Incoop: MapReduce for Incremental Computations

Bhatotia, P., Wieder, A., Rodrigues, R., Acar, U.A., and Pasquin, R. (2011).

Reviewed by Neil Satra

Why?

You are calculating PageRank at Google. Crawling petabytes of web pages.

1% of web pages have changed every time you crawl.



Iterative

Batch

Hard to scale efficiently

Need to redo entire computation for updated data



Iterative

Batch

Hard to scale efficiently

Need to redo entire computation for updated data

Incremental Batch Data Processing



Caching:

Option A: Give programmers the primitives Option B: Do it transparently

How?

	Not transparent	Transparent
Dryad and other tools	Yahoo! CBP	DryadIncl, Nectar
MapReduce	Google Percolator	Incoop

How?

3 optimizations:

- Partitioning of file system
- Fine-grained Reduce phase
- Memoization-aware scheduling

How?



Figure 1: Basic design of Incoop

Strengths

- Results: 10x to 1000x speedup, with a negligible processing overhead
- Evaluation: Used unmodified code for 5 realistic applications and

showed improvements both quantitatively and with mathematical proofs

- Optimizations show attention paid beyond surface-level

Weaknesses

- Evaluation: No quantitative comparison with non-transparent systems (Google Percolator)
- Insufficient discussion of the memoization server, which could be a bottleneck or central point of failure. No attempt to decentralize that component.
- Storage is linear in terms of input
- Assumptions about the application
- Garbage Collection of old cache entries
- Evaluation: Replaced part of data with equal sized chunks, rather than appending new data

Summary

Modified version of Hadoop (MapReduce)

• Efficient processing of large scale data, with incremental updates

• Works with existing code, transparently

 Memoizes computations, and tunes the operation of MapReduce to take maximum advantage of memoization

 Strong contributions, decently evaluated, number of potential concerns have been addressed



Bhatotia, P., Wieder, A., Rodrigues, R., Acar, U.A., and Pasquin, R. (2011a). Incoop: MapReduce for incremental computations. In Proceedings of the 2nd ACM Symposium on Cloud Computing, (ACM), p. 7.

Bhatotia, P., Wieder, A., Akkuş, \.Istemi Ekin, Rodrigues, R., and Acar, U.A. (2011b). Large-scale Incremental Data Processing with Change Propagation. In Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing, (Berkeley, CA, USA: USENIX Association), pp. 18–18.

Gunda, P.K., Ravindranath, L., Thekkath, R.A., Yu, Y., and Zhuang, L. (2010). Nectar: automatic management of data and computation in datacenters. In In OSDI '10,.

Logothetis, D., Olston, C., Reed, B., Webb, K.C., and Yocum, K. (2010). Stateful Bulk Processing for Incremental Analytics. In Proceedings of the 1st ACM Symposium on Cloud Computing, (New York, NY, USA: ACM), pp. 51–62.

Peng, D., and Dabek, F. (2010). Large-scale Incremental Processing Using Distributed Transactions and Notifications. In OSDI, pp. 1–15.

Popa, L., Budiu, M., Yu, Y., and Isard, M. DryadInc: Reusing work in large-scale computations.