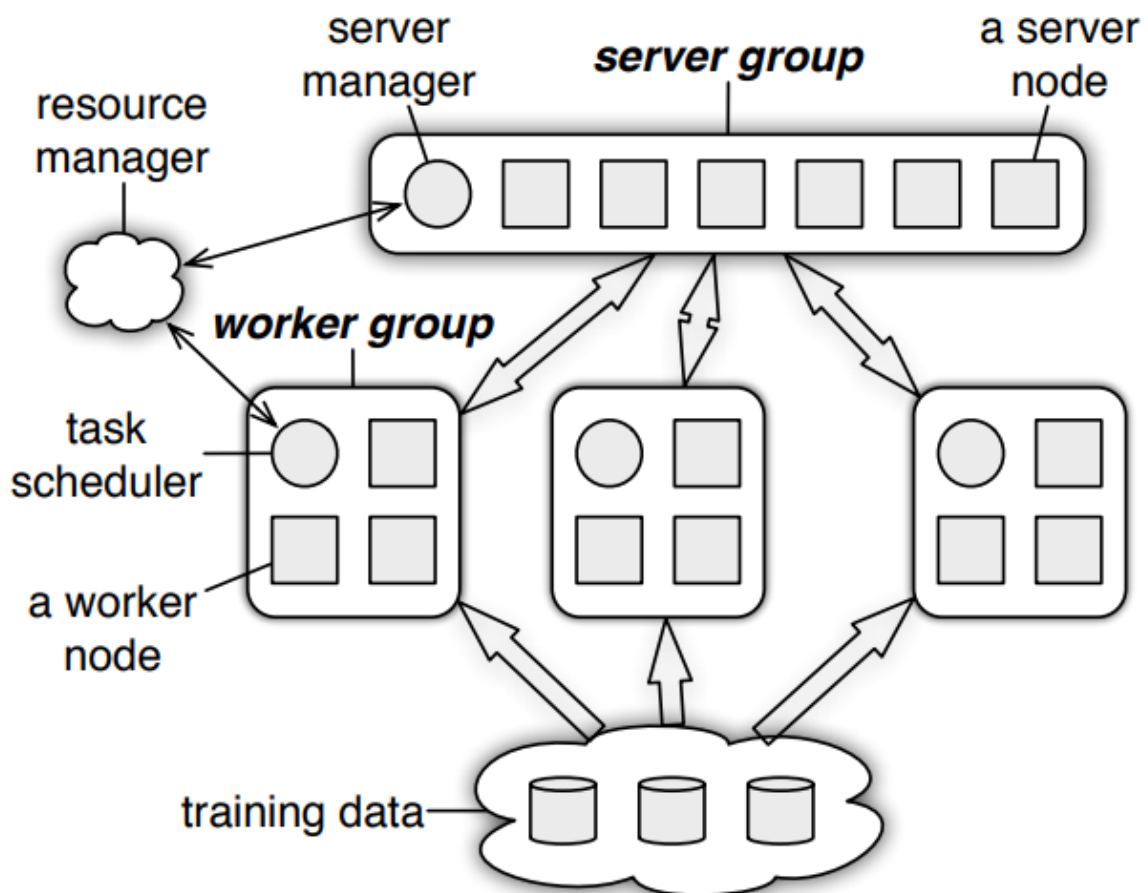
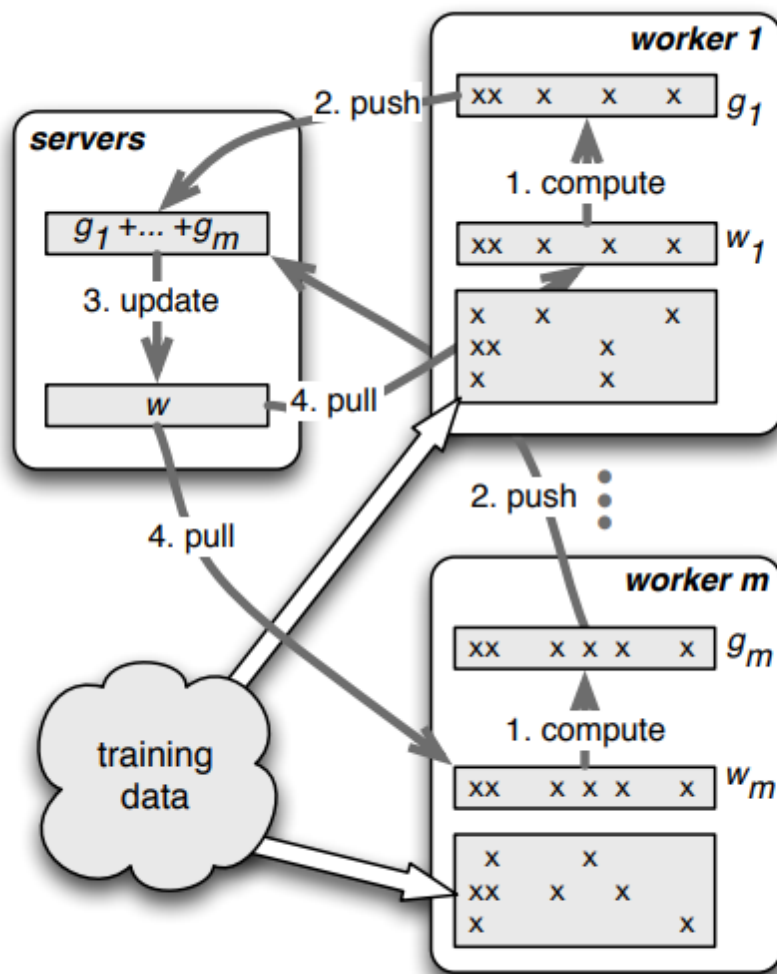
- 这篇论文可以说是分布式机器学习的milestone式的作品。并且吃惊的是，这篇论文发表于2014年（其在NIPS上投的那篇甚至是2013年），位于以深度神经网络为代表的AI大发展时期的最初期，其设计的parameter server的框架一直到现在还在应用，再一次感叹李沐大神的强大，，，
-

