IntegrityMR: Result Integrity Assurance Check Framework for Big Data Analytics and Management Applications

Yongzhi Wang*, Jinpeng Wei*, Mudhkar Srivatsa§, Yucong Duan◆, Wencai Du◆
* Florida International University § IBM T.J. Watson Research Center ◆ Hainan University

Motivation

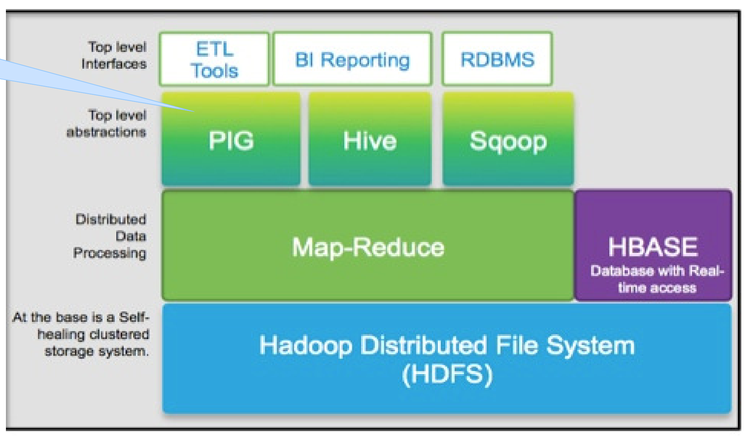
- How do we construct big data analytics infrastructure on the cloud that can provide high integrity assurance?

Our Focus

Application Layer Integrity

MapReduce Layer Integrity

Storage Layer Integrity



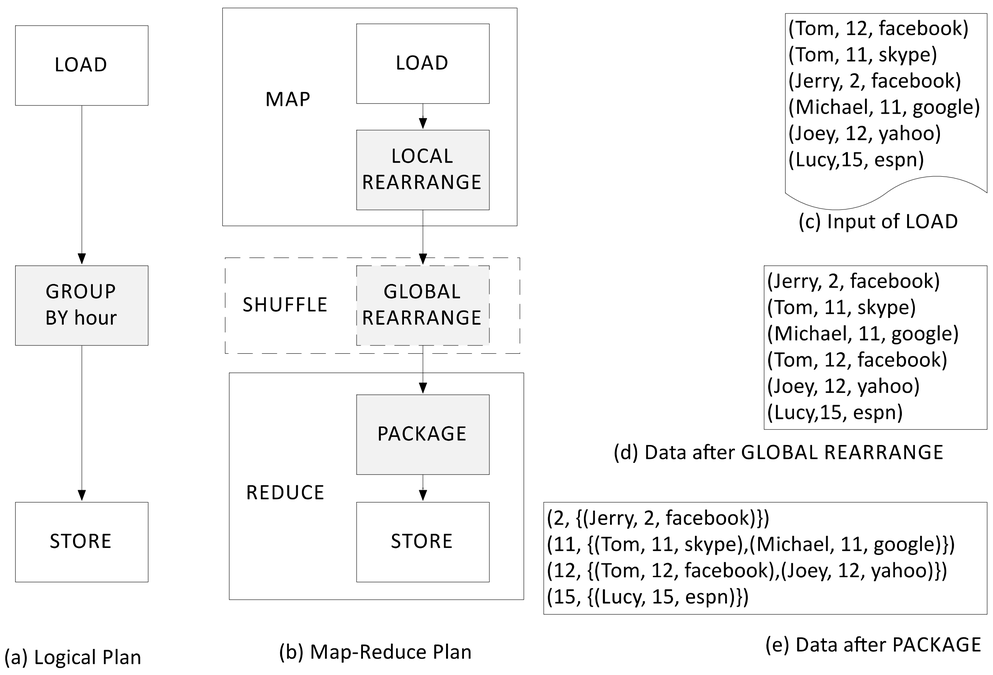
- Research goal: design and implement an integrity assurance framework for Apache Pig.

What is Apache Pig?

- Pig Latin is a scripting language designed to mimic the declarative style of SQL. The accompanying system, Pig, can compile Pig Latin scripts into physical plans that are executed over Hadoop MapReduce.
- Sample Pig Latin Script

```
-- Script 1: GROUP data in houred.txt by hour
raw_data = LOAD './houred.txt' USING PigStorage('\t')
                    AS (user, hour, query);
result = GROUP raw_data BY hour;
dump result;
```

- How Pig works: Transform logic plan into MapReduce plan



Yongzhi Wang, Jinpeng Wei, Mudhakar Srivatsa, Yucong Duan, and Wencai Du. "IntegrityMR: Integrity Assurance Framework for Big Data Analytics and Management Applications". The first IEEE Knowledge Management and Big Data Analytics Workshop, October 6-9, 2013, Santa Clara, CA

How to Ensure High Result Integrity for Pig?

Transform the Pig Latin script to change the plan

- Step1: Split an existing map task into two/more substitute tasks whose input data overlap.
- Step 2: Transform the reduce task to check an invariant: that the output of the substitute tasks agree on the part corresponding to the overlapped input.

