AMD

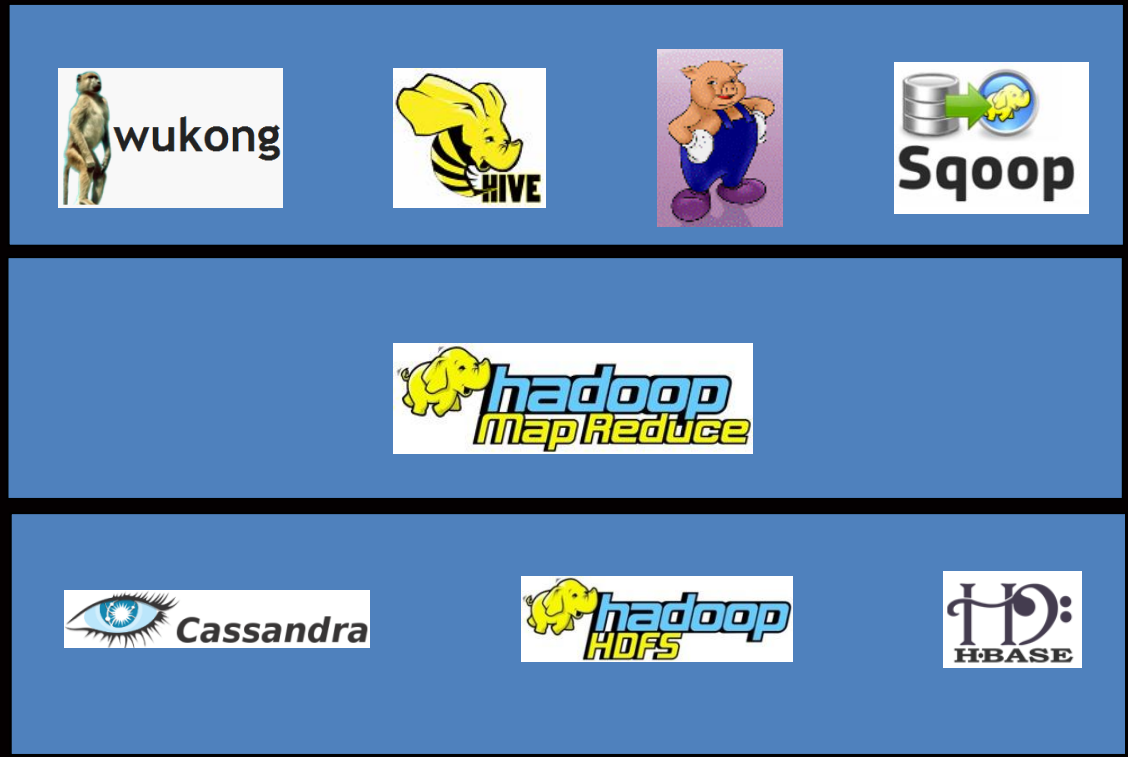
Fusion[11]

DEVELOPER SUMMIT

Leveraging Multicore Systems for Hadoop and HPC Workloads

Jim Falgout | Pervasive Software | DataRush Chief Technologist

Evolving Hadoop Stack



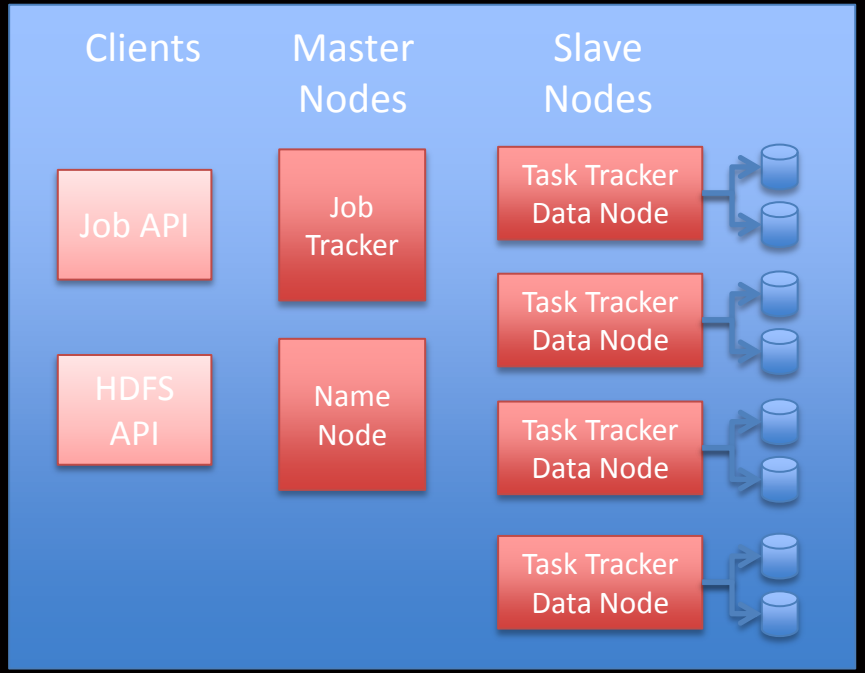
MapReduce & HDFS Architecture

Job Tracker
- Schedules and manages jobs
- Splits input files
- Manages shuffle of map output

