Lustre User Group 2013 | China and Japan
Hosted by OpenSFS
Beijing - October 15 Tokyo - October 17

Sponsored by: intel

MapReduce and Lustre*: Running Hadoop* in a High Performance Computing Environment

Ralph H. Castain – Senior Architect, Intel Corporation
Omkar Kulkarni – Software Developer, Intel Corporation
Xu, Zhenyu – Software Engineer, Intel Corporation

Agenda

- Hadoop* and HPC: State of the Union
- Enabling MapReduce on Slurm
- Enabling MapReduce on Lustre*
- Summary and Q&A

Agenda

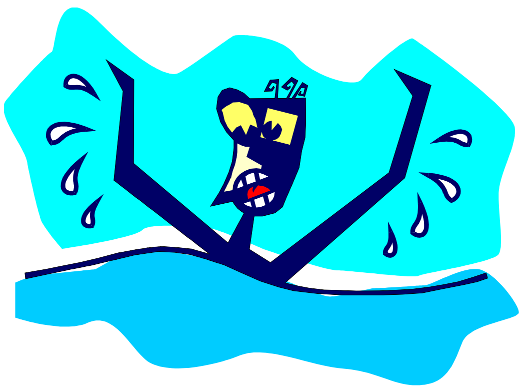
- Hadoop* and HPC: State of the Union
- Enabling MapReduce on Slurm
- Enabling MapReduce on Lustre*
- Summary and Q&A

A Tale of Two Communities

HPC grew out of a need for computational speed

- Scientific programming
- Small input, large output
- High-speed networks, parallel file systems, diskless nodes

