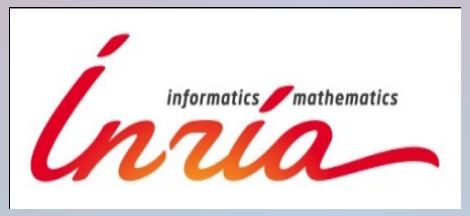
Management & Analysis of Big Data in Zenith Team

Reza Akbarinia

Zenith Team,
INRIA & LIRMM



Outline

- Introduction to MapReduce
- Dealing with Data Skew in Big Data Processing
- Data Partitioning for MapReduce
- Frequent Sequence Mining
- Frequent Itemset Mining
- Conclusion

MapReduce : A Framework for Big Data Processing

Introduced by Google to support parallel processing of highly distributable problems, e.g. PageRank

- Hadoop: an open source (Apache) distribution of MR
- Used by many companies for Big Data processing, e.g. Yahoo, Facebook

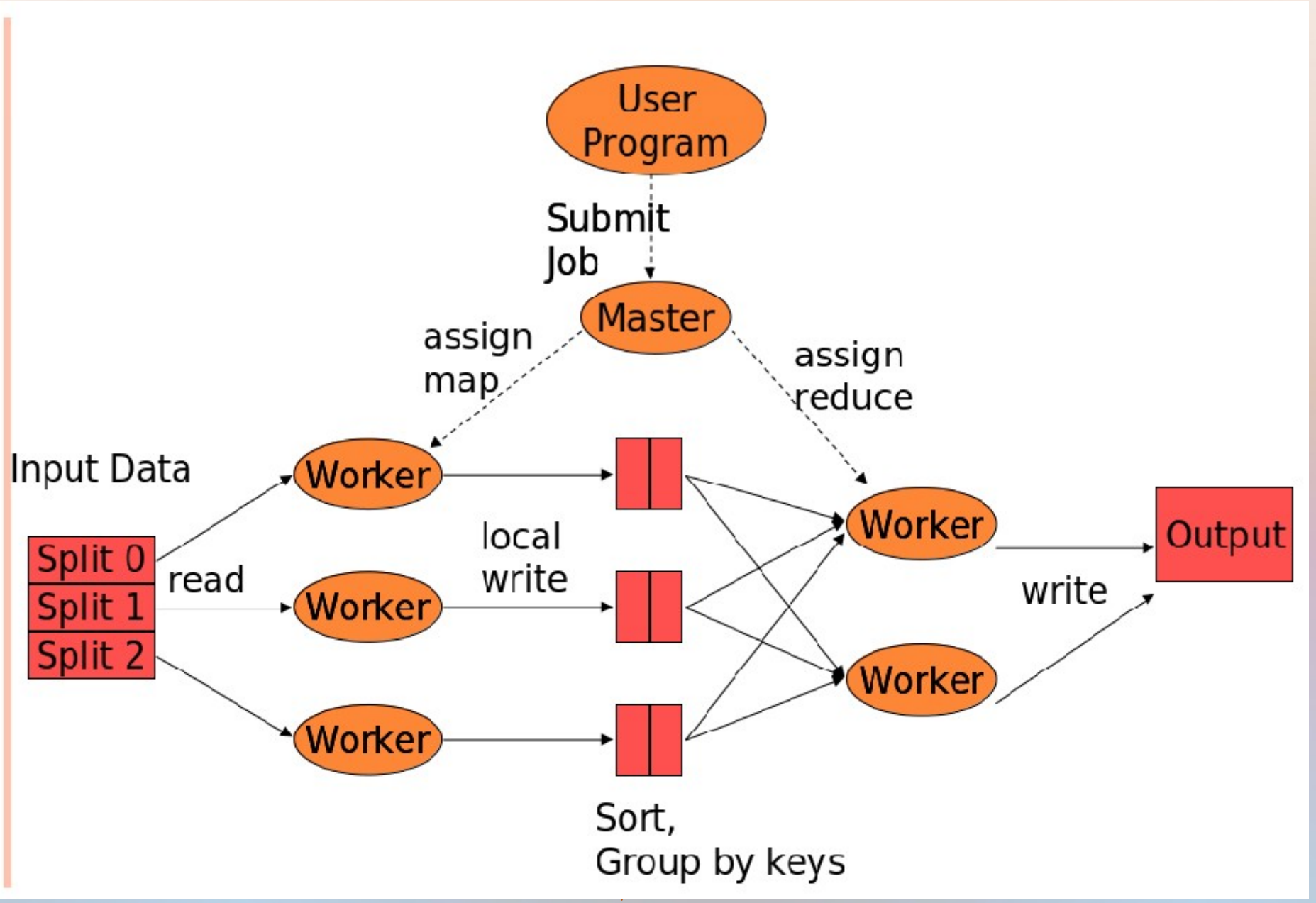
Programming principle:

- **Map step**: The input data is divided to smaller split, and each split is processed by a map worker to produce a set of intermediate key-values

  Reduce step: all values of each key are transferred to one reduce worker where a reduce function is applied on them to produce the final results

Shuffle: process of sorting, grouping and transferring intermediate key-values from map to reduce workers.

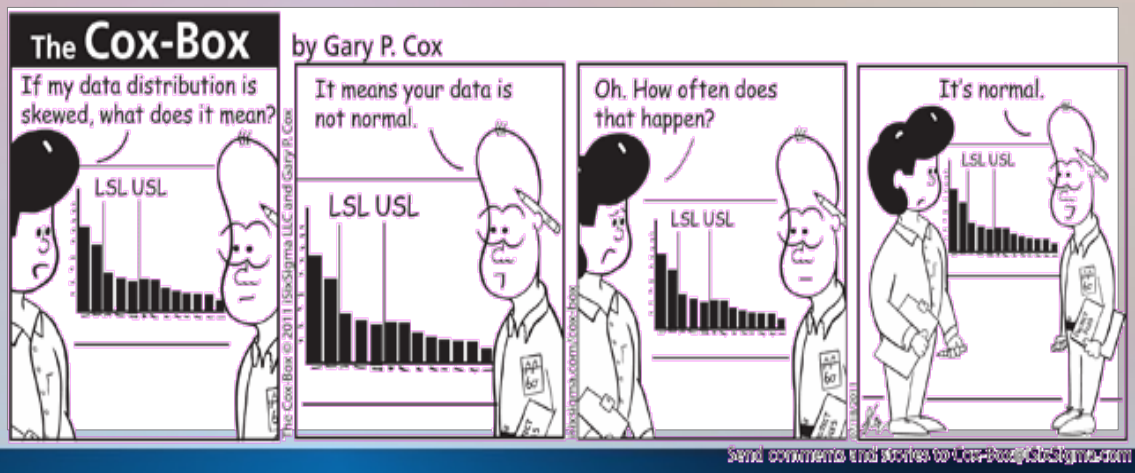
MapReduce Architecture

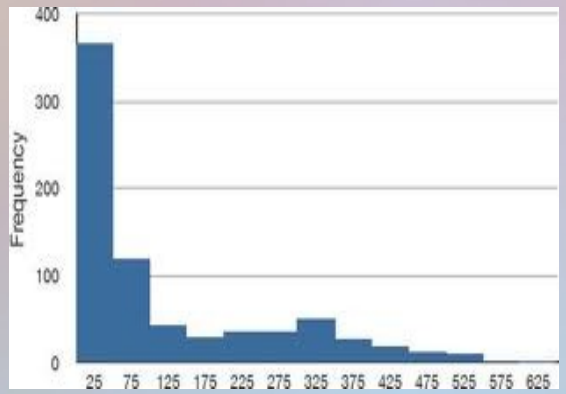


Dealing With Data Skew in MapReduce

Problem:

- Poor performance in case of skew in reduce phase
- In some applications, a high percentage of values is processed by one reducer
- E.g. Top-k, Skyline, some Aggregate queries

