Winning a 60 Second Dash with a Yellow Elephant

Arun C Murthy
Owen O'Malley
{oom,acm}@yahoo-inc.com

Existential Angst: Who Am I?

- **Yahoo! Engineer on Hadoop Map/Reduce**
  - Design, review, and implement features in Hadoop
  - Working on Hadoop full time since April 2006

- **Hadoop Core Committer** and **Member of the Hadoop Project Management Committee`**

Jim Gray's Sort Benchmark

- **Started by Jim Gray at Microsoft in 1998**
- **Currently managed by 3 of the previous winners**
- **Sorting different numbers of 100 byte records**
  - 10 byte key
  - 90 byte value
- **Multiple variants:**
  - **Minute Sort**: sort must finish < 60.0 secs
  - **Terabyte Sort**: 10^12 bytes, won in 2008, deprecated
  - **Gray Sort**: ≥ 10^14 bytes and ≥ 1 hour

Rules of the Benchmark

- **Rules**
  - Must use official data set, defined by their program
  - The input must not start in the file cache
  - The input and output must not be compressed
  - The output must not overwrite the input
  - The 128 bit sum of the crc32's of each key/value pair must match between input and output
  - The output must be totally ordered.
  - Output must be synced to disk.
  - Sampling, starting and distributing the application count toward the run time.

Hadoop Implementation

- **Four Map/Reduce Programs:**
  - **TeraGen** – Generate the dataset. Includes the number of 100 byte records to generate.
  - **TeraSort** – Sort the input data. This is the benchmark.
  - **TeraSum** – Sum (128 bits) the crc32 of each key/value
  - **TeraValidate** – Check the sort order of the output
    - Each reduce's output file is totally sorted.
    - The last key in reduce N is less than the first key of reduce N+1
    - Also calculates the 128 bit sum of the crc32 of each key/value

Hammer Cluster Specifications

- **Hammer was brand new and now is in production**
  - 3879 nodes (in theory, but in practice 3400-3700)
  - 2 quad-core Xeons @ 2.5 Ghz / node
  - 4 SATA disks / node
  - 8 GB ram / node (upgraded to 16 GB)
  - 1 gb ethernet / node
  - 40 nodes / rack
  - 8 gb ethernet uplink / rack
  - Red Hat Enterprise Linux Server Release 5.1 (kernel 2.6.18)
  - Sun Java JDK (1.6.0_05-b13 and 1.6.0_13-b03) (32 and 64 bit)

Results

Bytes	Nodes	Maps	Reduces	Repl	Time
5*10^11	1,406	8,000	2,600	1	59 sec
10^12	1,460	8,000	2,700	1	62 sec
10^14	3,452	60,000	7,200	2	98 min
10^15	3,658	80,000	20,000	2	975 min

- **Small runs used a subset of nodes**
  - **Higher cross-section bandwidth (500 MBPS)**
  - **Lower overhead for TaskTracker reporting**
- **Large runs need replication 2 to survive failures.**
- **100 TB and 1PB sort rates are 1.03 TB/min.**