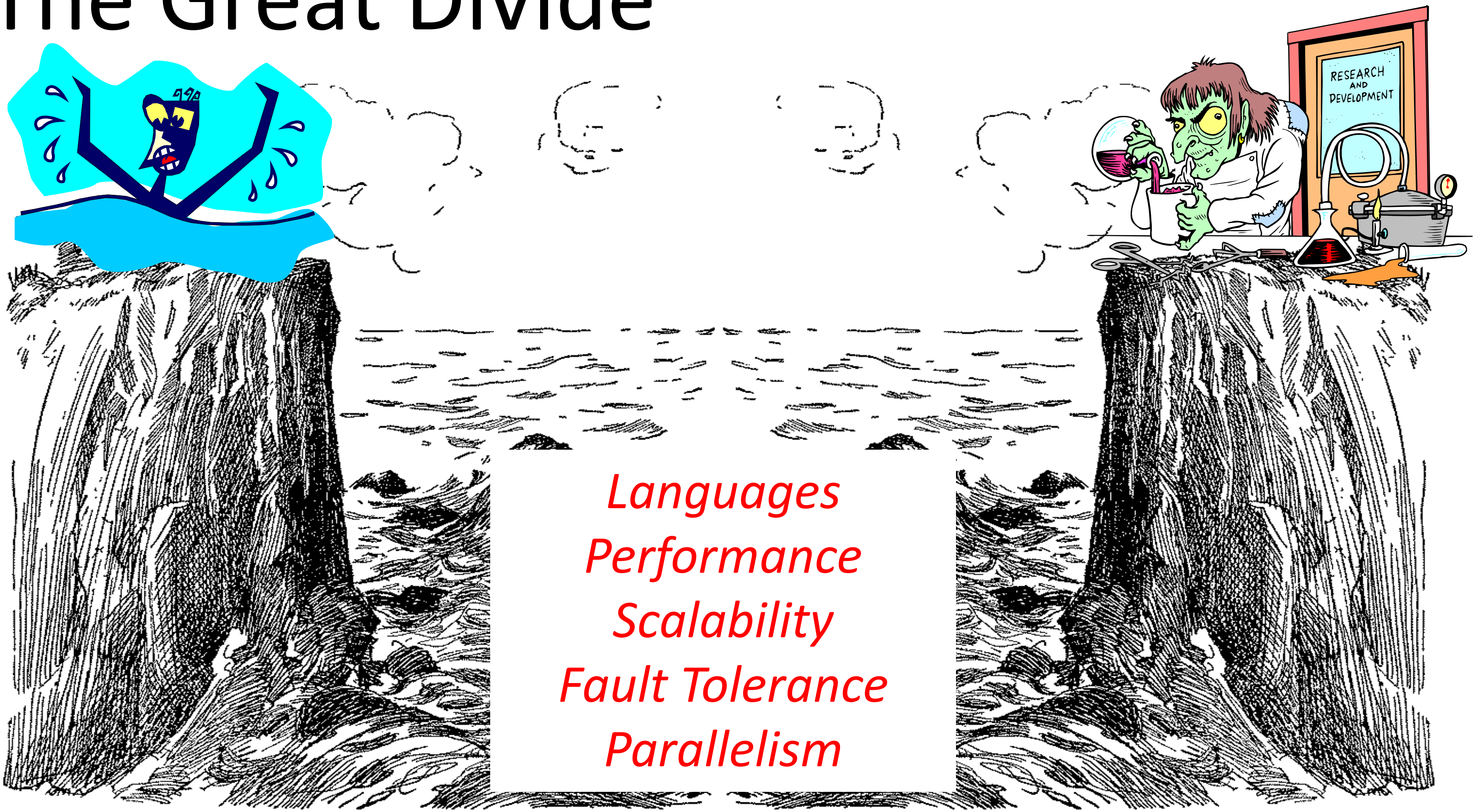
A Tale of Two Communities



Hadoop* grew out of a need to process large volumes of data

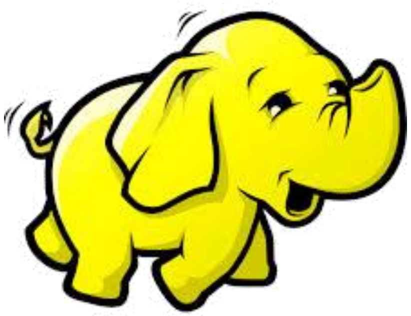
- Web programming
- Large input, small output
- Low-speed networks, local file systems, diskful nodes

The Great Divide



Languages
Performance
Scalability
Fault Tolerance
Parallelism

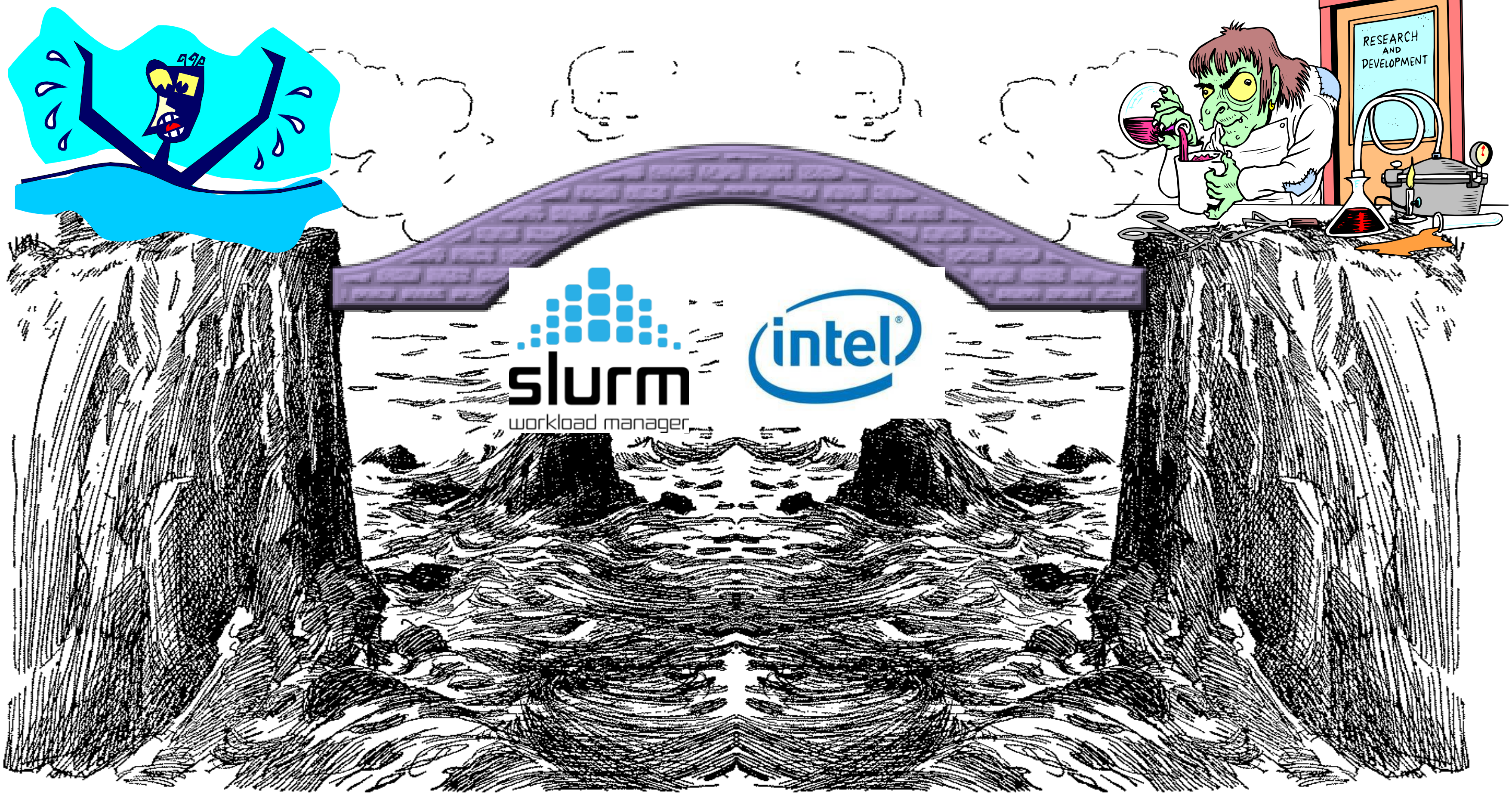
The Gorge is Deep



- Embedded resource manager
  - Retrain IT staff, users
  - Very slow application start
  - No hardware awareness
  - Cannot share with typical HPC applications
- HDFS*
  - Retrain IT staff, users
  - Requires local disks on every node
  - Poor support for non-Hadoop* uses

