Criteria to Compare Cloud Computing with Current Database Technology

Jean-Daniel Cryans
Dr. Alain April
Dr. Alain Abran
École de technologie supérieure, Montréal

IWSM08, Munich, Germany
November 18th

Université du Québec
École de technologie supérieure
Département de génie logiciel et des TI

Overview

- Introduction
- HBase insfrastructure
- HBase
- Comparison elements
- Transforming Comparison Elements into Comparison Criteria
- Summary

Disclaimer

- The author became an official HBase developer after writing this paper.

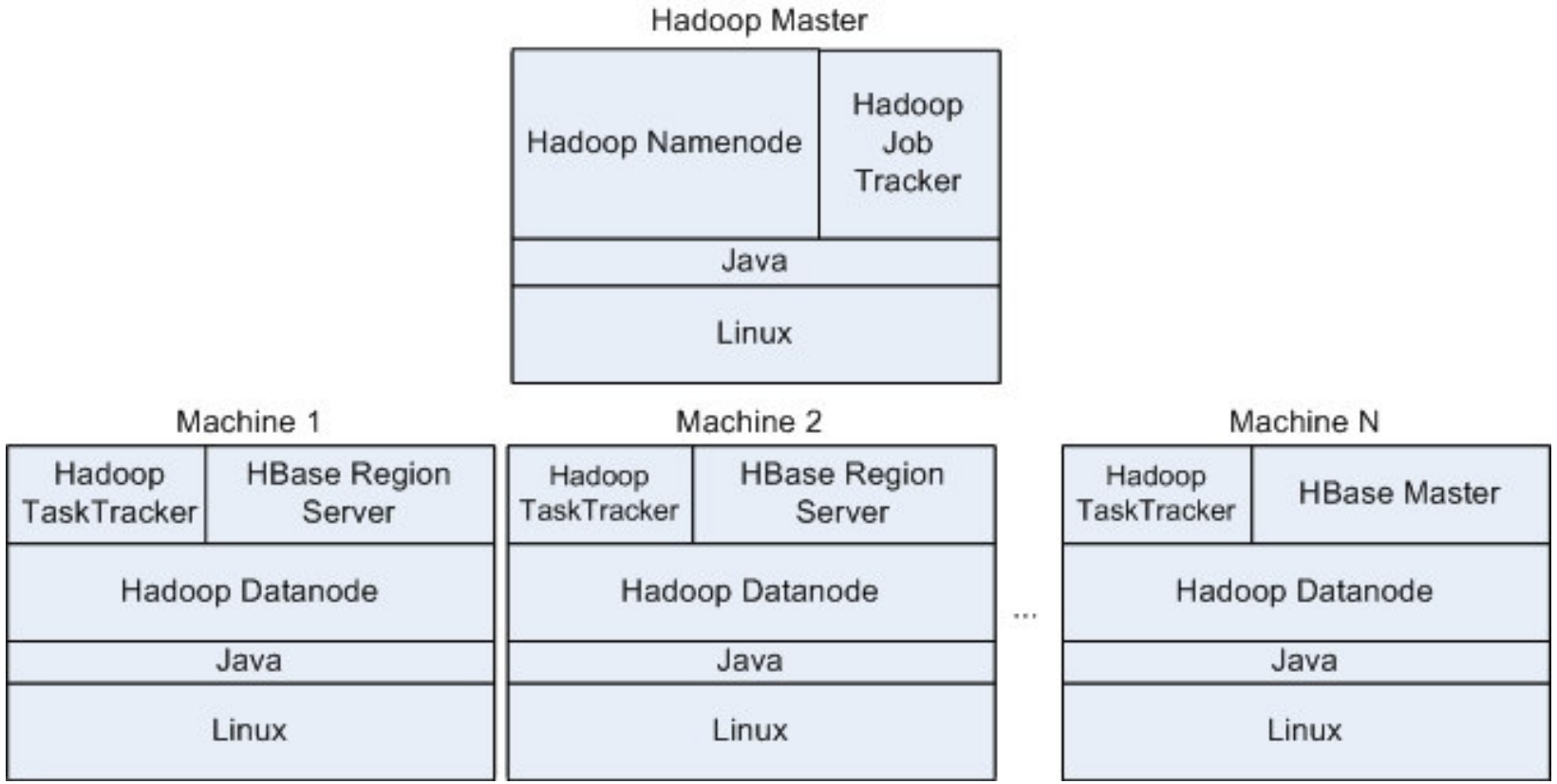
Introduction

- Today's information-heavy world faces a new problem: processing huge data sets reliably and efficiently.
- Since the publication of Google's infrastructure (Google File System, Map Reduce, Bigtable), the open-source community followed with it's own set of components (Hadoop, HBase).

