IBM Analytics

The Evolution of Big Data Platforms and Data Science

ECC Conference 2016

Brandon MacKenzie

June 13, 2016

Hello,

I’m Brandon MacKenzie. I work at IBM.

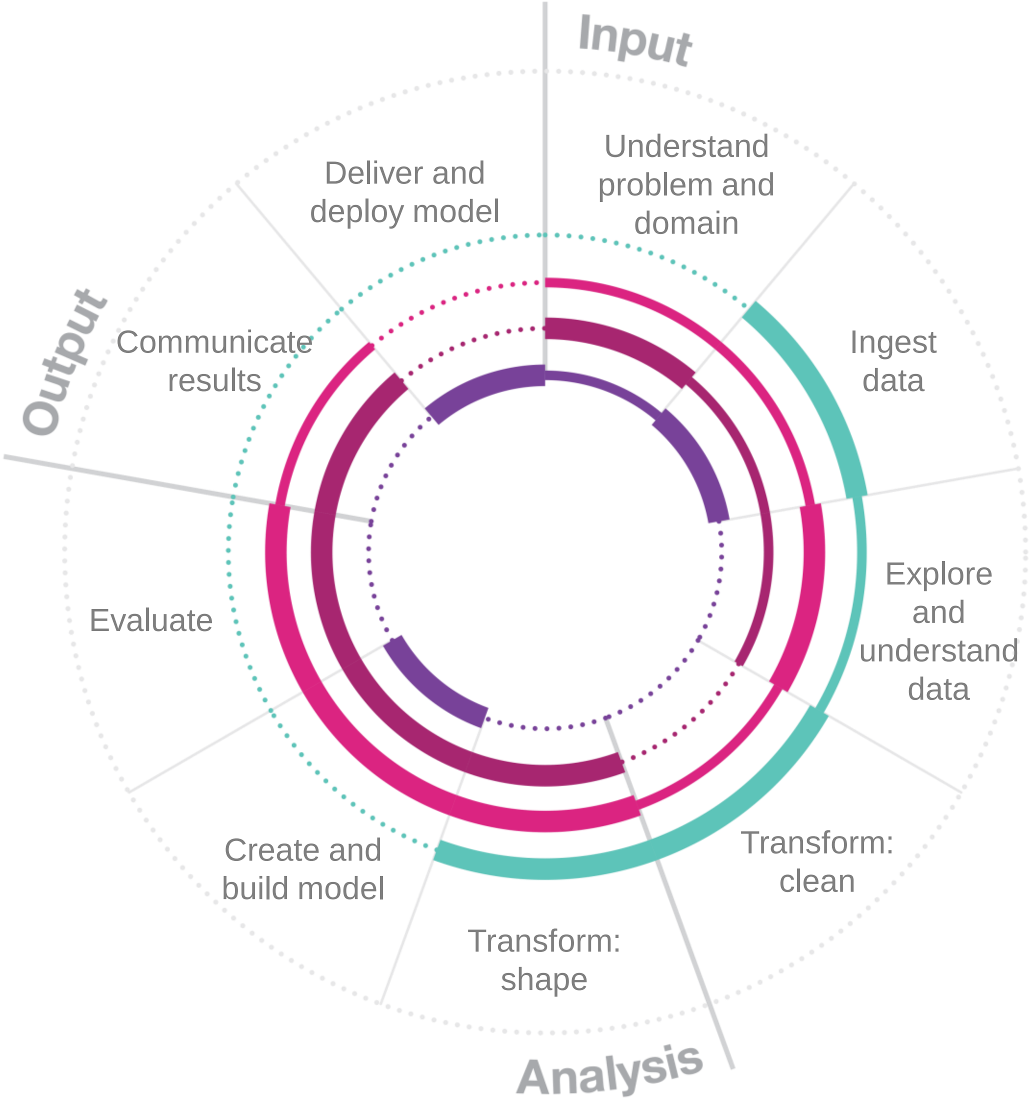
→ **Data Science** - Offering Management & Worldwide Technical Sales

→ **Advisory Member** - Spark Technology Center Advisory Council

→ **Co-author** - IBM Apache Spark Professional Certification Program

→ **MSc** - University of Edinburgh

Data volume, variety, and velocity change – but the goal of the Data Science remains the same