Cannot leverage existing infrastructure requires up-front expense for dedicated hardware

The Great Divide



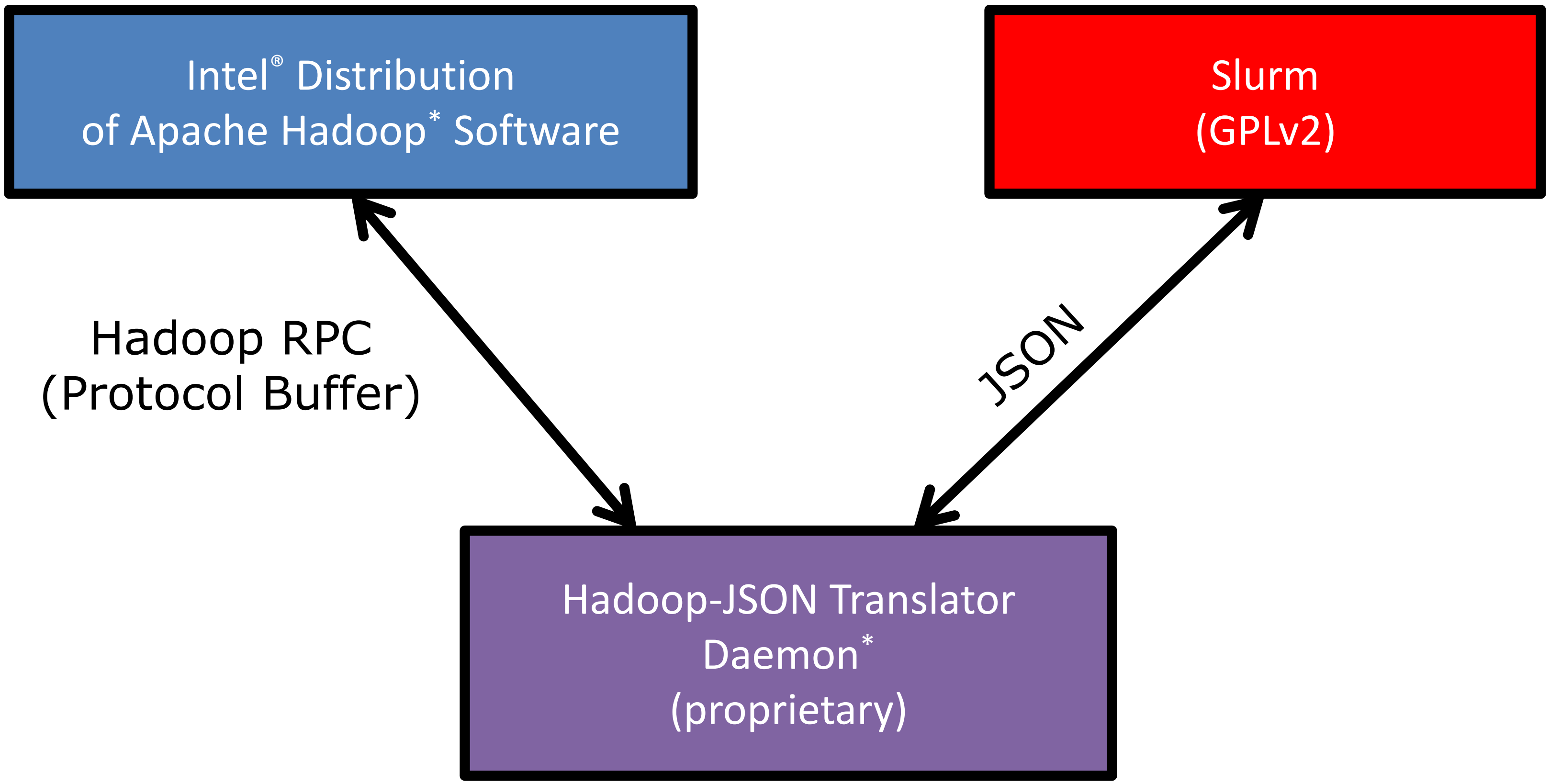
Agenda

- Hadoop* and HPC: State of the Union
- Enabling MapReduce on Slurm
- Enabling MapReduce on Lustre*
- Summary and Q&A