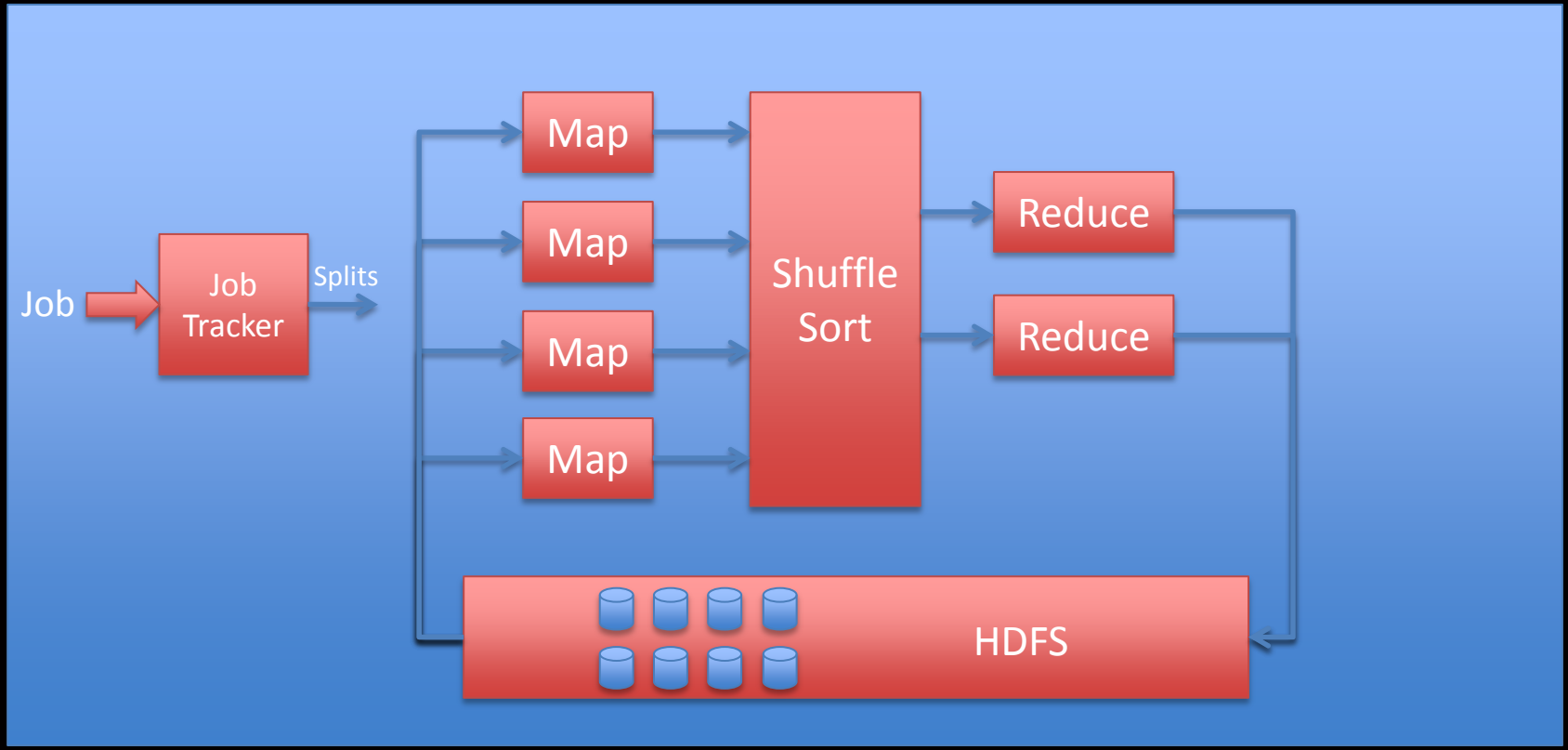
Name Node
- Manages HDFS name space
- Manages data replication

Task Tracker
- Manages tasks local to worker node

Data Node
- Manages local space on worker node
- Handles data transfers



Map/Reduce Job Flow



MapReduce Parts

Mapper
- User provided map function: maps input key-value pairs into output key-value pairs

Reducer
- User provided reduce function: aggregates input key-value sets into output key-value pairs

Input/Output Formats
- Determines how to split input files; how to write output files

Partitioner
- Partition function that determines how to distribute output of mapper to reducer(s)

Comparator
- Compares key values for sorting

Serialize/Deserialize
- Serialize key/value data for storage and network transfer
- Deserialize for injection into reducer function

The Need for Other Compute Paradigms

"… in addition, these data stores often expose a proprietary interface for application programming (e.g. PL/SQL or TSQL), but not the full power of procedural programming. More programmer-friendly parallel dataflow languages await discovery, I think. MapReduce is one (small) step in that direction."

