Power Pig with Spark

Kelly Zhang (liyun.zhang@intel.com)

Apache Big Data Europe 2016

Agenda

- **Background**
- Why Pig on Spark ?
- Design Architecture
- Benchmark
- Optimization
- Current Status & Future Work
- Q&A

Background


cloudera
Ask Bigger Questions


intel


UBER


SIGMOID


INMOBI
The Largest Independent Mobile Ad Network

Apache Pig

- Procedural scripting language
- Pig Latin: similar to sql
- Heavily used for ETL
- Schema / No schema data, Pig eats everything

Spark

- Faster
- Generality
- Easy of use

Agenda

- Background
- **Why Pig on Spark ?**
- Design Architecture
- Benchmark
- Optimization
- Current Status & Future Work
- Q&A

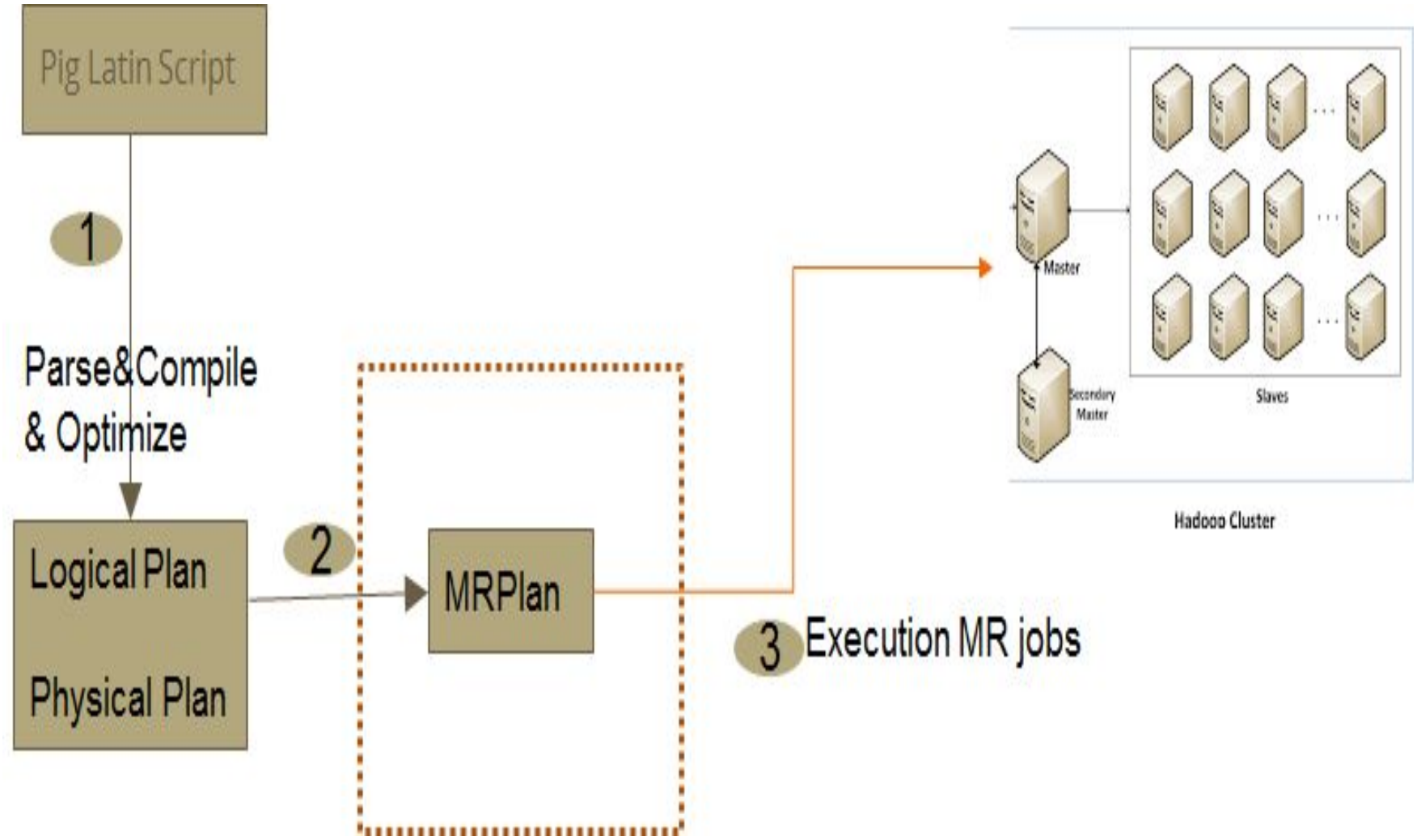
Why Pig on Spark

- Better Performance
  - No intermediate data between stages
  - In-memory caching abstraction
  - Executor JVM Reuse
- Support Pig users to experience Spark conveniently

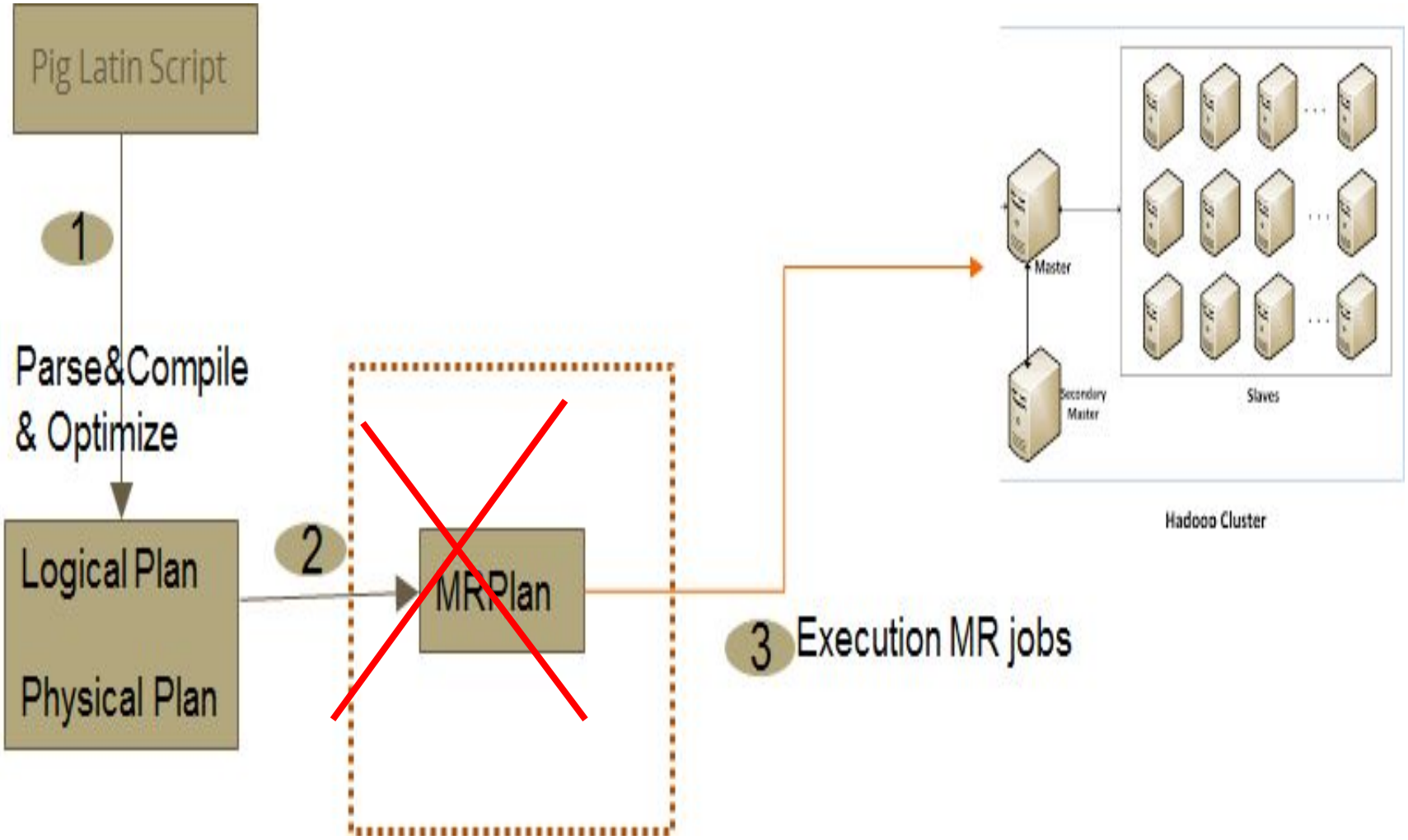
Agenda

- Background
- Why Pig on Spark ?
- **Design Architecture**
- Benchmark
- Optimization
- Current Status & Future Work
- Q&A

