HBase on top of HDFS

Seminar Software Systems Engineering
"Mobile, Security, Cloud Computing"

Kevin Böckler

Institut für Telematik

February 19, 2015

Outline

Storage in Cloud Computing

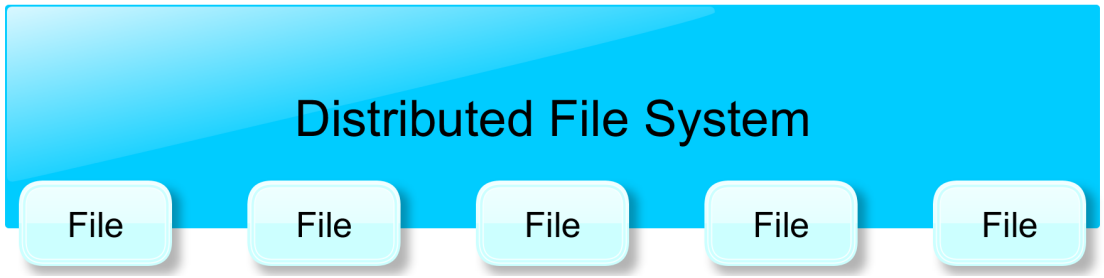
Requirements

- Millions of users
- Realtime access
- Reduce loss of data

Use cases

- Cloud Storage
- Collaborative Platforms
- Social Platforms
- Messengers

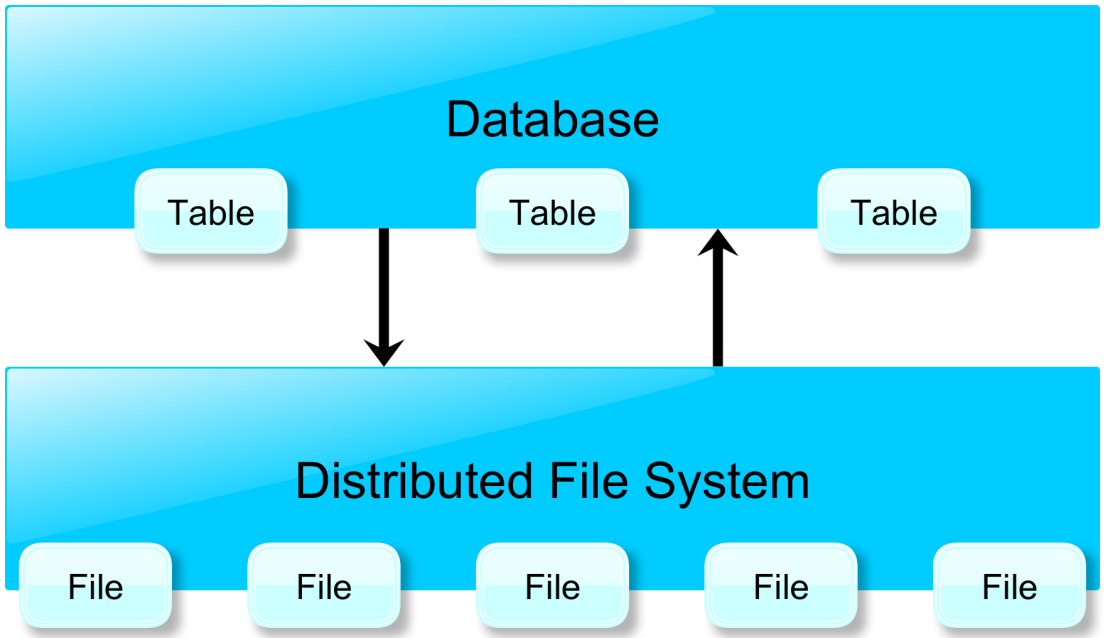
Hadoop implementation



HDFS = Hadoop Distributed File System

Hadoop implementation

HBase = Hadoop implementation of a database



HDFS = Hadoop Distributed File System

HDFS

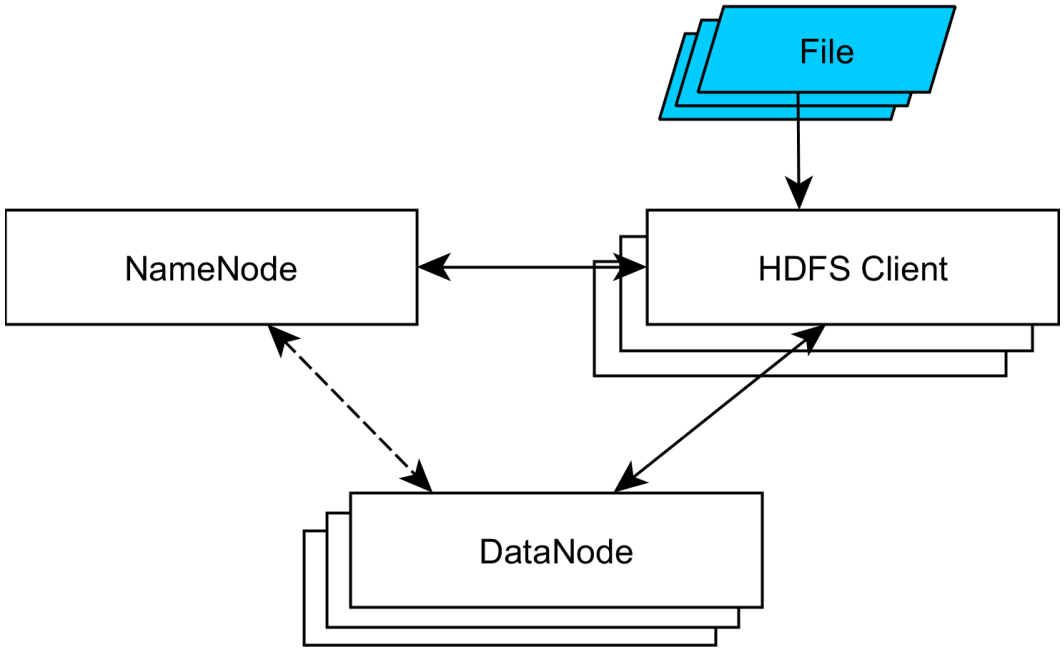


Figure: Participants in a HDFS Cluster

Properties of HDFS

Scalability

- Multiple Nodes distributed
- NameNode for Metadata, DataNode for actual payload
- ”Moving Computation is Cheaper than Moving Data”

Transparency

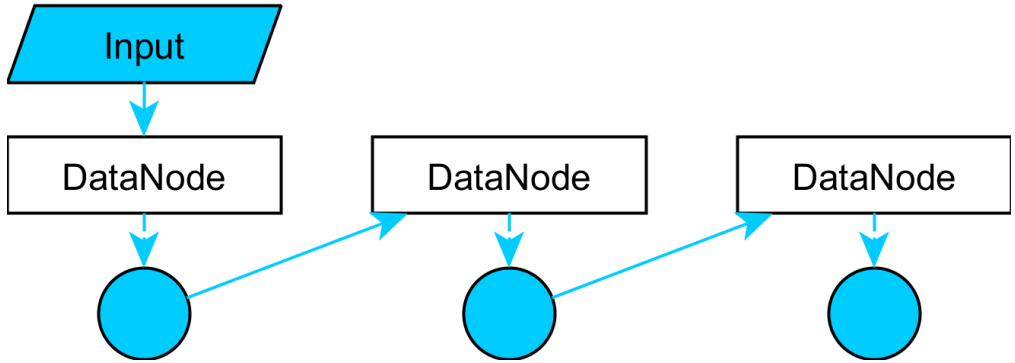
- UNIX paths (`/files/seminar/hbase.pdf`)
  - Location Transparency
  - Location Independency
- Hidden replication
- Hidden fail-over

Fault-Tolerance in HDFS

Replication

at write process: Pipelining

$\rightarrow$ Robustness



Availability

