High-Performance Hadoop and Spark MapReduce on Modern HPC Systems

Md. Wasi-ur- Rahman, Advisor: Dhabaleswar K. Panda. Email: {rahmanmd, panda} @cse.ohio-state.edu
Done as a part of the High-Performance Big Data (HiBD) project (http://hibd.cse.ohio-state.edu)

MapReduce Programming Model

Limitations of MapReduce
- MapReduce implementations (e.g. Hadoop, Spark) use Ethernet infrastructure with Java sockets
- Large number of disk operations for Hadoop
- Concerns for performance and scalability

Applications
- Hadoop MapReduce
- Sockets
- Storage
  - Temporary Data (Local Disks)
  - Persistent Data (HDFS)

Number of InfiniBand Systems
- 47%
- 24%
- 15%
- 12%
- 2%

- HPC adopted advanced interconnects and protocols
- Low latency, high bandwidth
- HPC deploys vast parallel file systems, e.g. Lustre
- In-memory based computing is more popular

Broad Challenge
Can MapReduce execution frameworks (Hadoop, Spark) take advantage of HPC resources?

Selected Contributions
- RDMA-enhanced Hadoop MapReduce (ICS’14)
- RDMA-enhanced HDFS (SC’12)
- MapReduce over Luster (TPDS’16)
- Hybrid Hierarchical HDFS (CCGrid’15)
- RDMA-enhanced Spark (HotI’14)
High-Performance Hadoop and Spark MapReduce on Modern HPC Systems
Md. Wasi-ur Rahman, Advisor: Dhabaleswar K. Panda. Email: {rahmanmd, panda}@cse.ohio-state.edu
Done as a part of the High-Performance Big Data (HiBD) project (http://hibd.cse.ohio-state.edu)

Applications
- Resource Manager
- Node Manager
- App Master
- Map Reduce
- Java Sockets Interface
- Java Native Interface (JNI)
- MRoIB
- IB Verbs
- RDMA Capable Networks (IB, 10GE/iWARP, RoCE ...)

Enhanced phase overlapping
- Map
- Shuffle
- Merge
- Reduce
- Implicit Barrier

IB Verbs
- RDMA Capable Networks (IB, 10GE/iWARP, RoCE ...)
- MRoIB
- Java Sockets Interface (JNI)
- Resource Manager
- Node Manager
- App Master
- Map Reduce
- Java Native Interface (JNI)

MapReduce with Advanced Features
- MapReduce over Lustre
- Spark Applications (Scala/Java/Python)
- Spark BlockManager (Scala/Java)
- Spark BlockFetcherTerator (Scala/Java)
- Java Socket
- RDMA-based Shuffle Engine (Java/JNI)
- Native InfiniBand (QDR/FDR)

Normalized Execution Time
- AdjList (30 GB)
- SeuJoin (80 GB)
- SeqCount (30 GB)
- WordCount (30 GB)
- InvertIndex (30 GB)

Job Execution Time (sec)
- Cluster Size
- Data Size (GB)
- Time (sec)

Enhanced phase overlapping
- Map
- Shuffle
- Merge
- Reduce
- Implicit Barrier

Job Execution Time (sec)
- Cluster Size
- Data Size (GB)
- Time (sec)