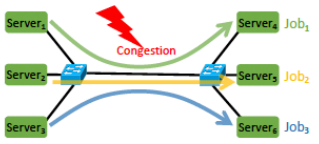
CASSINI: Network-Aware Job Scheduling in Machine Learning Clusters

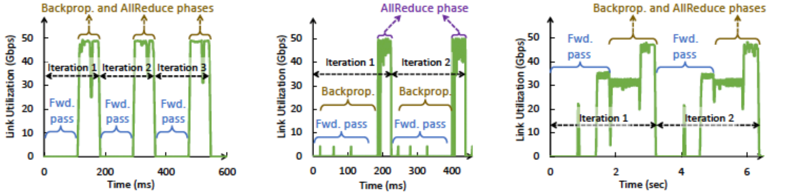
Sudarsanan Rajasekaran, Manya Ghobadi, Aditya Akella

CONGESTION IN ML CLUSTERS

- Machine Learning training is becoming a dominating workload in datacenters
- There is a significant need to *efficiently* train large Deep Neural Networks (DNNs)
- Today's ML clusters ignore the impact of congestion when multiple jobs share a network link
- In our work CASSINI, we develop simple and effective approach to place multiple jobs on network links while minimizing congestion
- CASSINI, integrates with existing ML schedulers to find optimal job placements that are network aware



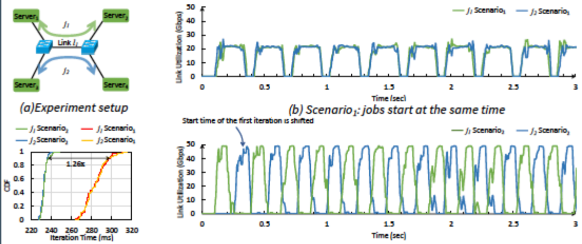
PERIODIC COMMUNICATION PATTERN IN DNN JOBS



(a) Data parallelism in GPT-1 (b) Pipeline parallelism in GPT-2 (c) Hybrid data/pipeline/tensor parallelism in GPT-3

- In DNN training the network demand repeats itself across all iterations, as long as the hyper-parameters remain the same
- The network demand of an iteration may consist of multiple Up and Down phases, depending on the parallelization strategy

INTERLEAVING COMMUNICATION ACCELERATES TRAINING!



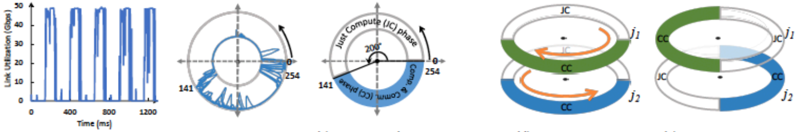
(a)Experiment setup
(b) Scenario$_1$: jobs start at the same time
(c) Scenario$_2$: the start time of j_2 is shifted by 120 ms
(d) CDF of iteration time

This experiment demonstrates the power of interleaving network demands in DNN jobs.

Two DNN training jobs share a 50Gbps network link.
In Scenario$_1$ both jobs start at same time, while in Scenario$_2$ job j_2 is delayed by 120ms.

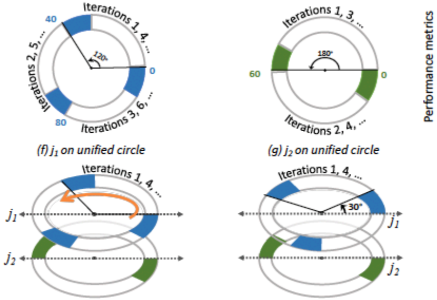
Interleaving of network demand, accelerates training of both jobs in (c).

CASSINI USES GEOMETRIC ABSTRACTION TO FIND COMPATIBILITY OF JOBS



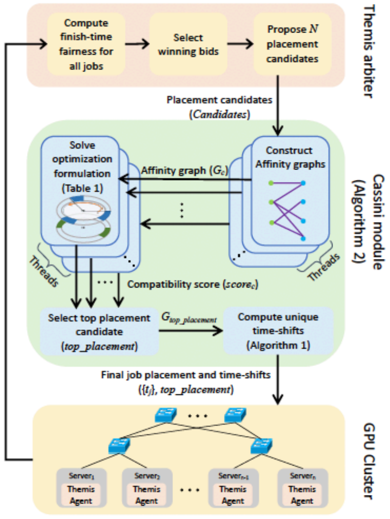
(a) Network demand (b) Network demand rolled around a circle (c) Geometric abstraction (d) Communication demands colliding (e) Communication demands are interleaved

- Perfectly interleaving the Up and Down phases of different jobs is not always possible
- Finding which job combinations can interleave network demands is challenging, since
  - jobs have thousands of iterations
  - jobs have varying iteration times, communication patterns
- CASSINI uses a novel geometric abstraction to efficiently determine the optimal interleaving of network demands
- CASSINI defines a metric called *compatibility score* that captures the potential degree of interleaving of jobs
- We convert our geometric abstraction into an optimization formulation, which computes the compatibility score for a given job combination
- Using optimizations based on simple geometry and graph theory, CASSINI acts as a pluggable module for existing ML schedulers and ranks different placement candidates based on compatibility scores of jobs sharing network links



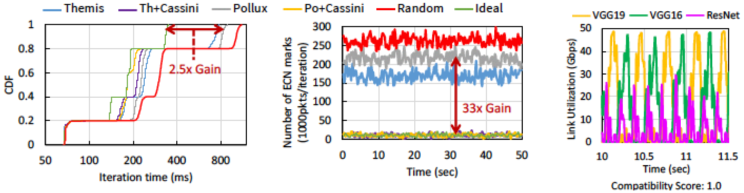
(f) j_1 on unified circle (g) j_2 on unified circle
(h) Overlaying the unified circles (i) Final representation with rotation

LCM of iteration times is used when jobs have different iteration times, unified circle has perimeter set to LCM of the iteration times



(j) System design of using CASSINI to augment Themis

RESULTS



(a) CDF of iteration times (b) ECN marks for DLRM (c) Interleaving of network demands by CASSINI

Experiments with 13 common ML models on a 24-server testbed demonstrate that compared to the state-of-the-art ML schedulers, CASSINI improves the average and tail completion time of jobs by up to 1.6x and 2.5x, respectively. Moreover, we show that CASSINI reduces the number of ECN-marked packets in the cluster by up to 33x.

CONCLUSION

- CASSINI is a simple but effective approach that can integrate existing cluster schedulers to allow them to accommodate multiple ML jobs' network needs
- We introduce a novel metric, called compatibility score, to rank different GPU placements when jobs compete on network links
- Our evaluations show that CASSINI improves the average and tail job completion times by up to 1.6x and 2.5x, respectively
- We also show that CASSINI reduces the number of ECN-marked packets by up to 33x in the network
- Our experiments show significant gains in various types of parallelization strategies and also improve training on hybrid clusters with multiple GPUs per server

This work was supported in part by the ACE Center for Evolvable Computing, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.