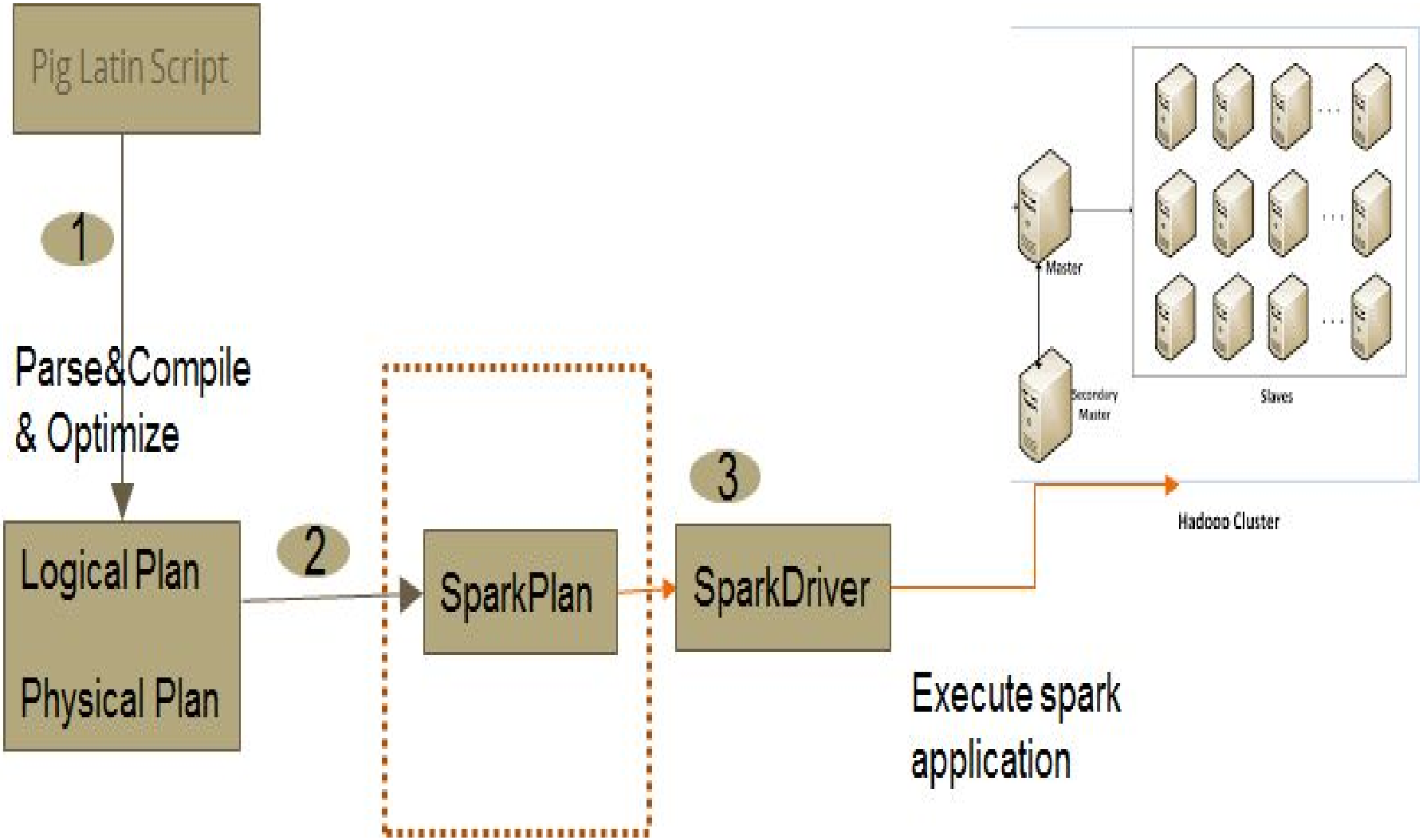
Design Architecture



Design Architecture

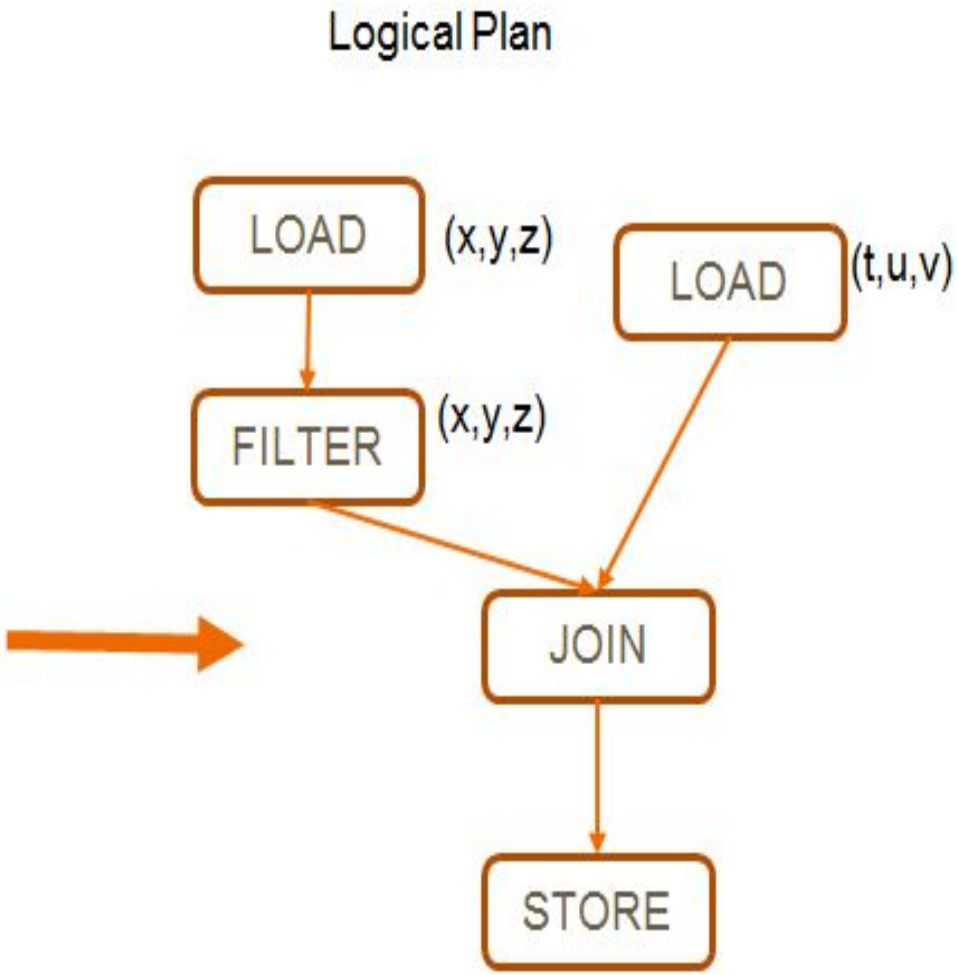


Design Architecture

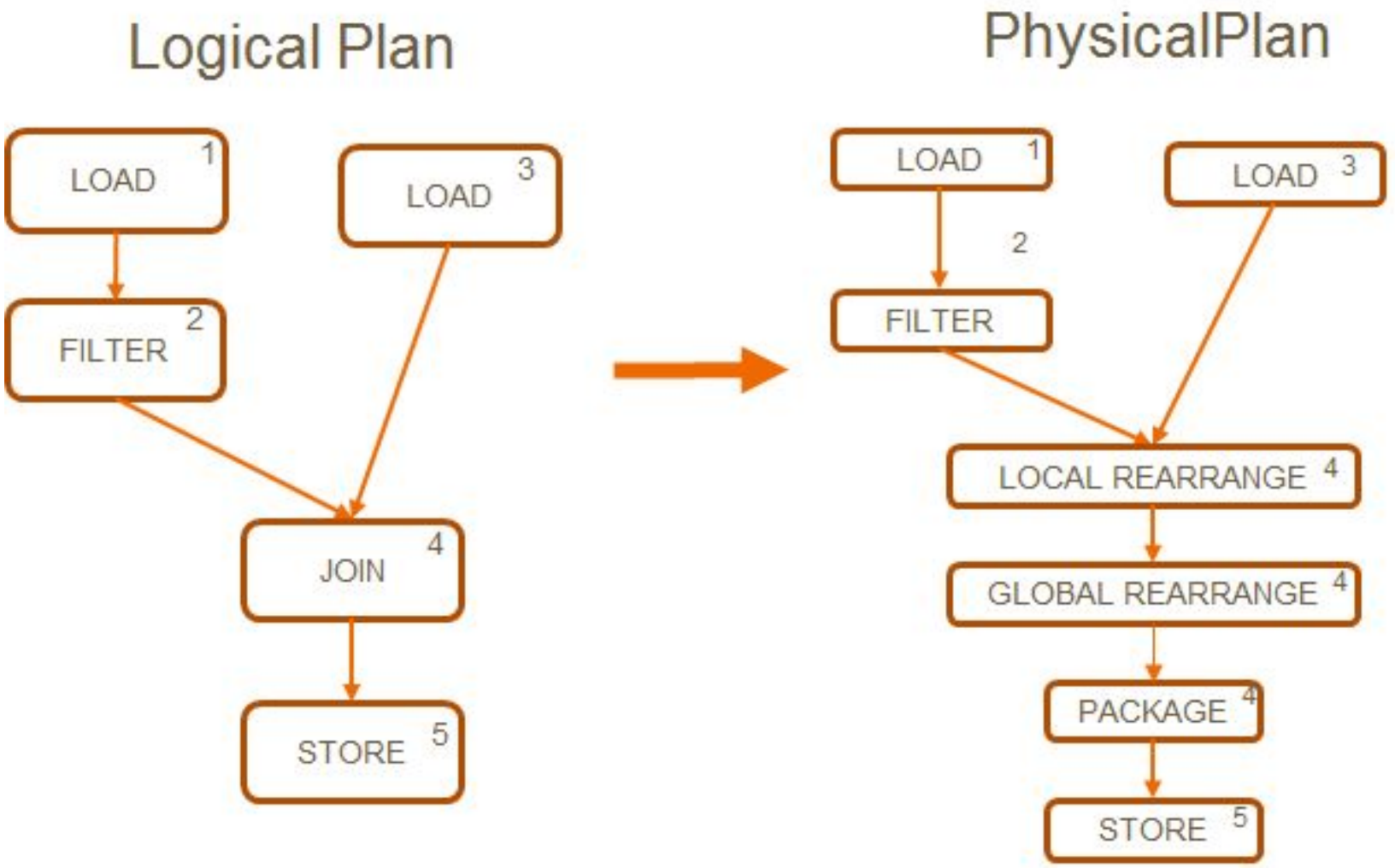


Pig Latin to RDD<Tuple> transformations

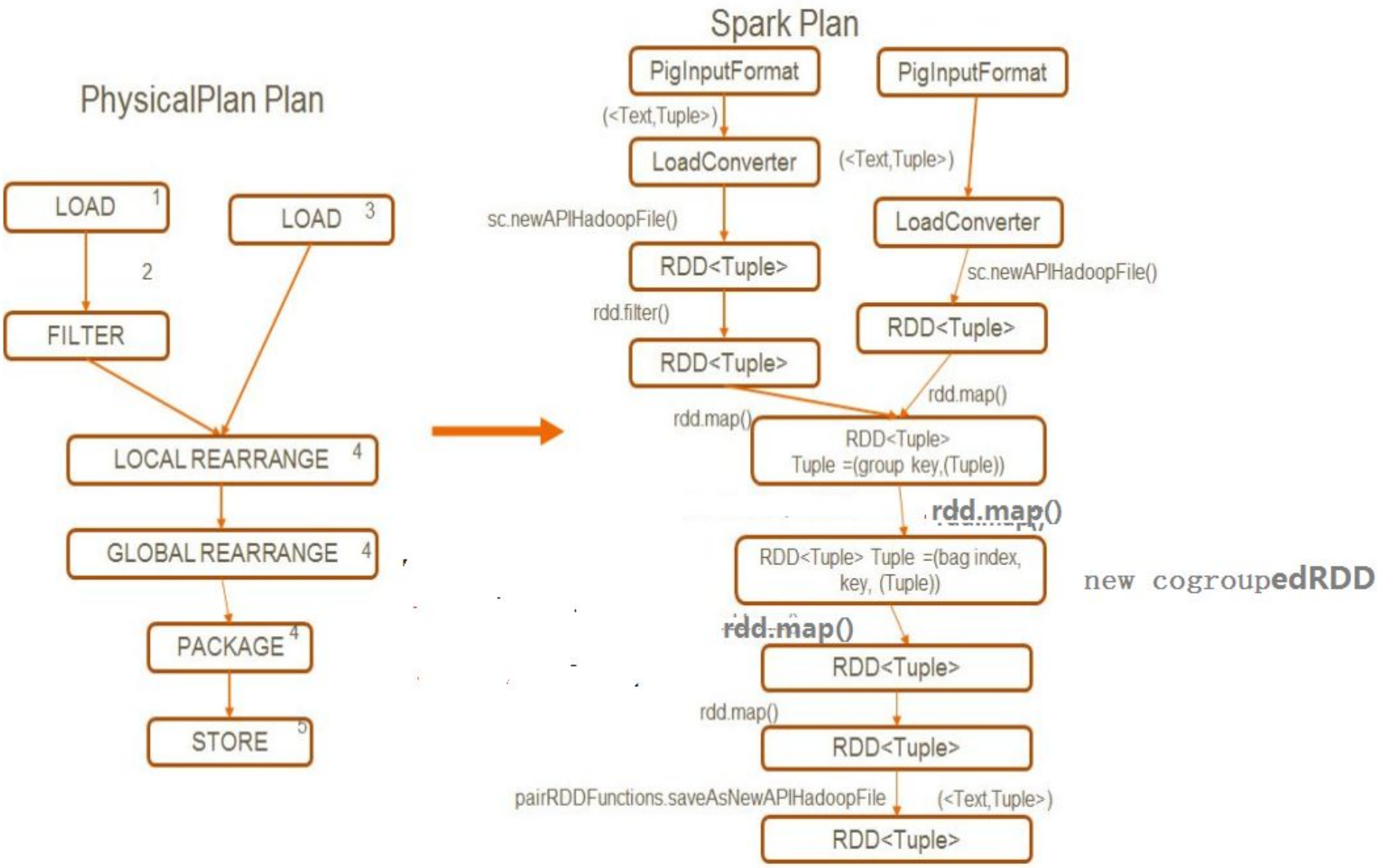
```
Pig Latin

A = LOAD 'file1' AS (x,y,z);

B = LOAD 'file2' AS (t,u,v);

C = FILTER A by y >0;

D = JOIN C BY x, B BY u;

STORE D INTO 'output';
```



Pig Latin to RDD<Tuple> transformations



Pig Latin to RDD<Tuple> transformations



Operator Mapping

Pig Operator	Spark Operator
Load	newAPIHadoopFile
Store	saveAsNewAPIHadoopFile
Filter	filter
GroupBy	groupby/reduceBy
Join	CoGroupRDD
ForEach	mapPartitions
Sort	sortByKey

Agenda

- Background
- Why Pig on Spark ?
- Design Architecture
- **Benchmark**
- Optimization
- Current Status & Future Work
- Q&A

