

Prototype of xrootd monitor with hadoop backend

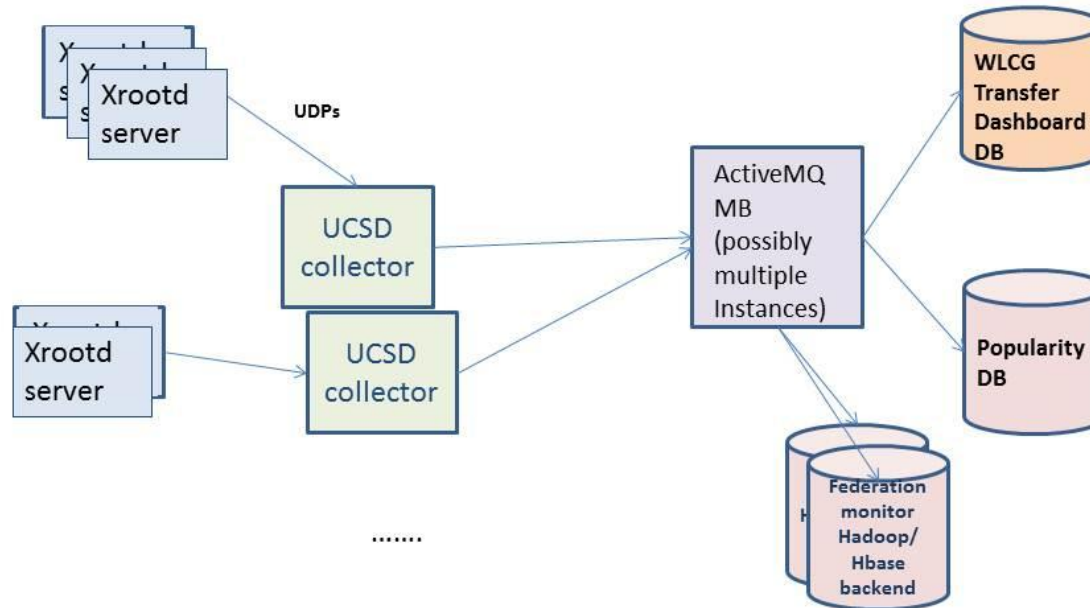
Sergey Mitsyn 2012-10-23

Plan

- 1) xrootd central monitor with Oracle backend intro
- 2) Hadoop and Hbase microintroduction
- 3) Experience and expectations
- 4) Current state of hadoop deployment