Intro. : Problem statement

- A growing number of companies are considering moving from typical open-source RDBMS to those technologies but don't have any tools or methodologies to compare both solutions.
- Our paper is aimed at helping them by presenting a list of comparison elements and how they translate into assessment criteria.

HBase infrastucture



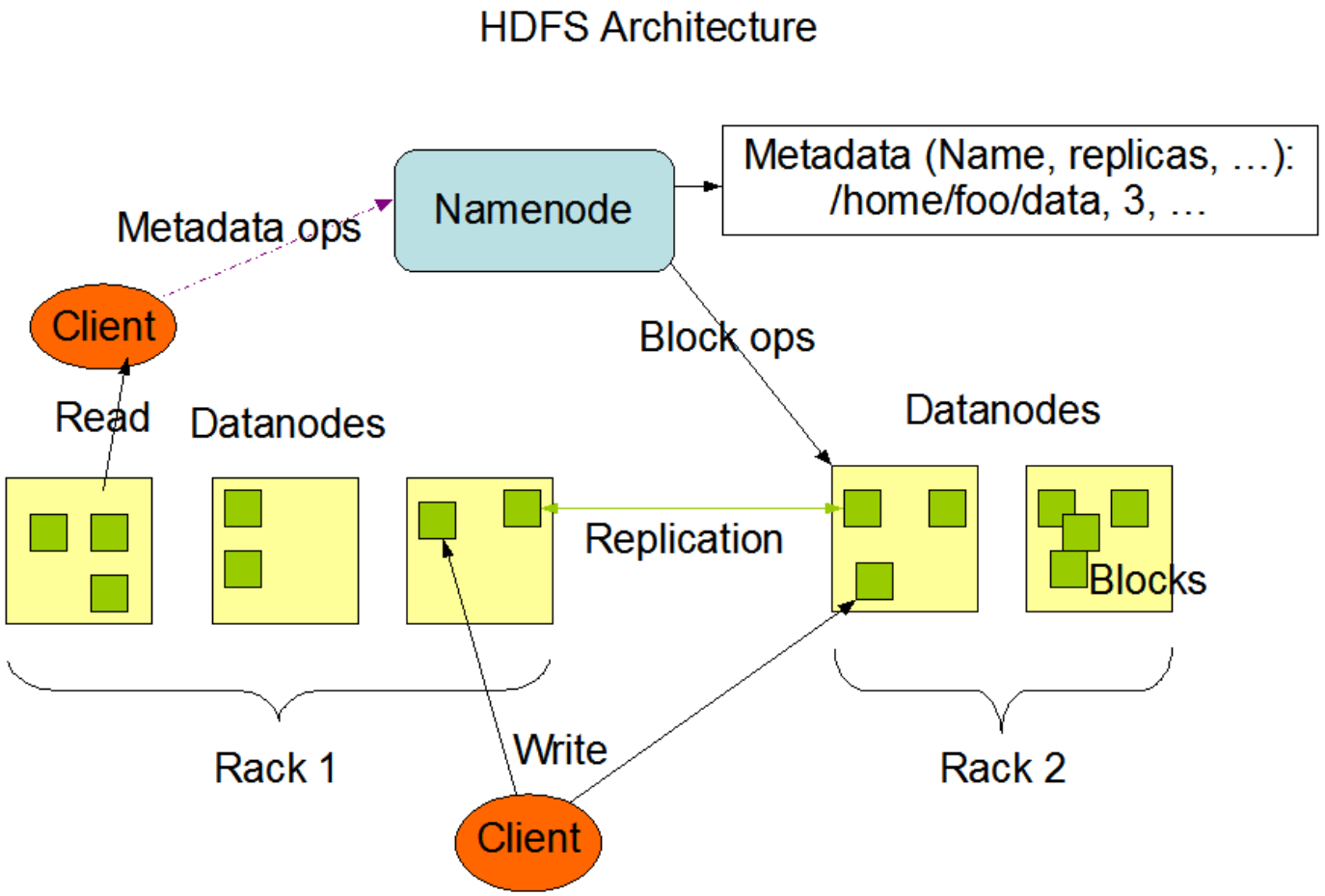
HBase infra.: Hadoop

- Hadoop is an Apache Software Foundation top-level project backed primarily by Yahoo!.
- Consists of two components:
  - Hadoop Distributed File System (HDFS)
  - An implementation of MapReduce

HBase infra.: HDFS

- The Hadoop Distributed File System can
  - reliably
  - store petabytes
  - of replicated data
  - over thousands of nodes.
- Master/slave architecture, built on commodity machines.