Integrating Hadoop* with Slurm

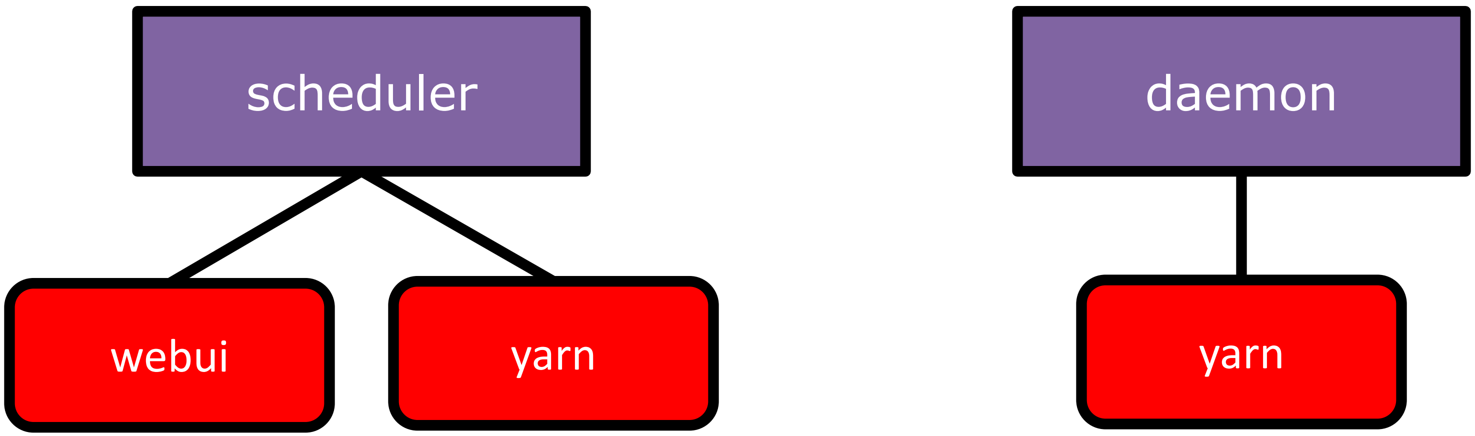
- Why Slurm
  - Widely used open source RM
  - Provides reference implementation for other RMs to model
- Objectives
  - No modifications to Hadoop* or its APIs
  - Enable all Hadoop applications to execute without modification
  - Maintain license separation
    - Apache*, GPLv2
  - Fully and transparently share HPC resources
  - Improve performance

Methodology



No change to Hadoop applications
Fully support Hadoop 2.0 series

Slurm Plugins



webui: Web interface for Hadoop*-like job/system status

yarn (scheduler): Accept and process JSON RM requests from HJTD

yarn (daemon): Accept and process JSON NM requests from HJTD

Create a Hadoop-like environment for similar user experience

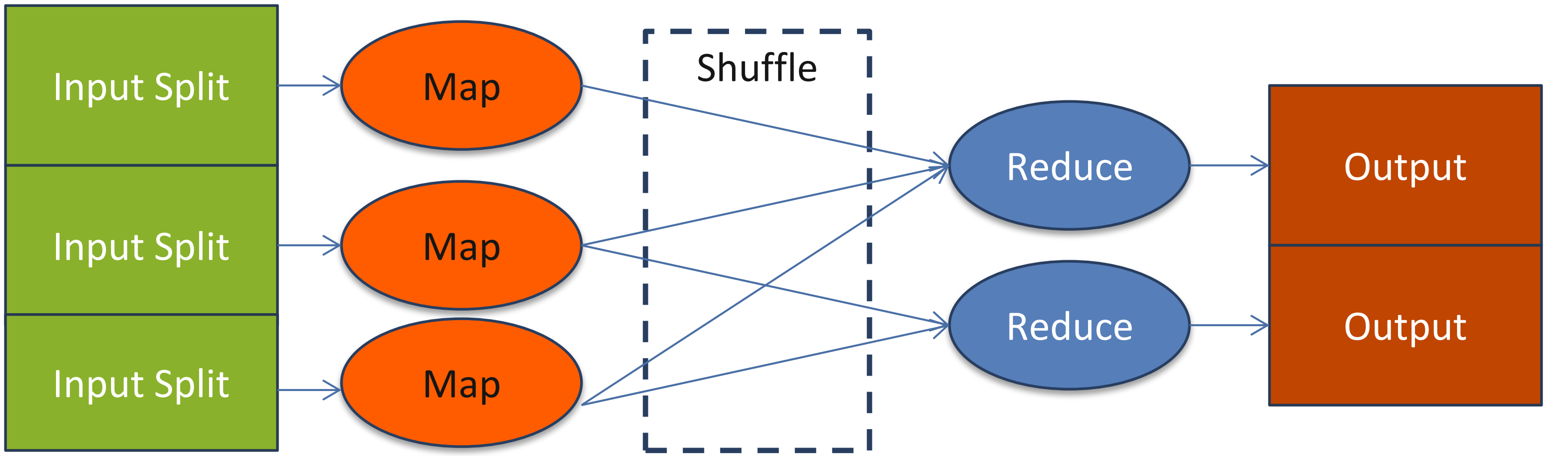
Agenda

- Hadoop* and HPC: State of the Union
- Enabling MapReduce on Slurm
- Enabling MapReduce on Lustre*
- Summary and Q&A

Why Hadoop* with Lustre*?