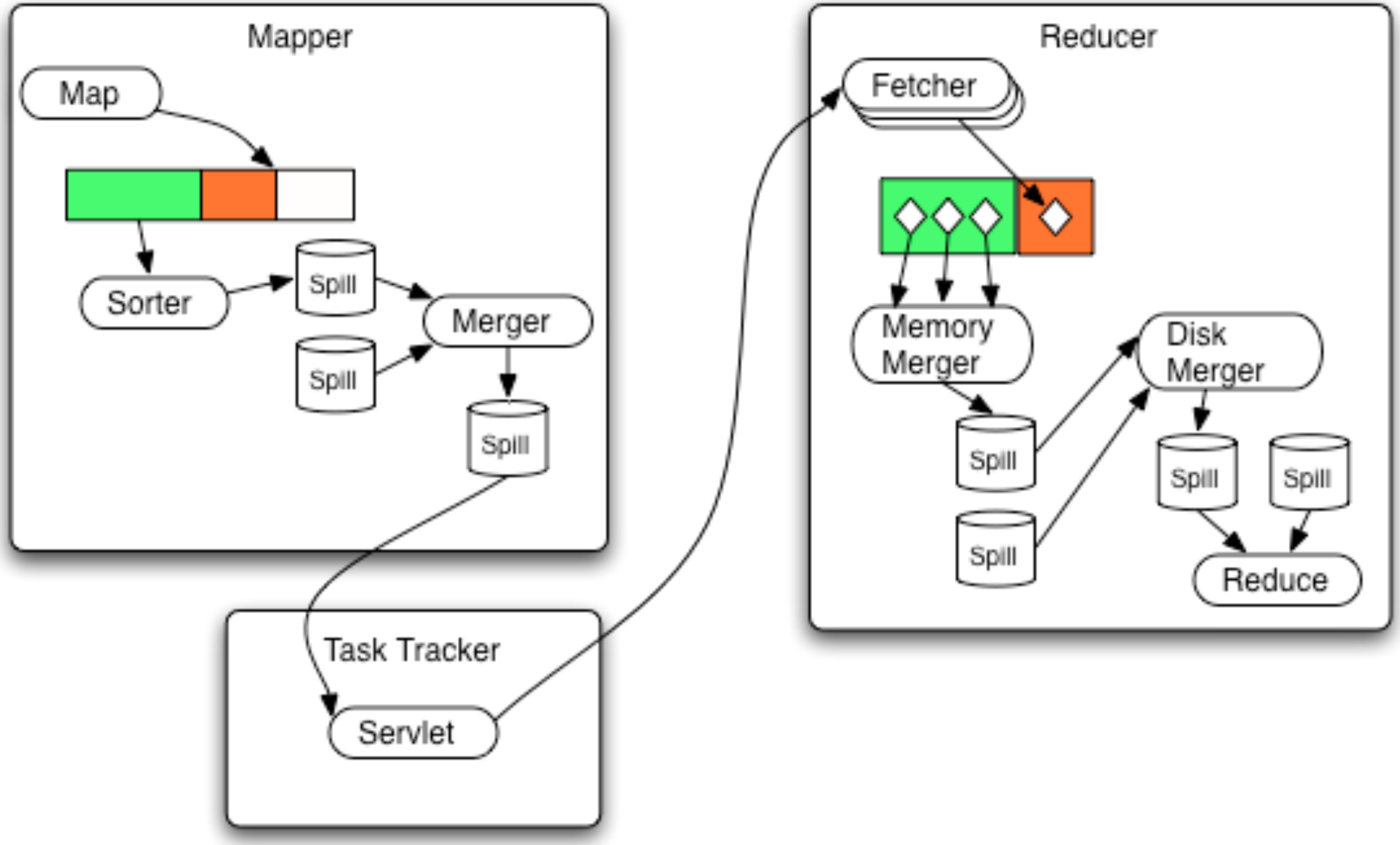
Throughput versus Latency

- Speed means different things
- Freight trains move a lot of cargo, but start slowly
- Sports cars move little cargo, but start very fast.
- Hadoop was designed to maximize throughput, not minimize latency.
- Minute sort was a challenge!

Changes to Hadoop

- **Re-implemented the shuffle**
  - Refactored code to be more maintainable. (Required so that we could work on it together without stepping on toes!)
  - Fetch multiple map outputs in the same request.
  - Allow configuration of timeouts on shuffle connections. We saw some connections hang until the timeout.
- **Set TCP_NODELAY and more frequent pings between Task and TaskTracker.**
- **Used LZO compression on the map outputs.**
- **Made the heap size of maps and reduces configurable separately.**