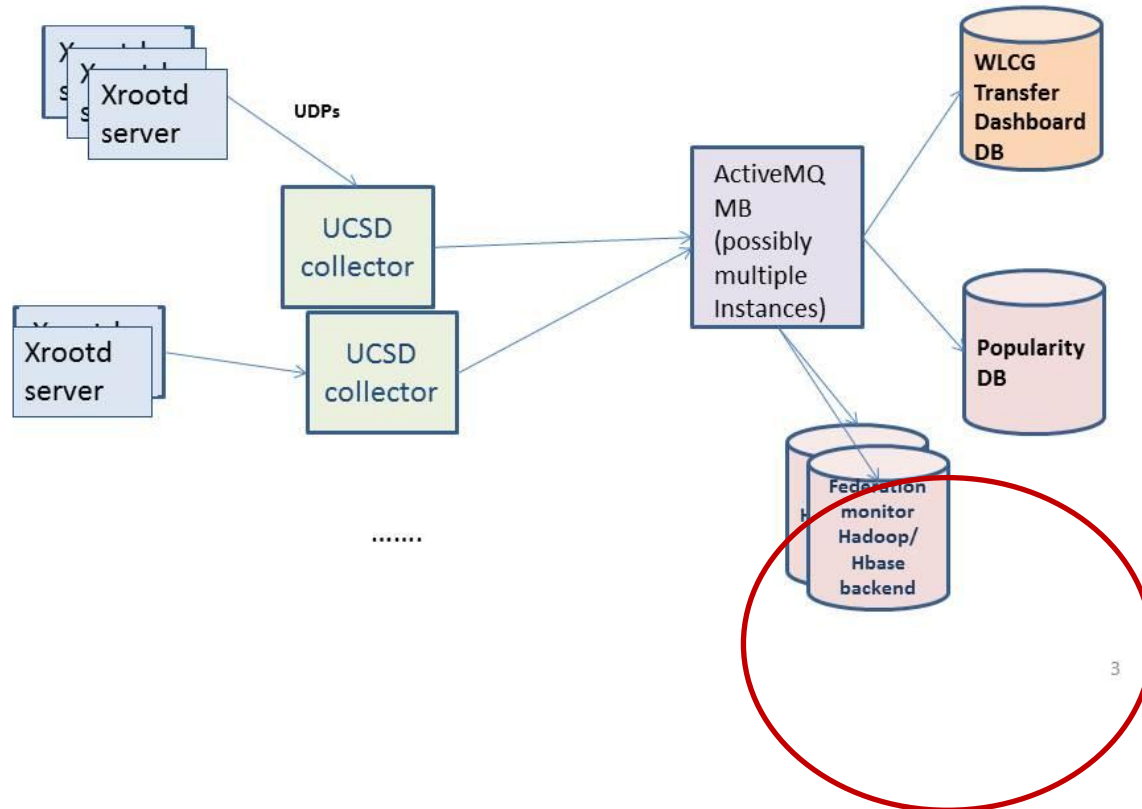
xrootd monitoring/central storage

Data flow for the xrootd monitoring applications

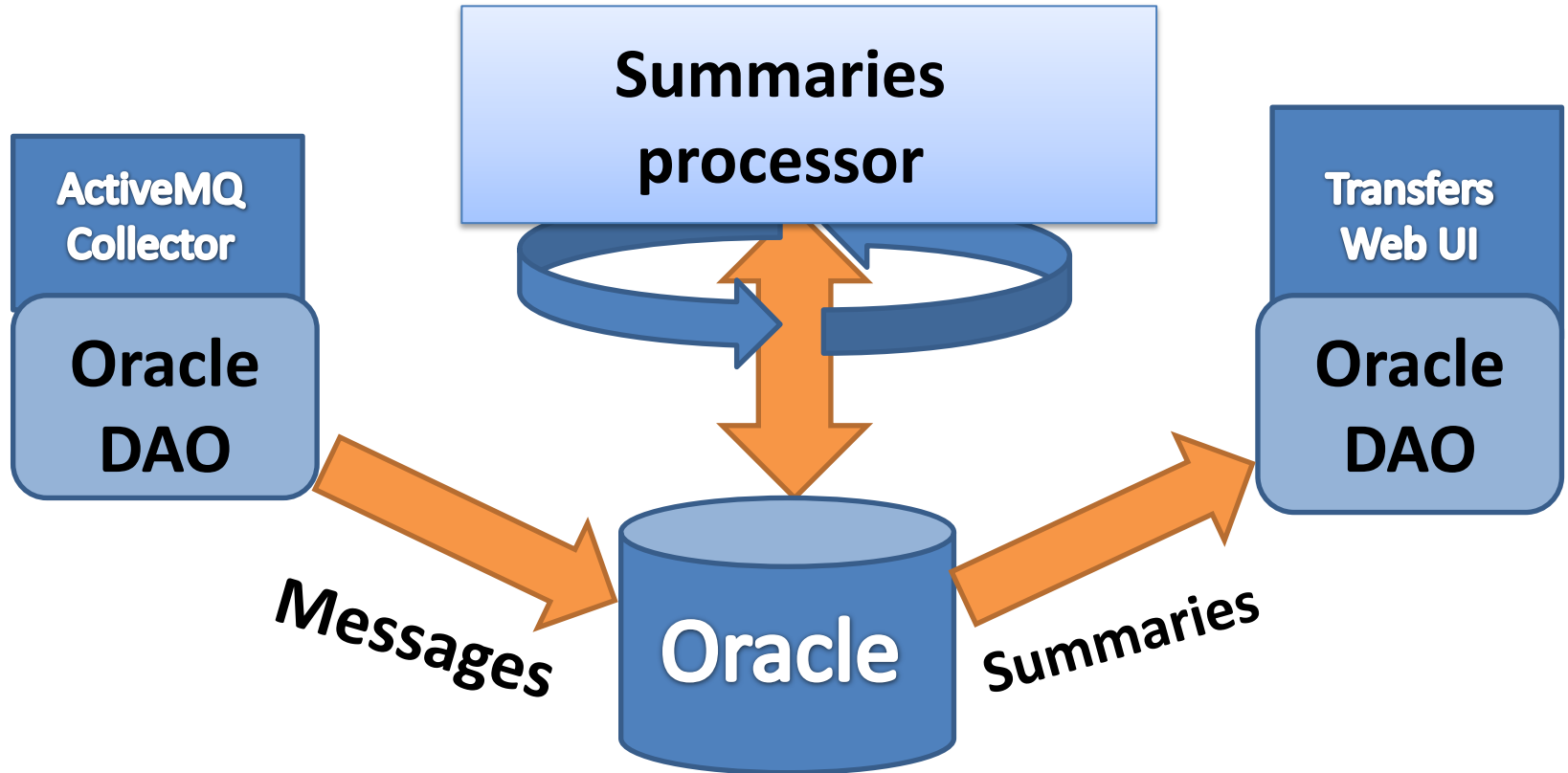