- HPC moving towards Exascale. Simulations will only get bigger.
- Need tools to run analyses on resulting massive datasets
- Natural allies:
  - Hadoop* is the most popular software stack for big data analytics
  - Lustre* is the file system of choice for most HPC clusters
- Lustre is POSIX compliant: use the Java* native file-system support
- Easier to manage a single storage platform
  - No data transfer overhead for staging inputs and extracting results
  - No need to partition storage into HPC (Lustre) and Analytics (HDFS*)
- HDFS expects nodes with locally attached disks, while most HPC clusters have decoupled storage and compute nodes

The Anatomy of MapReduce

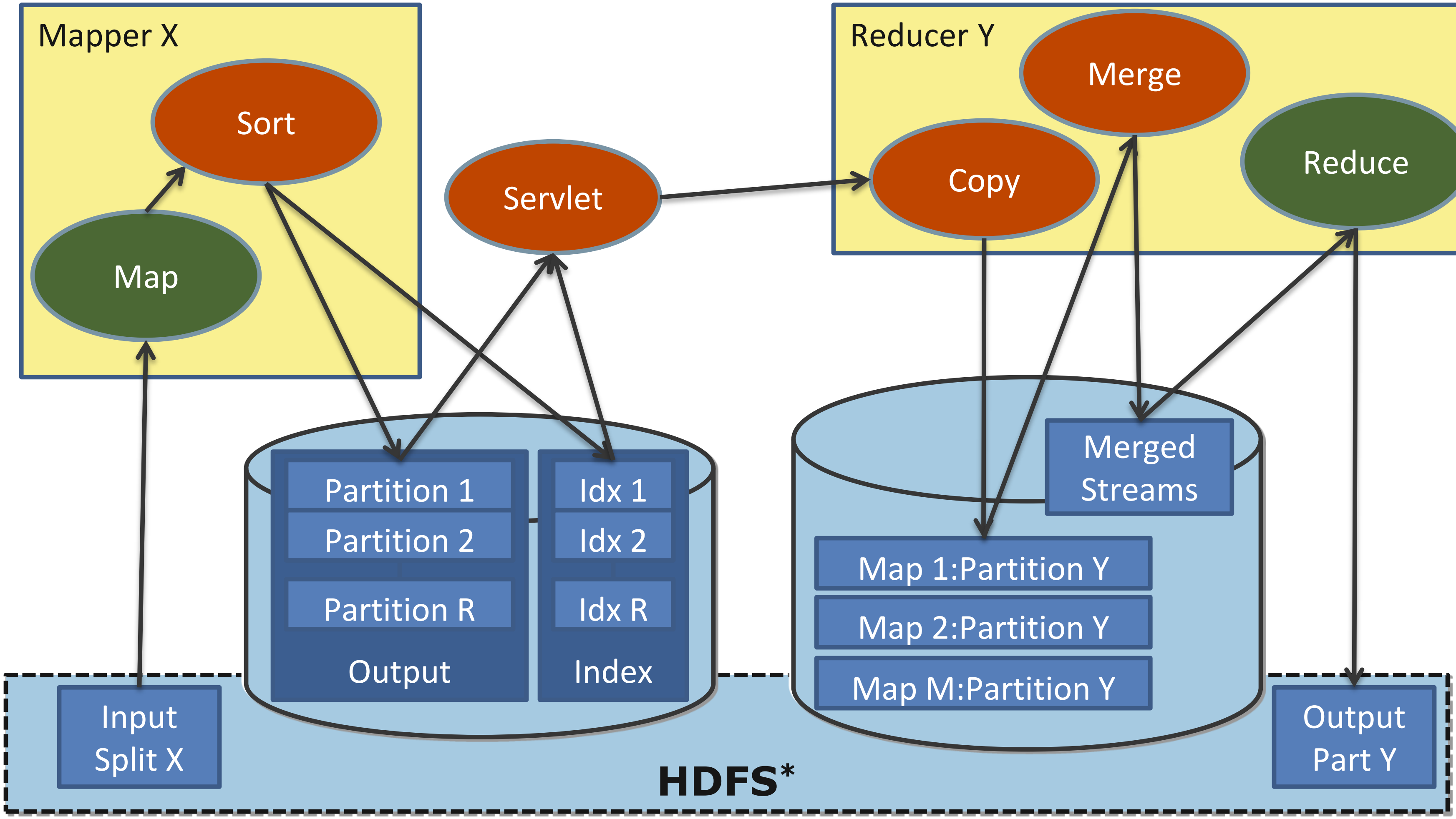


- Framework handles most of the execution
- Splits input logically and feeds mappers
- Partitions and sorts map outputs (Sort)
- Transports map outputs to reducers (Shuffle)
- Merges output obtained from each mapper (Merge)