Shuffle Dataflow



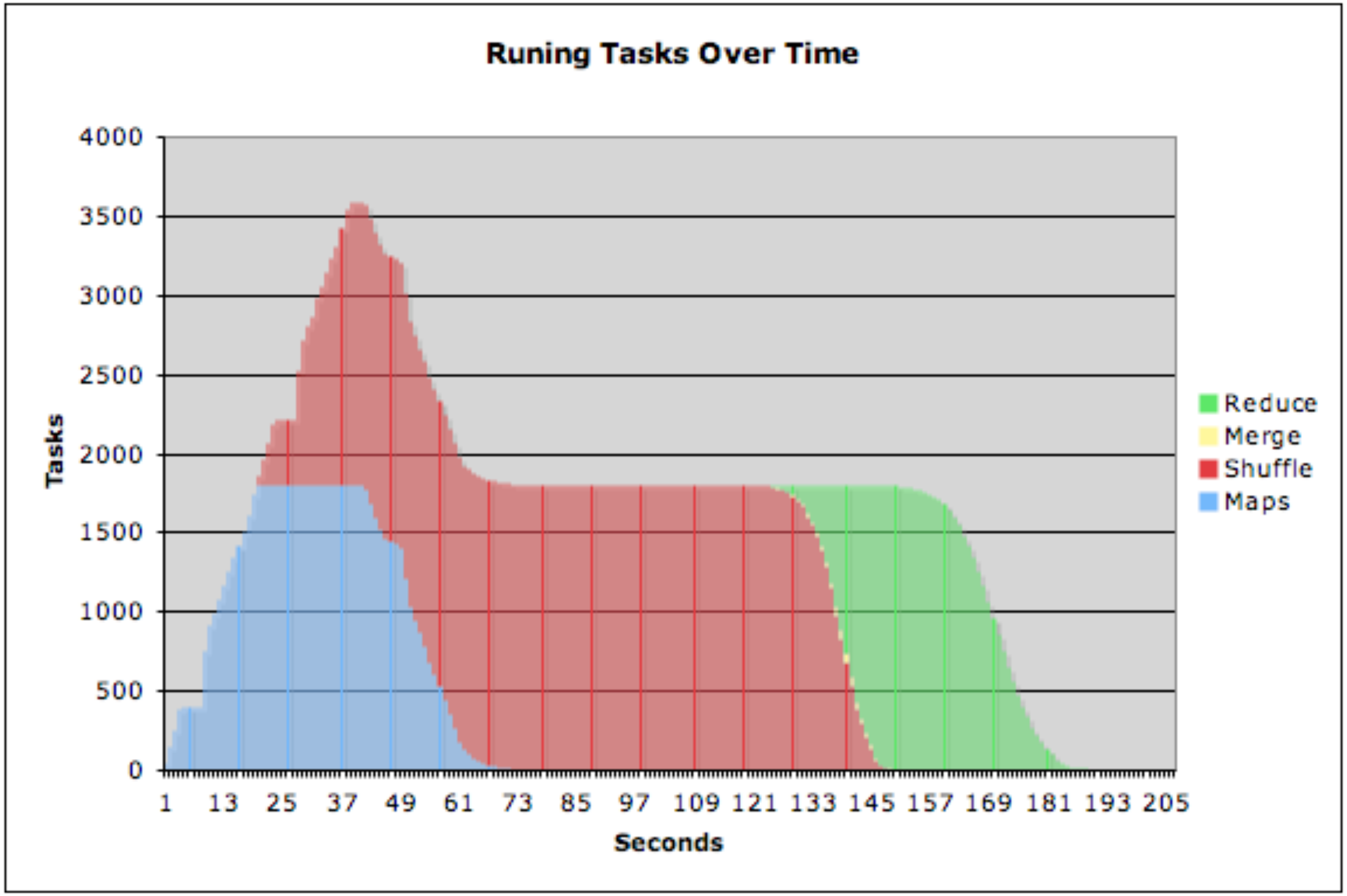
Changes to Hadoop (cont)

- **Found and worked around JVM bug that caused data corruption in shuffle. (Fixed in latest JVM!)**
  - Took most of a week to track down cause of dropped records
- **Made the heartbeat interval configurable for lower latency.**
- **Made the Job setup and cleanup tasks optional.**
- **Made the logging level for tasks configurable.**
- **Implemented memory to memory merge in shuffle.**
- **All of the changes have Jiras and will be rolled into Hadoop trunk.**