xrootd monitoring/central storage

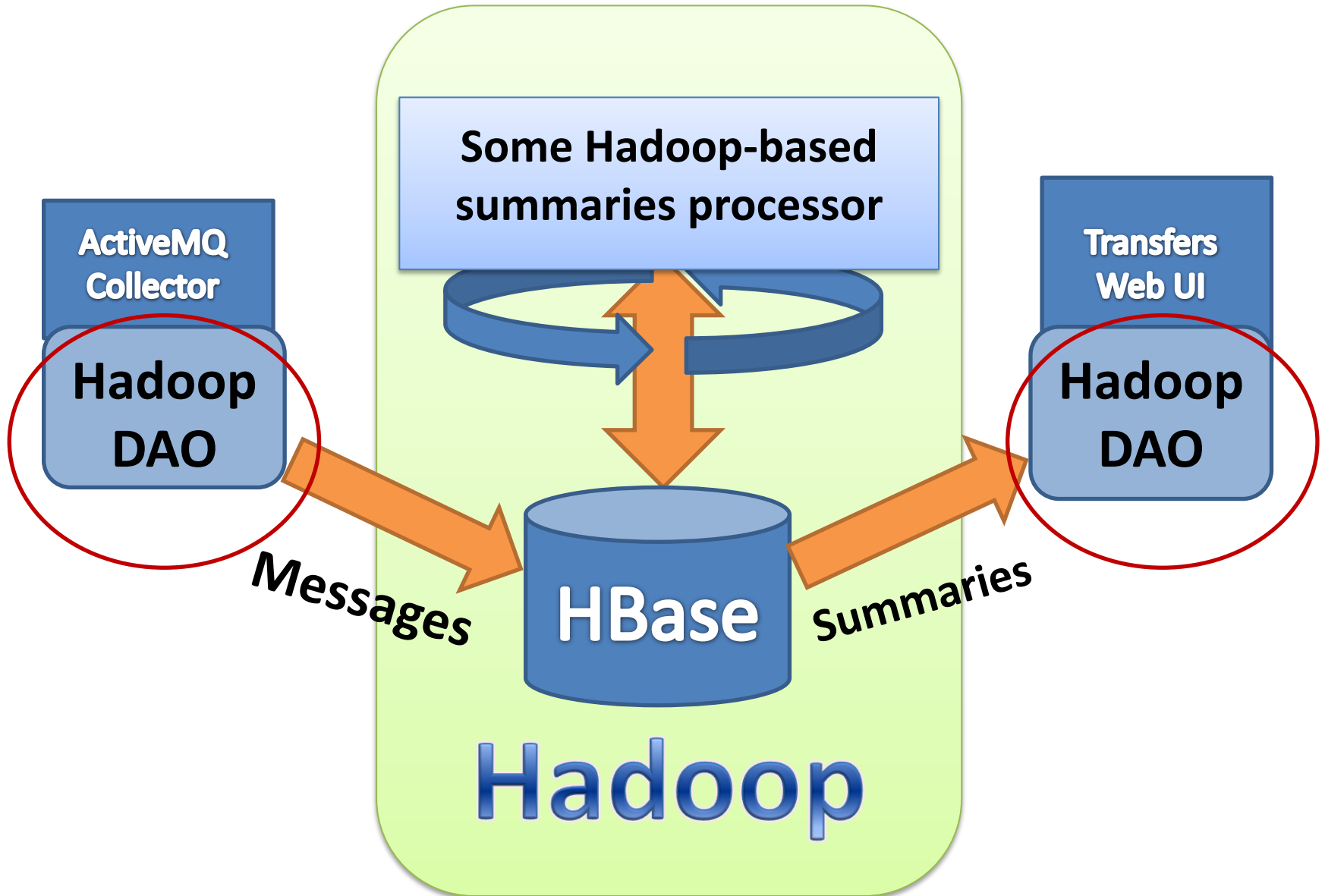
Data flow for the xrootd monitoring applications



