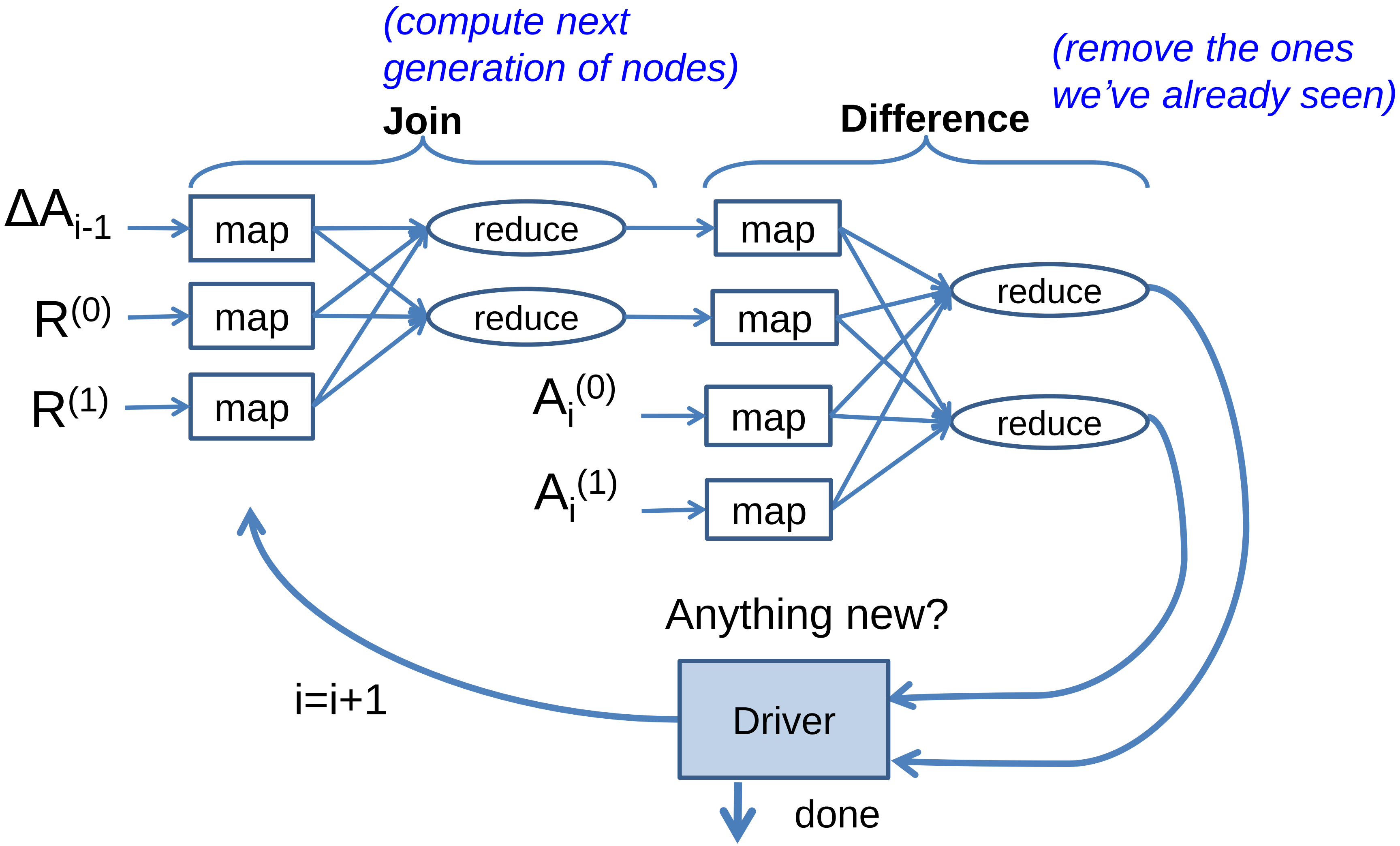
In MapReduce



(compute next generation of nodes)

