Toward Efficient Provisioning and Performance Tuning for Hadoop

Jason Dai
jason.dai@intel.com
Cloud Computing Architect
Intel China Software Center

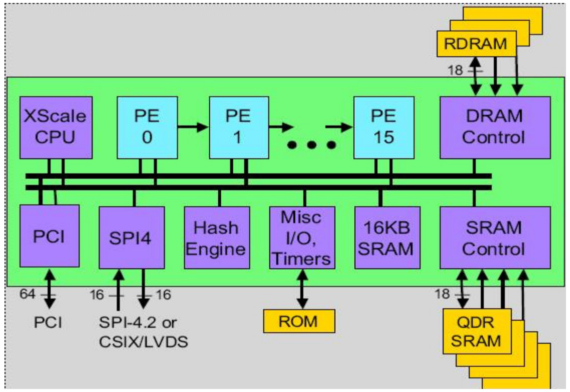
Apache Asia Roadshow 2010

Background & Bias

5 years in compiler development

- **Lead architect on Intel network processor compiler**
  - Intel network processors
    - 16 cores, 8 hardware threads per core @ Year 2002
    - Foreshadow the general trend to multi-core, multi-thread architectures
  - Focused on parallel processing, performance and scalability

Intel IXP2800



Currently Cloud Computing architect

- **Lead the work on massively distributed cloud platforms**
  - Cloud storage
  - Big Data analytics
  - Virtualized utility cloud
  - Online web service
  - Cloud datacenter building blocks

Agenda

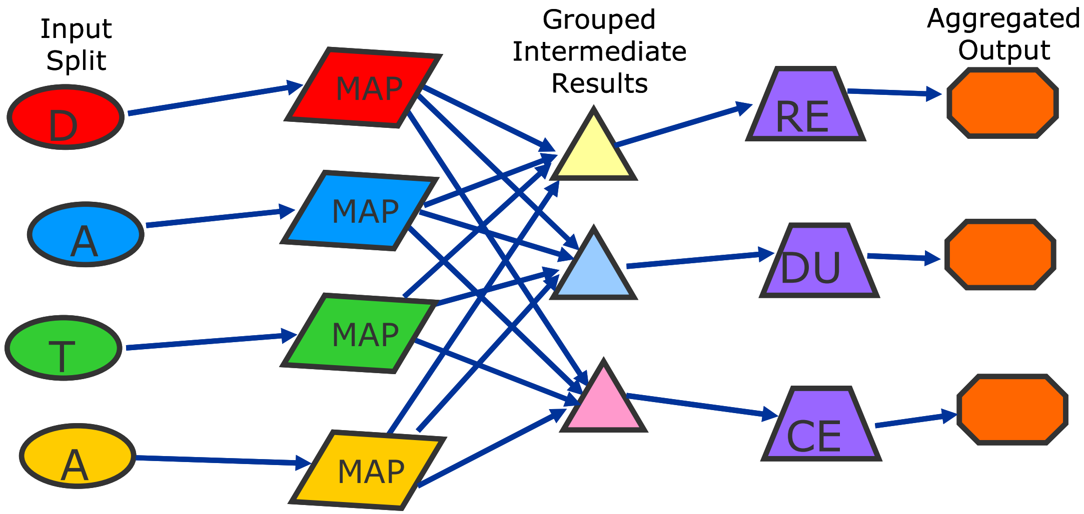
MapReduce/Hadoop overview
- Dataflow model of MapReduce/Hadoop
- Why MapReduce/Hadoop?

The challenges
- Efficient provisioning and performance tuning for Hadoop
- HiTune: dataflow-based Hadoop performance analyzer

Performance analysis
- HiBench: a realistic and comprehensive Hadoop benchmark suite
- Balanced architecture design for Hadoop clusters

Summary

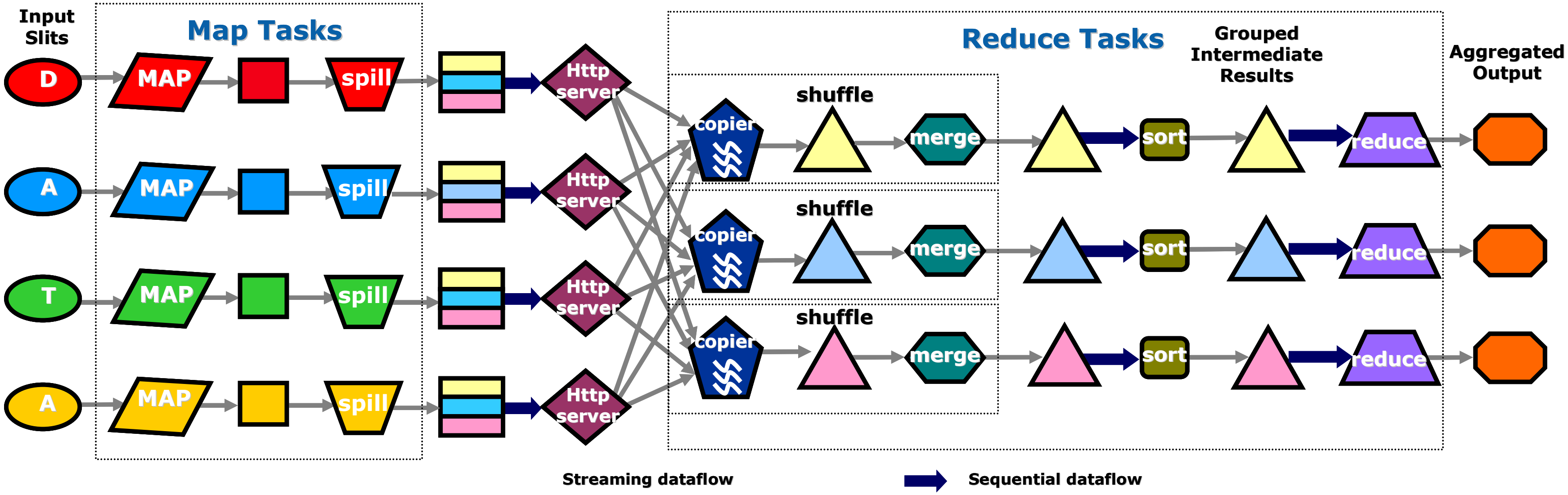
MapReduce



MapReduce (Google OSDI'04 paper)