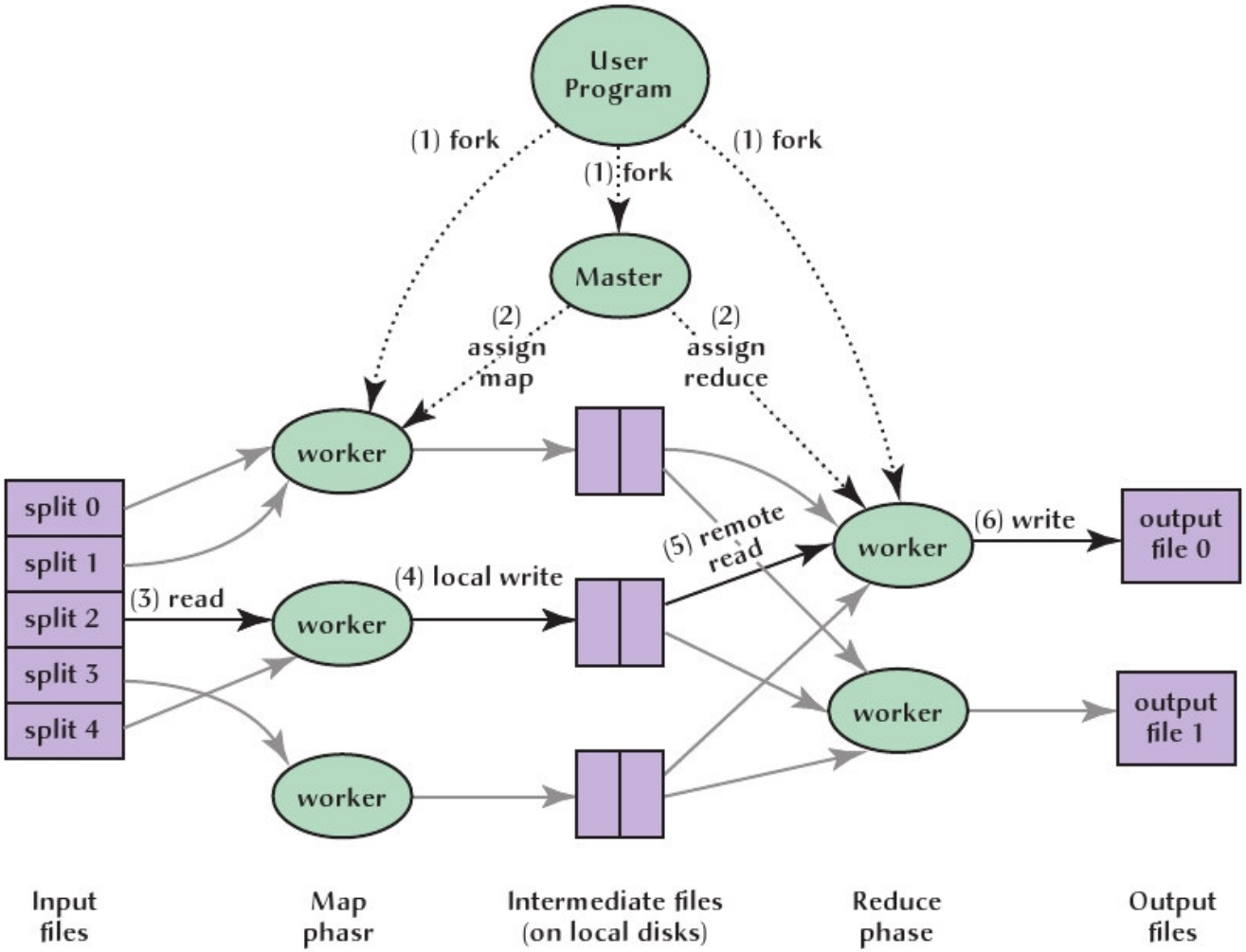
HBase infra.: HDFS



HBase infra.: MapReduce

- MapReduce is a Google-designed
  - programming model
  - to reliably
  - process petabytes of data
  - using it's locality.
- Hadoop has it's own implementation.
- Major bindings in Java and C, accessible from other languages with crippled functionality.