Benchmark Overview

Component	Version
Pig	Spark branch
Hadoop	2.6.0
Spark	1.6.2
PigMix	Trunk

Basic Configuration

spark.master=yarn-client

spark.executor.memory=6553m

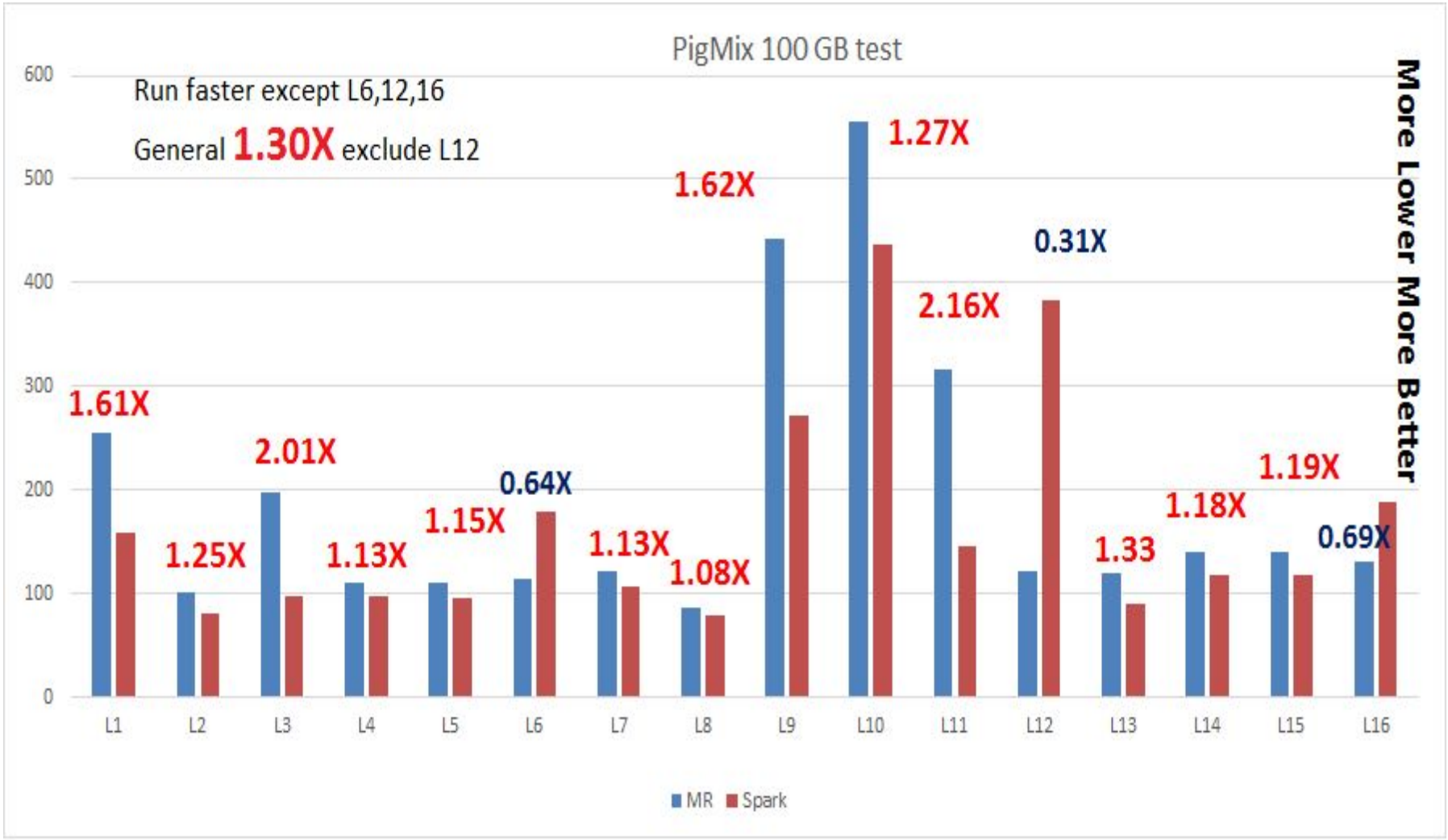
spark.yarn.executor.memoryOverhead=1638

spark.executor.cores=8

spark.dynamicAllocation.enabled=true

spark.network.timeout=1200000

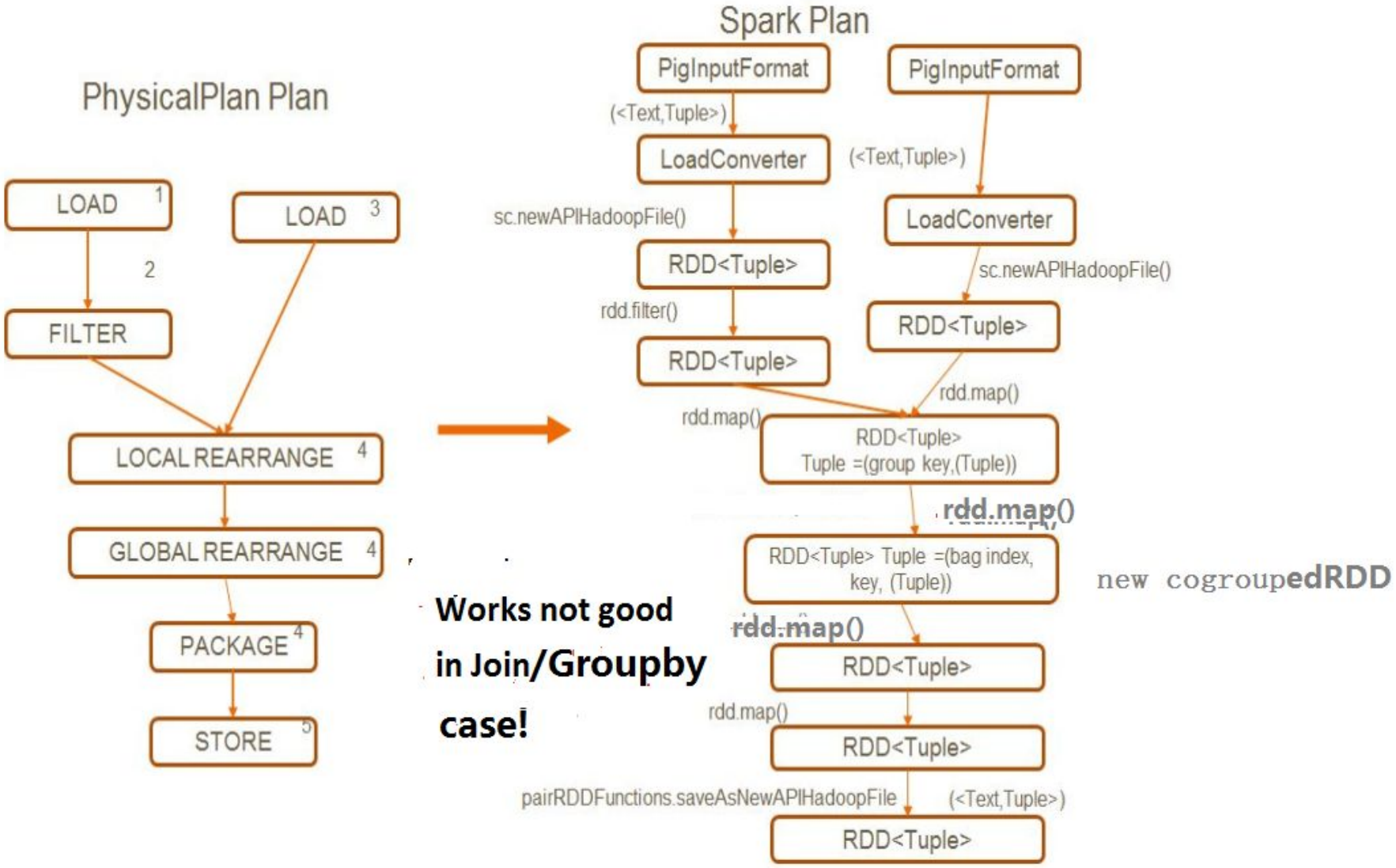
Benchmark Overview (cont'd)