- Heartbeat (NameNode $\leftrightarrow$ DataNode)
- NameNode issues Re-replications

Fileaccess in HDFS

File Access

Write-once-read-many (WORM): Immutable

Remote Access

1. Ask the NameNode for *filename* $\rightarrow$ DataNode-Connection
2. Open connection to DataNode
3. Transfer payload

Features of HBase

Efficiency

- Bulk loading
- Sequential and Random reads
- Distributed MapReduce-Tasks

Scalability

- Column Families
- Concurrency Model: File Locks

Fault Tolerance

- inherited by HDFS
- Additionally Heartbeat (HBaseMaster $\leftrightarrow$ HRegionServer)

HBase: HRegionServer

Three Abstract Components

HRegionServer $\leftrightarrow$ HRegion $\leftrightarrow$ Store

- gets connection from HClient
- receives Table-Requests (GET, PUT, DELETE, ...) from HClient
- manages HRegions

HBase: HRegion

Three Abstract Components

HRegionServer $\leftrightarrow$ HRegion $\leftrightarrow$ Store

ID	a	b	c	d	e
1	42	1	world	9	hello
2	43	3	npe	9	hadoop
3	19	3	java	9	ping
4	22	7	easy	9	bye

- HRegion $\subseteq$ Table
- receives Requests from HRegionServer
- manages Stores
- Write-ahead-Log (WAL) of Column Writes
  ($\rightarrow$ eventually flushed to Store)