Engineer-to-Engineer Lectures

Jeff Hammerbacher

June 2010

Dataflow is:

Based on operators that provide a specific function (nodes)

Data queues (edges) connecting operators

Composition of directed, acyclic graphs (DAG)

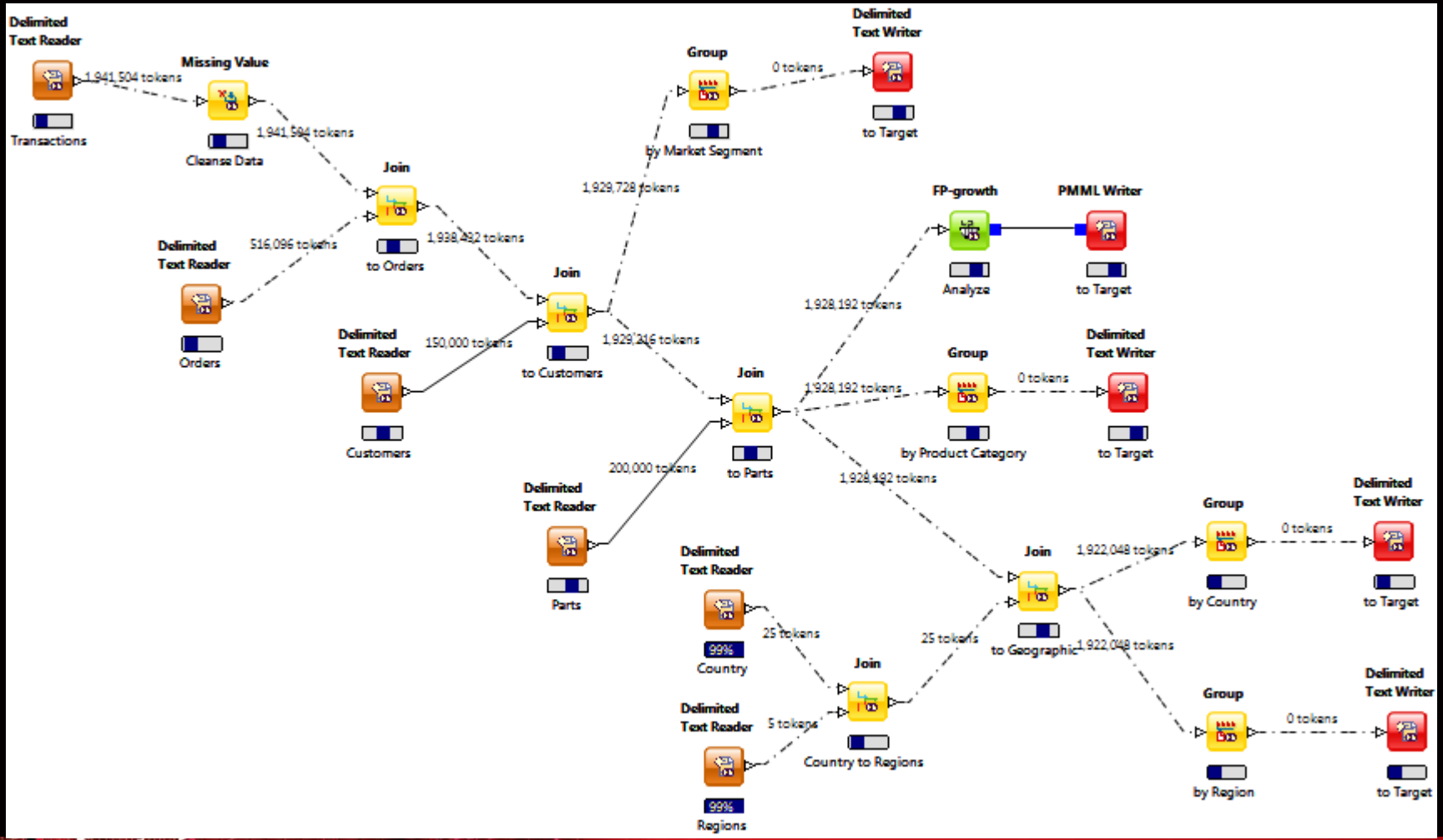
- Operators connected via queues
- A graph instance represents a “program” or “application”

Flow control

Scheduling to prevent dead locks

Focused on data parallelism

Implemented by the Pervasive DataRush engine


Delimited Text Reader
Transactions
1,941,504 tokens
Missing Value
Cleanse Data
1,941,504 tokens
Delimited Text Reader
Orders
516,096 tokens
Join
to Orders
1,938,492 tokens
Delimited Text Reader
Customers
150,000 tokens
Join
to Customers
1,929,728 tokens
1,929,216 tokens
Group
by Market Segment
0 tokens
Delimited Text Writer
to Target
FP-growth
Analyze
PMML Writer
to Target
1,928,192 tokens
Join
to Parts
1,928,192 tokens
Group
by Product Category
0 tokens
Delimited Text Writer
to Target
1,928,192 tokens
200,000 tokens
Delimited Text Reader
Parts
Delimited Text Reader
99%
Country
25 tokens
Delimited Text Reader
99%
Regions
5 tokens
Join
Country to Regions
25 tokens
Join
to Geographic
1,922,048 tokens
Group
by Country
0 tokens
Delimited Text Writer
to Target
1,922,048 tokens
Group
by Region
0 tokens
Delimited Text Writer
to Target

Dataflow and GPU's

DataRush is 100% Java

- No explicit GPU support other than Aparapi from AMD (alpha)
- Under consideration for our roadmap

Usage Scenarios

- Data mining algorithms
  - Many are floating point intensive
- Next Generation Sequencing (NGS)
  - Smith-Waterman
  - BFast

Is it a fit?