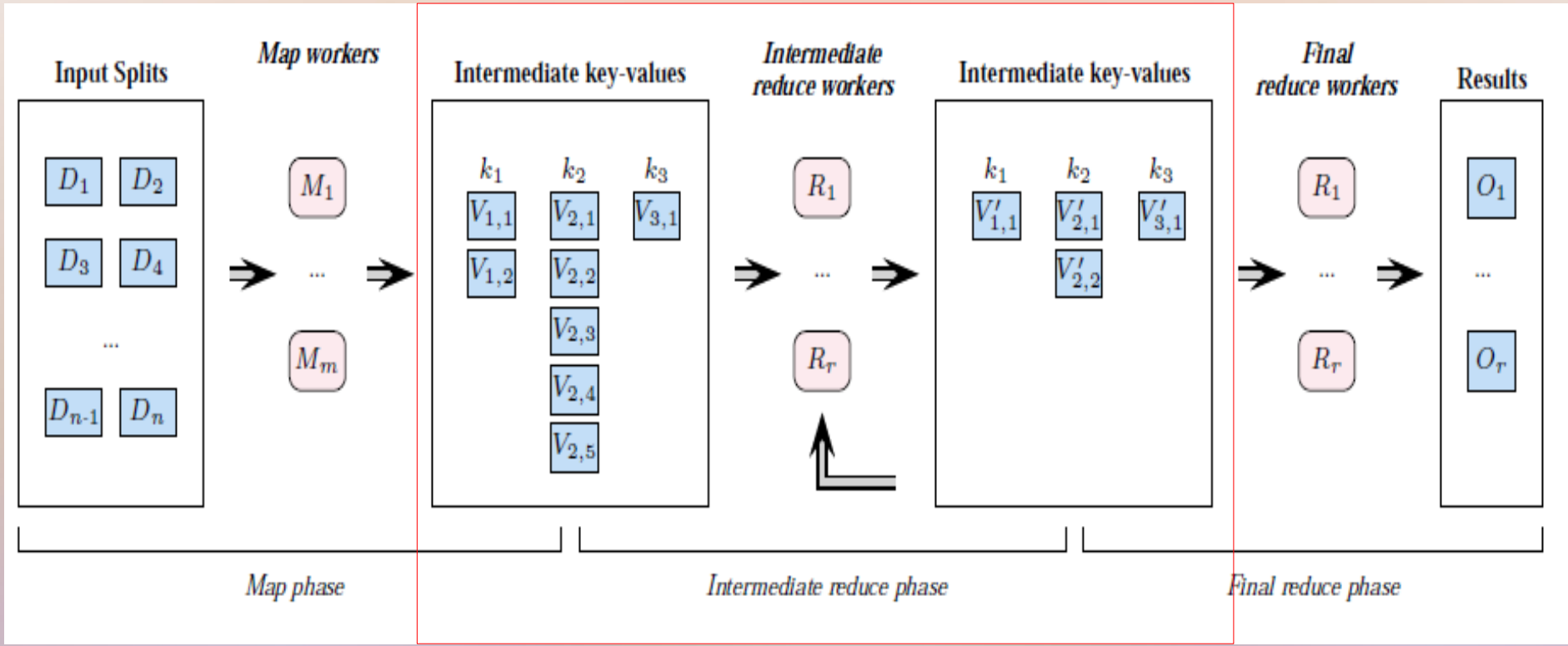




Our Solution: FP-Hadoop

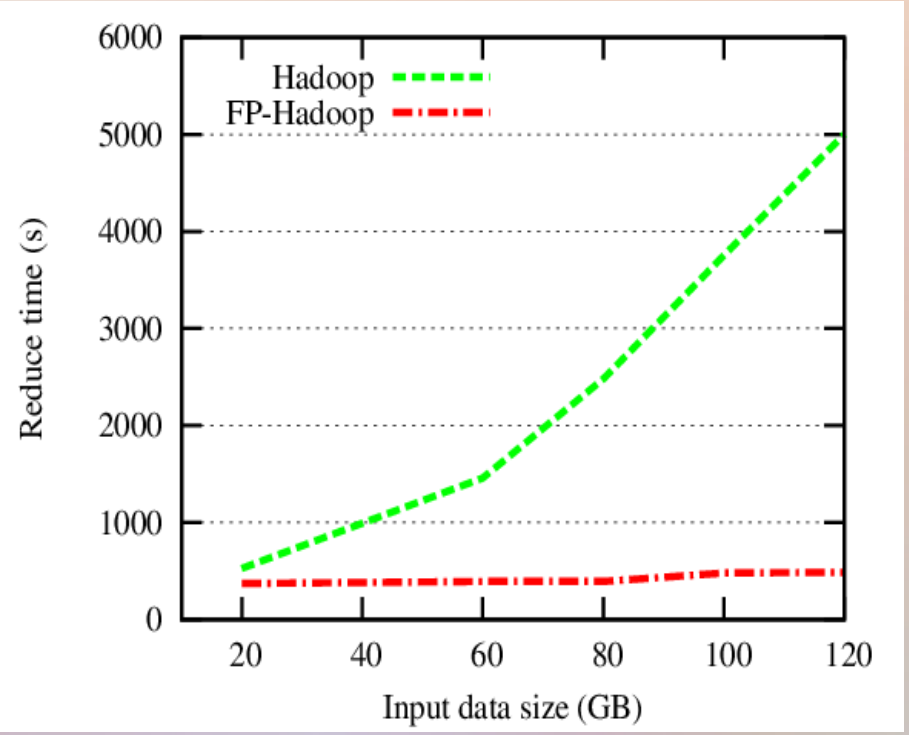
- Main idea: an intermediate reduce (IR) phase
  - An IR function (e.g. combiner) is executed over blocks of intermediate values
  - Using all workers of the system
- Features of IR phase
  - Collaborative reducing of each key
  - Hierarchical execution plans
  - Optimized scheduling of distributed blocks
    - In contrast to combinerf unction, the IR function is applied over distributed intermediate blocks

