PETUUM: An ML-Centric System for Big Learning

Wei Dai, Jinliang Wei, Jin Kyu Kim, Qirong Ho, Xun Zheng, Pengtao Xie, Seunghak Lee, Eric Xing (CMU)

Unique Nature of ML

1. ML algorithms create order from chaotic process
   - Iterative, approximate, hill-climbing algorithms
     - Find a good-enough solution from random starting point
   - ML algorithms "self-heal"; mitigate errors in previous iterations to some degree
     - No need for fine-grained but costly fault tolerance
     - Can also exploit error-tolerance to increase speed!
2. ML is dynamic and perpetually rewiring
   - Ever-changing, “soft” dependencies in model state
     - Continuously analyze model structure, to find best parallel compute strategy at every moment
   - Not all parts of ML model equally important
     - Prioritize most important areas for speedup

Iterative Algorithms

- Iterative-convergent (IC) algorithms start from a non-meaningful model state, and climb towards a good, possibly non-unique, solution
- IC algos are approximation algorithms
  - Never get "best" solution; settle for a "good-enough" one
  - ML practice is to stop when additional computation produces diminishing returns
- Even the IC algo trajectory can be approximated --- in other words, execution is error-tolerant

Iterative Algorithm

$\Delta = \Delta(A^{(t-1)}, D)$

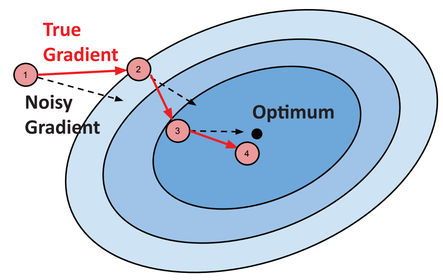
$A^{(t)} = F(A^{(t-1)}, \Delta)$

$F()$ Aggregate + Transform

Δ Intermediate Updates

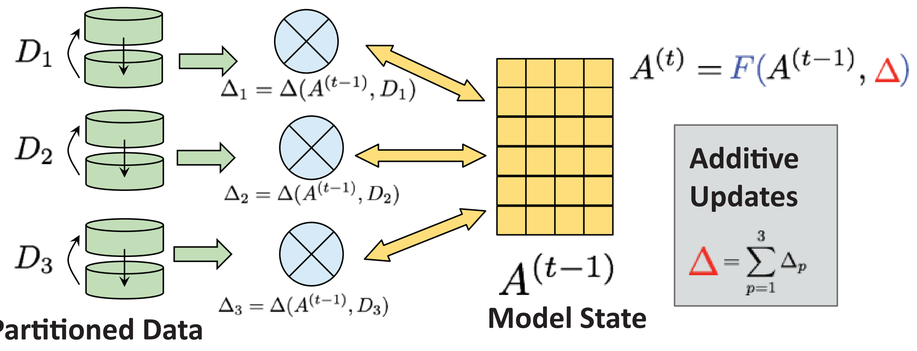
D = Data

$A^{(t-1)}$ = Model State at iteration (t-1)



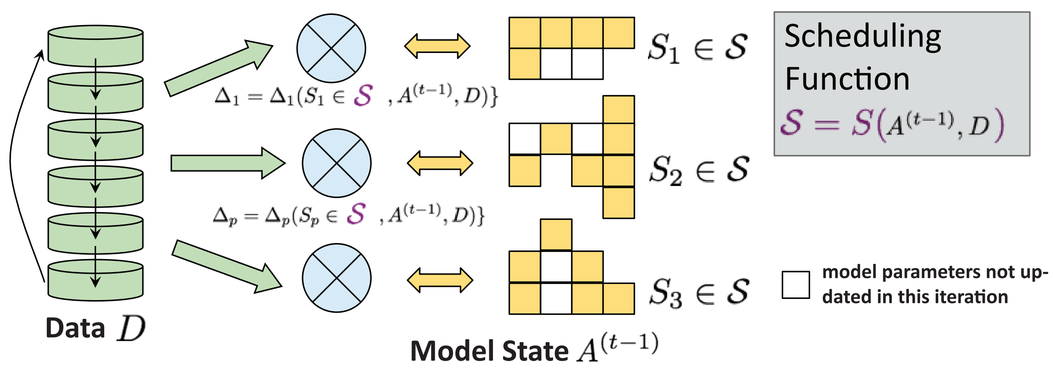
Data Parallel Controller

- A high-throughput distributed key-value store that schedules communications (reads/writes) for improved ML algorithm effectiveness under limited bandwidth
- Novel synchronization protocol supporting bounded staleness
- Exploit error tolerance in iterative-convergent ML algorithms; prioritize communications using both staleness (age) and impact on the ML algorithm, with sound guarantees

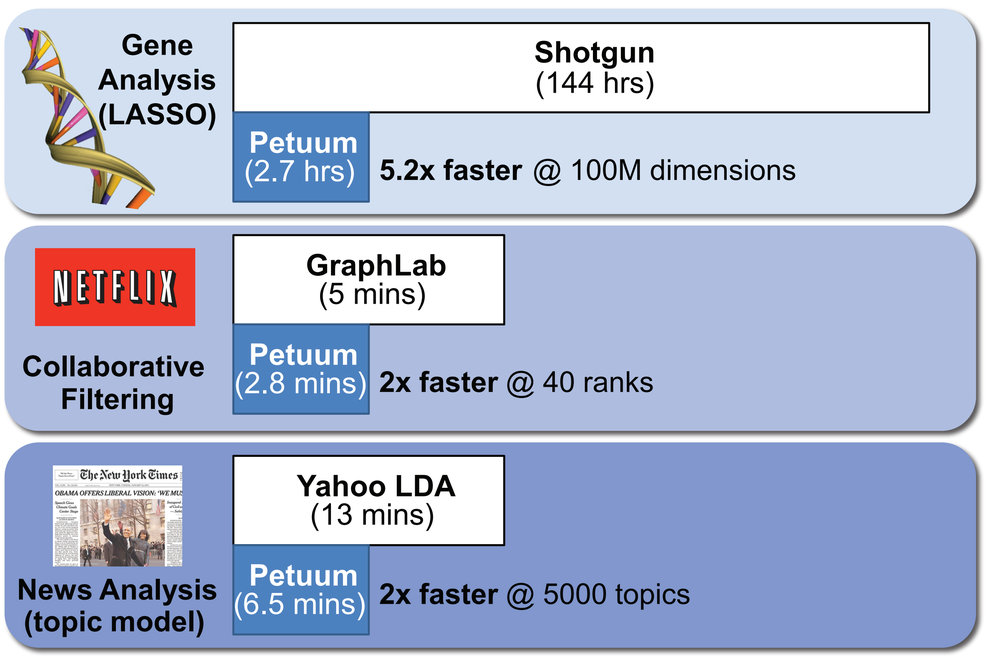


Model Parallel Controller

- Continuously analyzes the "soft", statistical dependencies between parts of an ML model.
  - Avoids breaking such dependencies during parallelization, which can cause algorithm divergence/failure
- Prioritize variables to maximize progress
  - E.g., Prioritize variables that cause large reduction in objective function in the last iteration



Performance



Open Source

- http://petuum.org/
- Available Apps:
  - Latent Dirichlet Allocation (Topic Modeling)
  - Lasso
  - Deep Neural Network
  - Matrix Factorization
- More apps to be added quarterly

Acknowledgement

Petuum is a SailingLab project inspired by collaboration with Cui Henggang, Phillip Gibbons, Greg Ganger, and Garth Gibson.

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley · University of Washington