xrootd monitoring/central storage



xrootd monitoring/central storage



Hadoop, a microintroduction

- Is an implementation of Google's Map/Reduce framework, a framework for distributed computing.
- Consists of multiple services, including:
 - HDFS – a distributed high-latency file system
 - HBase – a distributed low-latency columns-based database (based on HDFS)
 - Data storage.
 - mapreduce – a framework for Map/Reduce jobs (processing)
 - Data processing backend.

Hbase 1

- Column-based:
 - **Rowkey – Table – Column** denotes a single value.
 - Slightly more features than key-value based store.
 - All values are uninterpreted bytes;
 - Rowkey: global index, fast lookup;
 - Columns: independent values for single row.
- Operations:
 - Put/get/delete:
 - Single row, multiple columns, *atomic*
 - Batch put/delete:
 - Multiple rows, *non-atomic* across multiple rows
- Naïve usage:
 - Message ~ Row;
 - Message field ~ Column.

HBase 2

- Fully consistent (as in the CAP-theorem, single copy) (more strict than eventual consistency).
 - Though eventual consistency would be fine too:
 - The messages are never deleted, only inserted.
 - Summaries are updated (overwritten) without dependency on old values.
- **No** joins, **no** transactions, **no** global locks.
- No need of transactions:
 - Denormalized data -> no secondary keys:
 - Available atomic operations transfer database from one consistent state to another (in terms of invariant).

Messages table format

| Row Key | Single column family 'cf' | | | |
|--|--------------------------------------|-------------------------|-----|---------------------|
| A concatenation of "end_time" and sha1 hash of json string | Fields/columns <i>cf:end_time</i> | <i>cf:server_domain</i> | ... | <i>cf:file_size</i> |
| 1352000001:2e79ac6823b7d64 | 1352000001 | www.com | ... | 3133700 |
| 1352000001:a278399fb980a3cb | 1352000001 | www.com | ... | 7654321 |
| 1352000002:... | 1352000002 | cern.ch | ... | ... |

A concatenation of end_time and SHA1 hash of message enables us to:

- 1) Have unique row key for each message.
- 2) Have them sorted in "end_time" order for faster processing.

Processing

- Deduce source and destination from *server* and *client* addresses, bytes read and written.
- Group by fields
 - Source, destination, traffic type, etc
- Sum over transferred bytes.
- Write summaries to summaries table.

Processing 2

- Alternatives:
 - Map/Reduce:
 - code in Java ☹️
 - External processing: load data to python, process, put summaries back.
 - Does not scale well (but with current load it doesn't matter, also may utilize more sophisticated schemes).
 - Hadoop Hive, Pig:
 - Hadoop services that translate their own code (HiveQL or Pig Latin) to Map/Reduce classes to run locally or on cluster.
 - No to code in Java, yes to code in “strange” languages.
 - Provide ***Joins!*** But we don't need them anyway...

Processing 3

- Hive:
 - HiveQL: a language that strongly resembles SQL.
 - Tables, views, SELECT ... FROM ... GROUP BY ..., etc.
 - Supports User-Defined Functions (UDF) written in Python.
- Pig:
 - Pig Latin: a language that resembles SQL a little.
 - More procedural than HiveQL; relations.