- Predominant model for Big Data analytics
- Essentially a group-by-aggregation in parallel over a cluster of servers
  - The input can be trivially divided into multiple splits
  - In the first phase, a map function (i.e., how to perform the grouping) is applied to each split
  - In the second phase, a reduce function (i.e., how to perform the aggregation) is applied to each group

Hadoop



Hadoop (open source implementation of MapReduce)

- Used by Yahoo, Facebook, Twitter, LinkedIn, China Mobile, Alibaba, Baidu, …
- Programmer specifies several methods in a Hadoop program:
  - Required:
    - map(k, v) → <k', v'>*
    - reduce(k', <v'>*) → <k'', v''>*
      - All v' with same k' are reduced together, in order
  - Optionally:
    - combine(k', <v'>*) → <k'', v''>*
      - Map-side "pre-aggregation" & often the same as reduce
    - partition(k', total partitions) -> partition for k'
      - Different partitions can be reduced in parallel & often a simple hash of the key

Why MapReduce/Hadoop?

The problem

- **Developing parallel and distributed programs is hard**

The MapReduce/Hadoop approach

- **Input program modeled as a directed acyclic dataflow graph**
- **User supplies subroutines running on the graph vertices**
- **Framework dynamically maps the dataflow graph to the cluster**

The benefits

- **Allow the user to easily develop massively distributed applications**
  - **User required to write the program considering data parallelisms exposed by the dataflow**
  - **System can distribute the execution of subroutines by exploiting data dependencies encoded in the dataflow**

Agenda

MapReduce/Hadoop overview

- Dataflow model of MapReduce/Hadoop
- Why MapReduce/Hadoop?

The challenges

- Efficient provisioning and performance tuning for Hadoop
- HiTune: dataflow-based Hadoop performance analyzer

Hadoop Performance analysis

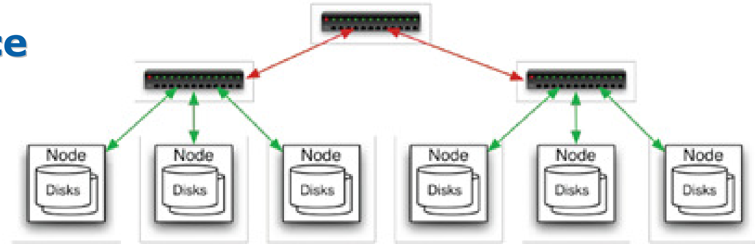
- HiBench: a realistic and comprehensive Hadoop benchmark suite
- Balanced architecture design for Hadoop clusters

Summary

Practical Issues

Provisioning the Hadoop cluster

- What are the desired hardware specs (or instance types on EC2) for the servers?
- How to connect those servers together?
- What's the bottleneck (e.g., CPU vs. memory vs. disk vs. network) in the cluster?
- …



Fine-tuning the Hadoop framework and application

- How to set the Hadoop configuration parameters appropriately?
- What's the hotspot (e.g., computing vs. disk I/O vs. network transfer vs. synchronizations) in the application?
- How to address the performance anomaly in the system?
- …

Fundamental Challenges

MapReduce/Hadoop greatly helps Big Data analytics

- **Abstract away low level details by a simple two-phase primitive**
  - Data partitioning, task scheduling, resource allocation, node communications, fault tolerance, …
- **Make it easy to develop and run massively scalable applications**

This abstraction often makes the system appear as a "black box"

- **Very difficult, if not impossible, for the user to understand the Hadoop runtime behaviors**
- **Efficient provisioning and fine-tuning of Hadoop systems remain a big challenge**
  - Request/allocate the optimal (physical or virtual) resources
  - Determine the best cluster architecture
  - Optimize the application and system for better resource utilizations

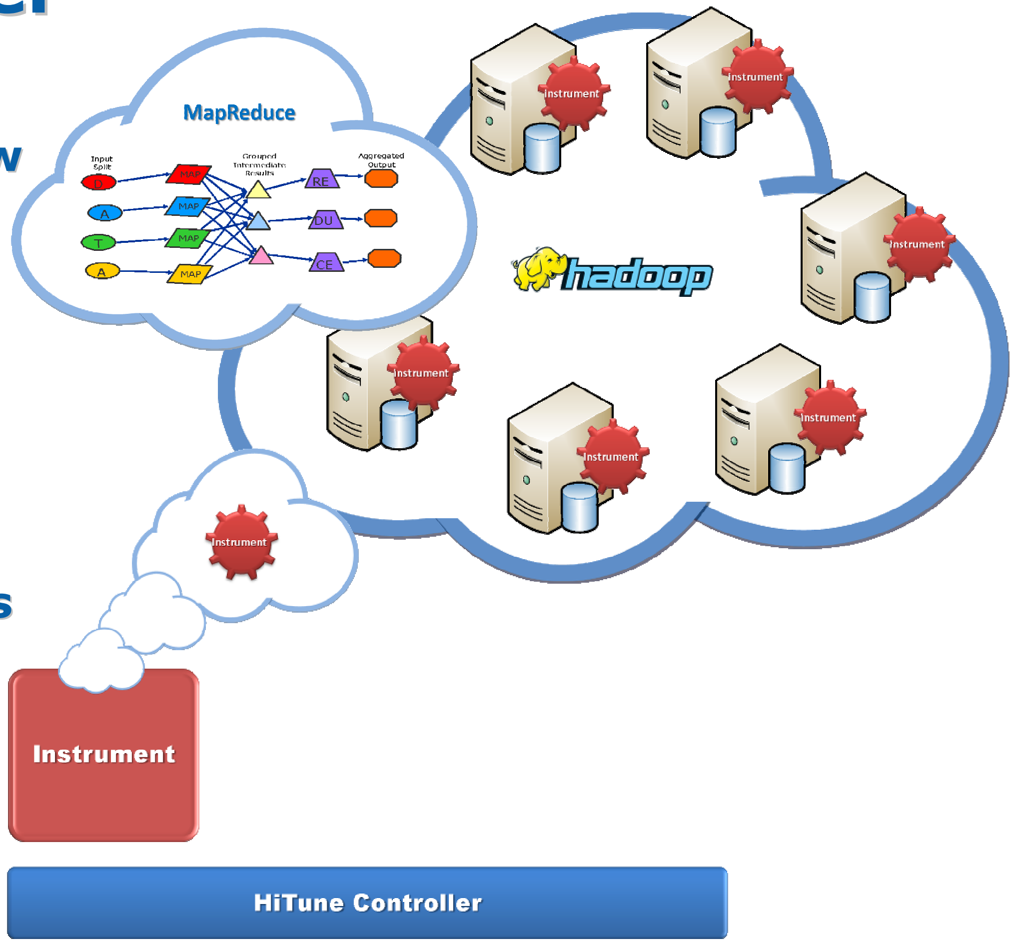
HiTune: Dataflow-Based Hadoop Performance Analyzer

