How ML on Spark differs from Tensorflow:

1. Spark abstracts complexities of distributing computations
Spark provides high level abstractions in multiple languages (Java, Scala, Python) that hide the underlying work distribution operations.
TF does not abstract work distribution like spark does and allows the user to specify where pieces of the graph are to be placed.
This can be seen as a Pro or Con for either system depending on how much control we want on our system.

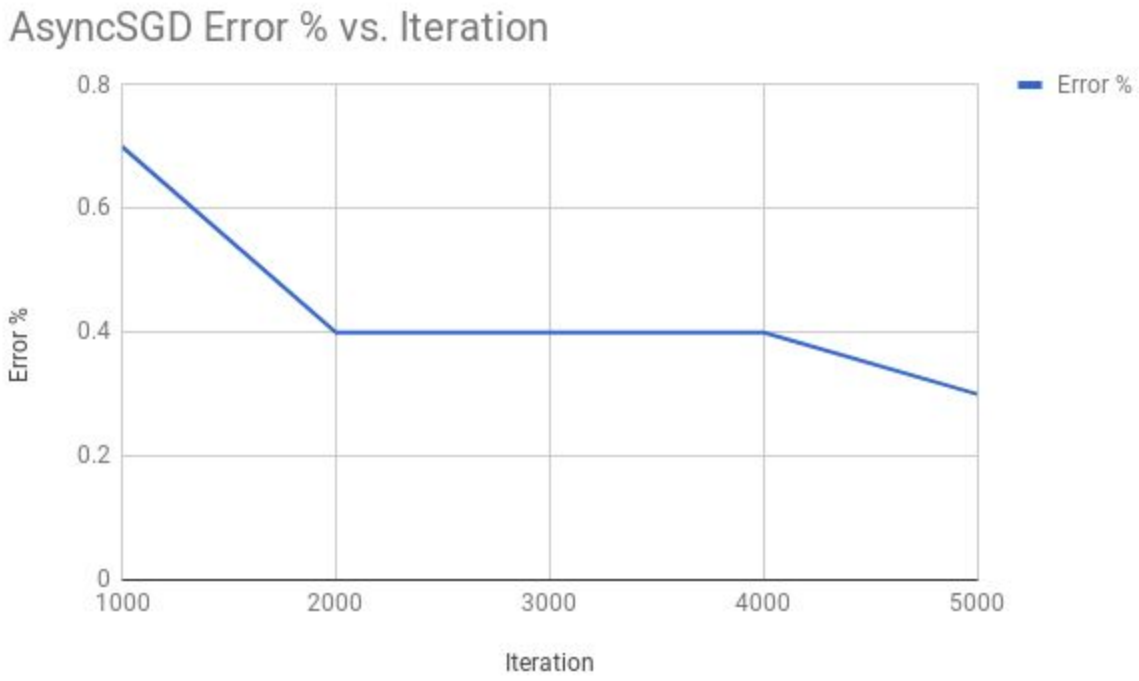
2. Spark automatically recovers from failures
Spark can recover from node failures automatically.
From the TF whitepaper: "When a failure is detected, the entire graph execution is aborted and restarted from scratch.". TF supports
checkpointing and unlike Spark, the recovery is only at the checkpoint rather than on an individual worker node failing.
This is a Pro for Spark and a Con for TF.

