MapReduce
Simplified Data Processing on Large Clusters

Dean J. and Ghemawat S.
Google, 2008

Presented by Robert Hoff
14 Feb 2012

MapReduce

- Distributed Execution Engine
- For Processing Large Datasets
- Provides a restrictive programming model to achieve this
By Google

Originated in 2003 to Solve search related problems
- Inverted Indices (Pagerank)
- Word Count
- Most Frequent Queries

Previously at Google
- Issues of parallelisation, fault-tolerance, load-balancing were specific for each problem
- Using ideas from functional programming, map and reduce don't have side effects and can be parallelised
- This method turned out to be applicable to most of their computational requirements
Related Work

- There existed systems that provided restricted programming models, and used these to parallelise the computations.

MapReduce main contributions at the time

- Fault Tolerance (running on top of commodity HW)
- Higher-Level of abstraction

Can consider separately:

![Diagram showing the separation between Programming Interface and Execution Engine (The Implementation)]
Map and reduce are client supplied functions (*may be anything*). These are applied to an input set that can be broken into n number of (k1, v1) pieces.
reduce

(k2, list(v2)) -> list(v3)

Word Count Example

```
"foo", 5
"bar," 0
"foo", 3
"bar," 4
"foo", 2
"bar," 3
```

Map must finish before reduce starts
Twitter Hashtag Count

- "#foo", 5
- "#bar", 0
- "#foo", 3
- "#bar", 4
- "#foo", 2
- "#bar", 3
- "#foo", 10
- "#bar", 7

Implementation

- Single Master
- Assigns Workers
- Fault Tolerant (includes failed and lagging workers)
Performance – Grep

- Searches for a \(10^{10}\) 100 byte records for a three character pattern
- \(10^{12}\) bytes = 1,000,000 MB = 15,000 x 64MB chunks
- 1800 Worker Machines

Experience

MapReduce Applied to an increasing number of useful Problems

- Machine learning (e.g. statistical translation)
- Clustering for Google News
- Graph Computations (social network data)
Further / Future Work

Since MapReduce programming model is restrictive and can only be applied to limited set of problems. Research is ongoing on execution engines that have higher generality

- DryadLINQ
- CIEL

Further / Future Work

The ideas of MapReduce, or any other Distributed Execution Engine may be applied to many-core architectures.

For example Open-Source version Phoenix (from Stanford). Automatically manages thread creation, dynamic task scheduling, data partitioning, and fault tolerance across processor nodes.
The paper - Remarks

● MapReduce solves Google's problems well.
● Results and ideas are highly replicable.
● But, somewhat disassociated from other research, lacks comparisons to other work (solves Google's problems well enough so why bother?)

Conclusion

● MapReduce is still in use by Google today, solving a growing number of problems.
● MapReduce has become the leading programming model of choice for processing large data sets
● Open-Source versions (e.g. Hadoop) are employed by many other organisations