An Example

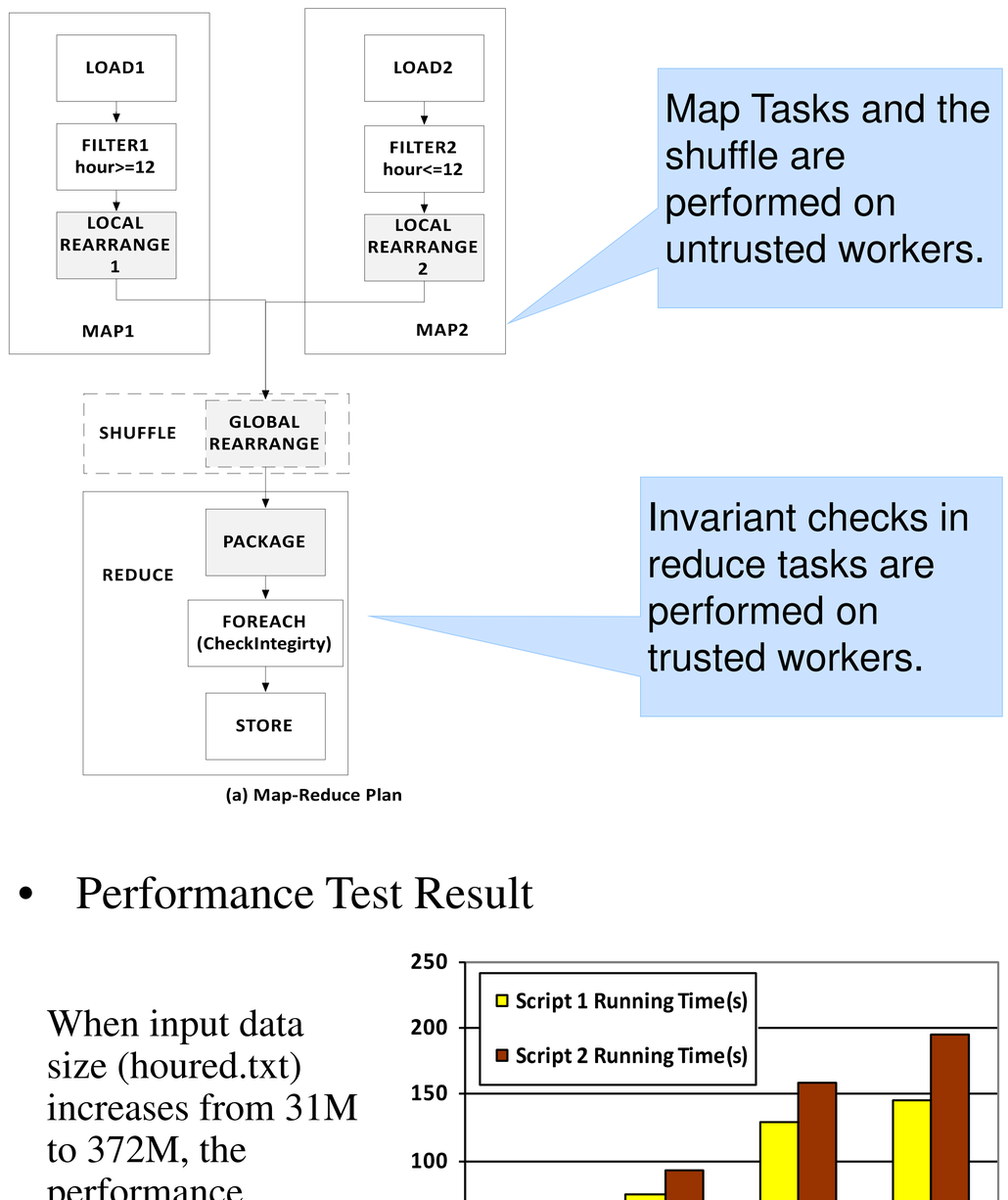
```
-- Script 1: GROUP data in houred.txt by hour
raw_data = LOAD './houred.txt' USING PigStorage('\t')
                    AS (user, hour, query);
result = GROUP raw_data BY hour;
dump result;
```

```
-- Script 2: invariant check is enforced
register ./tutorial.jar;
raw_data = LOAD './houred.txt' USING PigStorage('\t')
                AS (user, hour, query);
part1 = FILTER raw_data BY hour>=12;
part2 = FILTER raw_data BY hour<=12;
result = COGRUP part1 BY hour, part2 BY hour;
group_result=FOREACH result GENERATE
                group, org.apache.pig.tutorial.CheckInvariant($1,$2);
```

Step 1

Step 2

- Transformed MapReduce Plan



Map Tasks and the shuffle are performed on untrusted workers.

Invariant checks in reduce tasks are performed on trusted workers.

- Performance Test Result

When input data size (houred.txt) increases from 31M to 372M, the performance overhead increases from 0% to 35%.