The user develop the application based on the MapReduce dataflow graph

The Hadoop framework dynamically maps the dataflow graph to the underlying cluster

HiTune automatically instruments Hadoop tasks/framework to collect runtime information

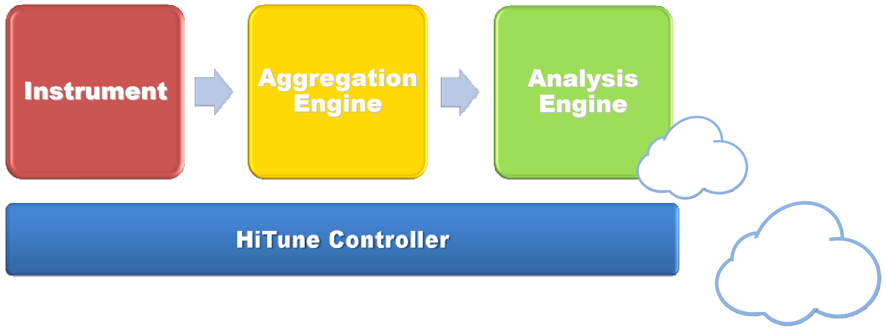
- **At binary level (i.e., no source code changes)**
- **Low overheads (<2%)**



HiTune: Dataflow-Based Hadoop Performance Analyzer

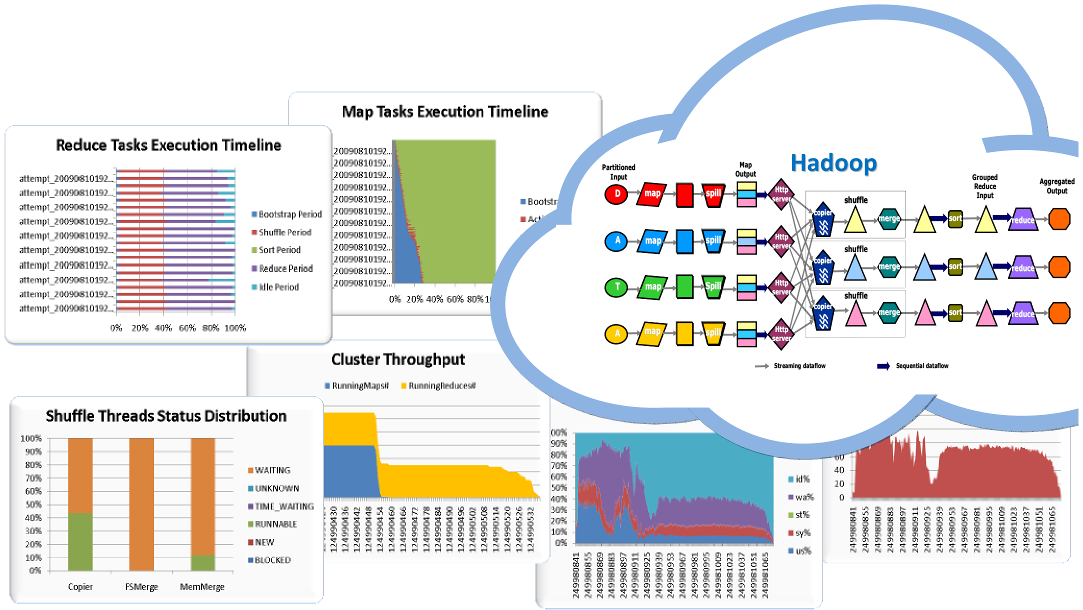
The aggregation engine merges instrumentation results in a distributed fashion

- **Implemented using the *Chukwa* framework**



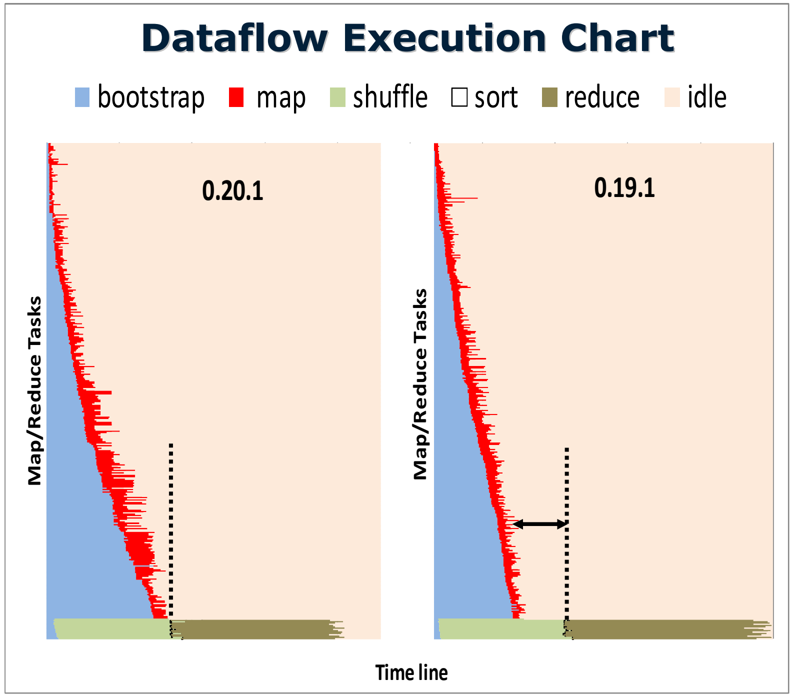
The analysis engine generates visualized report based on the Hadoop dataflow model