1

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

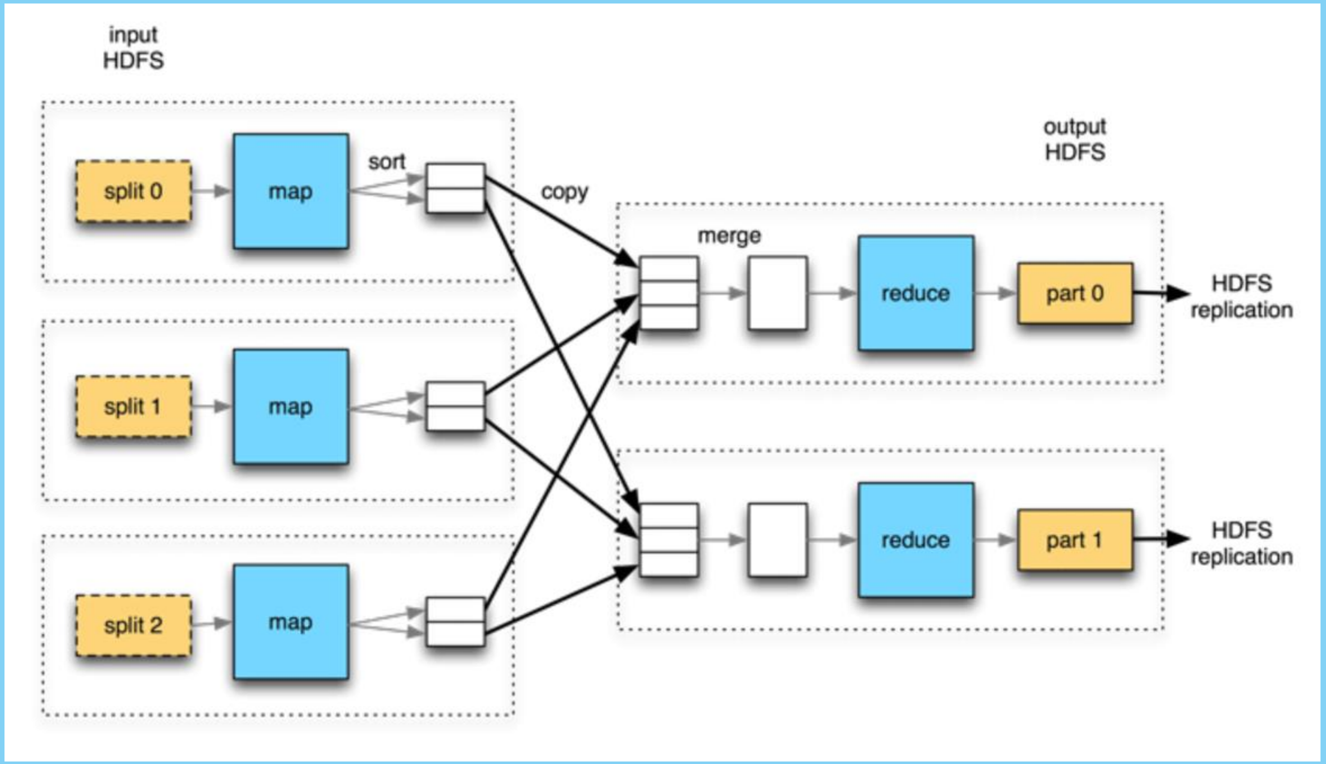
Our implementation of MapReduce runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of MapReduce pro-

given day, e
ally straight
large and the
hundreds or
a reasonable
allelize the c
failures cons
tation with l
these issues.

As a reac
abstraction t
tions we we
tails of para
and load ba
spired by the
and many o
most of our
eration to ea
compute a s
applying a *r*
the same ke
propriately.
specified ma
lelize large c