HBase infra.: MapReduce



HBase

- HBase is a
  - distributed
  - column-oriented
  - multi-dimensional
  - high-availability
  - high-performance
  - storage system.
- It also has a master/slave architecture.

HBase

- Motivation
  - A relational model hardly scales in the terabytes while providing built-in reliability.
- Goal
  - Usable in real-time and in MapReduce jobs.
- Data model
  - *(row:string,column:string,time:int64) → string*

HBase

- Some basic code:

```
HTable table = new HTable("myTable");
byte[] valueBytes = table.get("myRow",
  "myColumnFamily:columnQualifier1");
String valueStr = new String(valueBytes);
table.put("myColumnFamily:columnQualifier1",
  "columnQualifier1 value!");
table.delete(
  "myColumnFamily:cellIWantDeleted");
table.commit(lockId);
```

Comparison elements

- We investigated a case study where an existing system, based on RDBMS technology, has been migrated to HBase.
- In comparing the systems, many factors need to be taken into account.
- Initially, the comparison appeared difficult because some elements do not vary while others vary greatly.

Comparison elements (continued)

- Set of comparison elements we would like to assess:
  - Software architecture
  - Hardware (server VS PC class)
  - Operating system (anything VS Linux)
  - Data structure (relational VS Bigtable-like)
  - Data manipulation (SQL VS API/MapReduce)
  - Means to scale (custom VS built-in)
  - Where hardware reliability is handled (custom VS built-in)
  - How many systems are using the system

Comparison elements in Case Study

Comparison element	PostgreSQL impl.	HBase impl.
Software architecture	Three-tier	Three-tier
Hardware	Few, expensive	Scores, commodity
Operating system	Cent OS	Cent OS
Data structure	Relational tables	Bigtable-like structures
Data manipulation	ORM	HBase client API, MapReduce
Means to scale	Expensive	Inexpensive
Where hardware reliability is handled	Custom solution	HBase and Hadoop
How many systems are using it	One	One

Transforming Comparison Elements into Comparison Criteria

1. Identify impact: The impact that a comparison element has when it is not the same in the two implementations will lead to different measurement results.
2. Identify the direct effects on quality: Each comparison element has different effects on internal and external quality, as defined in the ISO/IEC 9126 models of software product quality

Transforming Comparison Elements into Comparison Criteria (continued)

Comparison element	Impact	Related ISO 9126 quality subcharacteristics
Software architecture	Changing the architecture implies completely different quality attributes that need to be evaluated.	All criteria
Hardware	Commodity hardware is less reliable and, on its own, performs poorly.	Fault tolerance Time behavior Scalability
Operating system	If the source operating system is not Linux, its efficiency and maturity are different. Restricting to Linux makes the system less adaptable.	Fault tolerance Time behavior Resource behavior Adaptability

Transforming Comparison Elements into Comparison Criteria (continued)

Comparison element	Impact	Related ISO 9126 quality subcharacteristics
Data structure	The target data structure is easy to change and provides constant performance. This data structure is not relational and not taught in classes.	Time behavior Changeability Replaceability Adaptability
Data manipulation	The target data manipulation provides limited functionalities for online processing, but is excellent for offline processing. MapReduce is unknown to most developers.	Time behavior Analyzability Changeability Replaceability

Transforming Comparison Elements into Comparison Criteria (continued)

Comparison element	Impact	Related ISO 9126 quality subcharacteristics
Means to scale	Scalability is built into HBase, provides for easier reactions to new specifications, and does not require the system to be shut down.	Scalability Stability Adaptability
Where hardware reliability is handled	Reliability is also built in, so there is no need for a custom layer to provide it. The system is simplified.	Reliability Analyzability Changeability
How many systems are using it	The target system may perform badly if other systems are heavy users of the resources.	Time behavior

Summary

- While new technologies have been developed to address the scalability and reliability problems inherent to data-intensive systems, little has been done to validate their impacts on quality.
- As described, HBase use may negatively affect maintainability at the expense of faster processing of large datasets and better scalability.

Questions?