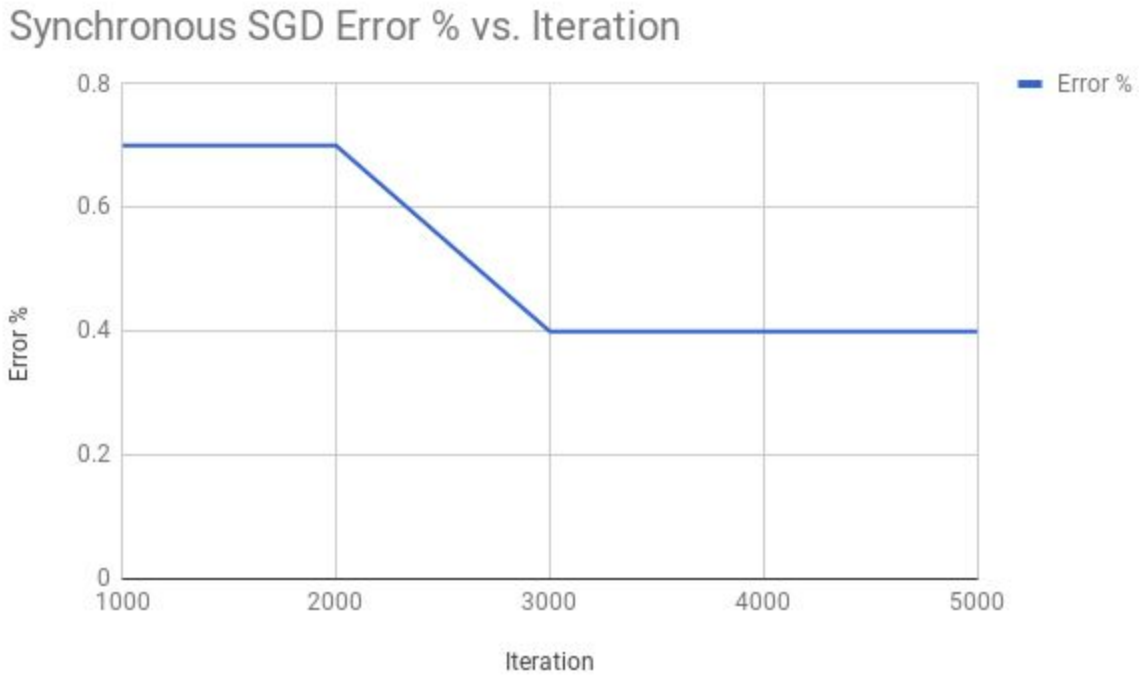
3. Spark records all the computation state in master whereas TF records state in parameter servers
Thus, TF provides more flexibility.

4. TensorFlow has the ability to perform partial subgraph computation. This allows partitioning of a model so that distributed training may be done.
This allows TF to support Model Parallelism.
This is a Pro for TF.

Graph of test errors

We ran the programs for ~5 hours since each iteration took a lot of time. Resource bottlenecks detailed in next section


AsyncSGD Error % vs. Iteration
Error %
0.8
0.6
0.4
0.2
0
1000
2000
3000
4000
5000
Iteration
Error %