Message storage formats

- Column-based:
 - One field – one value:
 - Fits the ActiveMQ message format.
 - Easy to work with Hive, Pig.
 - Big storage waste on metadata.
- “Flat” format:
 - Concatenate all values and put to single column.
 - Tightest storage utilization;
 - Still possible to work with Pig/Hive;
 - Dangerous in case of schema change.
- Json in single column:
 - Messages from collector are also in JSON format.
 - Preserves type info (no need to convert numbers to strings and back).
 - Impossible to work with Pig/Hive (or yes to code in Java).

Hbase external interfaces

2 protocols:

- REST:
 - pycurl/urllib is enough.
 - Cumbersome to use.
 - Buggy.
- Thrift:
 - Has external dependencies (*thrift-lib-python* and hbase bindings rpm, available in dashboard-externals repo for **current** HBase version).
 - Easy to use (happybase in SVN, adapted for python 2.4).
 - Binary (should be faster?).

Implemented

- DAO:
 - For collector: 100% (including 3 message formats)
 - For web UI: partly (transfers matrix and bins are ready)
- Processing:
 - In Python (external processing):
 - ready but not in SVN (will commit soon)
 - Pig, Hive:
 - In very early development stage; currently produce exactly the same results as python summaries processor.
 - To implement as script templates or use packages (PyPig)?
 - Results match these from Oracle backend.
 - ... more or less – some jobs are lost due to collector restarts.

More problems

- 1. Increasing keys for transfers are bad:
 - All insertion load is going to be on single node, and possibly summary queries also.
 - But we need timestamp-like keys for queries on time interval.
<http://ikaisays.com/2011/01/25/app-engine-datastore-tip-monotonically-increasing-values-are-bad/>
<http://hbase.apache.org/book/rowkey.design.html>
- 2. We need \geq **100 000 000** of entries for any improvements over RDBMS!
 - Currently no more than 100 000/day
<http://hbase.apache.org/book/architecture.html#arch.overview>
- 3. Optimizations ToDo: bloom filter, block caches, local data access...
<http://hbase.apache.org/book/perf.hdfs.configs.html>

Using HBase

- To begin learning, one should try HBase shell:
 - 1) Login to any host with HBase available (or install yourself)
 - See next slides...
 - 2) # hbase shell
 - 3) perform all of <http://hbase.apache.org/book/quickstart.html>
p.1.2.3 Shell Exercises.
 - Supports put, get, delete, create/drop table, etc...
 - Don't drop any of existing tables, please!

Local Hadoop installation

- Local installations for development:
 - Local or pseudo-distributed: every service on the same node.
 - Easy installation as in the Cloudera QuickStart guide:
 - <https://ccp.cloudera.com/display/CDH4DOC/CDH4+Quick+Start+Guide>
 - Nice to install for development.
 - Not suitable for application performance measurements (very low!).

The available Hadoop installations:

- CERN's at **lxfssm4401**:
 - *HDFS*: Namenode at lxfssm4401;
 - *Hbase*: master at the same host; *thrift*;
 - *mapred/YARN*: look tomorrow!
 - *Gateway* at **dashboard48**.
 - (just a node with software and configuration, but no data or mapred tasks)
- Dashboard's at **dashboard07**:
 - 4 hosts: 07, 08, 64, 65.
 - (currently) limited reliability:
 - Crash of dashboard07 stops everything (no SecondaryNameNode)
 - Crash of any 2 of 3 other machines at the same time is tolerable.
 - *mapred, HDFS, Pig, Hive, HBase* with *thrift* and *REST* interfaces;
 - Unstable ☹️
 - Future: development only? Deprecate & remove?