2

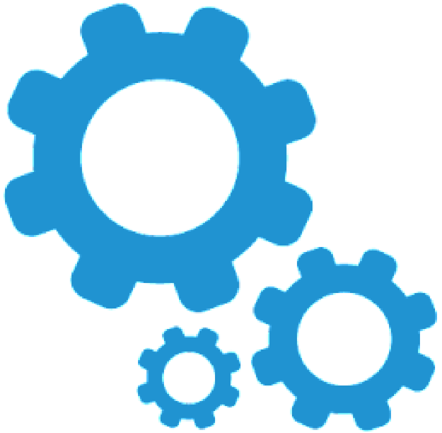
hadoop



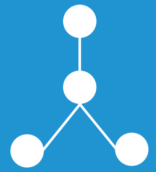
SQL: IBM SystemR 1970s

SQL Query:
SELECT…
FROM…
WHERE…

Optimizer

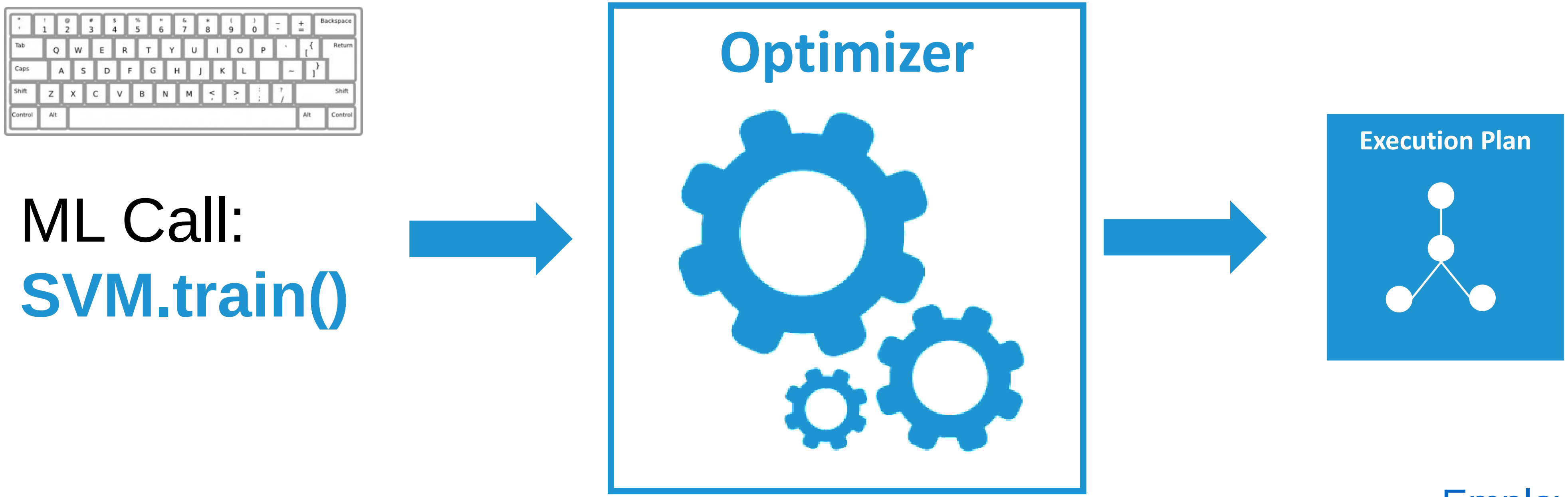


Execution Plan



Employee Skills ✓
Hardware ✓
Data ✓

Declarative Machine Learning: IBM SystemML 2000s



Employee Skills ✓

Hardware ✓

Data ✓

And then there was a movement in open source…

Apache Spark, finally, a chance to push declarative data science mainstream!

Apache Spark…"One of the most important open source projects in a <u>decade</u>"

- Significantly faster than MapReduce
- APIs for Python, R, Java, Scala
- Combine SQL, streaming, complex analytics
- Runs everywhere

World's largest investment in Apache Spark

Opened Spark Technology Center

3500 IBM Researchers and Developers to Spark

Apache Spark – Blends Multiple Data Types, Sources, & Workloads

Spark SQL	Spark Streaming	MLlib Machine Learning	Graph
Execute SQL Statements	Streaming Analytics via Micro-batch	M.L. and Statistical Algorithms	Distributed Graph Processing Framework

Spark Core

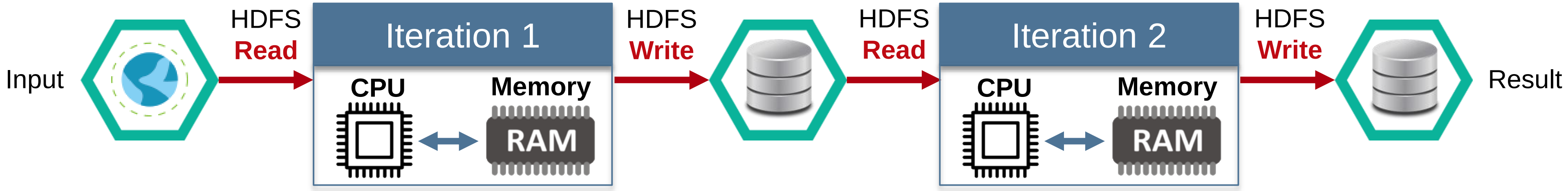
- General compute engine
- Basic I/O functions
- Task dispatching
- Scheduling

