Performance of map and reduce tasks are modeled from execution times of each phase in these tasks. For MACGYVER can also perform profiling and performance prediction using performance analytical models. Overlap Generalized configuration Future plan is to design advanced DAG execution framework (e.g. Tez) with modern HPC resources. HOMR MOFs are divided into small packets and are shuffled instead, based request response messages. Designing and map Reduce cannot take advantage of the underlying global file system in HPC clusters, such as Lustre. We propose an advanced design of HOMR, that can utilize Lustre and extract further benefits. The intermediate data directory can be configured to the local disks [4] or Lustre [5] or a combination of both [7].

MapReduce over Lustre

Default MapReduce cannot take advantage of the underlying global file system in HPC clusters, such as Lustre. We propose an advanced design of HOMR, that can utilize Lustre and extract further benefits. The intermediate data directory can be configured to the local disks [4] or Lustre [5] or a combination of both [7].

RDMA-based MapReduce

- MRDB [1] introduces RDMA-based shuffle, replacing the slower HTTP-based request response messages. MOFs are divided into small packets and are shuffled instead of shuffling the entire data at once as in default framework.
- No on-disk merge. Initially, small packets of data are required to create the Priority Queue (PQ); subsequent packets are inserted in this PQ for sorting operation.
- Merge and Reduce phase can run in an overlapping manner.
- Pre-fetching and caching of Map Output Files are introduced to accelerate the response from TaskTracker for each request of ReduceTasks. Performance evaluation shows 39% ($\Delta_{39}$) reduction in time with 2 HDD/node (1 HDD/node) for HDFS.

Hybrid Overlapping in MapReduce

- HOMR [2] (hybrid Overlapping in MapReduce) is designed to have maximum possible overlapping across all phases of MapReduce.
- HOMR also ensures faster job execution over other high performance interconnects (100G/40G) because of its new shuffle algorithm: provides the fastest execution over RDMA.
- HOMR assigns weights to different maps to signify how much data to shuffle on each request; this assignment can be greedy / all-average.
- Initial static weight assignment is updated by on-demand adjustment which makes each shuffle to bring only the map outputs needed; intelligent shuffling provides faster job execution pipeline.

Tuning, Profiling, and Prediction

- We design a generalized parameter tuning and prediction framework (MACGYVER) for any MapReduce implementation [4].
- Automatic tuning, profiling is performed for MapReduce implementations in Hadoop, Spark, and HOMR with file systems = HDFS, Lustre, and Tachyon.
- Generalized configuration parameter space is devised to facilitate different MapReduce implementations.
- MACGYVER can also perform profiling and performance prediction using performance analytical models.
- Performance of map and reduce tasks are modeled from execution times of each phase in these tasks. For example, execution time for a single Reduce task can be modeled as:
  \[ T_{map} = f_{shuffle} + f_{reduce} \]
- For ROMA-based MR, execution time can be re-modeled (2) $T_{map} = \max(f_{shuffle} + f_{reduce}) + \alpha \cdot f_{reduce}$
- Compared to Starfish, MACGYVER can achieve better speedups for different applications.

Conclusion and Future Work

- For large scale data processing, HOMR achieves significant performance benefits compared to default Hadoop MapReduce; leverages benefit from modern HPC resources (ROMA and Lustre).
- Future plan is to design advanced DAG execution framework (e.g. Tez) with modern HPC resources.

Software Distribution

- HOMR is publicly available in “RDMA for Apache Hadoop” public release (http://hbd.cse.ohio-state.edu)
- As of Sep’16, more than 17,850 downloads (190 different organizations) have taken place from this site.

References


Acknowledgements

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu

High-Performance Big Data (HiBD)
http://hbd.cse.ohio-state.edu

This research is supported in part by National Science Foundation grant #IIS-1447804.