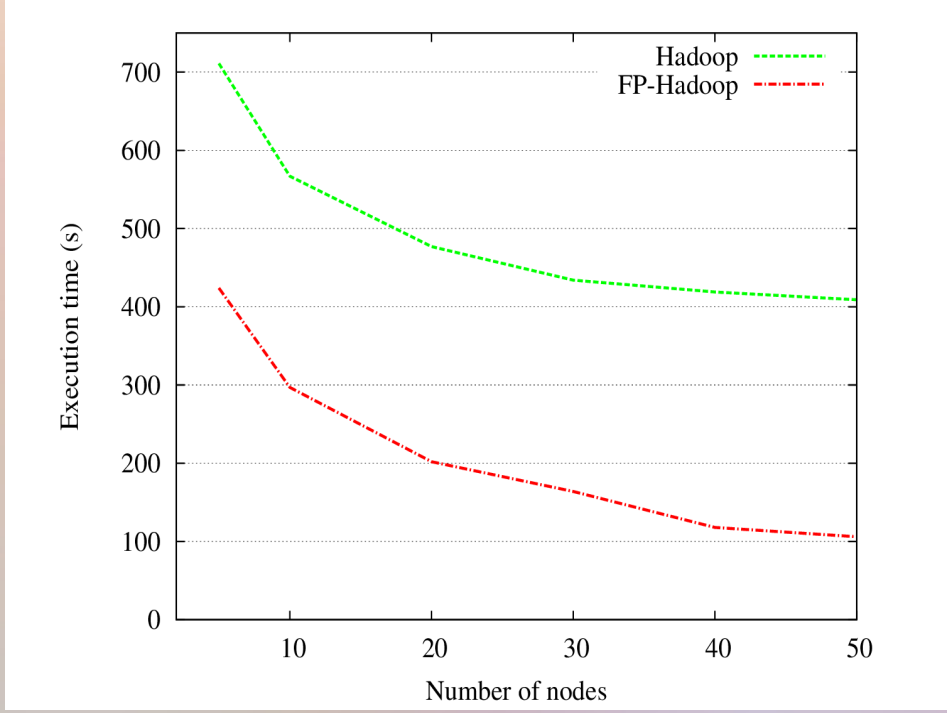
Data Flow in FP-Hadoop



Performance Evaluation

- Test platform: Grid5000

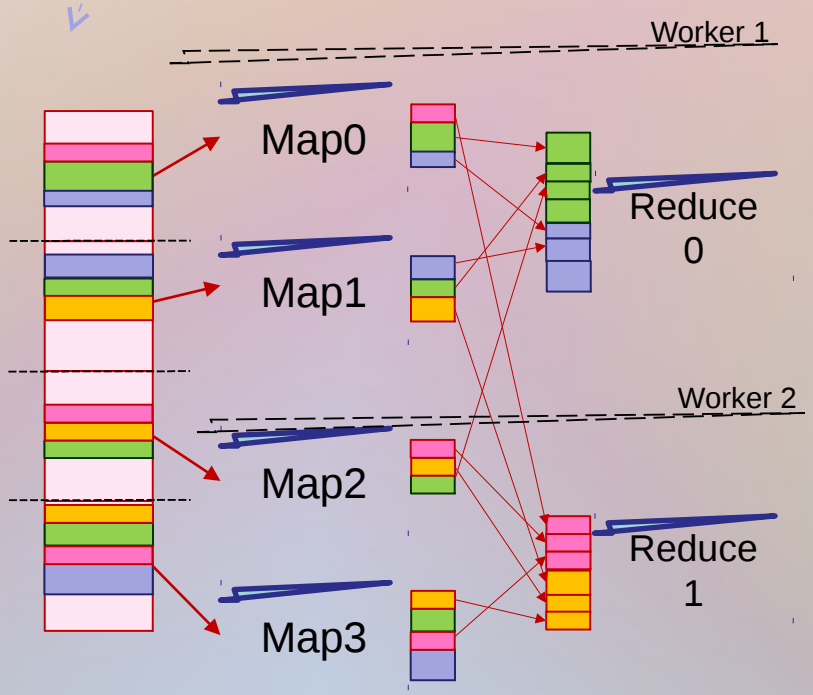




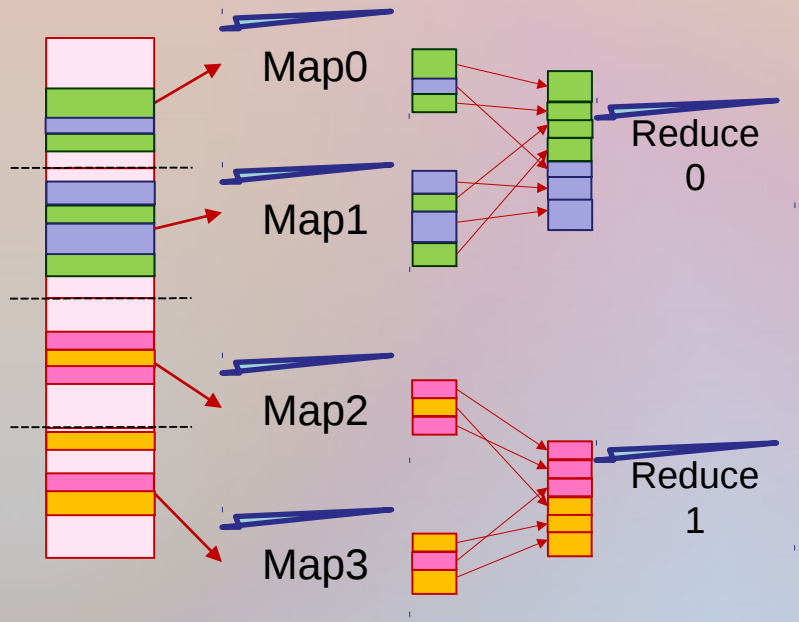
- FP-Hadoop : up to 10x faster than Hadoop (MR) in reduce time and 5x in total execution time

Data Partitioning for Reducing Data Transfer in MapReduce

- The shuffle phase may involve large data transfers
  - Because each mapper sends high data volumes to each reducer
- Result: high response time of some jobs because of slow shuffle
- **Ideal case:**
  - Values of each key are produced at one map worker, and are reduced by the same worker