- Seems to be: use DataRush for I/O and data preparation
- Invoke GPU's for computationally intensive functions

Dataflow goodness:

Concepts are easy to grasp
Abstracts parallelism details
Simple to express
- Composition based

Shared nothing, message passing
- Simplified programming model

Immutability of flows
Limits side effects
Functional style

Code Snippet

```
public final class StringLengthProcess extends DataflowProcess {
    private final StringInput input;
    private final IntOutput output;

    public StringLengthProcess(CompositionContext ctx, ScalarFlow input) {
        super(ctx);

        this.input = newStringInput(input);
        this.output = newIntOutput();
    }

    public ScalarFlow getOutput() {
        return getFlow(output);
    }

    @Override
    public void execute() {
        while (input.stepNext()) {
            output.push(input.asString().length());
         }
        output.pushEndOfData();
    }
}
```

Dataflow and big data

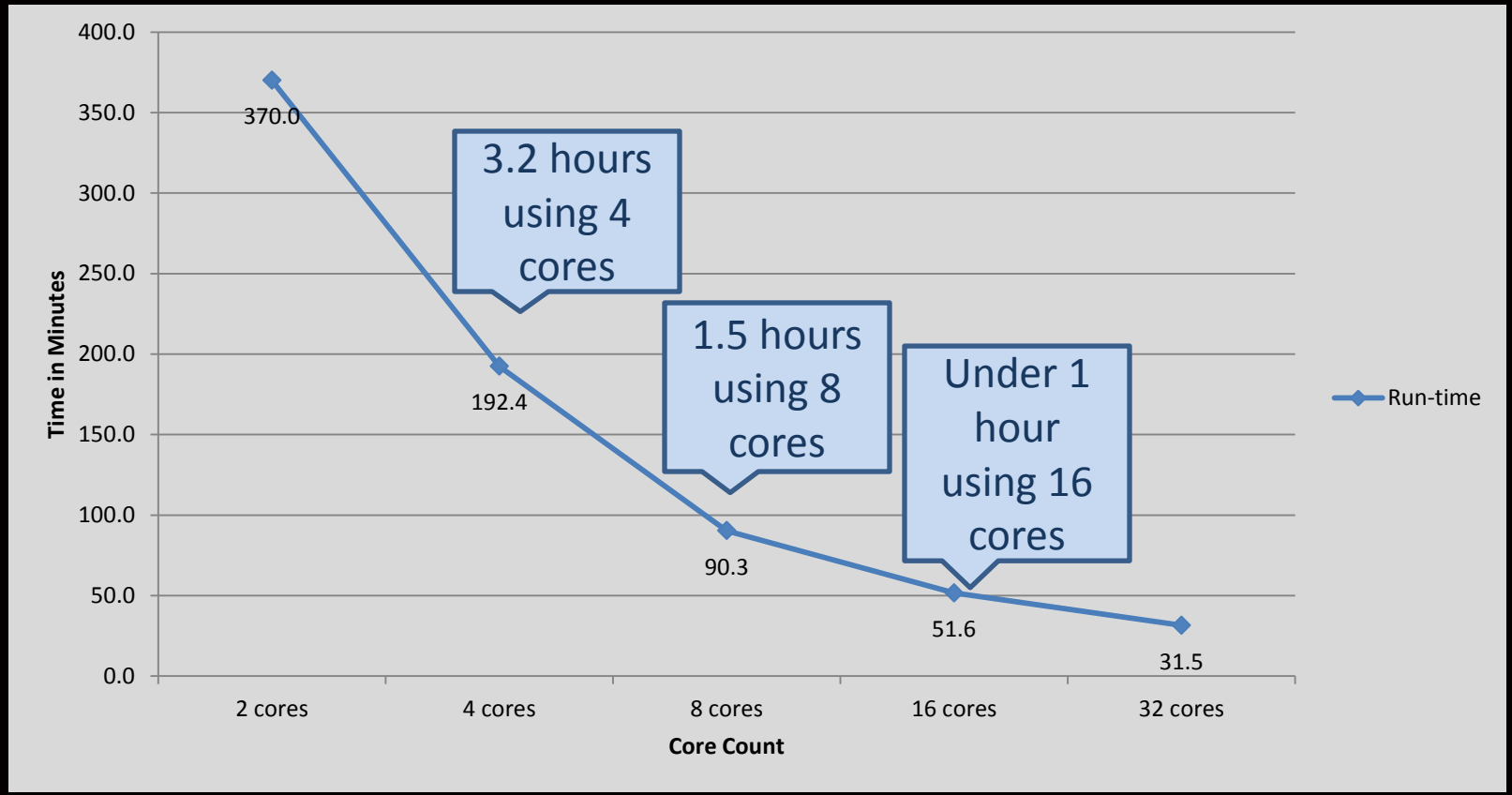
Pipelining
- Pipeline task based parallelism
- Overlap I/O and computation
- Can help optimize processor cache
- Whole application approach

Data scalable
- Virtually unlimited data size capacity
- Supports iterative data access