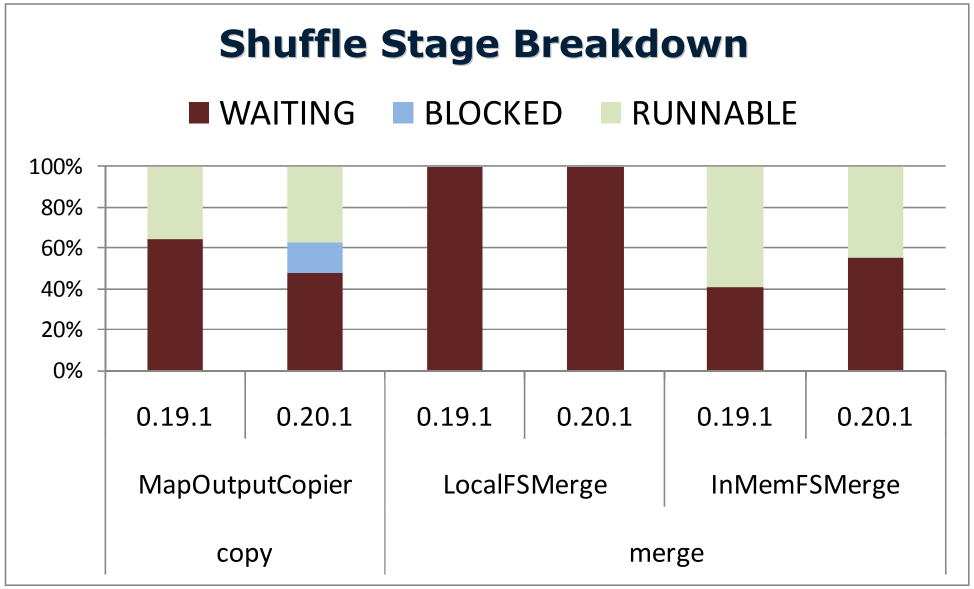
- **Task execution timeline**
- **Task hotspot breakdown**
- **Resource utilizations**
- **…**



HiTune Results: Sorting Large Files

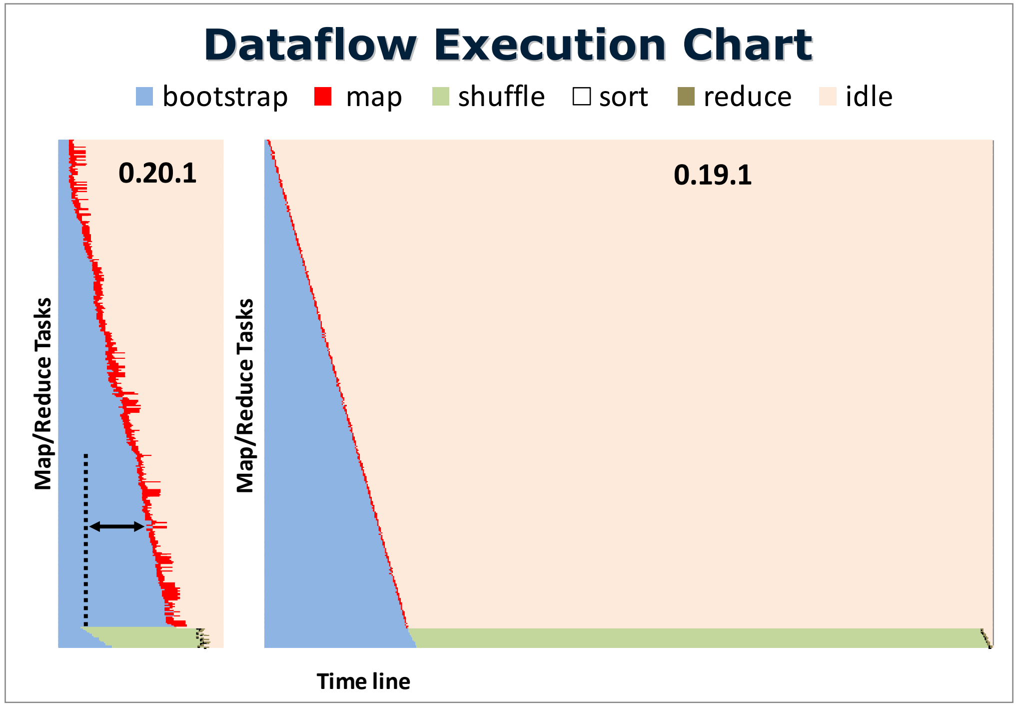
Sorting 60 1GB files





HiTune Results: Sorting Small Files

Sorting 480 33KB files



Agenda

MapReduce/Hadoop overview

- Dataflow model of MapReduce/Hadoop
- Why MapReduce/Hadoop?

The challenges

- Efficient provisioning and performance tuning for Hadoop
- HiTune: dataflow-based Hadoop performance analyzer

Hadoop Performance analysis

- HiBench: a realistic and comprehensive Hadoop benchmark suite
- Balanced architecture design for Hadoop clusters

