Managed Communication for Fast, Large-Scale, Iterative Analytics

Jinliang Wei, Wei Dai, Henggang Cui, Qirong Ho, Greg Ganger, Phil Gibbons*, Garth Gibson, Eric Xing (CMU, *Intel Labs)

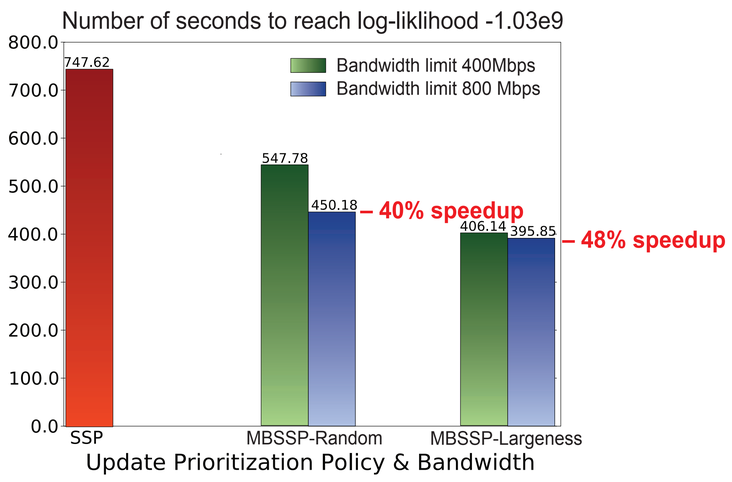
Background & Motivation

- Inter-machine communication is the bottleneck when scaling out
  - Parameter server & client stubs manage communication
- Application tolerance of bounded staleness allows delayed communication (Stale Synchronous Parallel)
  - Batching & coalescing reduce total communication
  - Communication becomes more bursty and efficient
- But reduced delay (freshness) improves convergence progress per update

Evaluation

- Hardware: 8 x 64-core machines with 1GE
- Application: Topic Modeling
  - Specifically, LDA with Gibbs sampling [1]
  - Using NYTimes dataset, deriving 1000 topics
  - Parameters and data fit in memory
  - Parameters (a_i) organized in rows, app requires atomic batch (row) updates
  - Importance of parameter update (u_i) becomes importance of row update
    - E.g. per row, max($|u_i|$) or sum($|u_i/a_i|$) (we use the latter)
  - Compare no early update, random row early update and "largest" row early update

Time to specific convergence threshold

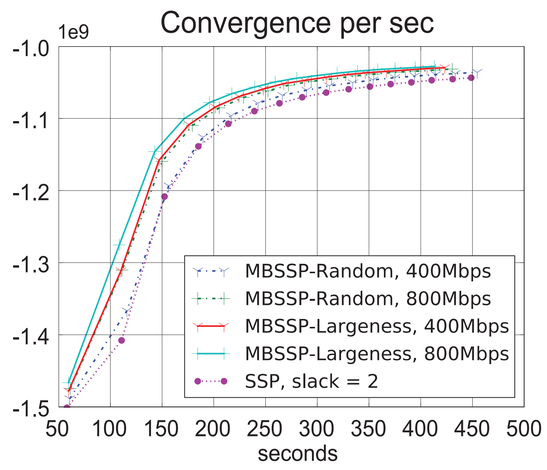


- 800 Mbps max bandwidth provides 40% speedup with random prioritization
- While prioritizing by row largeness achieves better speedup with 400 Mbps max

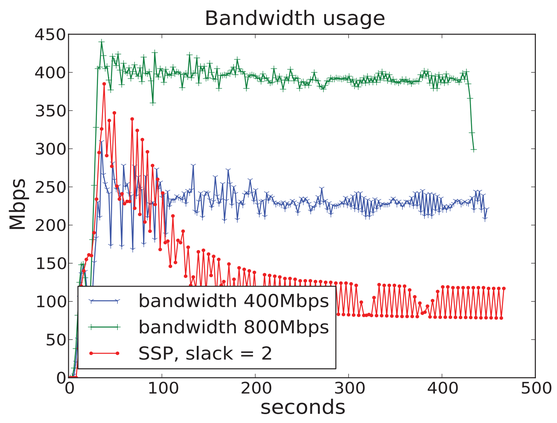
Managed Bandwidth Stale Synchronous Parallel (MBSSP)

- Communicate most important updates early and with bounded inter-machine bandwidth
  - Efficiency from potentially large delay, and convergence benefits of freshness where it counts
- Application-specific policies for prioritizing updates to communicate early
  - Eg., absolute magnitude of parameter delta
  - Eg., relative magnitude of delta over parameter

Absolute Convergence Rate Improves with Early Update



Bandwidth Usage



Conclusion & Future Work

Our vision:

- Updates bandwidth should be carefully managed
- Prioritization of "important" updates is app specific

Future work:

- Application-specific rules for suppressing updates
- Further delaying, dropping, lossy compression…
- Semantics for application-specific consistency requirements

[1] T. L. Griffiths and M. Steyvers. Finding scientific topics. Proc Natl Acad Sci U S A, 101(Suppl1):5228–5235, April 2004.

Intel Science & Technology Center for Cloud Computing

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