HBase: Store

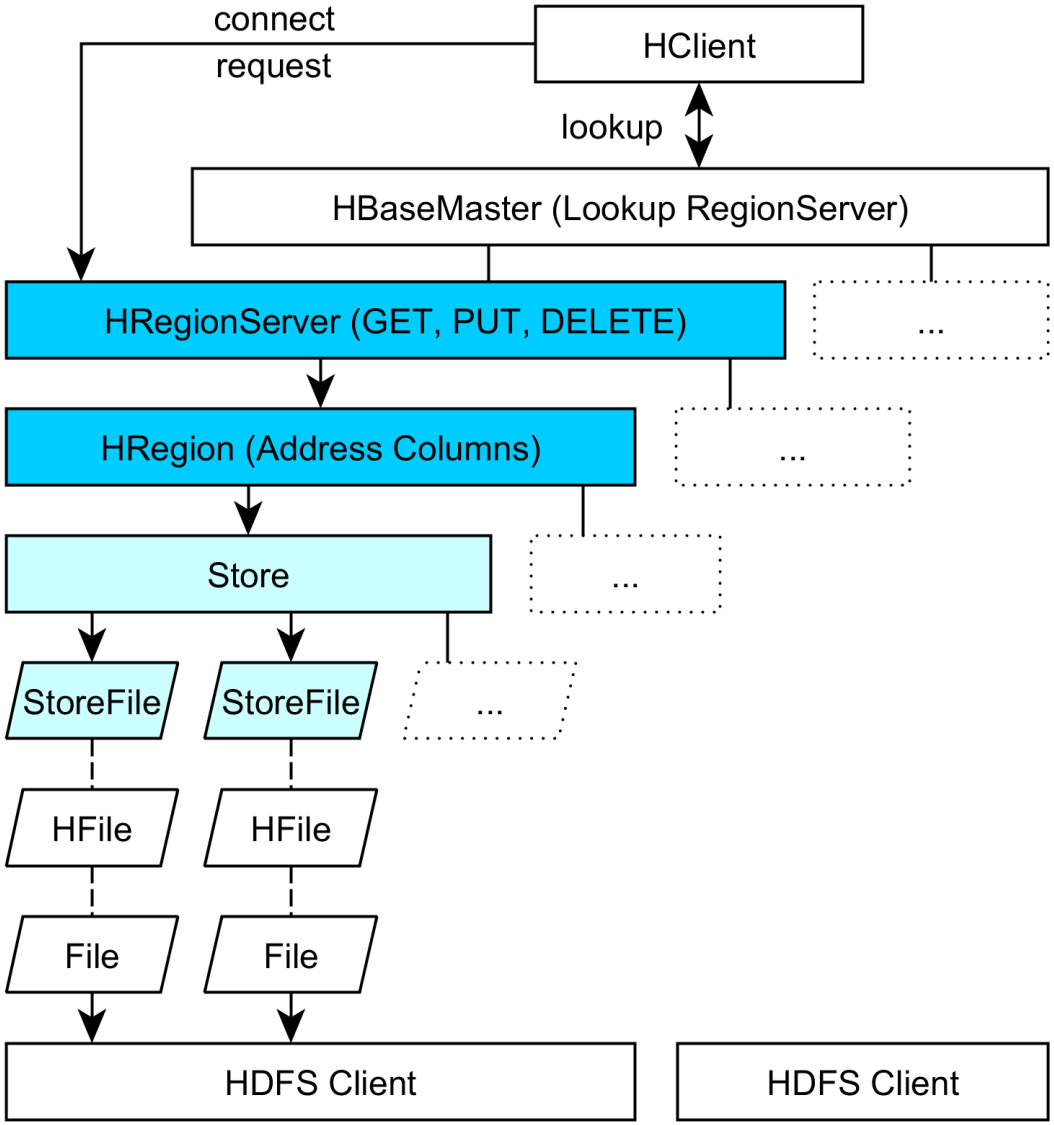
Three Abstract Components

HRegionServer $\leftrightarrow$ HRegion $\leftrightarrow$ Store

ID	a	b	c	d	e
1	42	1	world	9	hello
2	43	3	npe	9	hadoop
3	19	3	java	9	ping
4	22	7	easy	9	bye

- Store = ColumnFamily
- encapsulates one group of Columns and Rows
- holds its data in
  - MemStore (working Cache)
  - StoreFiles ($\rightarrow$ HFile $\rightarrow$ HDFS)
- compacts StoreFiles

Architecture of HBase



API

Java Usage

- Configuration

```
Configuration config = HBaseConfiguration.create();
config.set("hbase.zookeeper.quorum", "127.0.0.1");
config.set("hbase.zookeeper.property.clientPort", "2180");
```

- HTable

```
HTable table = new HTable(config, "someTableName");
```

- GET, SCAN, PUT, DELETE

```
Get get = new Get(Bytes.toBytes("someRowId"));
Result result = table.get(get);
```