Summary

HiBench: A Realistic and Comprehensive Hadoop Benchmark Suite

HiBench

1 *Micro Benchmarks*

- Sort
- WordCount
- TeraSort

2 *Web Search*

- Nutch Indexing
- Page Rank

3 *Machine Learning*

- Bayesian Classification
- K-Means Clustering

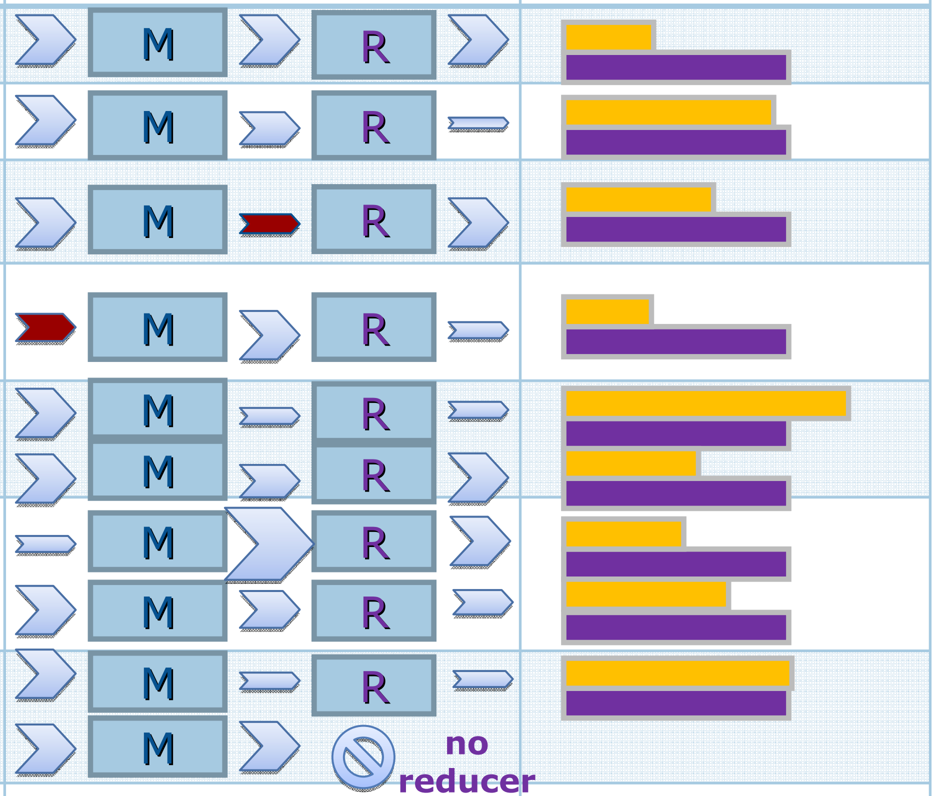
4 *HDFS*

- Enhanced DFSIO

See our paper "The HiBench Suite: Characterization of the MapReduce-Based Data Analysis" in ICDE'10 workshops (WISS'10)

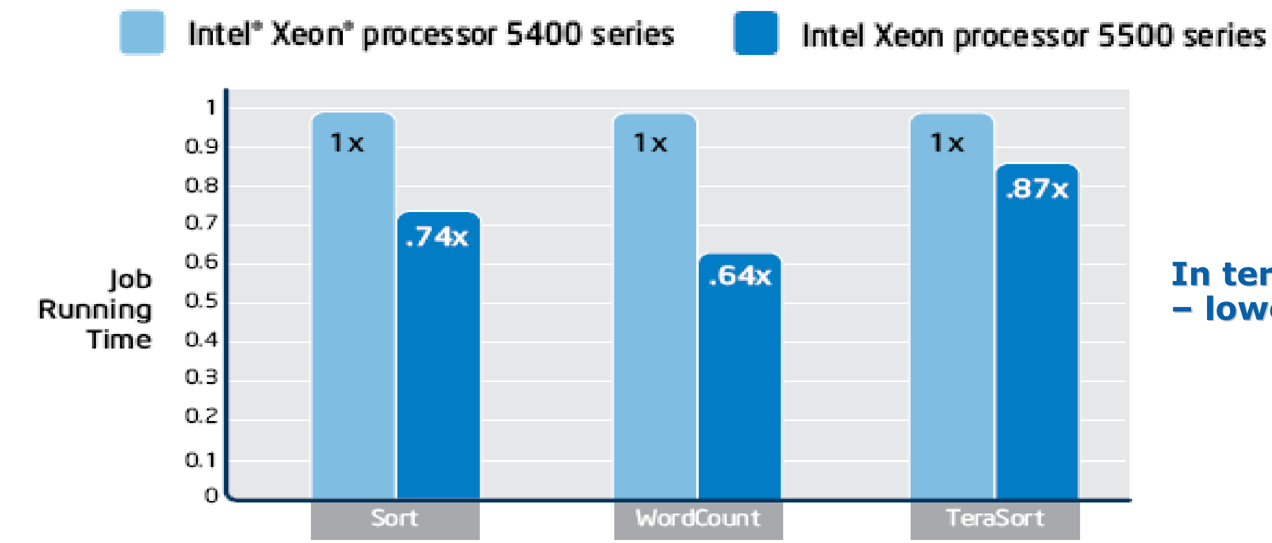
Characterization of HiBench Workloads

Workload	System Resource Utilization	Data Access Patterns	Map/Reduce Stage Time Ratio
Sort	I/O bound		
WordCount	CPU bound		
TeraSort	Map stage : CPU-bound; Red stage : I/O-bound		
Nutch Indexing	I/O bound, high CPU utilization in map stage		
Page Rank (1^{st} & 2^{nd} job)	CPU-bound in all jobs		
Bayesian Classification (1^{st} & 2^{nd} job)	I/O bound, with high CPU utilization in map stage in the 1^{st} job		
K-means Clustering	CPU bound in iteration; I/O bound in clustering	no reducer	
Enhanced DFSIO	I/O-bound	trivial	trivial

