Managed Communication for Fast Data-Parallel Iterative Analytics

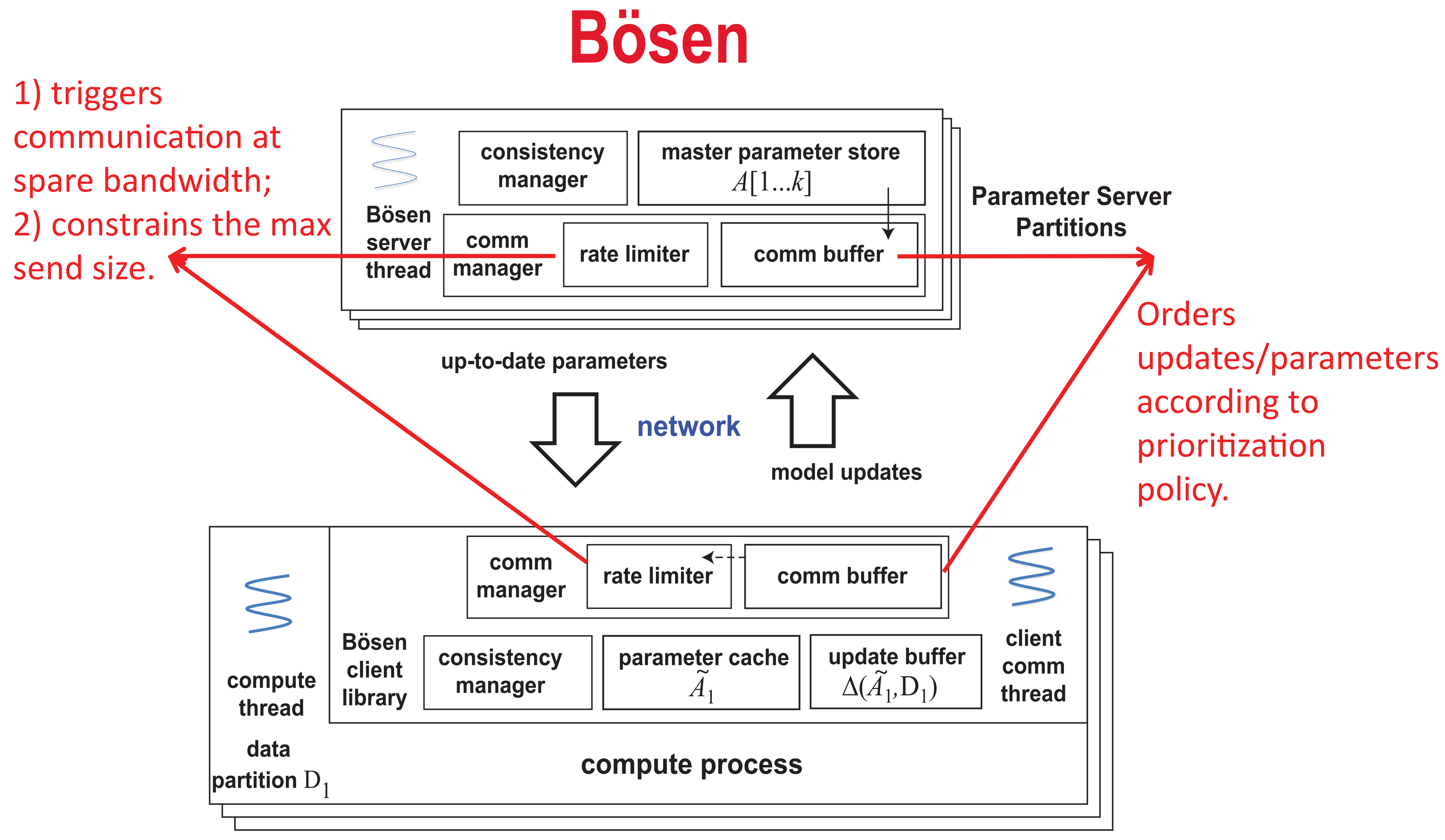
Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho*, Henggang Cui, Greg Ganger, Phil Gibbons†, Garth Gibson, Eric Xing
Carnegie Mellon University, *Institute for Infocomm Research, A*Star, †Intel Labs

Data-Parallel, Iterative-Convergent ML

- Data-parallel ML: distributed workers iteratively refine model parameters until convergence
  - › Partition the data among workers
  - › Workers share global model parameters
- Fast convergence comes from
  - › High # data samples per second
  - › High per-data-sample convergence rate
- Weak consistency (e.g bounded staleness):
  - › High # samples per second ✓
  - › Worst-case convergence guarantee ✓
  - › Lowered per-data-sample convergence rate ✗