Normal situation

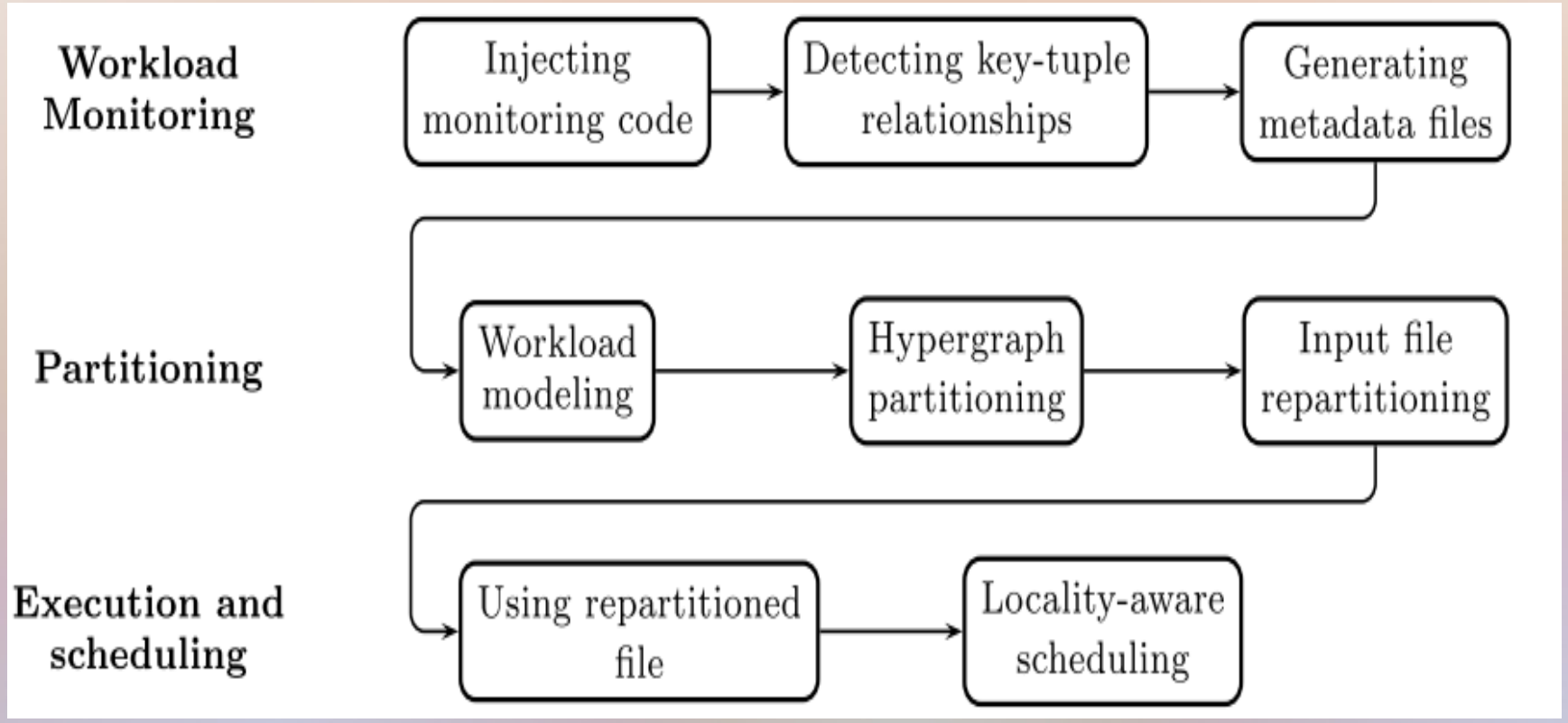


Ideal situation

Our Contribution

MR-Part: A new approach for minimizing data transfers in MapReduce

- Implemented on top of Hadoop



Experiments

Environment

- **Grid5000**

Comparison

- Native Hadoop (NAT)
- Hadoop + reduce locality-aware scheduling (RLS)
- MR-Part (MRP)

