Incoop: MapReduce for Incremental Computations

Mircea Dima
Vrije Universiteit Amsterdam
Monday, February 20, 2012
What is Incoop?

• Incoop is a generic MapReduce framework for incremental computations over Hadoop

• It detects modifications of the input and automatically updates the output in an efficient way
Basic Design of Incoop
Goals

- Transparency
- Efficiency
Goals

• **Transparency**
  – Input changes – store both input and output of consecutive runs in different HDFS files in order to determine the input differences
  – Programmer annotations – we use the Memoization Server in order to make sure we don’t re-compute the tasks that didn’t change from the previous run
Goals

• **Efficiency**
  – Stable input partitions – use partition techniques that enable maximal overlap between the set of data chunks of similar inputs
  – Granularity control – need to reduce the task size and eliminate potentially long dependencies between parts of Reduce tasks by using a Contraction phase
  – Scheduling – extend Hadoop scheduling algorithm in order to schedule tasks on the machine that stores the memoized results
Incremental HDFS

- Inc-HDFS extends the Hadoop Distributed File System to enable stable partitioning of the input using content-based chunking.
- Instead of using fixed-size chunks as provided by HDFS we would set boundaries for each chunk and resplit only when the upper bound is reached.
Incremental MapReduce

• Incremental Map – after each new task we store the result and for each new run where that task is needed, we fetch the result instead of re-computing it again

• Incremental Reduce – First we perform memoization at the entire Reduce tasks level, then we use the Contraction Phase for memoization of the Reduce sub-tasks
Incremental Map task
**Contraction Phase**

- The Contraction Phase uses Hadoop Combiners to break up the Reduce chunks into smaller sub-tasks.
- This is done in a recursive way until we reach the last level of the Combiners where we start compute the tasks and combine them back to the top.
- This way we can memoize sub-tasks and reuse them when needed.
Memoization-Aware Scheduler

• The Hadoop Scheduler is modified such that it will schedule tasks on the nodes where some or all the memoized results are stored.
Time Speedups for different applications
Conclusions

• Incoop is based on several novel techniques to maximize the reuse of results from previous computations
• It incorporates content-based chunking to detect incremental changes in the input file and to partition the data so that to maximize reuse
• It adds Contraction phase to control the granularity of the tasks in Reduce phase
• It adds an improved scheduler