Data Sources

IBM Cloud	Public Cloud	Cloud Apps	On-Premises
BigInsights (HDFS)	amazon web services S3	NETSUITE	ORACLE
Cloudant (DBaaS)	Cassandra	salesforce	SAP
dashDB (Analytics)	mongoDB	CSV	IBM DB2
SQDB (Managed DB2)	redis	JDBC	
Swift (Object Storage)	rackspace	{JSON}	
	Microsoft Azure	Parquet	
	MySQL	elasticsearch	
	HIVE	AVRO	
	PostgreSQL		
	HDFS		
	dBase		
	APACHE HBASE		

Motivation for Apache Spark

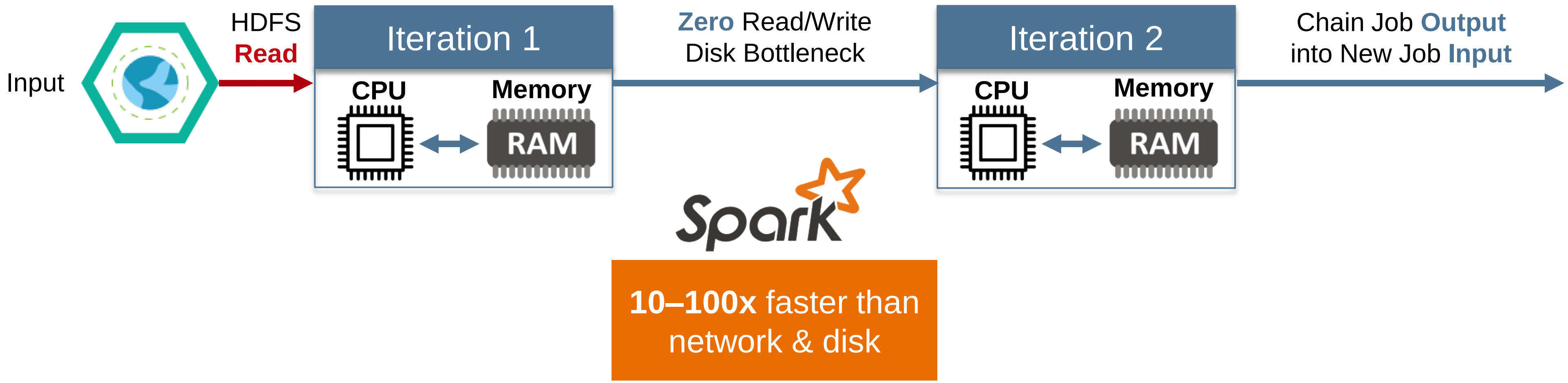
- **Traditional Approach**: MapReduce jobs for complex jobs, interactive query, and online event-hub processing involves lots of (slow) disk I/O



Motivation for Apache Spark

- **Traditional Approach**: MapReduce jobs for complex jobs, interactive query, and online event-hub processing involves lots of (slow) disk I/O

- **Solution**: Keep more data in-memory with a new distributed execution engine



Comparing MapReduce and Spark

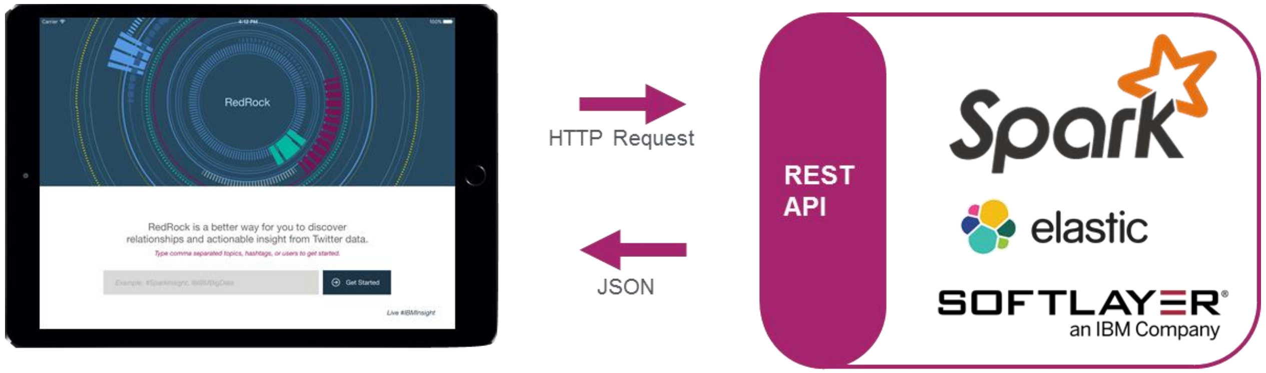
Spark	Spark	Hadoop MapReduce
Storage	No built-in storage (in-memory)	On-disk only
Operations	Map, Reduce, Join, Sample, etc.	Map and Reduce
Execution Model	Batch, interactive, streaming	Batch only
Programming Environments	Python, Scala, Java, and R	Java only