Sort, Shuffle & Merge

- Mappers generate records (Key-Value pairs) which are organized into partitions, one for each reducer
- Records within a partition are sorted in a memory buffer
- A background thread flushes the buffer to disk when full (Spill)
- Eventually, all spills are merged partition-wise into a single file
- An index file containing partition metadata is created
- Metadata = [Offset, Compressed Length, Original Length]
- Partitions are streamed to reducers over HTTP when requested
- All streams are merged into one before reducing

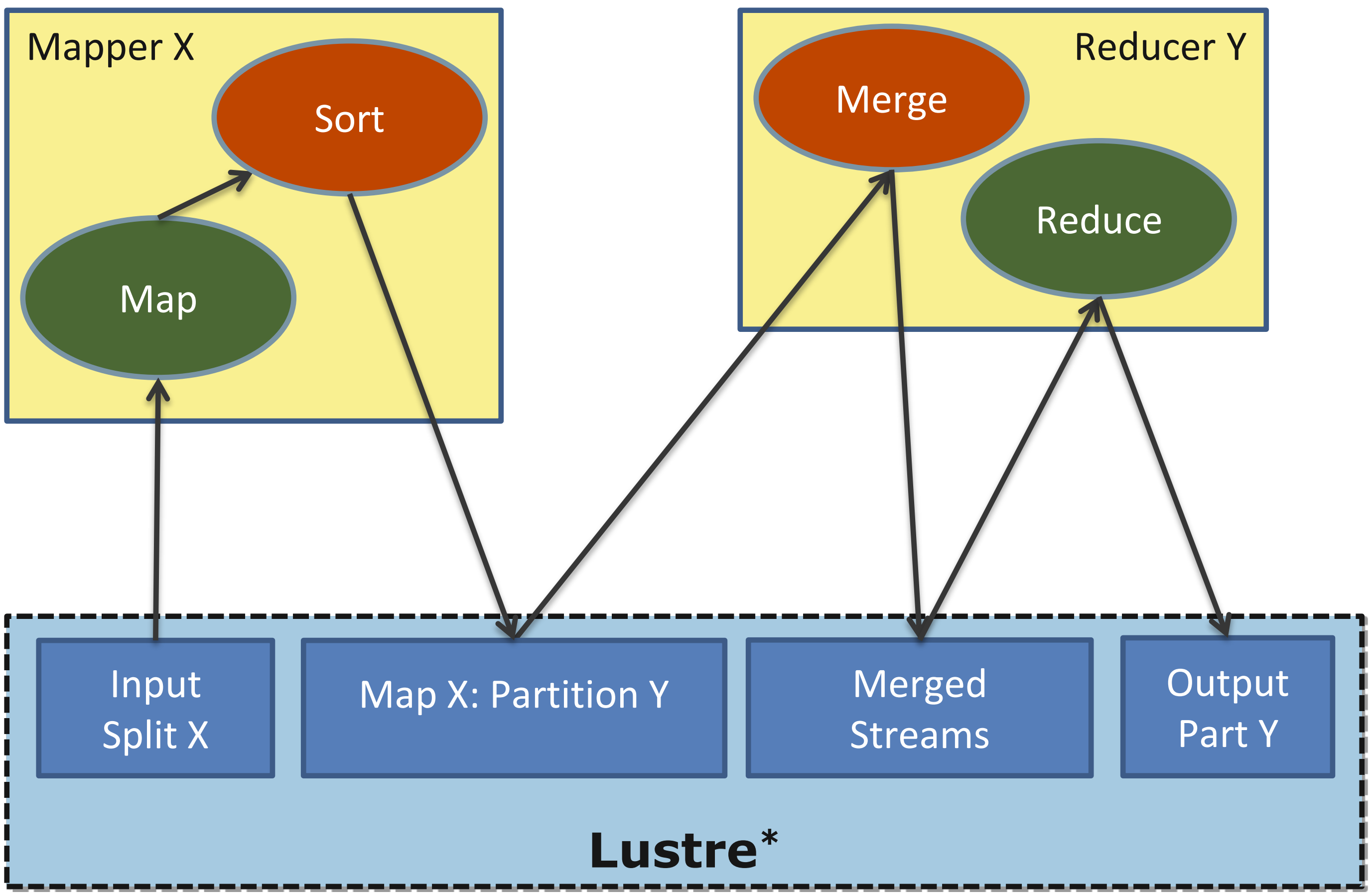
Sort, Shuffle & Merge



Optimizing for Lustre*: Eliminating Shuffle

- Why? Biggest bottleneck – bad for Lustre*!
- How? Since file system is shared across nodes, shuffle is redundant.
- Reducers may access map outputs directly, given the path
  - But, index information would still be needed to read partitions
- We could allow reducers to read index files, as well
  - Results in (M*R) small (24 bytes/record) IO operations