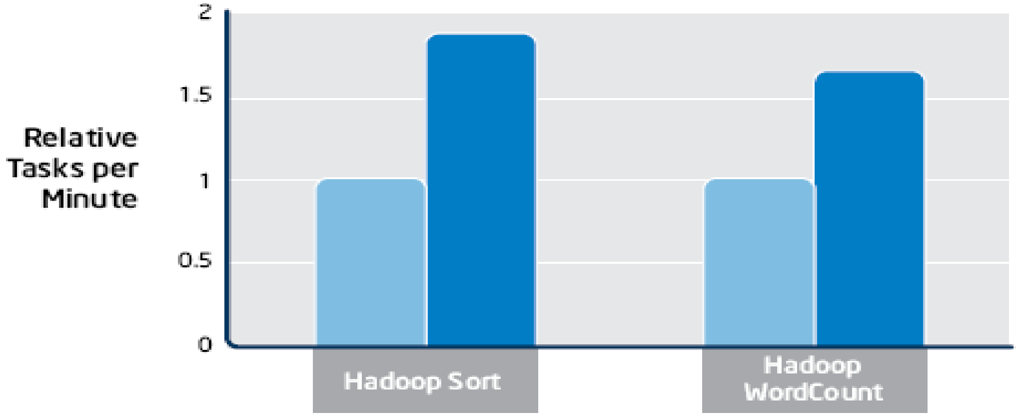


data fewer data even fewer data compressed

Server Platform Comparisons



In terms of job running time – lower is better

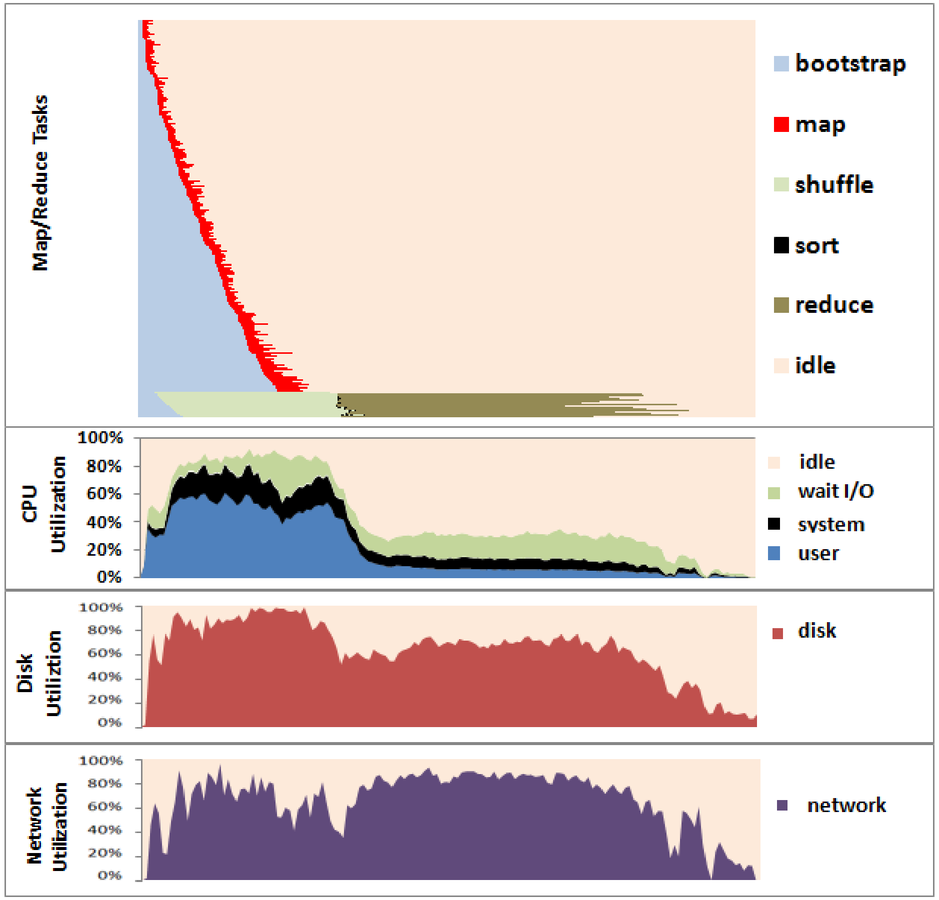


In terms of cluster throughput (*number of tasks completed per minute when cluster is at 100% utilization*) – higher is better

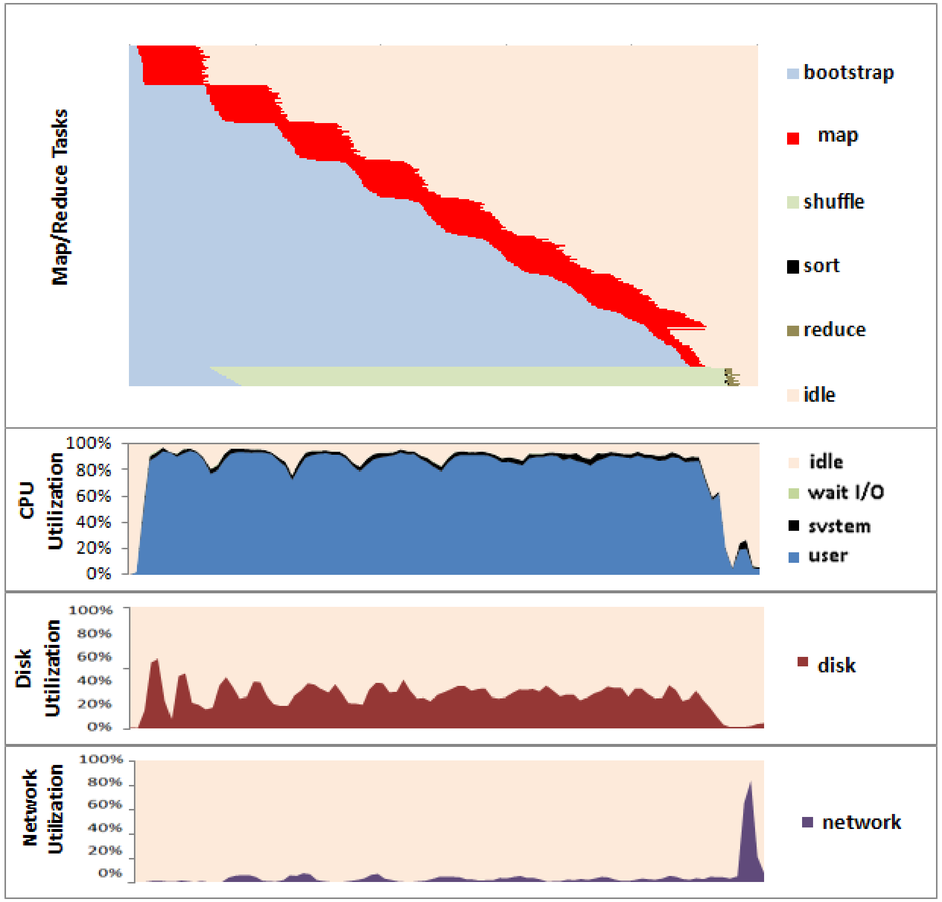
See our whitepaper "Optimizing Hadoop Deployment" released at Hadoop World: NYC 2009 (http://communities.intel.com/docs/DOC-4218)