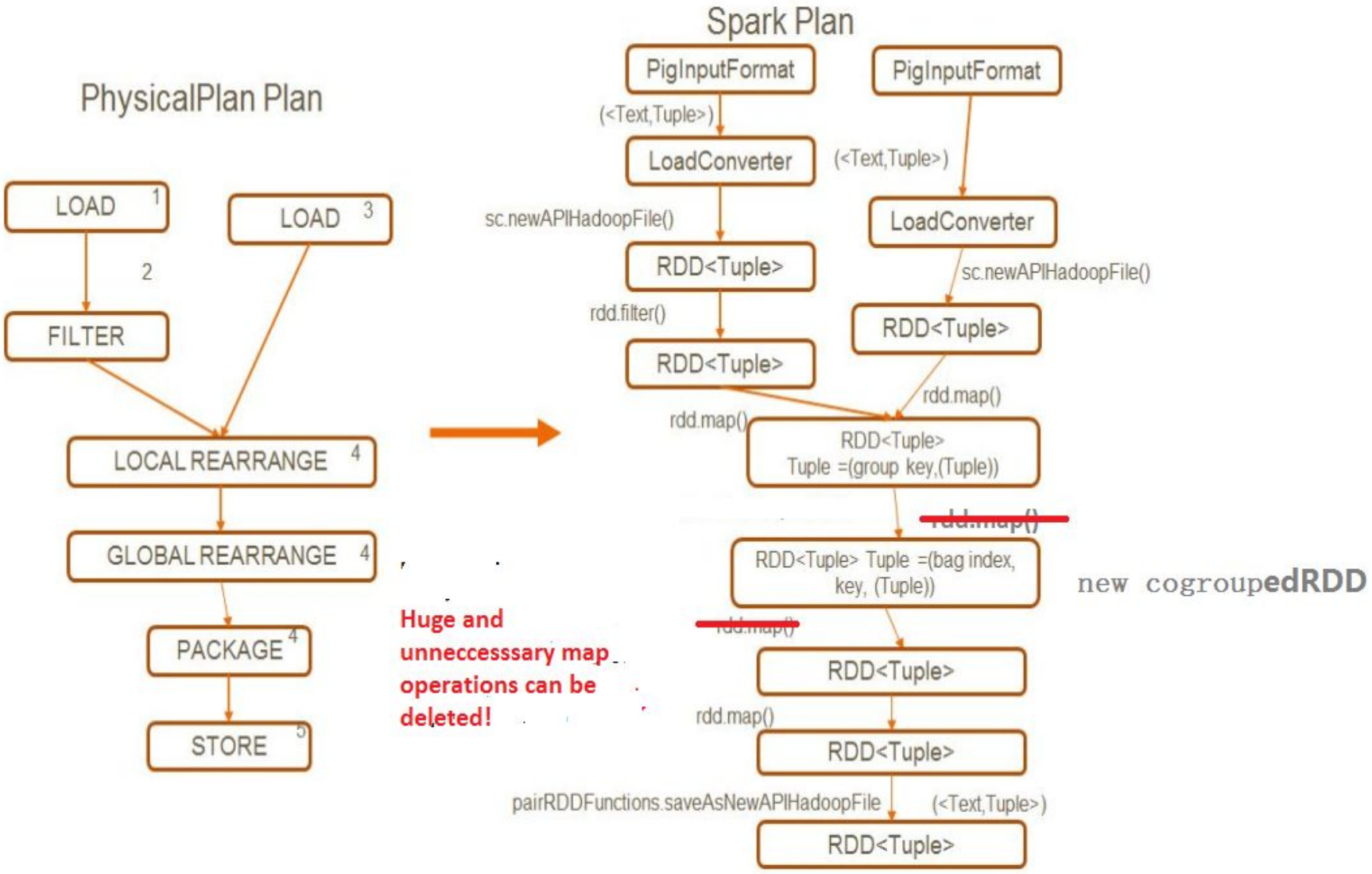
Agenda

- Background
- Why Pig on Spark ?
- Design Architecture
- Benchmark
- **Optimization**
- Current Status & Future Work
- Q&A