Benchmark

- TPC-H, MapReduce version

Parameters

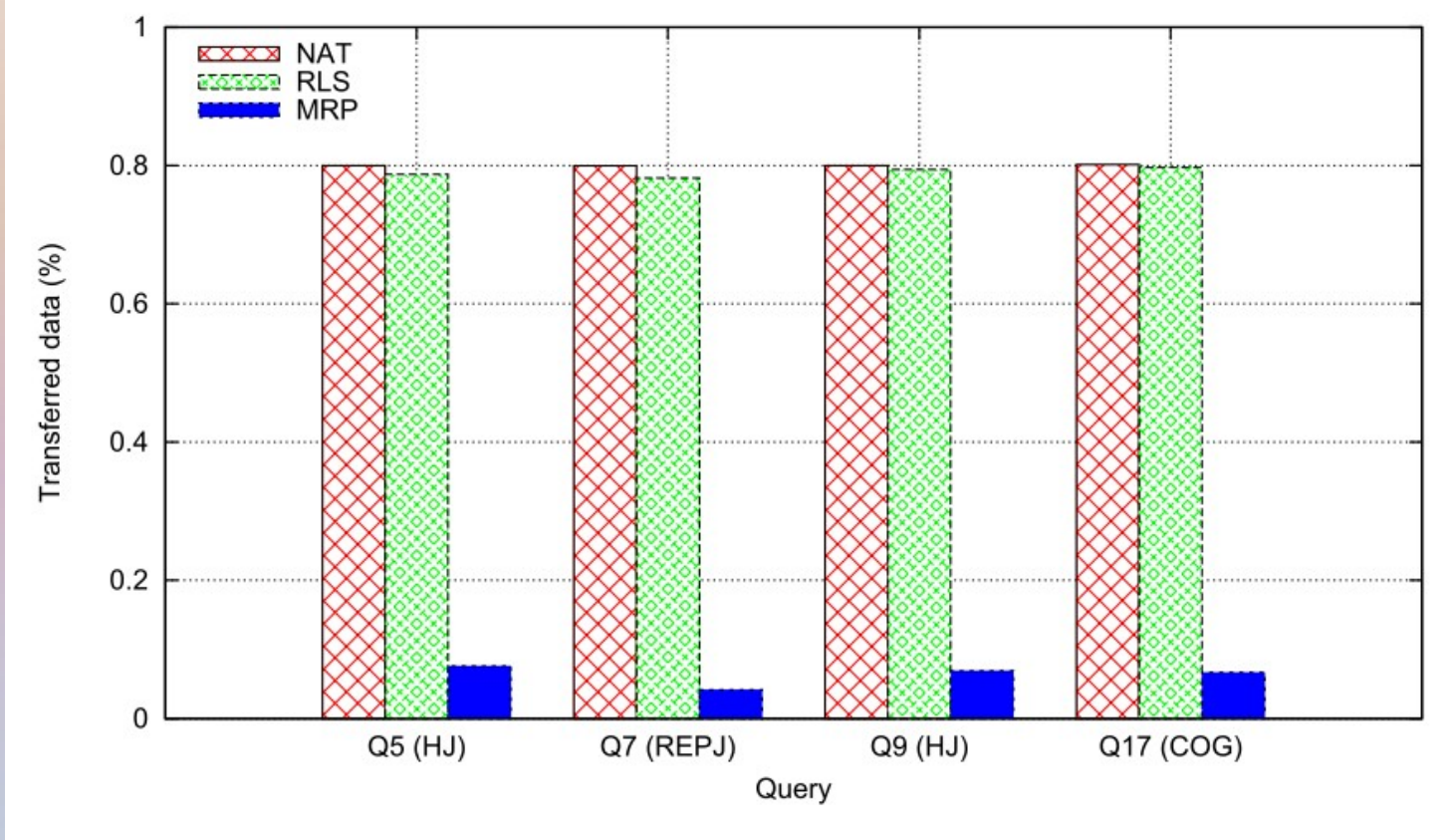
- Data size, cluster size, bandwidth

Metrics

- Transferred data
- Latency (response time)

Performance Evaluation

Percentage of transferred data



Frequent Sequences: TeraBytes of data ongoing work with Bull

- Bull's clusters (e.g. CURIE) need careful management and real-time monitoring.
- Clusters' nodes send lots of messages in log files that:
  - Cannot be explored by humans (hundreds of Tera-bytes)
- Zenith is designing massively distributed data mining methods that scale for analyzing this huge data
- Patterns discovered from these logs will feed rule bases that allow monitoring the clusters and trigger alarm in case of possible anomaly.

Frequent Sequences: TeraBytes of data

ongoing work with Bull

Message logs (hundreds of Tera-bytes)

47 2013 Jun 30 03:29:07 kay0 daemon info named error 1#53
47 2013 Jun 30 03:29:07 kay0 daemon info named error 2#53
48 2013 Jun 30 03:29:07 kay0 daemon info named error 1#53
49 2013 Jun 30 03:29:09 kay0 daemon info named error 5#53
50 2013 Jun 30 03:29:09 kay0 daemon info named error 5#53
50 2013 Jun 30 03:29:09 kay475 syslog err syslog-ng I/O
50 2013 Jun 30 03:29:09 kay475 syslog notice syslog-ng Error
51 2013 Jun 30 03:29:09 kay475 syslog notice syslog-ng Suspending
52 2013 Jun 30 03:29:10 kay0 daemon info named error 5#53
53 2013 Jun 30 03:29:10 kay0 daemon info named error 5#53
53 2013 Jun 30 03:29:10 kay0 daemon info named error 5#53
53 2013 Jun 30 03:29:11 kay0 daemon info named error f#53

Monitoring

Patterns

...
Daemon error + syslog error -> Suspending
User notice + IO warning -> syslog error
...

