Tensorflow

Summary :

Tensorflow is a machine learning system that operates at large scale and in heterogeneous environments
It uses dataflow graph to represent :

- Computation
- Shared state
- Operations that mutate on these state

It maps nodes of dataflow graph across many machines in a cluster.
It supports usage of CPUs, GPUs, TPUs (custom ASICs for machine learning)
It provides a lot of flexibility to application developer

Discussion on Section 1 (Introduction) :

- There are two approaches to make interfaces
  - Library based approach : tool is implemented as an additional library in any language (like EbbRT)
  - Language based approach

Discussion on Section 2 (Background) :

- What are the differences between Torch and Tensorflow?
  - Tensorflow is more high level (more abstractions) than torch.
  - In Torch we have scalability issue as Torch is a single machine framework

- Different features of DryadLINQ
  - DryadLINQ is generalisation of MapReduce
  - It is functional
  - Provides richer set of operators

- Whether the training algorithm is deterministic or non-deterministic?
  - Training algorithms are deterministic in nature. They generate same outputs on same inputs.

- Disadvantage of Spark :
  - It is immutable to input data
  - So provides static data analysis

- What are the different problems with the parameter server approach?

  - Dedicated machines are required for the parameter servers
  - Lot of manual work required
  - Scaling is difficult
  - Limited machine learning algorithm support

Discussion on Section 3 (Execution Model) :

- Dataflow graph vs Control flow graphs
  - In Dataflow graph, nodes represent operations while in control flow graph nodes represent a single instruction
  - Edges in dataflow graphs are streams of values unlike single value in case of control flow graph
  - In dataflow graphs we have model parallelism and data parallelism. In control flow graph we can have instruction parallelism (instruction pipelining)

- What is Shuffle queue and why is it required?
  - Shuffle queue reduces the probability of sticking in local maxima
  - Shuffle queue shuffles the input data before preprocessing.

- Strong Consistency and Weak Consistency?
  - Strong consistency requires more restrictions on updation of parameters.
  - Weak consistency:
    - Less restrictions on updation of parameters i.e. updates can be associative and commutative.
    - Can be done asynchronously
    - Neural Networks

- Tensor Flow as approximation System
  - TensorFlow is an approximation system.
  - TensorFlow doesn’t require strong consistency.

- What is Rendezvous Key?
  - Port no. for all devices connected to TensorFlow
  - Provides generalised send() and recv() for heterogeneous devices.

- Scan
  - Abstracts the entire loop in a single Apply node in the graph.
  - Keep the value for the last step of an output in memory if the whole sequence is not needed.

Discussion on Section 4 (Extensibility Case Study) :

- Why is Embedding matrix used?
    - Used for reducing large sparse features matrix to a smaller dense matrix.

- How are Static Graphs useful?
    - Static graphs cannot be changed while execution phase.
    - Provides a room for optimisation to software.

- What makes graph Non-Reducible Graph?
    - Graphs which have goto statements cannot be reduced due to complex control flows.

- Why are Backup Workers required?
    - Used for stragglers, works proactively
    - Keeps m out of n updates.

Discussion on Section 5 (Implementation) :

- What is cudaMemcpyAsync?
    - Asynchronously copies data from device to device.
    - Might return before copying

- Why is Fsync required?
    - Fsync() flushes all modified in-core data of the file referred to by the file descriptor fd to the disk device so that all changed information can be retrieved even if the system crashes or is rebooted.

Discussion on Section 6 (Evaluation) :

- Single machine benchmarks
    - TensorFlow’s performance is within 6% of Torch even after adding support for scalability
    - Neon outperforms because it uses hand-optimized convolutional kernels implemented in assembly
- Synchronous replica with backup servers clearly outperforms asynchronous replicas
- Are there any algorithms which require strict consistency and cannot be executed using asynchronous replicas?
    - Yes, if we consider just gradient descent method and not the stochastic gradient version, it needs strong consistency among updates and can be executed using synchronous replicas only.

Discussion on Section 7 (Conclusion) :

- Tensorflow offers a set of uniform abstractions that allow users to harness large-scale heterogeneous system, both for production tasks

and for experimenting with new approaches
- Its flexible dataflow graph representation enables power users to achieve excellent performance.
- Current implementations of mutable state and fault tolerance suffice for applications with weak consistency requirements