Configuring Hadoop: fail

```
smitsyn@lxplus443:~$
api-2.5.jar:/usr/lib/hadoop/lib/jackson-xc-1.8.8.jar:/usr/lib/hadoop/lib/mockito-all-1.8.5.jar:/usr/lib/hadoop/lib/commons-logging-1.1.1.jar:/usr/lib/hadoop/lib/jackson
-core-asl-1.8.8.jar:/usr/lib/hadoop/lib/jsr305-1.3.9.jar:/usr/lib/hadoop/lib/commons-lang-2.5.jar:/usr/lib/hadoop/lib/commons-httpclient-3.1.jar:/usr/lib/hadoop/lib/sna
ppy-java-1.0.4.1.jar:/usr/lib/hadoop/lib/junit-4.8.2.jar:/usr/lib/hadoop/lib/commons-net-3.1.jar:/usr/lib/hadoop/lib/commons-beanutils-core-1.8.0.jar:/usr/lib/hadoop/li
b/slf4j-log4j12-1.6.1.jar:/usr/lib/hadoop/lib/jetty-util-6.1.26.cloudera.2.jar:/usr/lib/hadoop/lib/jersey-server-1.8.jar:/usr/lib/hadoop/lib/log4j-1.2.17.jar:/usr/lib/h
adoop/lib/commons-cli-1.2.jar:/usr/lib/hadoop/lib/jets3t-0.6.1.jar:/usr/lib/hadoop/lib/protobuf-java-2.4.0a.jar:/usr/lib/hadoop/lib/commons-io-2.1.jar:/usr/lib/hadoop/l
ib/jetty-6.1.26.cloudera.2.jar:/usr/lib/hadoop/lib/jsch-0.1.42.jar:/usr/lib/hadoop/lib/jaxb-api-2.2.2.jar:/usr/lib/hadoop/lib/commons-el-1.0.jar:/usr/lib/hadoop/lib/jac
kson-jaxrs-1.8.8.jar:/usr/lib/hadoop/lib/jsp-api-2.1.jar:/usr/lib/hadoop/lib/commons-configuration-1.6.jar:/usr/lib/hadoop/lib/jasper-compiler-5.5.23.jar:/usr/lib/hadoo
p/lib/stax-api-1.0.1.jar:/usr/lib/hadoop/lib/kfs-0.3.jar:/usr/lib/hadoop/lib/commons-digester-1.8.jar:/usr/lib/hadoop/lib/jasper-runtime-5.5.23.jar:/usr/lib/hadoop/lib/
commons-beanutils-1.7.0.jar:/usr/lib/hadoop/lib/xmlenc-0.52.jar:/usr/lib/hadoop/lib/jackson-mapper-asl-1.8.8.jar:/usr/lib/hadoop/lib/commons-collections-3.2.1.jar:/usr/
lib/hadoop/lib/commons-math-2.1.jar:/usr/lib/hadoop/lib/commons-codec-1.4.jar:/usr/lib/hadoop/lib/jersey-core-1.8.jar:/usr/lib/hadoop/lib/asm-3.2.jar:/usr/lib/hadoop/li
b/guava-11.0.2.jar:/usr/lib/hadoop/lib/paranamer-2.3.jar:/usr/lib/hadoop/lib/activation-1.1.jar:/usr/lib/hadoop/lib/hadoop-common-2.0.0-cdh4.1.0.jar:/usr/lib/hadoop/lib/h
adoop-annotations.jar:/usr/lib/hadoop/lib/hadoop-common-2.0.0-cdh4.1.0-tests.jar:/usr/lib/hadoop/lib/hadoop-auth-2.0.0-cdh4.1.0.jar:/usr/lib/hadoop/lib/hadoop-auth.jar:/us
r/lib/hadoop/lib/hadoop-annotations-2.0.0-cdh4.1.0.jar:/usr/lib/hadoop/lib/hadoop-common.jar:/etc/hbase/etc/hbase/conf:/usr/java/jdk1.6.0_34/lib/tools.jar:/usr/lib/hbase
/bin/./conf:/usr/java/jdk1.6.0_34/lib/tools.jar:/usr/lib/hbase/bin/./usr/lib/hbase/bin/./hbase-0.92.1-cdh4.1.0-security.jar:/usr/lib/hbase/bin/./hbase-0.92.1-cdh4.
1.0-security-tests.jar:/usr/lib/hbase/bin/./hbase.jar:/usr/lib/hbase/bin/./lib/activation-1.1.jar:/usr/lib/hbase/bin/./lib/aopalliance-1.0.jar:/usr/lib/hbase/bin/./lib/