Exploits multicore
- Scalable
- High data throughput

Extendible to multi-node

Malstone-B10 Scalability



Multi-node DataRush

Extending dataflow to multi-node

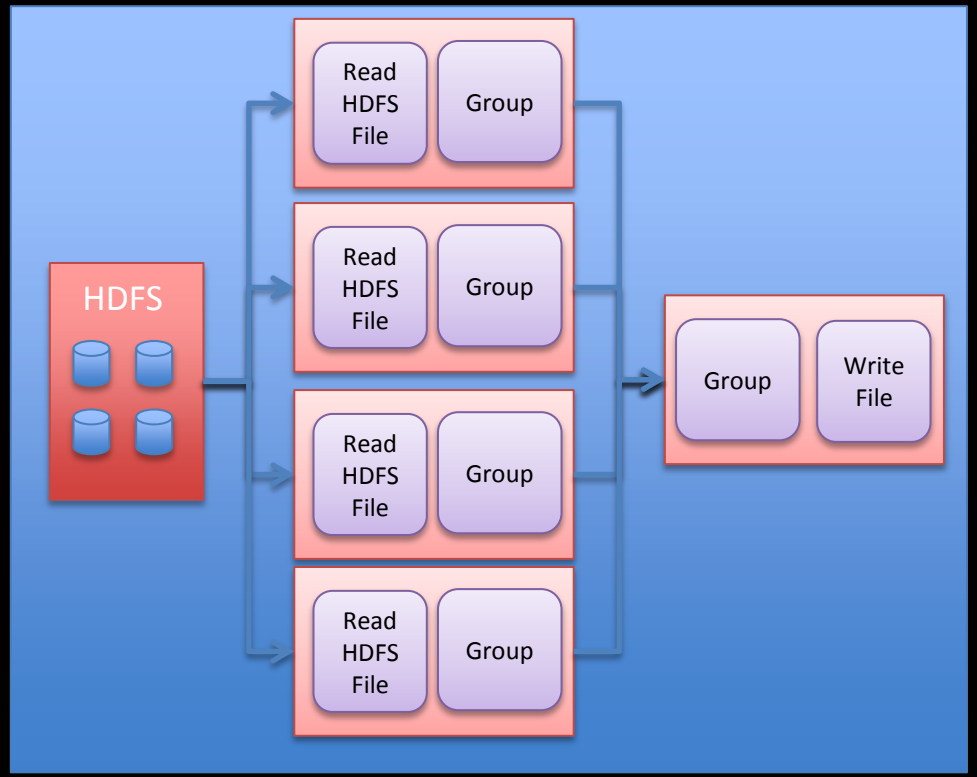
- Execute distributed graph fragments
- Fragments linked via socket-based queues
- Use distributed application graph

Specific patterns supported

- Scatter
- Gather
- Scatter-gather combined

Multi-node DataRush Example

Uses gather pattern from HDFS files
Reads file(s) from HDFS
Initial aggregation in distributed phase
Final aggregation with data from all nodes



Areas of Application

Bioinformatics
- Next Generation Sequencing
- Nearly 1 TCUP throughput using Smith Waterman
- Scalable BFAST implementation

Telecom
- Analyzing Call Data Records (network telemetry)
- Operational intelligence
- Fraud and waste detection