Managed Communication for Parameter Server

Bösen



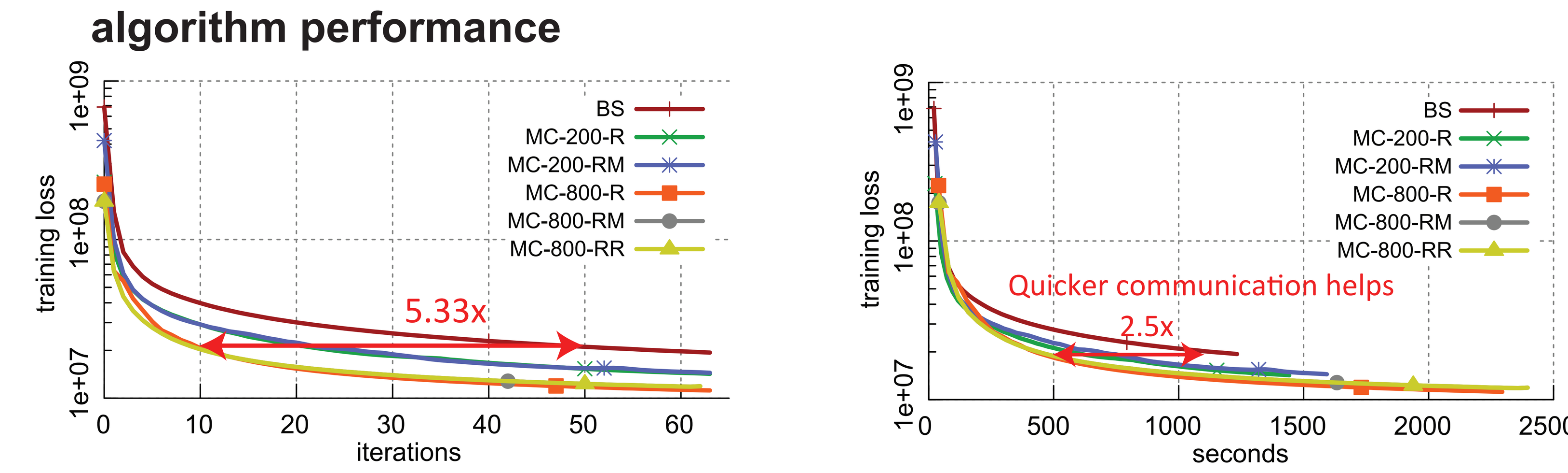
- The client library exposes a key-value abstraction, with communication management under the hood
  - › Get/GetRow for read; Inc/IncRow for incremental update; Clock for signaling end of an iteration
- When idle, the communication/server threads query the rate limiter for permission and max size to send
- The rate limiter employs a leaky bucket model for determining available bandwidth
- On sending, updates and dirty parameters are ordered according to prioritization policy, of which the top K are sent
- Exemplar prioritization policies:
  - › Random: a random subset of the entries
  - › RelativeMagnitude: magnitude of the delta change relative to the parameter value, $|\Delta/x|$

Improving Convergence per Data Sample

- Opportunities & Challenges
  - › Making updates visible sooner may improve convergence per data sample, but the network bandwidth is limited
  - › Model updates are of different significance
- Use all spare bandwidth and use it wisely!
  - › Bandwidth-driven communication and rate limiting
  - › Prioritization, i.e. bandwidth scheduling