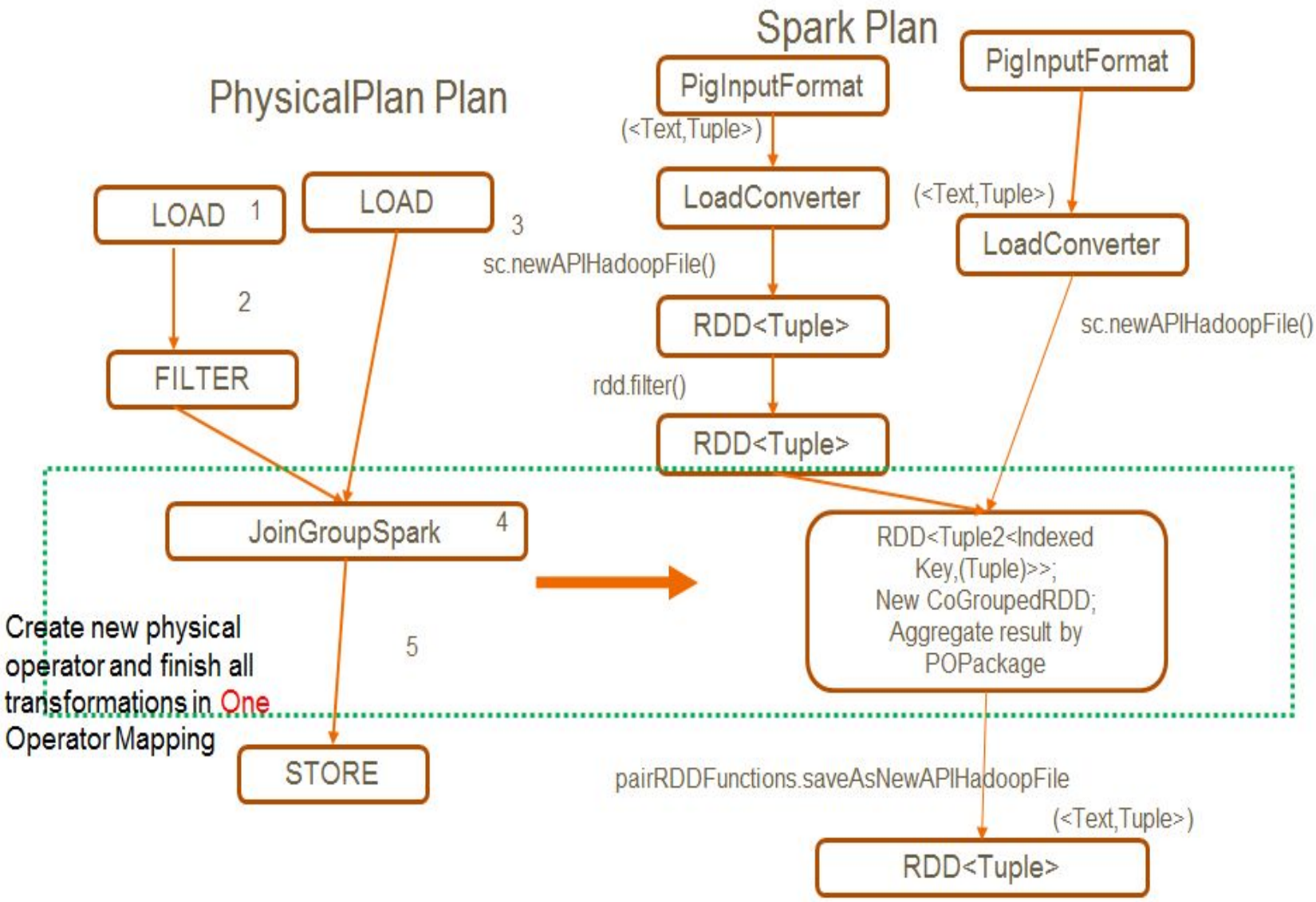
Optimize GroupBy/Join



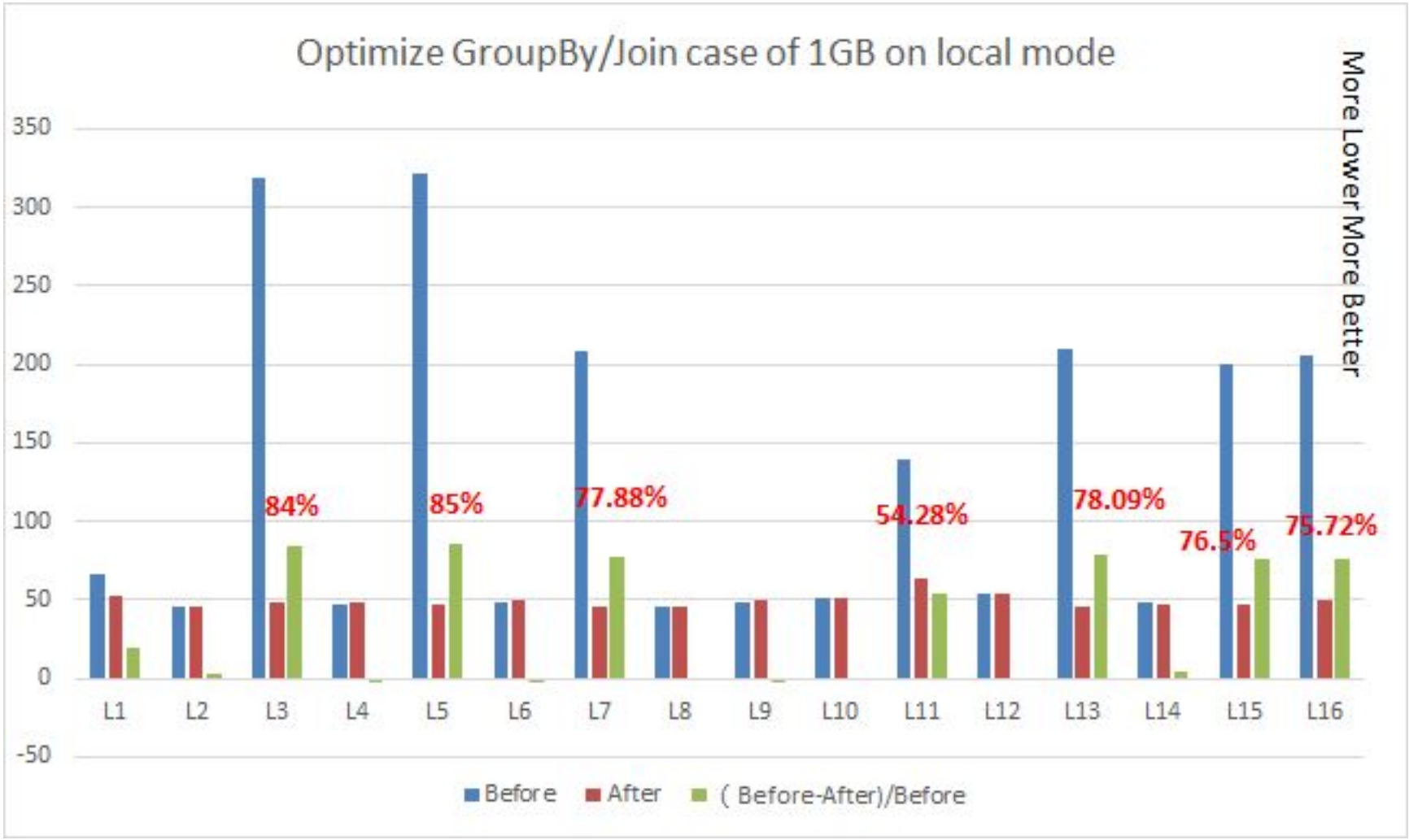
Optimize GroupBy/Join



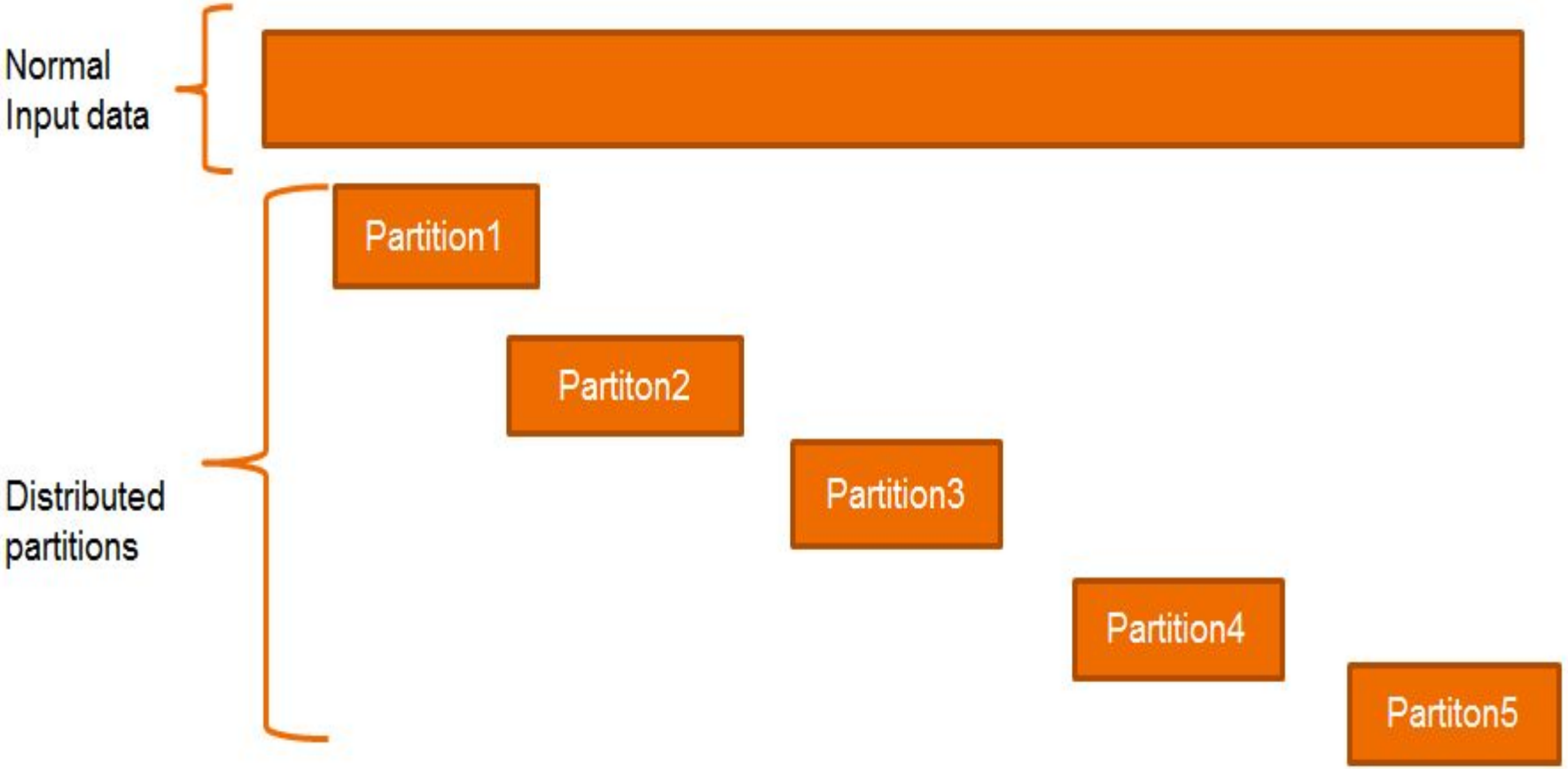
Optimize GroupBy/Join



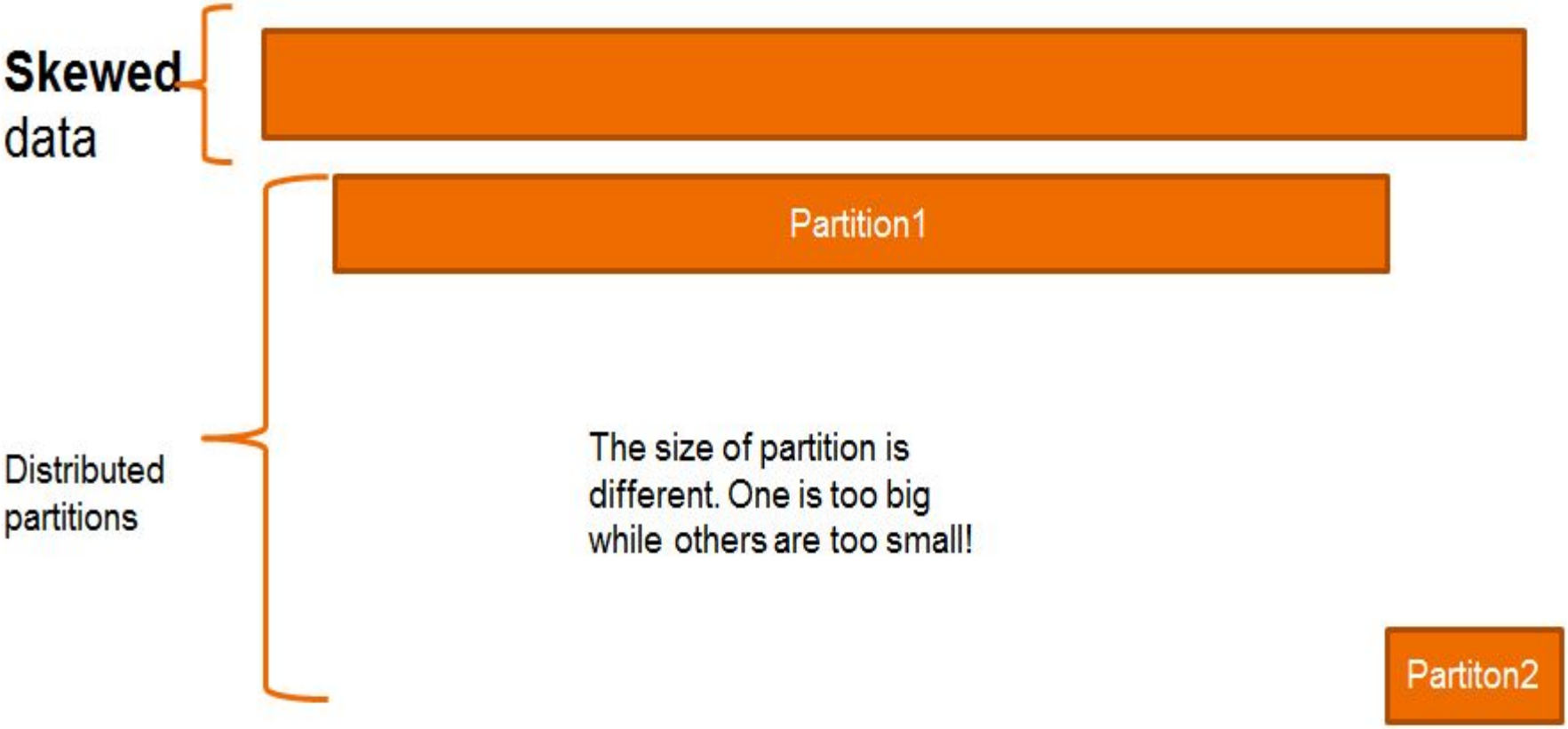
Optimize GroupBy/Join



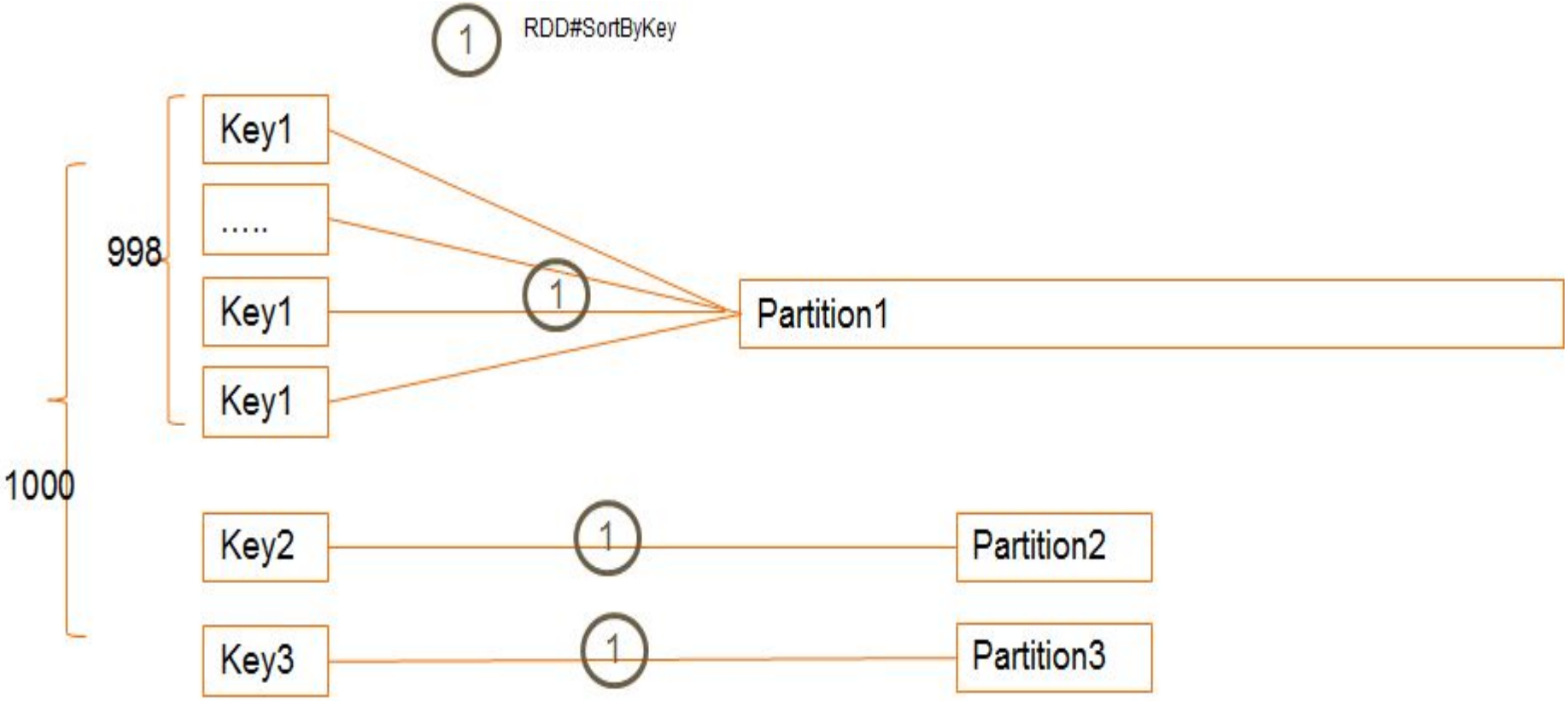
Skewed Key Sort



Skewed Key Sort

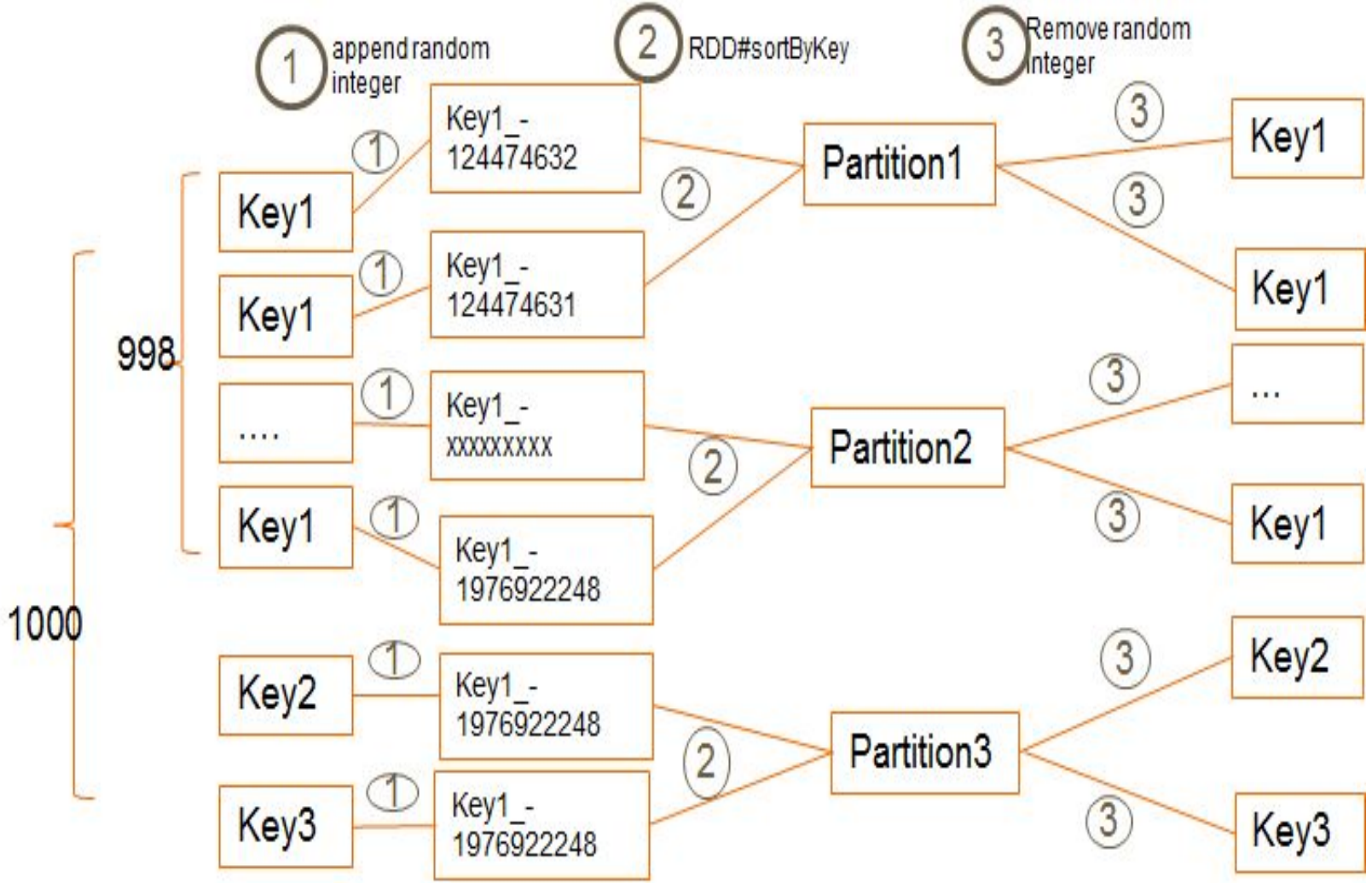


Skewed Key Sort

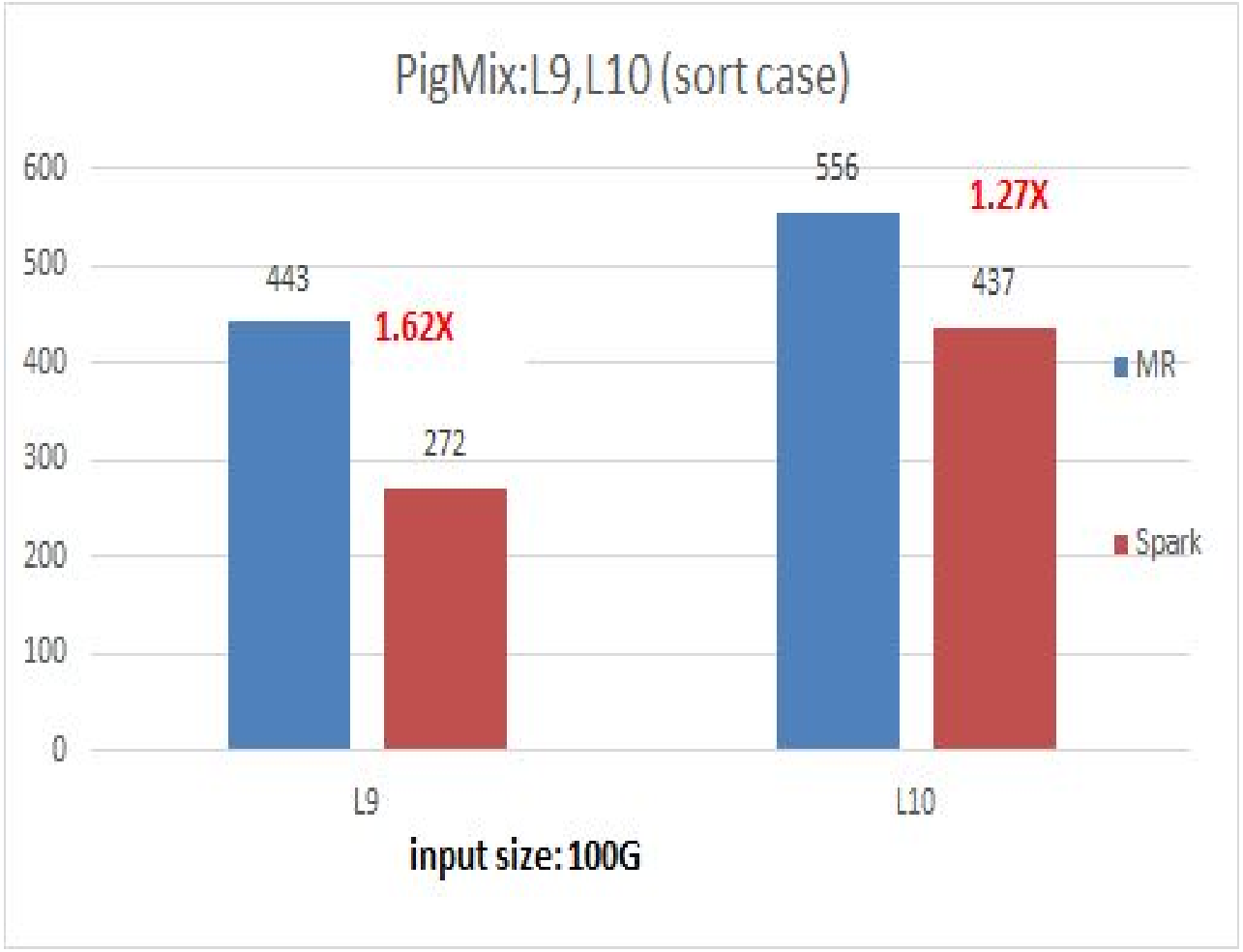


RangeParitioner used by
RDD.sortByKey works well ?

Salted Key Solution



Skewed Key Sort Performance