lib/asm-3.2.jar:/usr/lib/hbase/bin/./lib/avro-1.7.1.cloudera.2.jar:/usr/lib/hbase/bin/./lib/commons-beanutils-1.7.0.jar:/usr/lib/hbase/bin/./lib/commons-beanutils-co
re-1.8.0.jar:/usr/lib/hbase/bin/./lib/commons-cli-1.2.jar:/usr/lib/hbase/bin/./lib/commons-codec-1.4.jar:/usr/lib/hbase/bin/./lib/commons-collections-3.2.1.jar:/usr/
lib/hbase/bin/./lib/commons-configuration-1.6.jar:/usr/lib/hbase/bin/./lib/commons-daemon-1.0.3.jar:/usr/lib/hbase/bin/./lib/commons-digester-1.8.jar:/usr/lib/hbase/
bin/./lib/commons-el-1.0.jar:/usr/lib/hbase/bin/./lib/commons-httpclient-3.1.jar:/usr/lib/hbase/bin/./lib/commons-io-2.1.jar:/usr/lib/hbase/bin/./lib/commons-lang-2
.5.jar:/usr/lib/hbase/bin/./lib/commons-logging-1.1.1.jar:/usr/lib/hbase/bin/./lib/commons-net-3.1.jar:/usr/lib/hbase/bin/./lib/core-3.1.1.jar:/usr/lib/hbase/bin/./li
b/gmbal-api-only-3.0.0-b023.jar:/usr/lib/hbase/bin/./lib/grizzly-framework-2.1.1.jar:/usr/lib/hbase/bin/./lib/grizzly-framework-2.1.1-tests.jar:/usr/lib/hbase/bin/
./lib/grizzly-http-2.1.1.jar:/usr/lib/hbase/bin/./lib/grizzly-http-server-2.1.1.jar:/usr/lib/hbase/bin/./lib/grizzly-http-servlet-2.1.1.jar:/usr/lib/hbase/bin/./lib/
grizzly-rcm-2.1.1.jar:/usr/lib/hbase/bin/./lib/guava-11.0.2.jar:/usr/lib/hbase/bin/./lib/guice-3.0.jar:/usr/lib/hbase/bin/./lib/guice-servlet-3.0.jar:/usr/lib/hbase/
bin/./lib/high-scale-lib-1.1.1.jar:/usr/lib/hbase/bin/./lib/httpclient-4.0.1.jar:/usr/lib/hbase/bin/./lib/httpcore-4.0.1.jar:/usr/lib/hbase/bin/./lib/jackson-core-a
sl-1.8.8.jar:/usr/lib/hbase/bin/./lib/jackson-jaxrs-1.8.8.jar:/usr/lib/hbase/bin/./lib/jackson-mapper-asl-1.8.8.jar:/usr/lib/hbase/bin/./lib/jackson-xc-1.8.8.jar:/us
r/lib/hbase/bin/./lib/jamon-runtime-2.3.1.jar:/usr/lib/hbase/bin/./lib/jasper-compiler-5.5.23.jar:/usr/lib/hbase/bin/./lib/jasper-runtime-5.5.23.jar:/usr/lib/hbase/b
in/./lib/javax.inject-1.jar:/usr/lib/hbase/bin/./lib/javax.servlet-3.0.jar:/usr/lib/hbase/bin/./lib/jaxb-api-2.1.jar:/usr/lib/hbase/bin/./lib/jaxb-impl-2.2.3-1.jar/
usr/lib/hbase/bin/./lib/jersey-client-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-core-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-grizzly2-1.8.jar:/usr/lib/hbase/bin/./li
b/jersey-guice-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-json-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-server-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-test-framework-cor
e-1.8.jar:/usr/lib/hbase/bin/./lib/jersey-test-framework-grizzly2-1.8.jar:/usr/lib/hbase/bin/./lib/jets3t-0.6.1.jar:/usr/lib/hbase/bin/./lib/jettison-1.1.jar:/usr/li
