

Dumbo: Realistically Simulating MapReduce for Performance Analysis



Guanying Wang, Ali R. Butt
Virginia Tech
{wanggy, butta}@cs.vt.edu

Prashant Pandey, Karan Gupta
IBM Almaden Research Center
{ppandey, guptaka}@us.ibm.com



Motivation

Why simulate MapReduce?

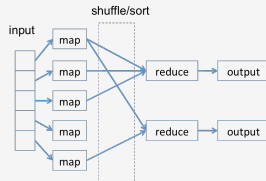
- Understanding MapReduce systems is critical given their increasing popularity and wide-spread use
 - Exact methods to design, build, and judiciously use resources in a MapReduce cluster are not clear
 - In practice, designers only have empirical insights to rely on, and follow the insights with little or no customization
 - Many, if not most, clusters yield low performance
- However, building and testing each and every cluster configuration is not feasible

Solution: Use realistic simulations to study behavior of MapReduce systems

Efficient means to study/test MapReduce cluster designs

Background: MapReduce

- A framework to run large-scale data-intensive applications
- A key enabler for “Cloud Computing”
- Map side
 - Map phase
 - Sort phase
 - Spill phase
- Reduce side
 - Shuffle/sort phase
 - Reduce phase



Dumbo Usage

Cluster planning tool

- Estimate a given MapReduce cluster’s performance before actually having to build the cluster
- Study alternate configurations, select most efficient design
- Make resource allocation decisions to balance computing power, storage, and network bandwidth constraints

Research based on simulator as well as on real systems

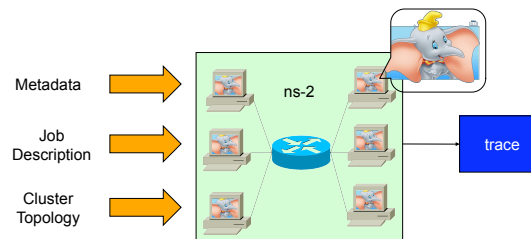
- Evaluate and extend design decisions in MapReduce implementations such as Hadoop
- Eliminate dependence on arbitrary rules of thumb

Challenges

- What is the right level of abstraction to use?
 - Fine-grain: simulation very slow for large applications
 - Coarse-grain: simulation not very accurate
- How to handle distribution of intermediate results between phases?
 - Reduce phase depends on distribution of keys
 - Distribution of keys depends on input data
- How to correctly model failures?
- How to validate Dumbo?
 - Use data from real setups with different scales & configurations

Design of Dumbo

- Simulate distributed communication at network-packet level
 - Employ ns-2
- Simulate node activities at phase-level
 - Computation node and disk modeled as process queues
 - Jobs modeled using compute and I/O characteristics
 - Metadata modeled randomly, as well as from real systems traces
- Assume uniform key distribution for reducers

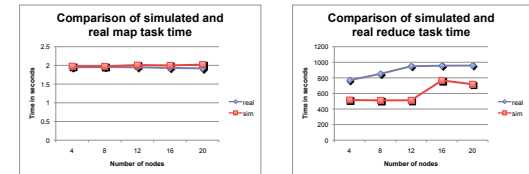


Simulator Validation

- Real experimental data collected using a small-sized test cluster:
 - 20-nodes organized in 2 racks with 10 nodes per rack connected using 1 Gbps Ethernet, running Hadoop
 - Run Terasort application
- Dumbo trained using a mini-setup of 4 nodes
 - Provide Dumbo information about test cluster
 - Provide compute-I/O description of the applications

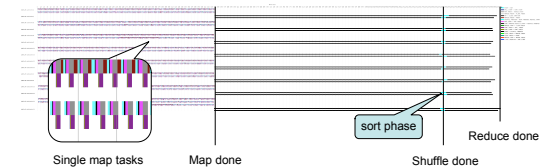
Results

- Use Dumbo to predict performance of test cluster
- Simulated Map results match observed results
- Reduce results need to be improved

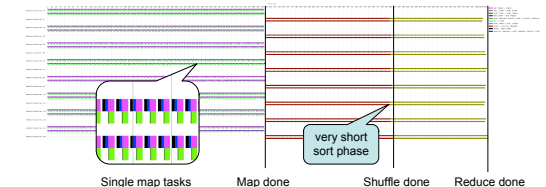


Phase-level View*

Real system



Simulator



Conclusion & Future Work

- Dumbo provides a good model for capturing complex MapReduce interactions and predicting the performance
- Dumbo aids in designing of emerging clusters for supporting MapReduce
- In our future work, we aim to:
 - Validate Dumbo for more scenarios at scale
 - Better model for reduce side phases
 - Add failure model
 - Replace/enhance ns-2 for better simulator performance

* Tool to generate phase-level views are created by Spyridon Blanas. Dumbo graphic courtesy of Disney.