Performance and Energy Efficiency of Hadoop deployment models
Contents

- Review: What is MapReduce
- Review: What is Hadoop
- Hadoop Deployment Models
- Metrics
- Experiment
- Results
- Summary
MapReduce

- Introduced by Google
- Programming model for generating and processing large data sets
- Popular framework for large scale data analysis
- Data generated are often handled as large graphs
MapReduce

- **Map()**
  - `map (in_key, in_value) -> list(out_key, intermediate_value)`
  - Processes input key/value pair
  - Produces set of intermediate pairs

- **Reduce()**
  - `reduce (out_key, list(intermediate_value)) -> list(out_value)`
  - Combines all values for a particular key
  - Produces a set of merged output values
Single Execution

MapReduce

Input

Intermediate

Group by Key

Grouped

Output
MapReduce
Parallel Execution
Apache Hadoop
Apache Hadoop

- Implementation of MapReduce
- An open source project
- Popular to the point of becoming the standard
Hadoop Deployment Models
Hadoop Deployment Models

- Traditional Model:
  - Collocated data and compute services

- Alternate Model:
  - Separate data and compute services
Hadoop Deployment Models

- Collocated Services
  - Physical Clusters
  - Virtual Clusters

- Separate Services
  - Physical Clusters
  - Virtual Clusters
Hadoop Deployment Models

Master and Slaves can be either servers or VMs
Hadoop Deployment Models

**Traditional**

- **Compute**: MapReduce Layer
  - **JobTracker** manages MapReduce jobs based on available map/reduce capacity.

- **Data**: Hadoop Distributed File System (HDFS)
  - **NameNode** system manages DataNode services.

(a) Traditional model: Collocated data and compute
Hadoop Deployment Models

Alternate

- **Compute**: MapReduce Layer
- **Data**: Hadoop Distributed File System (HDFS)
- TaskTracker and DataNode services run on separate dedicated sets of nodes.

(b) Alternate model: Separated data and compute
Hadoop Deployment Models

**Metrics**

- **Performance:**
  - Application Execution Time

- **Power Consumption:**
  - Energy efficiency
  - Power metered servers

- Performance-to-Power Ratio
Experiment
Benchmarks

- **TeraGen**
  - Generates large amounts of data blocks
  - Write intensive

- **TeraSort**
  - Sorts data generated by TeraGen
  - CPU bound during map phase
  - I/O bound during reduce phase

- **Wikipedia Data Processing**
  - Represents data intensive scientific application (filtering, reordering, merging)
Test Platform

- 33 HP DL165 G7 Servers
  - of parapluie cluster
- 3 Sun Fire X2270 Servers
  - for VM management under Snooze system
- 161 VMs
- External network file system (NFS) server hosting data sets for Wikipedia processing
Experiment

**Power Measurement**

- Total power consumption of parapluie cluster
Experiment

**Metrics**

- Application Execution Time
- Performance-to-Power Ratio
- Application progress correlation with power consumption
Metrics

Performance-to-Power Ratio

- Compare power efficiency of Hadoop models
- Performance: inverse of execution time
  - $1 / T_{\text{execution}}$
Application progress correlation with power consumption

- Workload’s power consumption profiles
Results
Results

Traditional Deployment (Execution Time)

Fig. 2. Hadoop Wikipedia data processing for three data-intensive operations on Wikipedia data with collocated data and compute services. Servers outperform VMs.
Results

**Traditional Deployment**
*(Execution Time)*

- Significant performance degradation on VMs

- On servers:
  - Filter 1.3 to **3.2** times faster
  - Reorder 2.1 to **2.5** times faster
  - Merge 2.3 to **3.3** times faster
  - TeraGen and TeraSort up to **2.7** times faster
I/O heavy benchmarks perform poorly in virtualized environments

Overhead compounded with multiple (read: 5) VMs per server

Competing for resources
Results

Alternate Deployment (Performance to Power Ratio)

Fig. 3. Hadoop Wikipedia data processing performance to power ratios for three data-intensive operations with separated data and compute services on servers. For filter with largest input size, the 16-8 data-compute ratio achieves the best results due to high write I/O. Reorder and merge perform the best with the 8-16 data-compute ratio. Adding more compute servers does not yield improvements.
Collocation consistently holds highest performance to power ratio

Impact of separating data and compute services heavily depends on data-compute ratio

Adding more compute servers did not yield significant improvement
Results

Application Power Consumption Profiles

TeraGen and TeraSort percentage of remaining map/reduce and power consumption with collocated data and compute layers on servers for 500GB. Map and reduce completion correlates with decrease in power consumption.

Trends similar for other data sets not shown.
Results

Application Power Consumption Profiles

- Remaining percentage of maps and reduces correlate with power consumption

- When map and reduce complete, power consumption decreases
  - Indication of underutilized servers
Results

Application Power Consumption Profiles

- **TeraGen:**
  - High, steady power consumption between 100% and 40%
  - high CPU utilization

- **TeraSort:**
  - Similar behavior
  - Long shuffle and reduce phase creates more fluctuations in power consumption
Results

Application Power Consumption Profiles

Different power profiles show granularity where energy saving mechanisms might be considered.
Results

Application Power Consumption Profiles

Remaining percentage of map/reduce and power consumption for Hadoop Wikipedia data processing with 80 data and 30 compute VMs. Power consumption drops as the map and reduce complete.
Results

Application Power Consumption Profiles

- Similar results for *collocated scenario* and other ratios of separated data and compute services
Results

**Application Power Consumption Profiles**

- Power consumption profile is significantly different from TeraGen and TeraSort
  - Steady *map* phase
  - Smooth *reduce* phase

- Indicates power consumption profiles are heavily application specific
Summary
Key Findings

- Hadoop on VMs yields significant performance degradation with increasing data scales for both compute and data intensive applications
Summary

Key Findings

- Separation of data and compute layers reduces the performance-to-power ratio

- Degree of reduction depends on:
  - Application
  - Data Size
  - Data to Compute ratio
Key Findings

Power consumption profiles are application specific and correlate with the map and reduce phases.

Opportunities for applying energy saving mechanisms.
Thank you

The End