Hadoop Sort vs. Hadoop WordCount

Sort



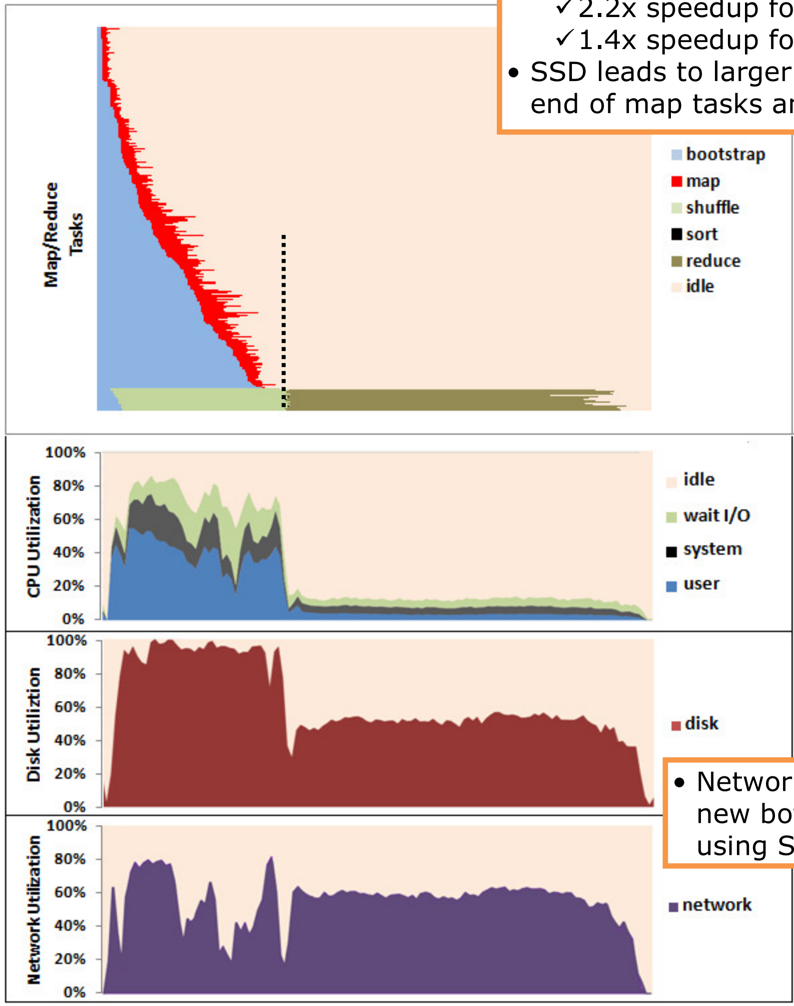
WordCount



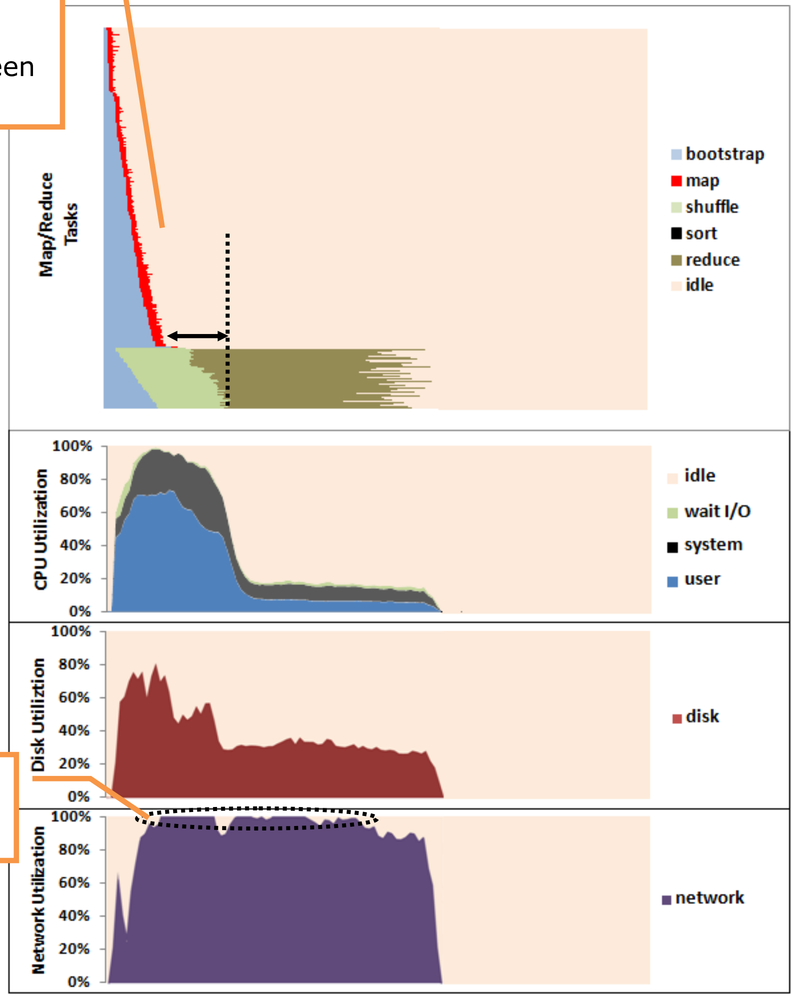
HDD vs. SSD for Hadoop Sort

- SSD brings 1.66x speedup over HDD
  - ✓2.2x speedup for map tasks
  - ✓1.4x speedup for reduce stages
- SSD leads to larger (5.3x) gap between end of map tasks and shuffle stages