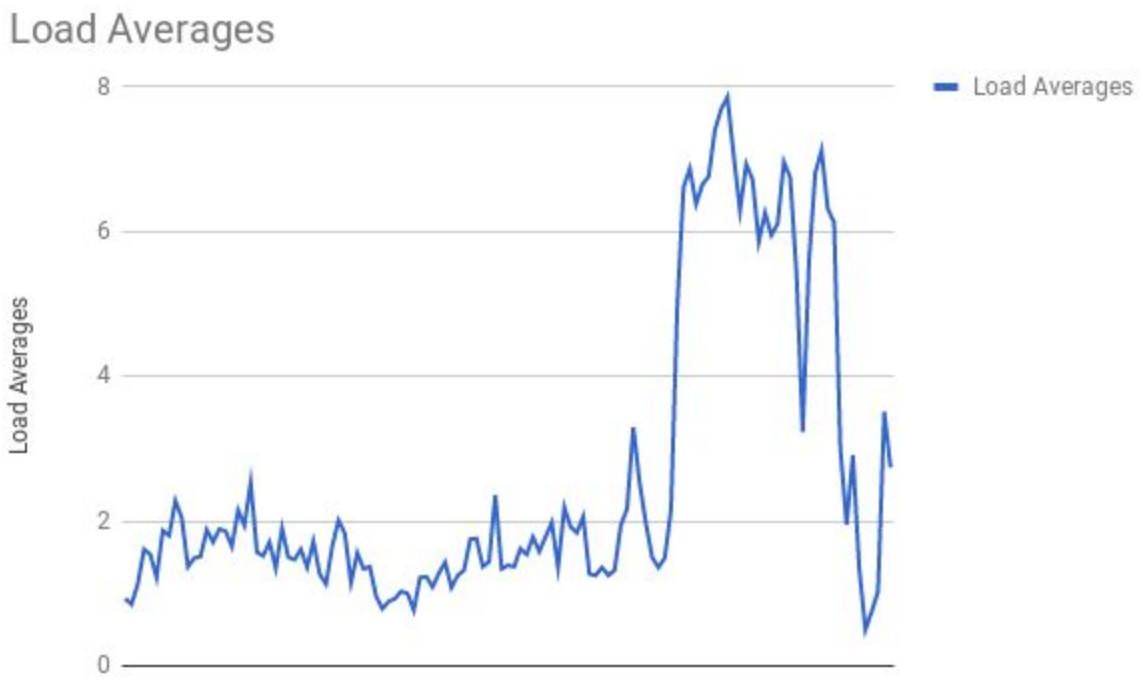

Synchronous SGD Error % vs. Iteration
Error %
0.8
0.6
0.4
0.2
0
1000
2000
3000
4000
5000
Iteration
Error %

Bottlenecks

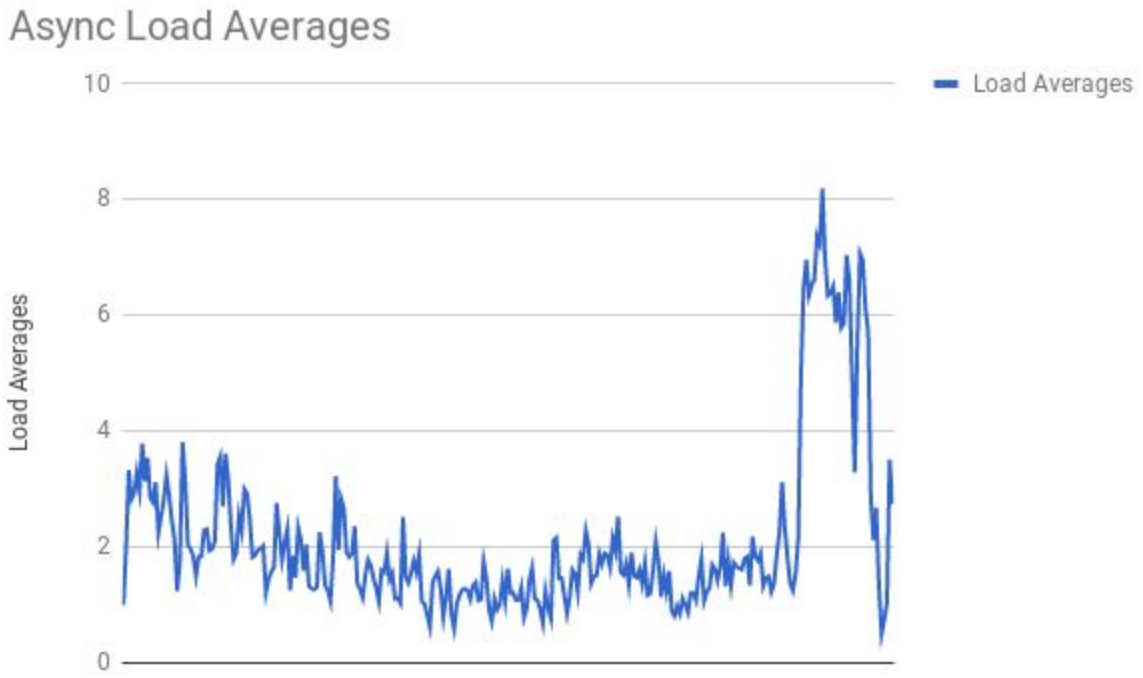
There were no other processes running on the cluster when TF model training was executing

- Load Average i.e. processes waiting for CPU

Synchronous SGD:
We can see that there is are many processes waiting for CPU. At times there being 8 processes waiting for CPU => There was contention for CPU