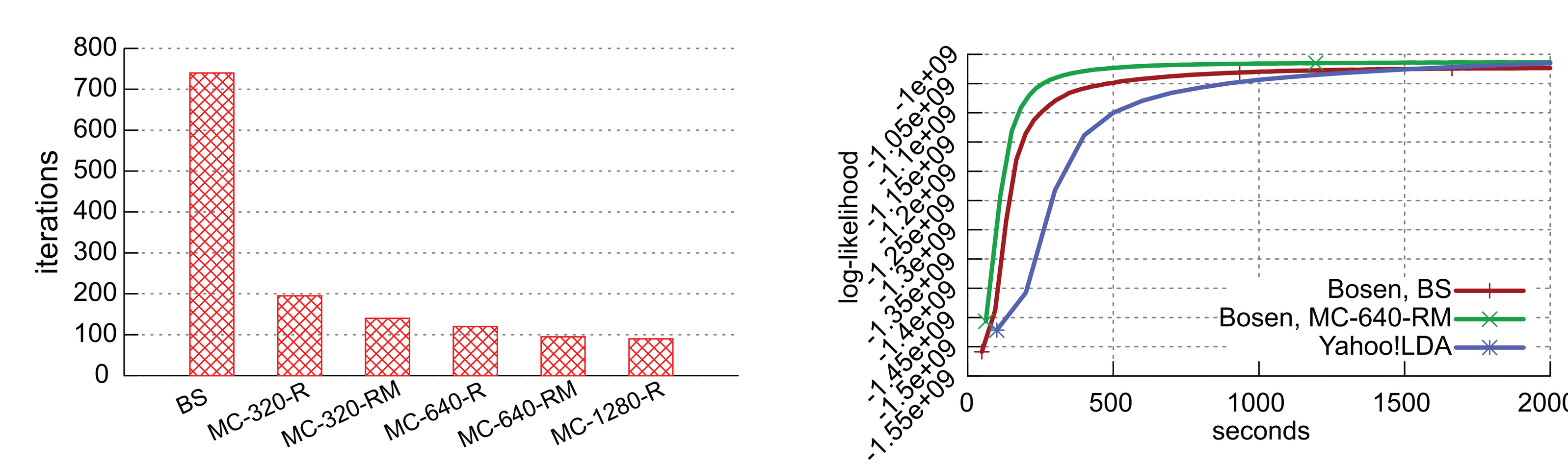
Evaluation

- Bösen's modes of execution:
  - BS-X: Bounded Staleness w/ X clock ticks/data pass (def=1)
  - MC-X-Y: BS + Managed Communication with bandwidth budget X Mbps and prioritization policy Y; R – Randomized, RM – Relative Magnitude, RR – Round Robin
- Automatically takes advantage of spare bandwidth to improve algorithm performance



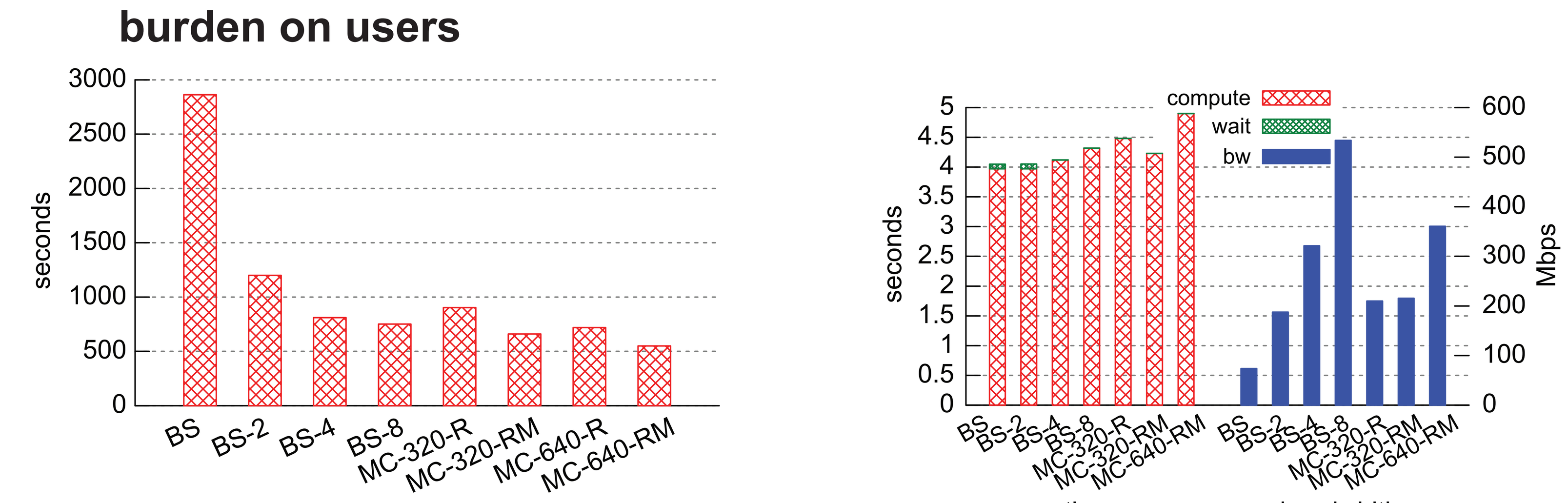
MF, 8x16 cores, 1GbE, Netflix data, rank=400.

- Allocating bandwidth based on message importance makes a difference
- Better convergence rate than a popular specialized LDA implementation



LDA, NYTimes, # topics = 1000, 16x16 cores, 1GbE

- Manual clock tick size tuning may also improve algorithm performance but fails to prioritize important messages and it imposes additional burden on users



LDA, NYTimes, # topics = 1000, 16x16 cores, 1GbE

Conclusion

- When used carefully, spare network bandwidth can be taken advantage of to improve ML convergence rate (2-3X)
- Scheduling of bandwidth (prioritization) improves communication efficiency and thus further improves convergence rate

Carnegie Mellon University

Related Work

[Power'10] R. Power and J. Li. Piccolo: Building fast, distributed programs with partitioned tables. OSDI'10.

[Ahmed'12] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy and A. J. Smola. Scalable inference in latent variable models. WSDM'12.

[Ho'13] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger and E. P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. NIPS'13.

Parallel Data Laboratory