Join

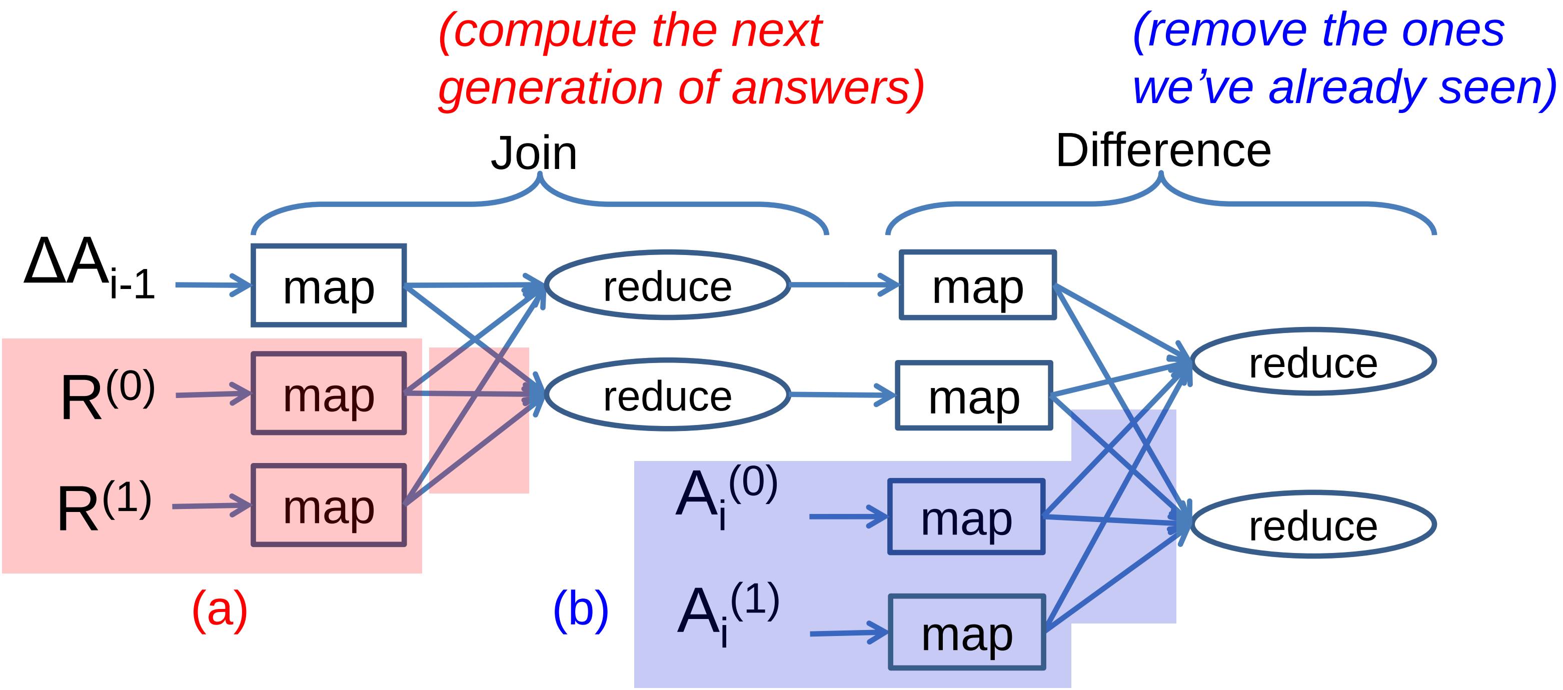
(remove the ones we've already seen)

Difference

Anything new?