Public Sector
- State income tax revenue recovery
- Cyber security

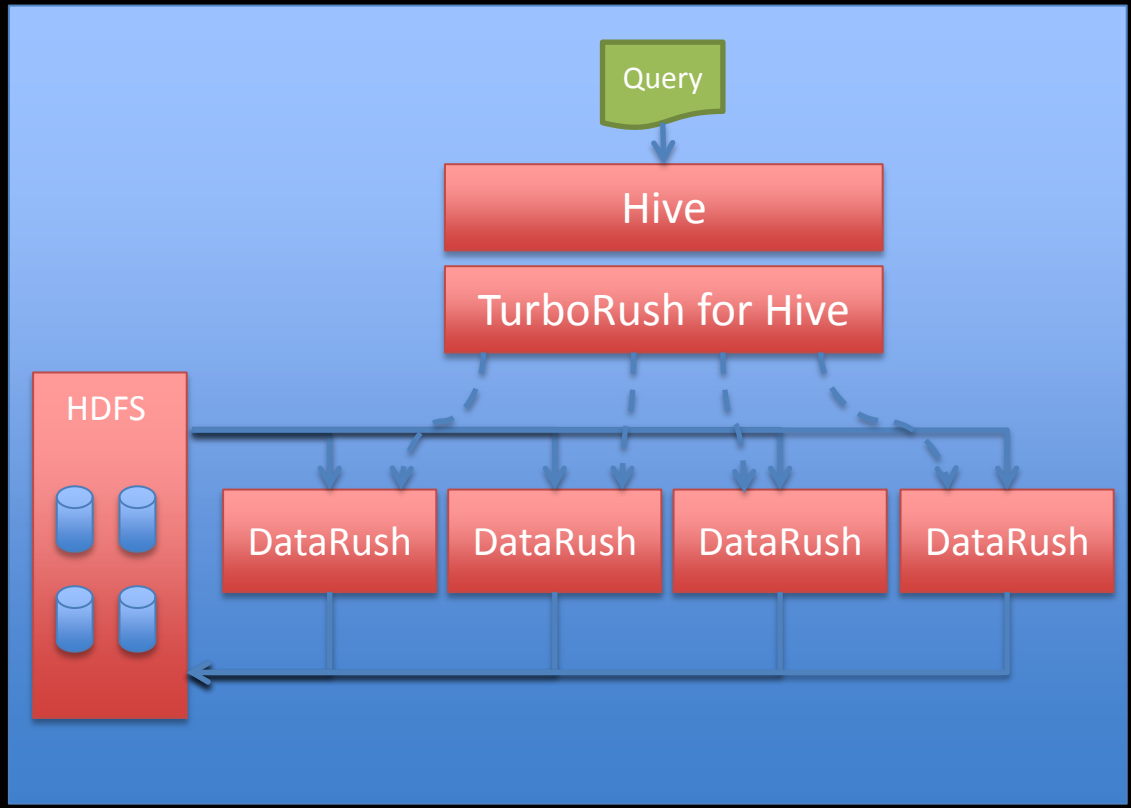
Financial Services
- Mortgage analysis

Healthcare
- Claims processing and analysis
- Fraud detection

Network
- Analyzing network log data
- Cyber security

Adding Dataflow to Hive

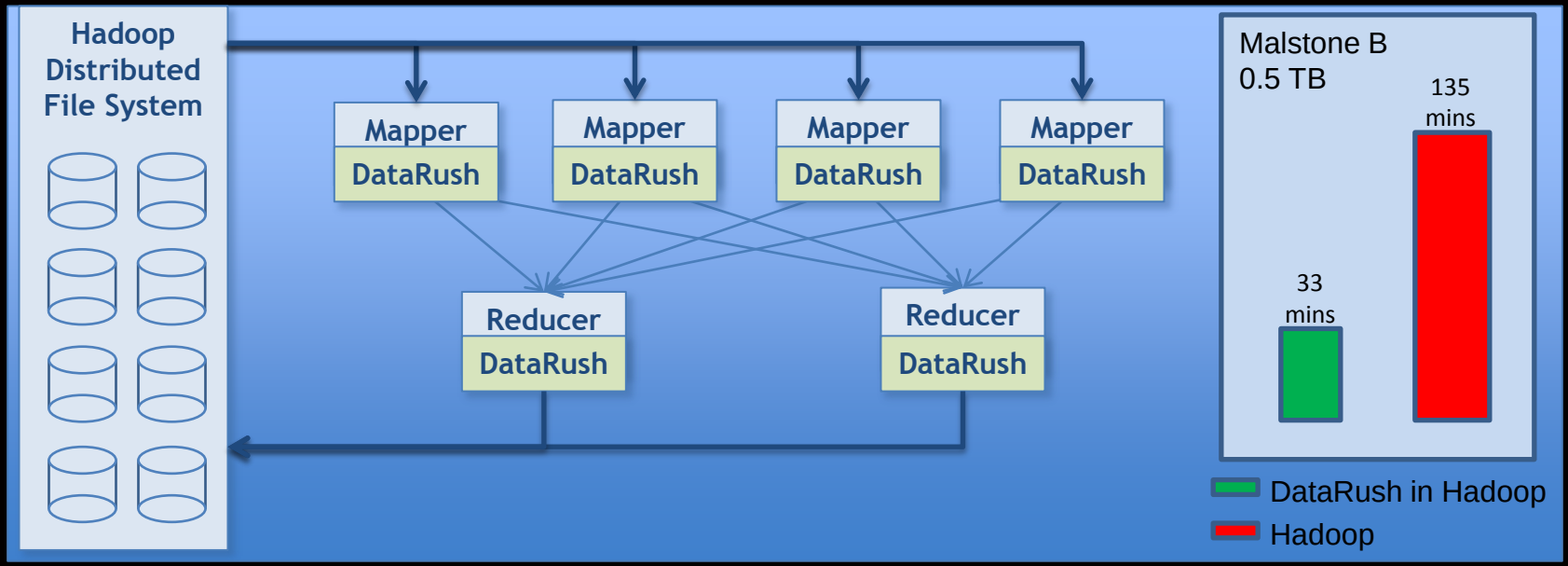
- Extension to Hive allows executing queries using the dataflow based DataRush framework
- Queries source data from HDFS
- Testing with TPC-H data and queries has show a significant performance boost