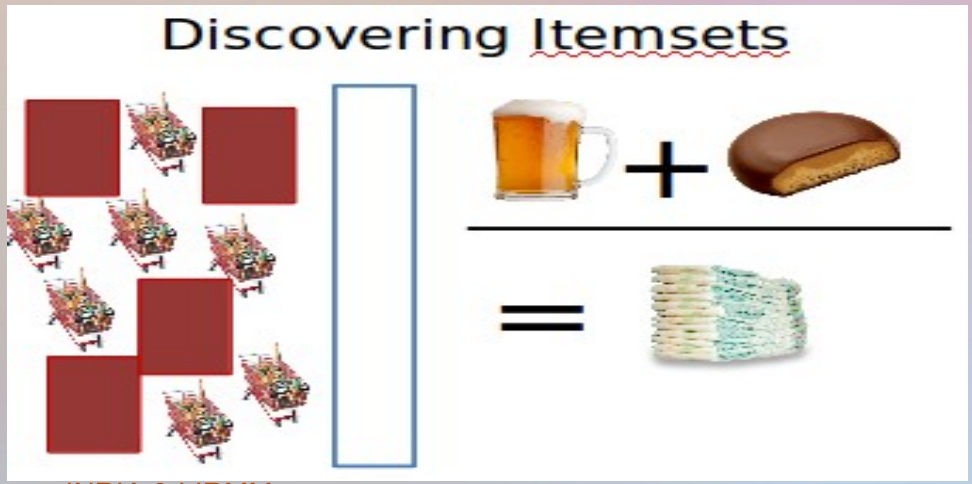
Rules

Data Partitioning for Frequent Itemset Mining ongoing work

- Improving frequent itemset mining algorithms based on input data partitioning
  - Mappers of MapReduce work on partitions

- A new data partitioning technique: **Item based data partitioning**

  Objective: Mining several Terabytes of data (Clouweb)



Hadoop_g5k: a Tool for Easy Hadoop Deployment in Grid5000

Overview

- Initiated by Miguel Liroz (research engineer in Zenith)
- Scripts and APIs to deploy Hadoop applications in G5K
- Based on execo library
- Documentation available in G5K wiki, and sources in GitHub

Features

- Manages whole life-cycle of Hadoop clusters
- Supports several versions of Hadoop and hides configuration details
- Automatic configuration based on best practices
  - Topology, number of slots, memory per task, etc.

Conclusion

Big Data Processing and Analysis

- Dealing with skew in big data processing
- Data partitioning for reducing network traffic in MapReduce framework
- Error pattern detection in Bull super-computer logst
- Large scale frequent itemset mining

To evaluate all these solutions

- We use Grid5000
- Requirement: more storage capacity

Questions ?

Web Site: search “Zenith Team” in Google
Email: Reza.Akbarinia@inria.fr

hg5k

An example

- Creation + installation + initialization/start
- Job execution
- Destruction

```
hg5k --create $OAR_FILE_NODES
hg5k --bootstrap $LIB_HOME/hadoop-1.2.1.tar.gz
hg5k --initialize –start
hg5k --jarjob $LIB_HOME/hadoop-test-1.2.1.jar mrbench
hg5k --delete
```

hadoop_engine

Features

- Test automatization for Hadoop based experiments
- Optimizes dataset creation and/or deployment
- Generates ds/xp configuration + statistics for all combinations
- (to appear) Automatic generation of figures from results

Example (from wiki)

```
[test_parameters]
test.summary_file = ./test/summary.csv
test.ds_summary_file = ./test/ds-summary.csv
test.stats_path = ./test/stats

[ds_parameters]
ds.class = hadoop_g5k.dataset.StaticDataset
ds.class.local_path = datasets/ds1
ds.dest = ${data_dir}
ds.size = 1073741824, 2147483648 # 1 | 2 GB

[xp_parameters]
io.sort.factor = 10, 100
xp.combiner = true, false
xp.job = program.jar || ${xp.combiner} other_job_options ${xp.input} ${xp.output}
```

Where results are stored

Dataset configuration

Experiments configuration