- Or convey index information to reducer via HTTP
  - Advantage: Read entire index file at once, and cache it
  - Disadvantage: Still (M*R) Disk seeks to read partitions + HTTP latency
- Our approach: Put each map output partition in a separate file
  - Three birds with one stone: No index files, no disk seeks, no HTTP

Optimizing for Lustre*: Eliminating Shuffle



Hadoop* Adaptor for Lustre*

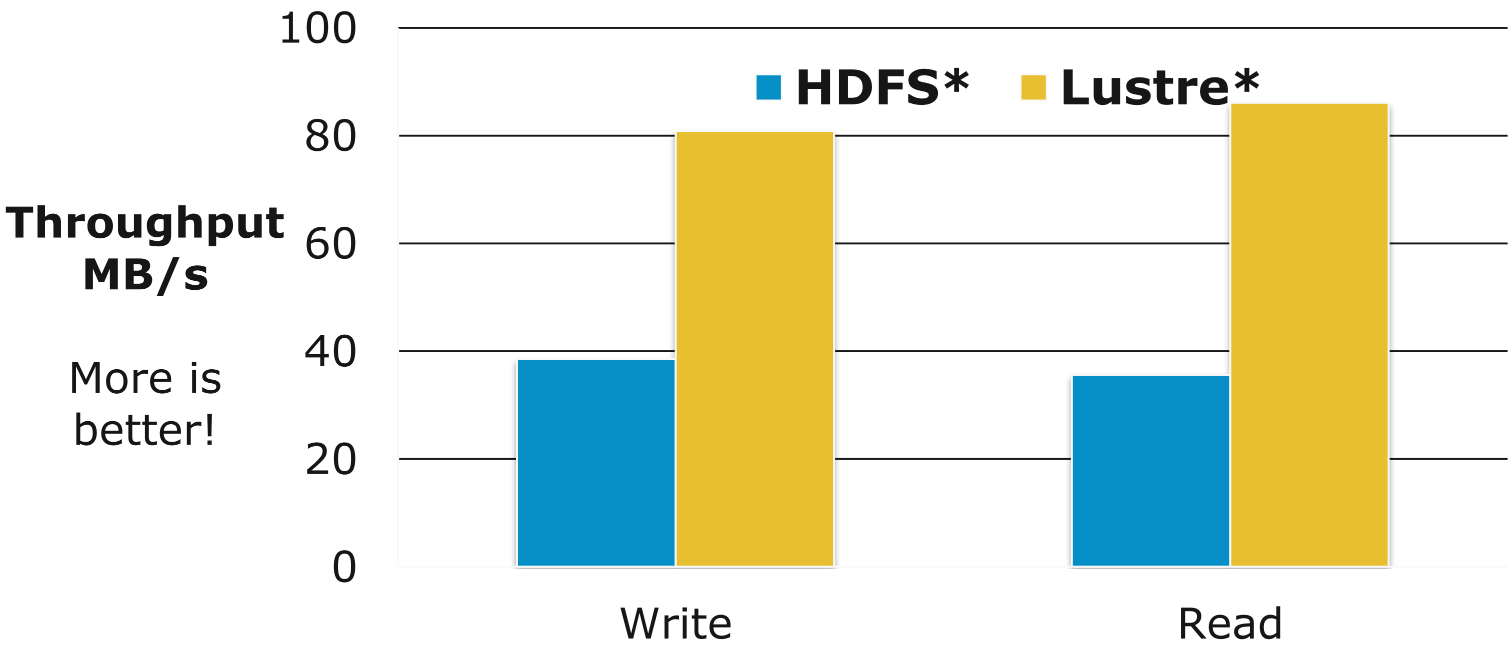
- Based on the new Hadoop* architecture AKA MapReduce NextGen based on Apache* Hadoop 2.0.4
- Packaged as a single Java* library (JAR)
  - Classes for accessing data on Lustre* in a Hadoop compliant manner. Users can configure Lustre Striping.
  - Classes for "Null Shuffle", i.e., shuffle with zero-copy
- Easily deployable with minimal changes in Hadoop configuration
- No change in the way jobs are submitted
- Part of Intel® Enterprise Edition for Lustre Software 1.0 and Intel® Distribution of Apache Hadoop Software 3.0