Adding DataRush to MapReduce

DataRush embedded within Hadoop

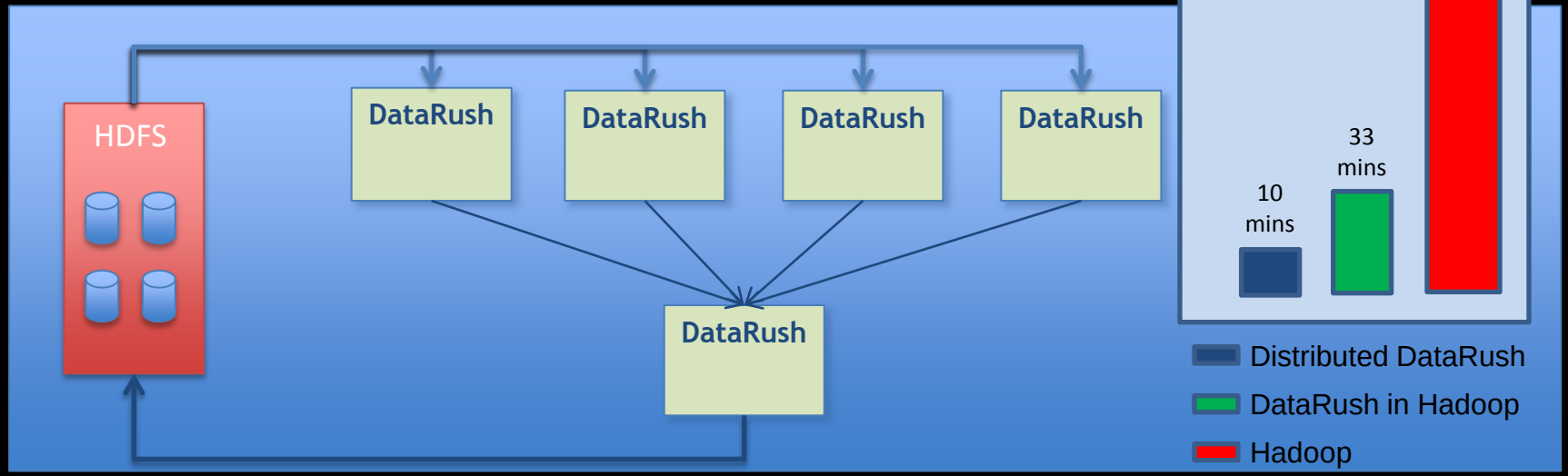
- Reduce complexities of MapReduce experience
- Increased efficiencies can lead to significantly faster run times



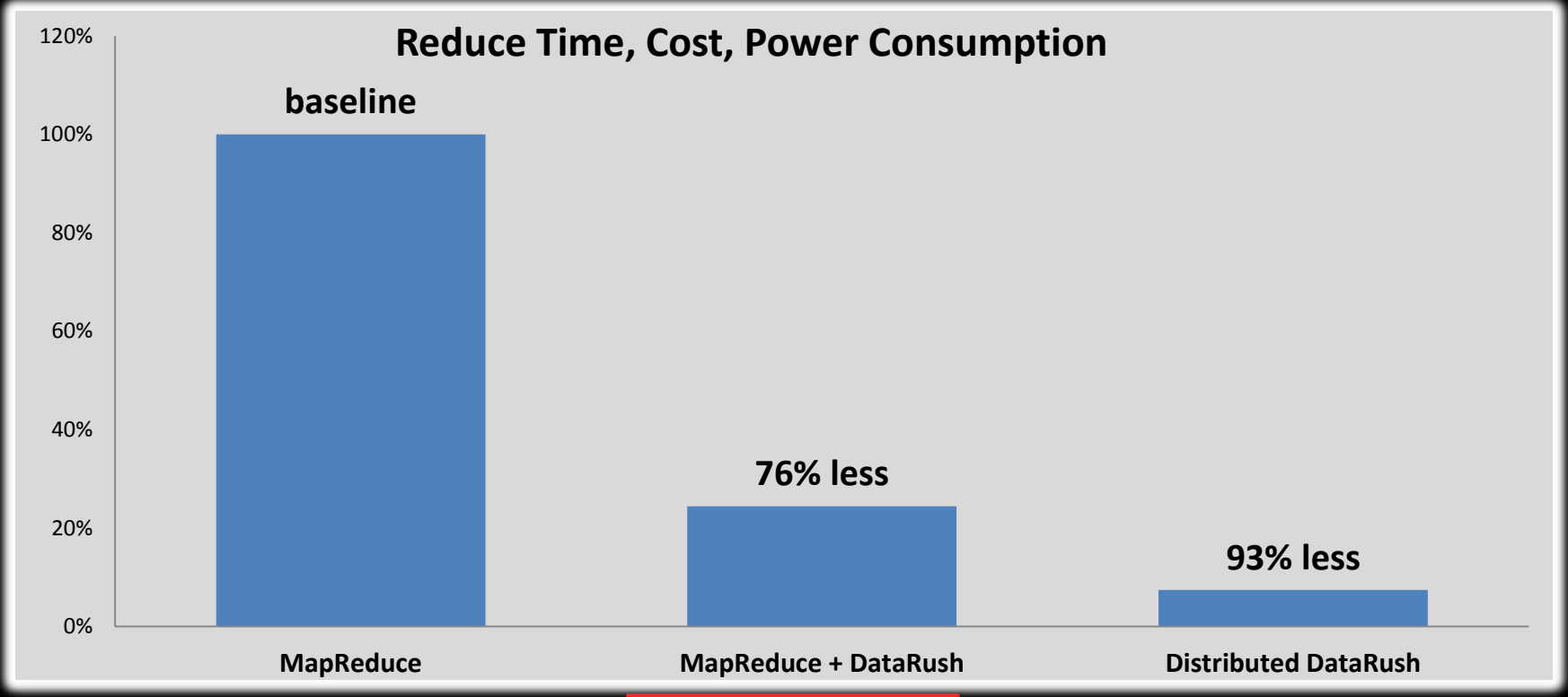
Distributed Dataflow With HDFS

Access existing files in Hadoop file system (HDFS) using dataflow readers/writers