Bu et al. VLDB10, VLDBJ12
Shaw et al. Datalog12

Evaluating Recursive Queries at Scale



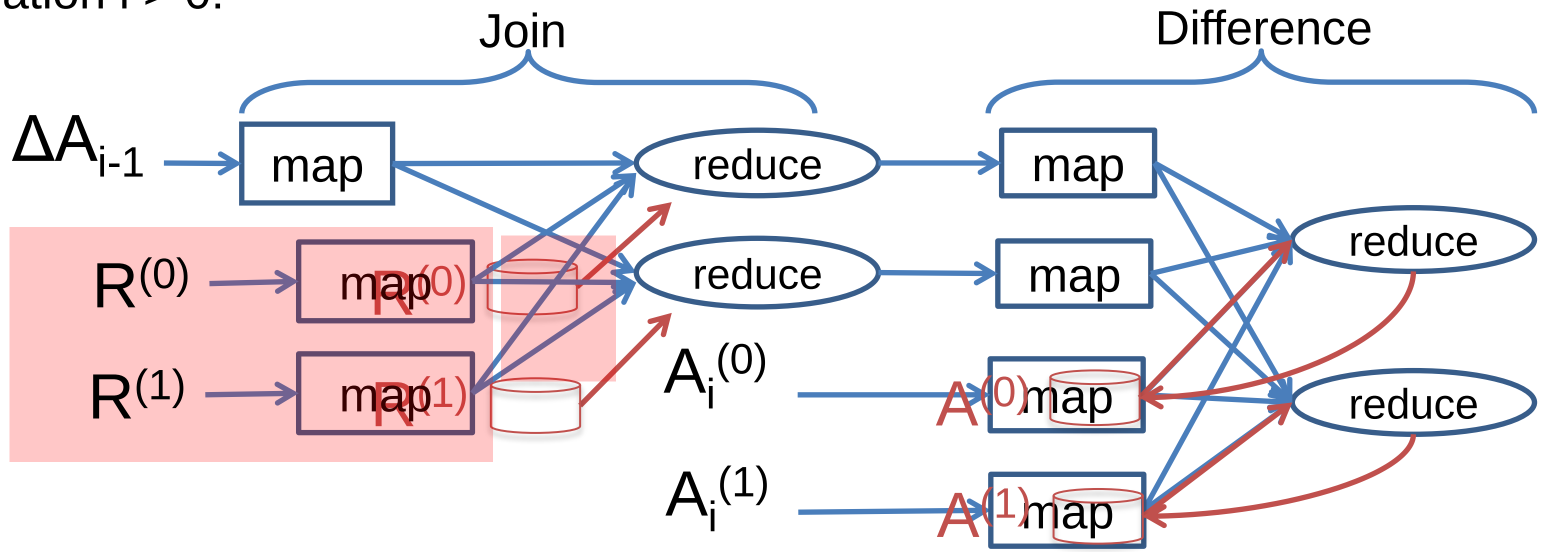
(a) R is loop invariant, but gets loaded and shuffled on each iteration

(b) A_i grows slowly and monotonically, but is loaded and shuffled on each iteration.