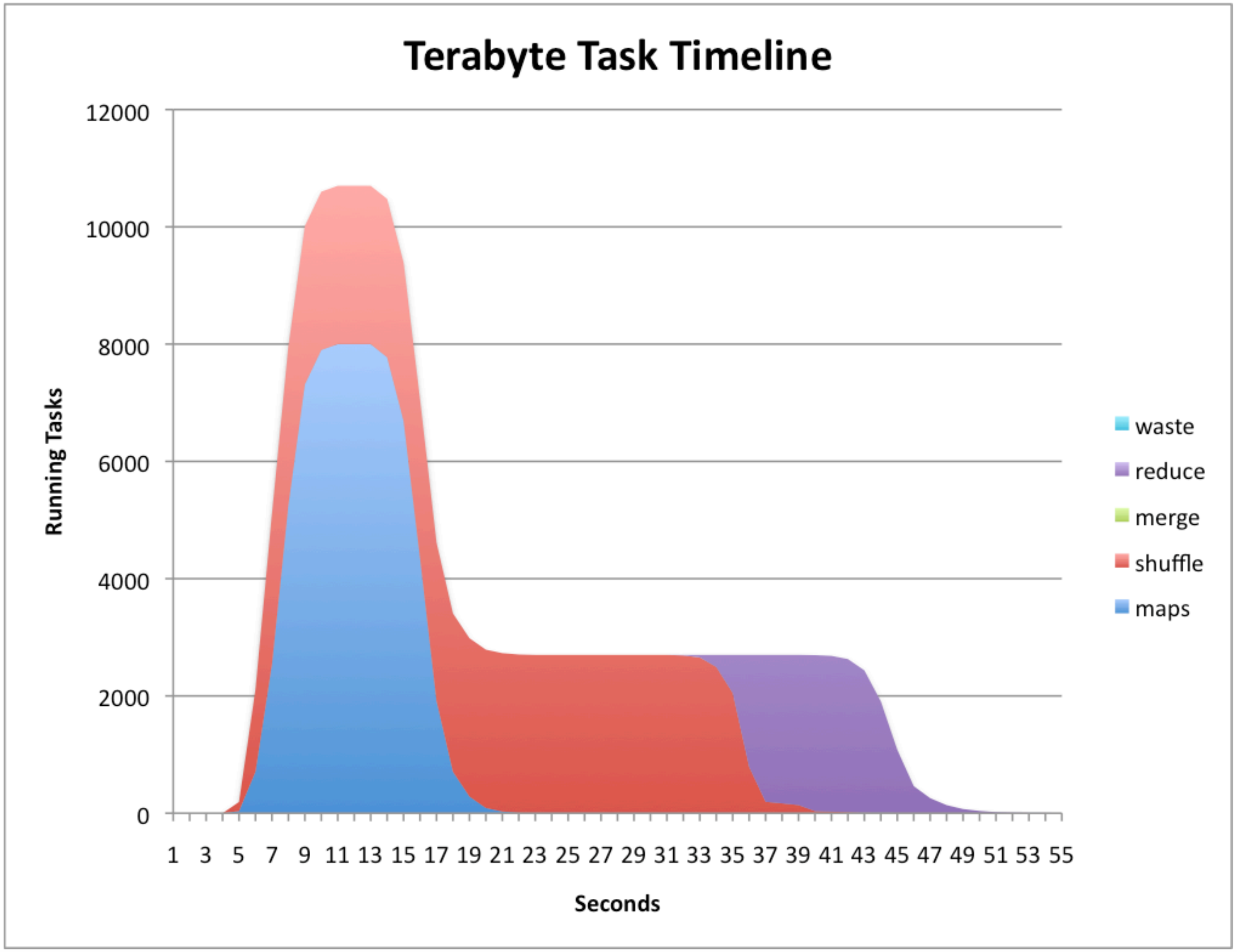
Changes to Benchmark Code

- **Updated tools to reflect new rules in 2009.**
  - Data files now binary instead of text
  - Random number generator 128 bit, so no overflow after 4 billion rows.
  - Added TeraSum to calculate checksum
- **Made the input sampling code multi-threaded.**
  - Each thread reads one range of the input
- **Made a global scheduler for the map Tasks.**
  - Assign each map to a node
  - For each node (starting with node with fewest maps)
    - Choose blocks with fewest nodes
    - Remove node from blocks not chosen