- Utilize existing data with fewer resources
- Distributed or single-node

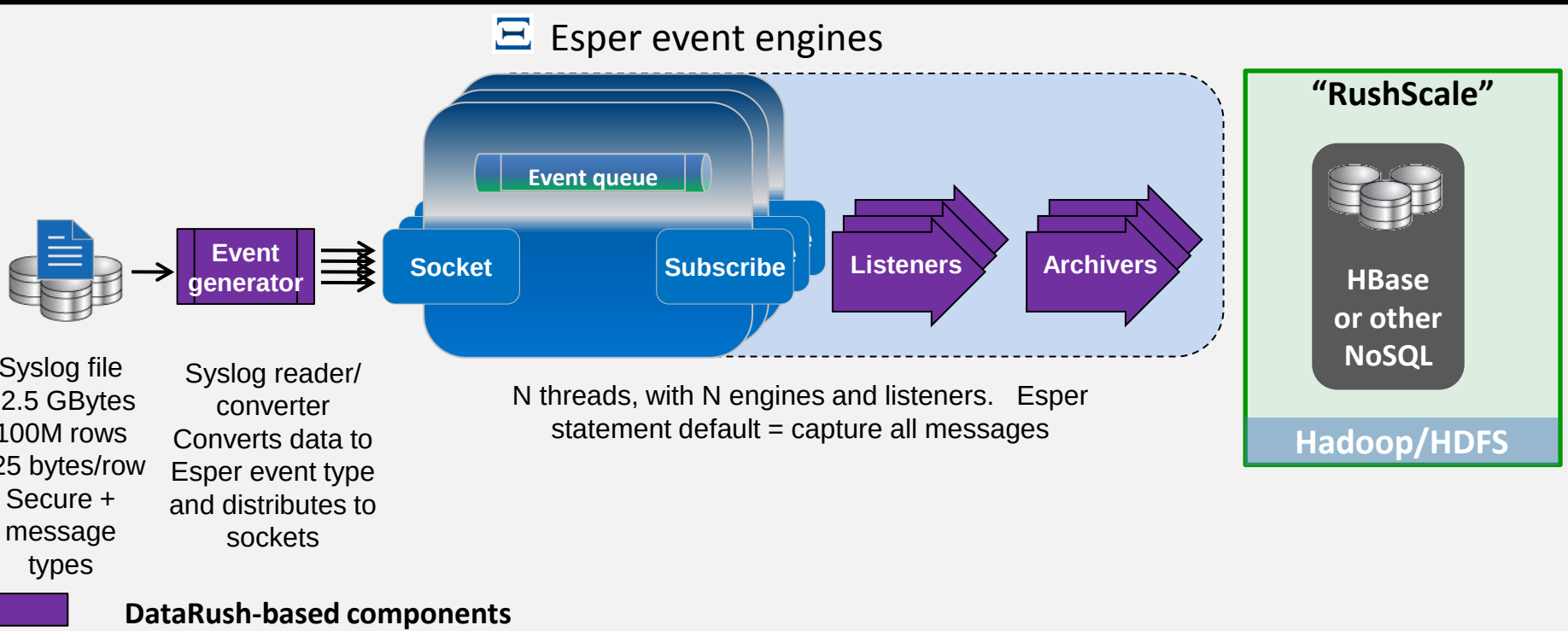


Malstone-B10 Benchmark Summary
Web log file analysis



10 node cluster
8 cores per node

Scalable Event Archiving



Summary

Hadoop

- Set of open source projects from Apache for big data handling & processing
- Complex to learn (MapReduce)
- Not optimal for small (Terabytes) work loads
- Scales out to enormous workloads; thousands of nodes, Petabytes of data

Dataflow

- Scalable, data focused architecture
- Easy to grasp and simple to express
- Simple programming model
- Enables full utilization of multicore and multi-node resources
- Integrated into MapReduce and Hive

Disclaimer & Attribution

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. There is no obligation to update or otherwise correct or revise this information. However, we reserve the right to revise this information and to make changes from time to time to the content hereof without obligation to notify any person of such revisions or changes.

NO REPRESENTATIONS OR WARRANTIES ARE MADE WITH RESPECT TO THE CONTENTS HEREOF AND NO RESPONSIBILITY IS ASSUMED FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT WILL ANY LIABILITY TO ANY PERSON BE INCURRED FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. All other names used in this presentation are for informational purposes only and may be trademarks of their respective owners.

The contents of this presentation were provided by individual(s) and/or company listed on the title page. The information and opinions presented in this presentation may not represent AMD's positions, strategies or opinions. Unless explicitly stated, AMD is not responsible for the content herein and no endorsements are implied.

Jim Falgout jfalgout@pervasive.com
Joe Dubin jdubin@pervasive.com

http://www.pervasivedatarush.com

Stop by the Pervasive DataRush demo pod in the Experience Zone.

We're hiring!