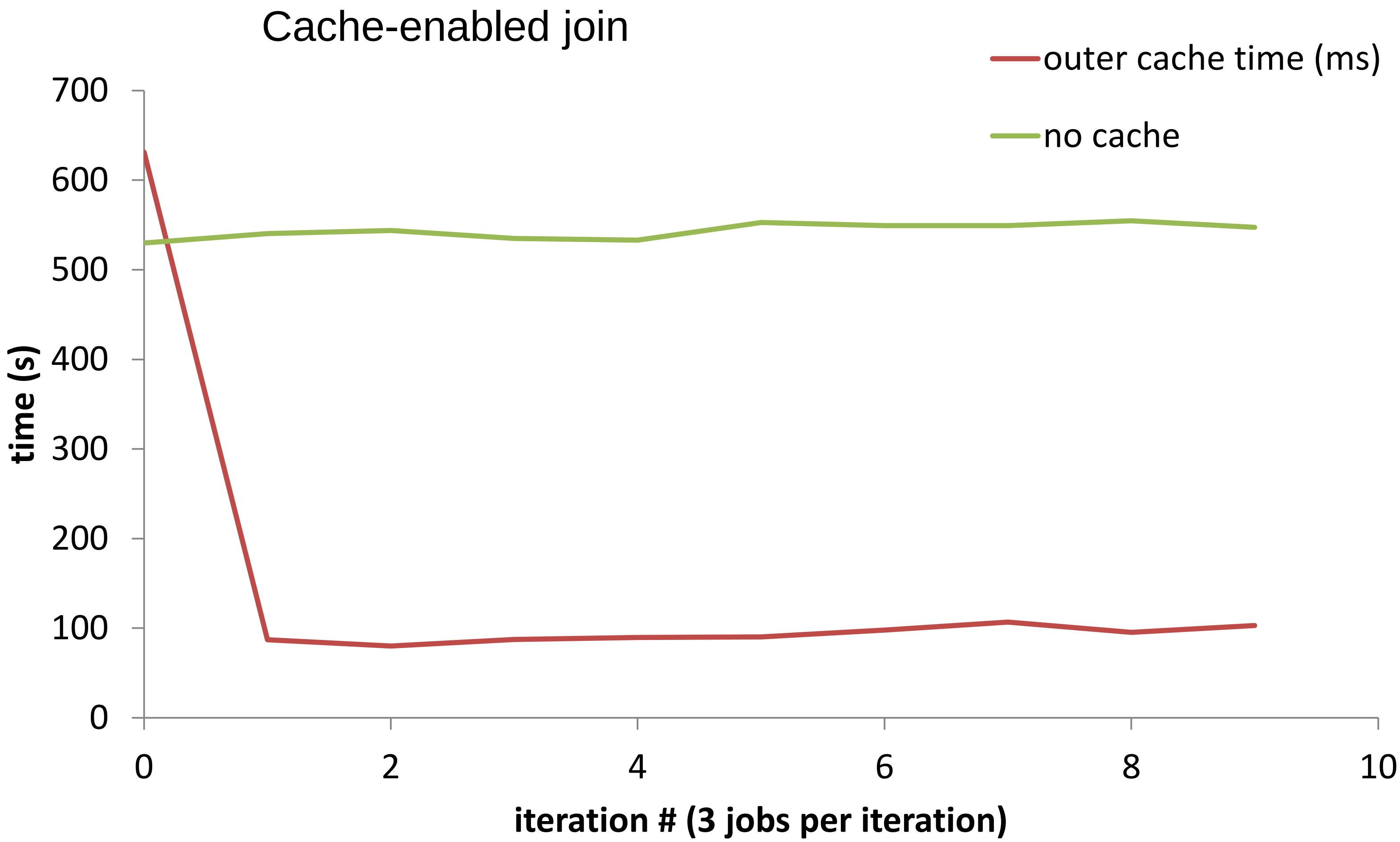
Idea: Cache Loop-Invariant Data

Iteration i = 0: Load a distributed cache

Iteration i > 0:

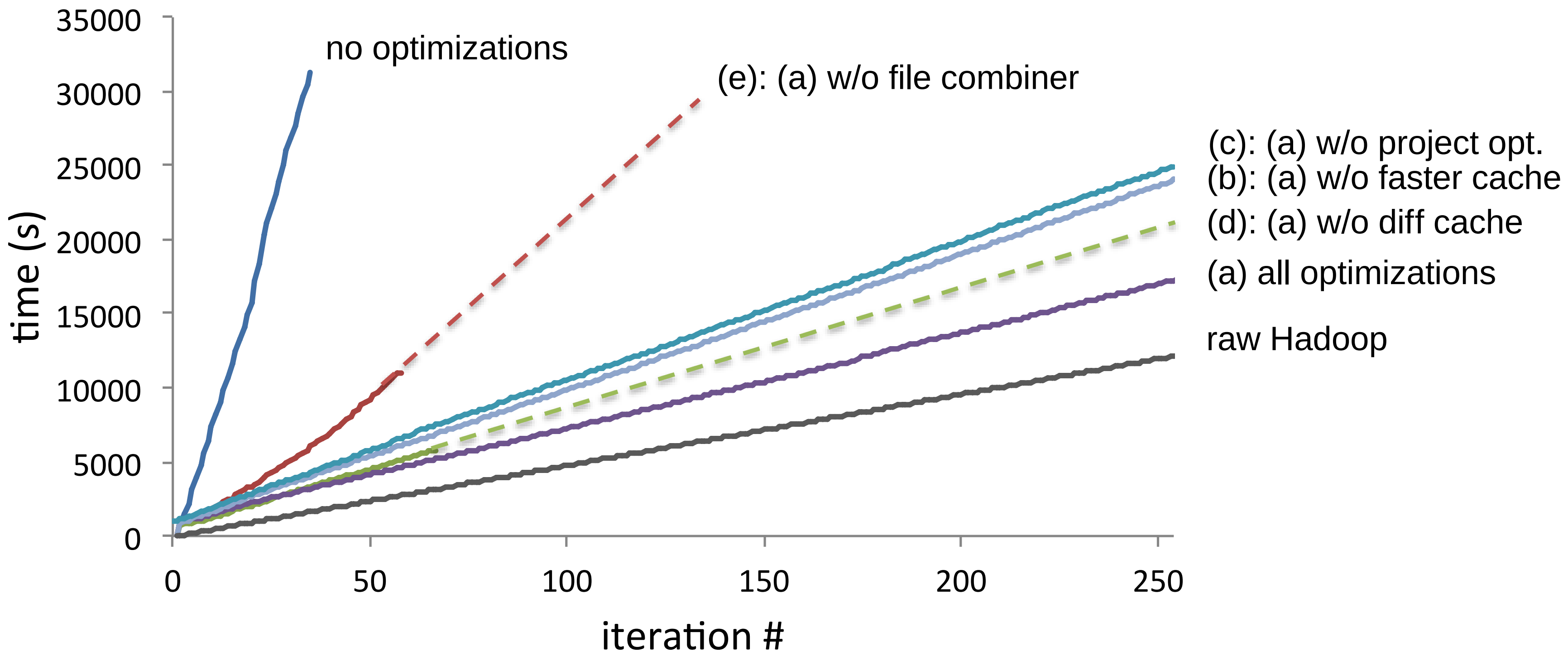


BTC2010, 680GB



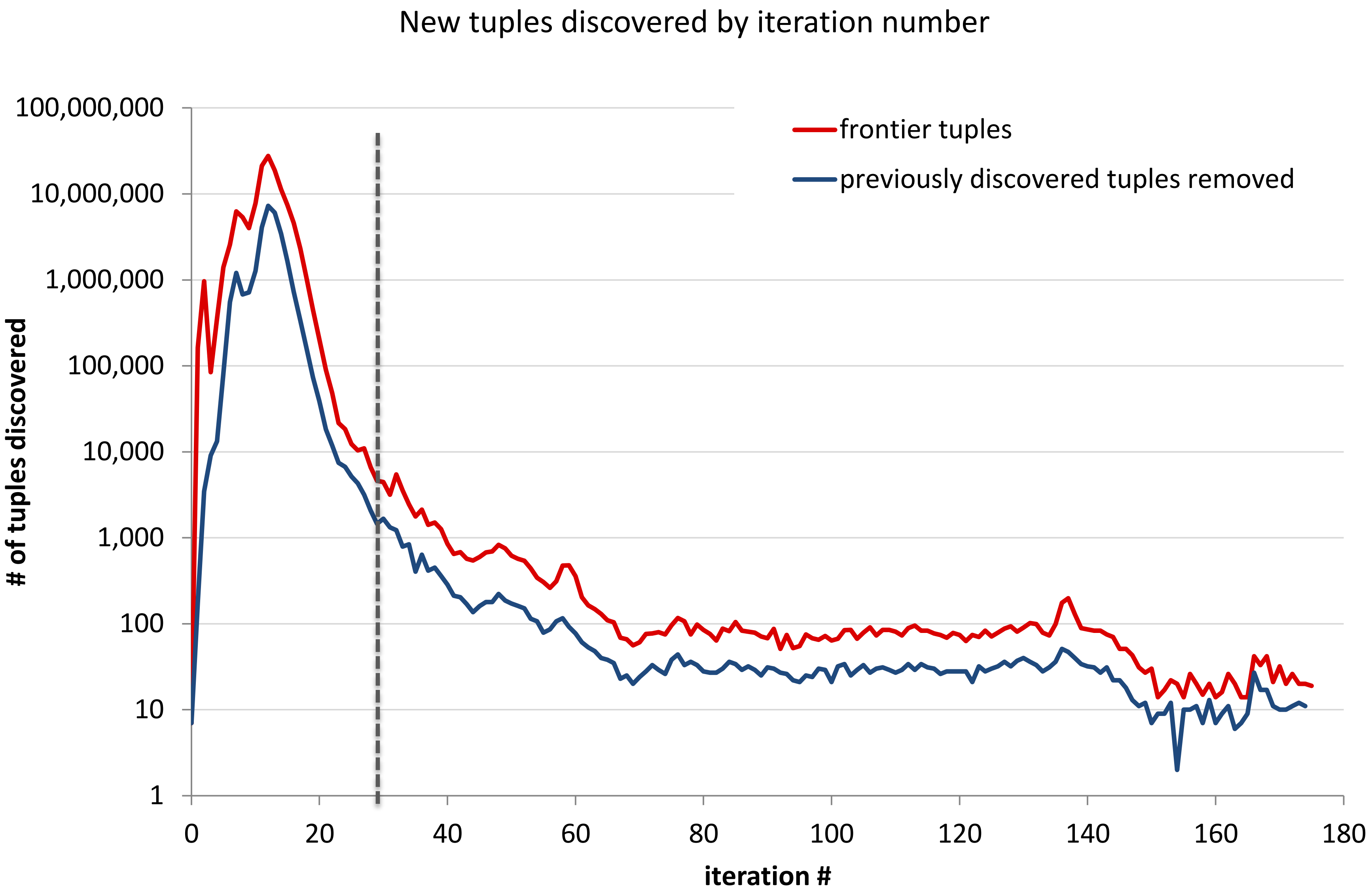
Effect of various optimizations for a recursive graph query on BTC 2010

(query: transitive reachability from 7 nodes)



Takeaways:

- 10x improvement over no optimizations.
- All optimizations are useful
- We’re approaching the raw overhead of Hadoop (bottom gray line)


New tuples discovered by iteration number
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
20
40
60
80
100
120
140
160
180
# of tuples discovered
iteration #
frontier tuples
previously discovered tuples removed