- **Spark is a unified platform for data integration**
  → Contrast with the many distinct distributions & tools for Hadoop
- **Spark follows lazy evaluation of execution graphs**
  → Optimizes jobs, reduces wait states, and allows easier pipelining of tasks
- **Spark lowers the resource overhead for starting or shuffling jobs**
  → Less expensive than MapReduce

Infusing Analytics Applications with Apache Spark

Spark is a unified framework for building analytics applications

- Performant Architecture
- Productive Workflows
- Leverages Existing Investments
- Continually Improving



Client Side

Server Side

Project RedRock

https://github.com/SparkTC/redrock

http://www.spark.tc/project-redrock-design-data/

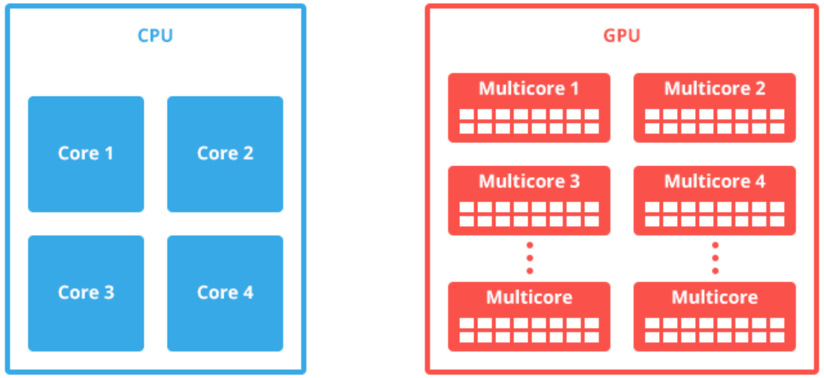
IBM donates SystemML to open source to strengthen Spark machine learning