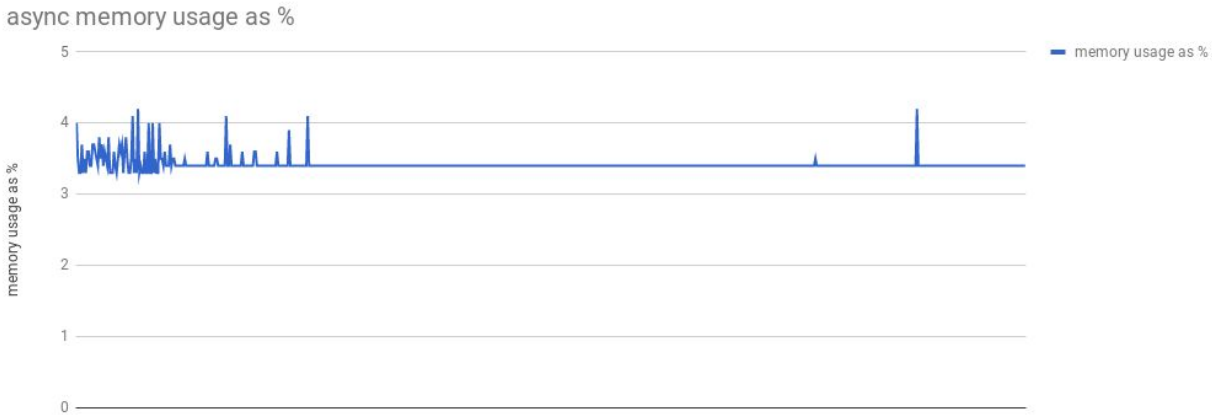


Async SGD:
With asynchronous processing as well there was a lot of contention for CPU, the number of processes waiting for CPU going above 8

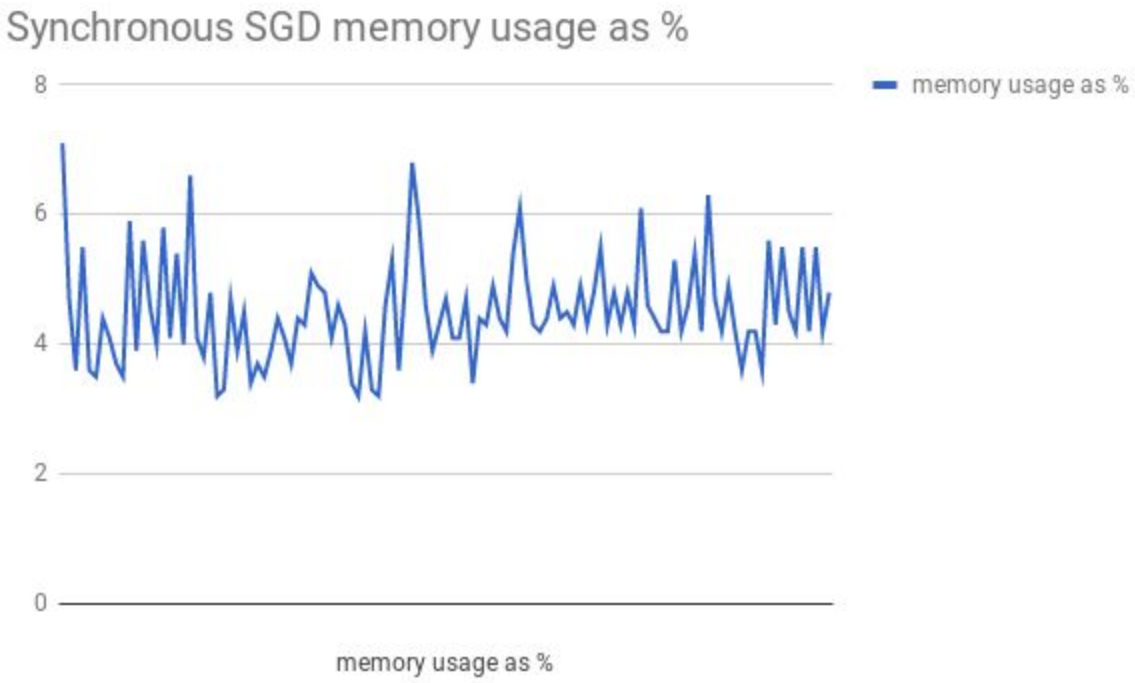


- Memory usage

Async SGD:This was not a bottleneck for the Async execution.



Synchronous SGD: Memory usage was higher than in async execution but it was not high enough to be a bottleneck for execution


Synchronous SGD memory usage as %
8
6
4
2
0
memory usage as %
memory usage as %