2008 Terabyte Sort Task Timeline



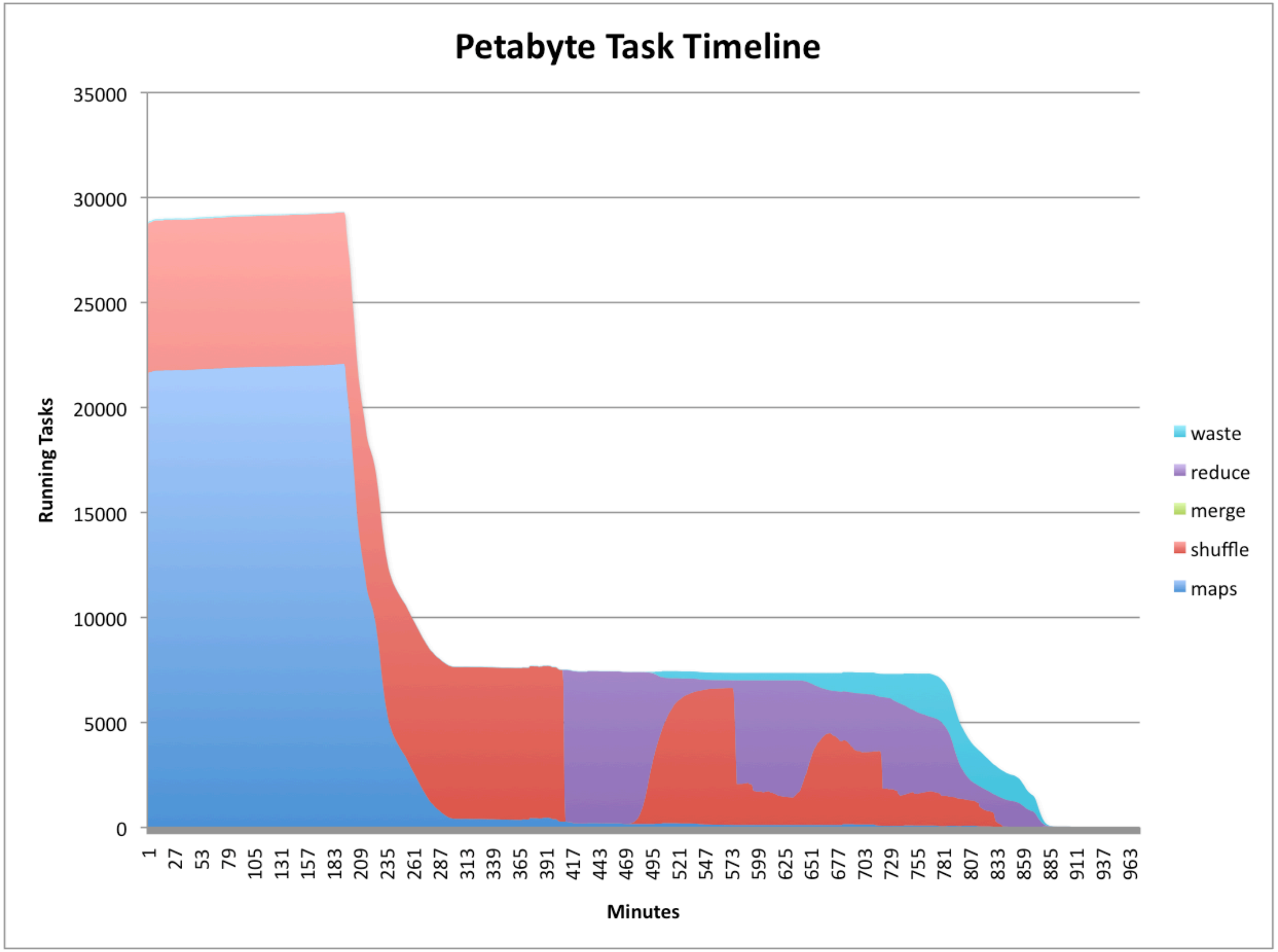
2009 Terabyte Sort Task Timeline



Speed Ups from 2008

- **Ran with 50% more nodes**
- **Ran with 2.5x cross-section bandwidth**
- **Faster task launching.**
  - 2008 timeline is from first task launch, 2009 from job submission
  - In 2008, reduces didn't finish launching until 40 sec
- **Compression of transient data**
  - LZO got 2x on the dataset
  - Last year's shuffle couldn't use compression on large in memory shuffle.
- **Other framework improvements**

Petabyte Sort Task Timeline



Notes on Petabyte Sort

- 80,000 maps and 20,000 reduces
- Each node ran 2 maps and 2 reduces at a time
- So 11 waves of maps and 3 waves of reduces
- Tail of maps was 100 minutes
- Tail of reduces was 80 minutes
  - Caused by one slow node!
- Used speculative execution, but it must do better.
- The “waste” tasks at the end are mostly speculative execution.

Future Improvements

- **Better Speculative Execution**
  - Launches duplicate tasks when the original is being slow.
  - Current heuristic helps, but is not good enough.
  - HADOOP-2141
- **Progress reporting isn't smooth enough**
  - Map progress tracks input consumption, doesn't include sort
  - Reduce progress miscounted when compression used.
- **Better handling of shuffle failures**
- **Better handling of task failures**
- **Automatic detection of bad and slow nodes**

Coverage

- **Yahoo Hadoop blog:**
  - http://developer.yahoo.net/blogs/hadoop/2009/05/hadoop_sorts_a_petabyte_in_162.html
- **Slashdot:**
  - http://tech.slashdot.org/story/09/05/16/1316242/Open-Source-Solution-Breaks-World-Sorting-Records?art_pos=1
- **Cnet:**
  - http://news.cnet.com/8301-13846_3-10242392-62.html
- **Sort benchmark:**
  - http://sortbenchmark.org/