HDD



SSD

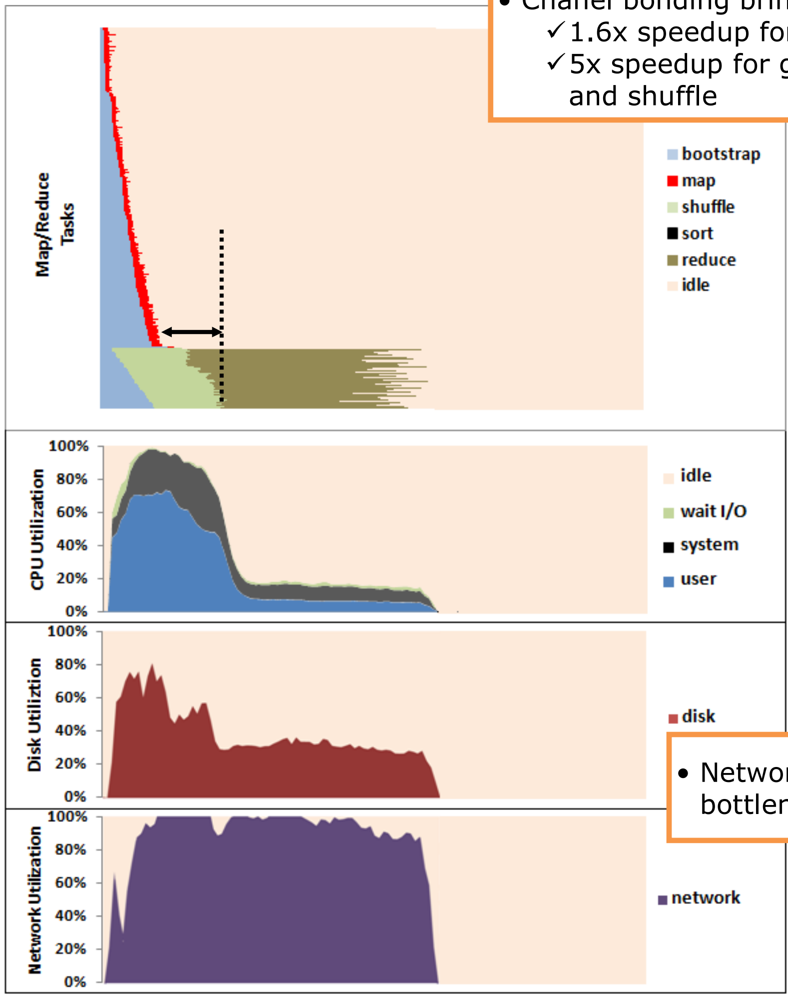


- Network becomes the new bottleneck when using SSD

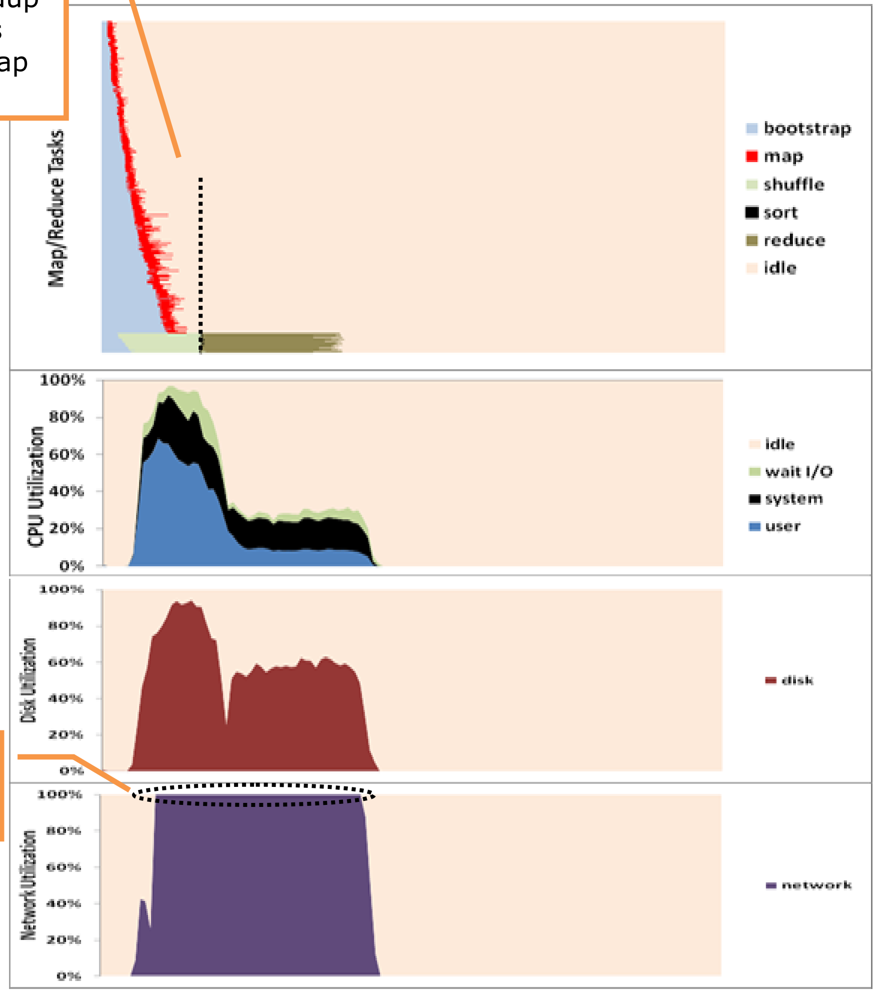
Channel Bonding for Hadoop Sort

- Chanel bonding brings 1.49x speedup
  - ✓1.6x speedup for reduce stages
  - ✓5x speedup for gap between map and shuffle

SSD



SSD + Channel Bonding



- Network is still the bottleneck

Summary

MapReduce/Hadoop

- **Allow the user to work at the right level of abstraction**
  - Make it easy to develop and run massively scalable applications
  - Make it vey challenging to efficiently provision and fine-tune Hadoop systems

HiTune: dataflow-based Hadoop performance analyzer

- **Provide valuable insights into Hadoop runtime behaviors**
  - Instrument Hadoop tasks and framework in a distributed fashion
  - Aggregate instrumentation results to generate visualized analysis report

HiBench: a realistic and comprehensive Hadoop benchmark suite

- **Different Hadoop workloads have bottlenecks in different components of the Hadoop cluster**
- **Improving just one component in the cluster often shift the bottleneck to other components**
  - A balanced cluster architecture design is important to improve Hadoop efficiency

intel®

Leap ahead™