IBM Research

THE APACHE SOFTWARE FOUNDATION

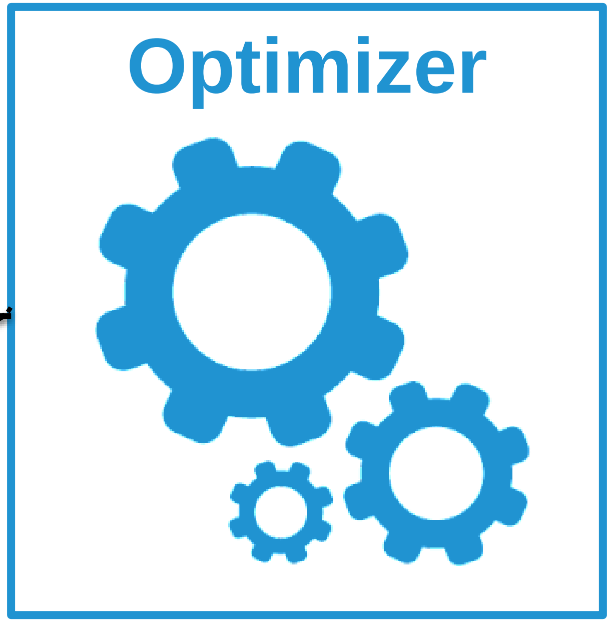
Apache SystemML
systemml.apache.org

Cost Models for Heterogeneous Architectures

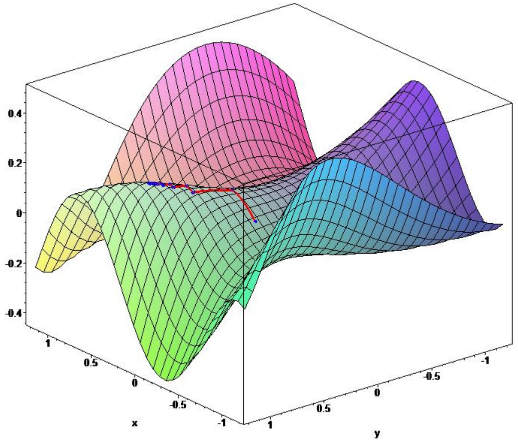


Improving Scala

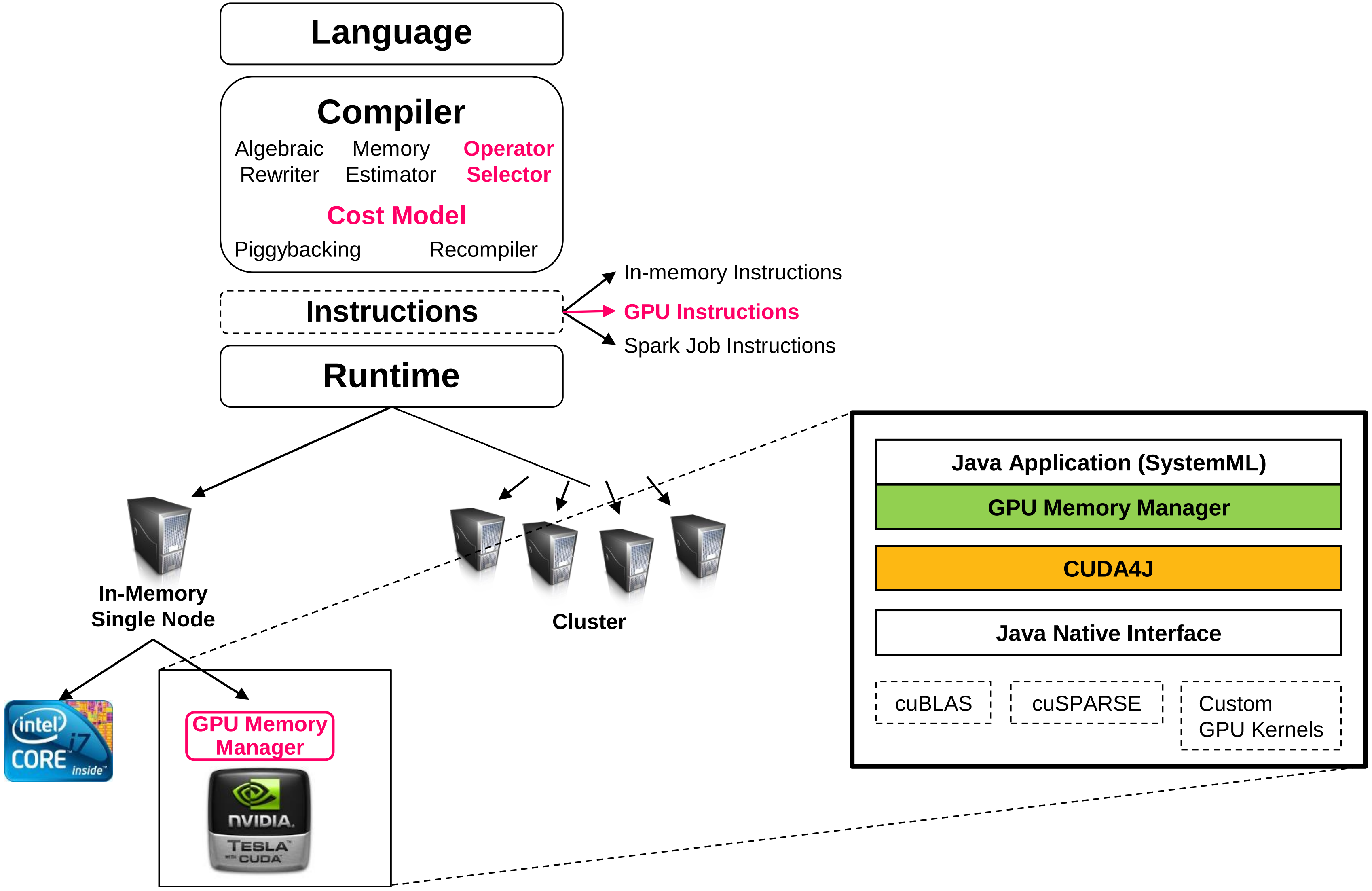
scalacenter



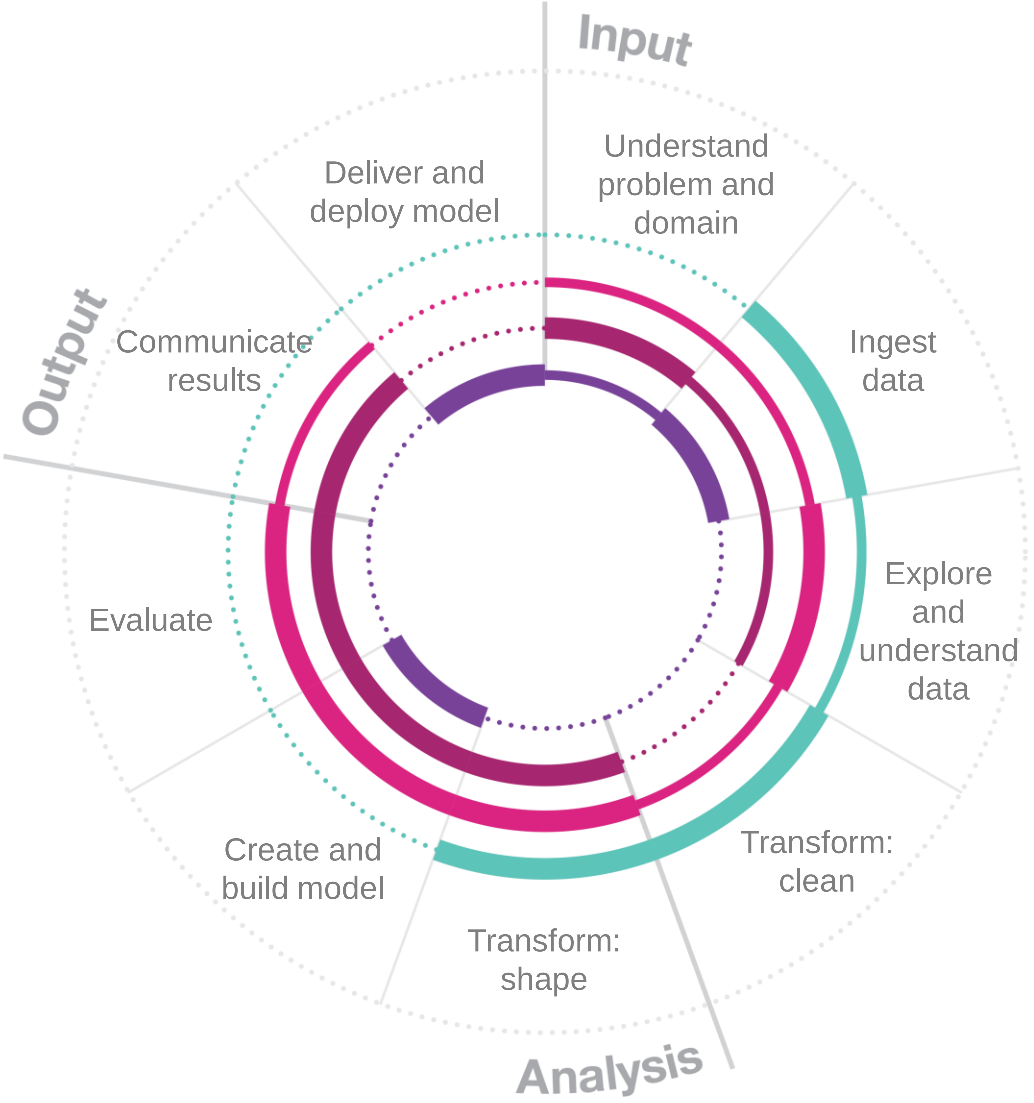
Automating Model Selection via Hyperparameter Heuristics



Example of how GPU Acceleration can be added to SystemML Optimizer



All of this innovation in Big Data Platforms will continue to flow into **OPTIMIZERs** so that Data Scientists can focus on being Data Scientists



So, where can I Learn More?

1. Spark Technology Center
   spark.tc

2. Apache SystemML
   systemml.apache.org

3. IBM Data Science Experience
   datascience.ibm.com

IBM®

For Apache Spark news and innovation from IBM's Spark Technology Center —

{spark.tc}

IBM | Apache Spark

Power of data. Simplicity of design. Speed of innovation.