b/hbase/bin/./lib/jetty-6.1.26.cloudera.2.jar:/usr/lib/hbase/bin/./lib/jetty-util-6.1.26.cloudera.2.jar:/usr/lib/hbase/bin/./lib/jruby-complete-1.6.5.jar:/usr/lib/hb
ase/bin/./lib/jsch-0.1.42.jar:/usr/lib/hbase/bin/./lib/jsp-2.1-6.1.14.jar:/usr/lib/hbase/bin/./lib/jsp-api-2.1-6.1.14.jar:/usr/lib/hbase/bin/./lib/jsp-api-2.1.jar/
usr/lib/hbase/bin/./lib/jsr305-1.3.9.jar:/usr/lib/hbase/bin/./lib/kfs-0.3.jar:/usr/lib/hbase/bin/./lib/libthrift-0.7.0.jar:/usr/lib/hbase/bin/./lib/log4j-1.2.17.jar
:/usr/lib/hbase/bin/./lib/management-api-3.0.0-b012.jar:/usr/lib/hbase/bin/./lib/metrics-core-2.1.2.jar:/usr/lib/hbase/bin/./lib/netty-3.2.4.Final.jar:/usr/lib/hbase/
bin/./lib/paranamer-2.3.jar:/usr/lib/hbase/bin/./lib/protobuf-java-2.4.0a.jar:/usr/lib/hbase/bin/./lib/servlet-api-2.5-6.1.14.jar:/usr/lib/hbase/bin/./lib/servlet-
api-2.5.jar:/usr/lib/hbase/bin/./lib/slf4j-api-1.6.1.jar:/usr/lib/hbase/bin/./lib/snappy-java-1.0.4.1.jar:/usr/lib/hbase/bin/./lib/stax-api-1.0.1.jar:/usr/lib/hbase/
bin/./lib/xmlenc-0.52.jar:/usr/lib/hbase/bin/./lib/zookeeper.jar:/etc/hadoop/conf/*:/lib/*:/usr/lib/zookeeper/zookeeper-3.4.3-cdh4.1.0.jar:/usr/lib/zookeeper/zookeep
er.jar:/usr/lib/zookeeper/lib/jline-0.9.94.jar:/usr/lib/zookeeper/lib/slf4j-api-1.6.1.jar:/usr/lib/zookeeper/lib/netty-3.2.2.Final.jar:/usr/lib/zookeeper/lib/slf4j-log4
j12-1.6.1.jar:/usr/lib/zookeeper/lib/log4j-1.2.15.jar:/etc/hadoop/conf:/usr/lib/hadoop/lib/zookeeper-3.4.3-cdh4.1.0.jar:/usr/lib/hadoop/lib/jersey-json-1.8.jar:/usr/li
b/hadoop/lib/jline-0.9.94.jar:/usr/lib/hadoop/lib/slf4j-api-1.6.1.jar:/usr/lib/hadoop/lib/jettison-1.1.jar:/usr/lib/hadoop/lib/avro-1.7.1.cloudera.2.jar:/usr/lib/hadoop
/lib/jaxb-impl-2.2.3-1.jar:/usr/lib/hadoop/lib/servlet-api-2.5.jar:/usr/lib/hadoop/lib/jackson-xc-1.8.8.jar:/usr/lib/hadoop/lib/mockito-all-1.8.5.jar:/usr/lib/hadoop/li
b/commons-logging-1.1.1.jar:/usr/lib/hadoop/lib/jackson-core-asl-1.8.8.jar:/usr/lib/hadoop/lib/jsr305-1.3.9.jar:/usr/lib/hadoop/lib/commons-lang-2.5.jar:/usr/lib/hadoop
/lib/commons-httpclient-3.1.jar:/usr/lib/hadoop/lib/snappy-java-1.0.4.1.jar:/usr/lib/hadoop/lib/junit-4.8.2.jar:/usr/lib/hadoop/lib/commons-net-3.1.jar:/usr/lib/hadoop/
```

Hadoop Gateway

- Is simply a node without any data or mapred services running,
 - ... but may run any data access services like REST or Thrift;
 - may run Java applications for Hadoop.
- All you need are:
 - client packages (e.g. yum install hbase-rest)
 - Copy-paste HBase config (I hope someday you may find it in Dashboard wiki)
- Why you may need that for:
 - An HBase shell without ssh access to the cluster;
 - To run arbitrary service.

The end

- Questions?