Performance Tests

- Standard Hadoop* benchmarks were run on the Rosso cluster
- Configuration – Intel® Distribution of Apache Hadoop* Software v2.0.4:
  - 8 nodes, 2 SATA disks per node (used only for HDFS*)
  - One with dual configuration, i.e., master and slave
- Configuration – Lustre* (v2.3.0):
  - 4 OSS nodes, 4 SATA disks per node (OSTs)
  - 1 MDS, 4GB SSD MDT
  - All storage handled by Lustre, local disks not used

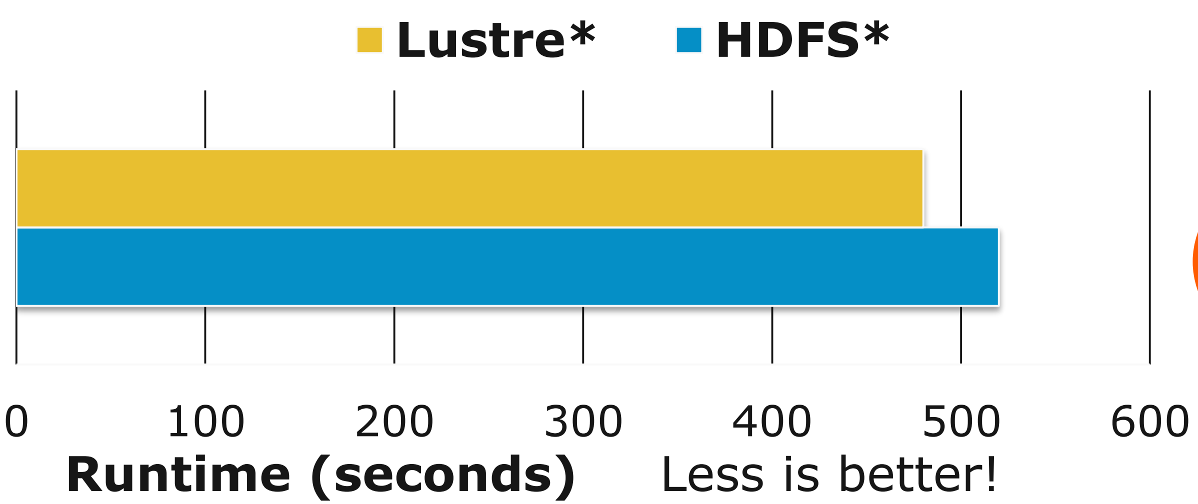
TestDFSIO Benchmark

- Tests the raw performance of a file system
- Write and read very large files (35G each) in parallel
- One mapper per file. Single reducer to collect stats.
- Embarrassingly parallel, does not test shuffle & sort



Terasort Benchmark

- Distributed sort: The primary Map-Reduce primitive
- Sort a 1 Billion records, i.e., approximately 100G
  - Record: Randomly generated 10 byte key + 90 bytes garbage data
- Terasort only supplies a custom partitioner for keys, the rest is just default map-reduce behavior
- Block Size: 128M, Maps: 752 @ 4/node, Reduces: 16 @ 2/node



Lustre ~10% Faster

Further Work

- Test at scale (100+ Nodes): Verify that large scale jobs don't throttle MDS
- Scenarios with other tools in the Hadoop* Stack: Hive*, Hbase*, etc.
- Introduce locality and run mappers/reducers directly on OSS nodes
- Experiment with caching (e.g., Lustre* with ZFS)

Agenda

- Hadoop* and HPC: State of the Union
- Enabling MapReduce on Slurm
- Enabling MapReduce on Lustre*
- Summary and Q&A

Summary

- Intel is working to enable leveraging of existing HPC resources for Hadoop*
  - Integrating Hadoop to Lustre*
- Full support for Hadoop 2.0
  - No change to Hadoop applications
  - Support use of MPI-based libraries by Hadoop

Q&A

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.
A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.
Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.
The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel, Look Inside and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the third quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “believes,” “seeks,” “estimates,” “may,” “will,” “should” and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel’s actual results, and variances from Intel’s current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the company’s expectations. Demand could be different from Intel's expectations due to factors including changes in business and economic conditions; customer acceptance of Intel’s and competitors’ products; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Uncertainty in global economic and financial conditions poses a risk that consumers and businesses may defer purchases in response to negative financial events, which could negatively affect product demand and other related matters. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel’s response to such actions; and Intel’s ability to respond quickly to technological developments and to incorporate new features into its products. The gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; start-up costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; product manufacturing quality/yields; and impairments of long-lived assets, including manufacturing, assembly/test and intangible assets. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. Intel’s results could be affected by the timing of closing of acquisitions and divestitures. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues, such as the litigation and regulatory matters described in Intel's SEC reports. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel’s ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. A detailed discussion of these and other factors that could affect Intel’s results is included in Intel’s SEC filings, including the company’s most recent reports on Form 10-Q, Form 10-K and earnings release.