Overview

- Parameter server is composed of two kinds of nodes:
 - server node: store the global shared parameters, perform the works of gradient aggregation, and seems it has a better view of all the parameters, it is better to perform some specific user defined works
 - worker nodes: store the related parameters to perform the works, store partition of the data. push the sub-gradient to the server node and pull the update of parameter back.
 - Contributions:
 - efficient communication: the asynchronous communication model does not block computation.
 - flexible consistency models: relax the strict requirement on the data consistency, improve on the system efficiency (while I am very curious about its detailed trade-off in the convergence and system efficiency)
 - Elastic scalability: Enable Adding server nodes and worker nodes during training. (while the paper mention that the adding worker nodes will cost a lot, so it will not add a new worker node if one fail: whether we can research in a better way to add a worker node?)
 - Fault tolerance: the time to recover from failure is short.
-

Some interesting points

- Architecture
 - The architecture can be well illustrated by the following two graphs.



- Range push and pull
 - Since the communication between server nodes and worker nodes is crucial in the whole performance, it is better to transfer a batch of data instead of transferring one data one time. The parameter server enables push/pull a range of data in one time.
- User Defined functions on server
- Asynchronous Tasks and dependency
 - The iterations are performed in asynchronous way for the better parallelism thus better system efficiency. However, user can also add dependency to ensure the logic of algorithm.
- Flexible consistency
 - As mentioned above, the parameter server relax the strict requirement on the data consistency, because the inconsistency will not affect the final result of model(it can be proved), but only potentially slows down the convergence progress of training. Facing this trade-off, the parameter server decide to make it flexible.
- Choice of tasks dependency:

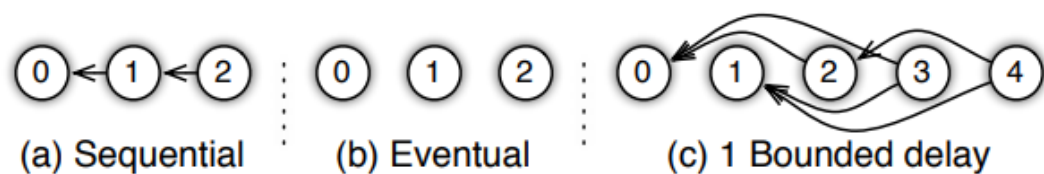


Figure 6: Directed acyclic graphs for different consistency models. The size of the DAG increases with the delay.

- Boundary Delay: when a maximal delay time t is set, a new task will be blocked until all previous tasks t times ago have been finished.
- User-defined filters
 - 指用户可以用过滤器选择性地同步部分参数