There are **significant** performance Improvement in sort case(L10) and skewed key sort case(L9)

Agenda

- Background
- Why Pig on Spark ?
- Design Architecture
- Benchmark
- Optimization
- **Current Status & Future Work**
- Q&A

Current Status: Nearing end of Milestone 1

- Functional completeness: DONE
- All Unit Tests Pass: DONE
- Merge Spark Branch to Master: In Code Review

Ongoing Work towards Milestone 2

- Implement Optimizations
  - Optimize Group by/Join - PIG-4797: DONE
  - FR Join - PIG-4771: DONE
  - Merge Join - PIG-4810: DONE
  - Skewed Join: UNDER REVIEW
- Enhance Test Infrastructure
  - Use “local-cluster” mode to run unit tests
- Spark Integration
  - Improved error, progress, stats reporting
  - YARN Cluster Mode

Future work: Milestone 3

- Implement More Optimizations
  - Split / MultiQuery using RDD.cache()
  - Compute optimal Shuffle Parallelism
  - Optimize/Redesign Spark Plan
- Code Stablization, Bug Fixes

Contribution welcomed

- **Git:**
  - https://github.com/apache/pig/tree/spark
- **Wiki :**
  - https://cwiki.apache.org/confluence/display/PIG/Pig+on+Spark
- **Umbrella jira:**
  - PIG-4059

Q&A