- Filter

```
SingleColumnValueFilter filter = new SingleColumnValueFilter(
  someColumnFamily,
  someColumn,
  CompareOp.EQUAL,
  Bytes.toBytes("someColumnNameValue")
);
```

Demo

1. Using the HBase Shell
2. HDFS - Filesystem and Influence of Compactions
3. Using Java-Implementation

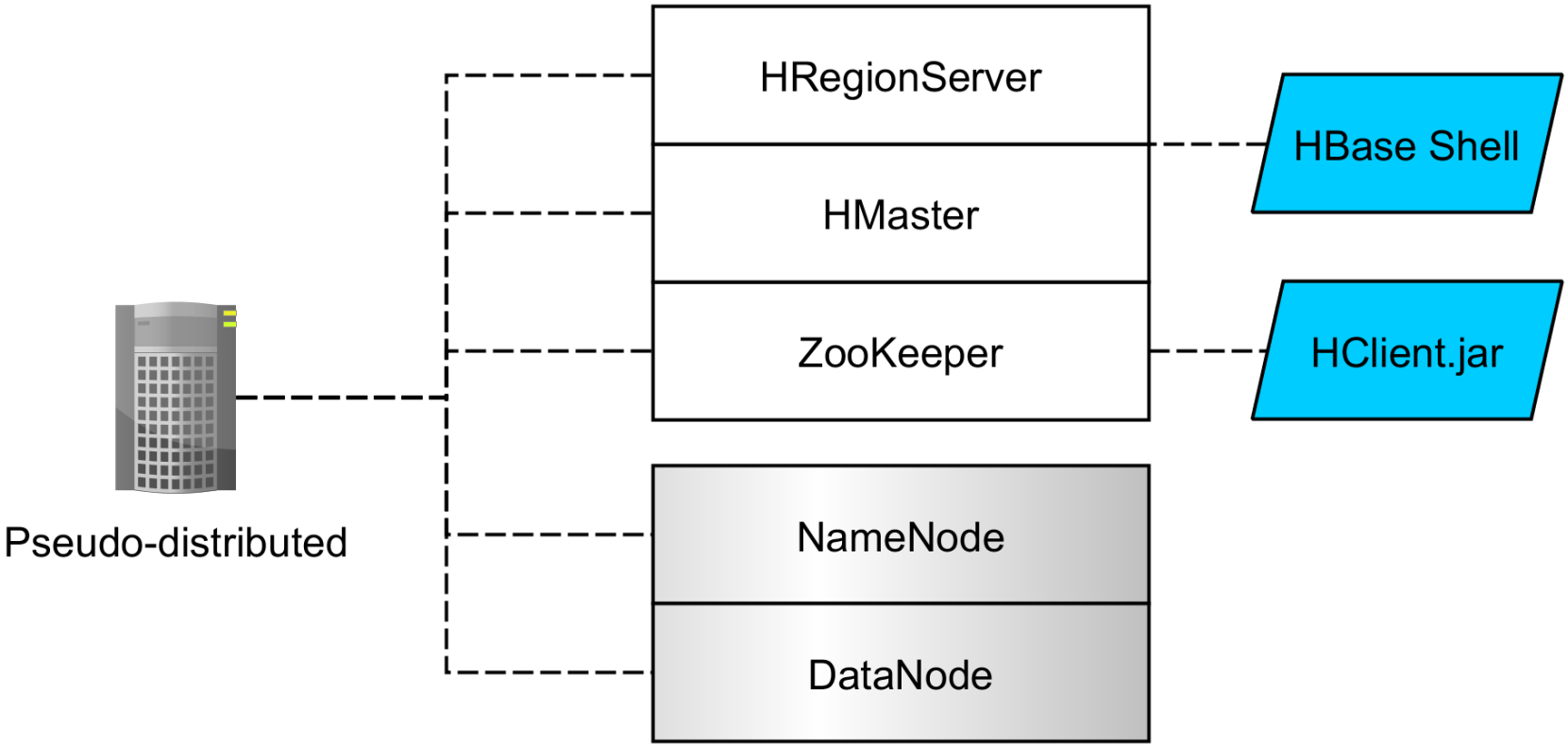


Figure: Process stack of the single-machine-cluster

Questions?

Outline
Storage in Cloud Computing
Hadoop implementation
Hadoop implementation
HDFS
Properties of HDFS
Fault-Tolerance in HDFS
Fileaccess in HDFS
Features of HBase
HBase: HRegionServer
HBase: HRegion
HBase: Store
Architecture of HBase
API
Demo
Questions?

HBase on top of HDFS

Seminar Software Systems Engineering
"Mobile, Security, Cloud Computing"

Kevin Böckler

Institut für Telematik

February 19, 2015