

CS 378 – Big Data Programming

Lecture 27

Hadoop Ecosystem

Assignment 12

- Compute statistics on price ranges
- Utilize multiple patterns/techniques
 - Filtering, inverted index
 - Reduce-side join
 - Summarization
 - Job chaining
- Questions/issues

Hadoop Ecosystem

- Many other tools have been implemented on
 - Hadoop
 - HDFS (Hadoop Distributed File System)
- We'll discuss a few
 - HBase
 - ZooKeeper
 - Pig, Impala
 - Hive

HBase

- Column-orient database
 - Implemented on top of HDFS
 - Distributed
- Goal is to scale to very large datasets
 - With real-time read/write access

Column-oriented Database

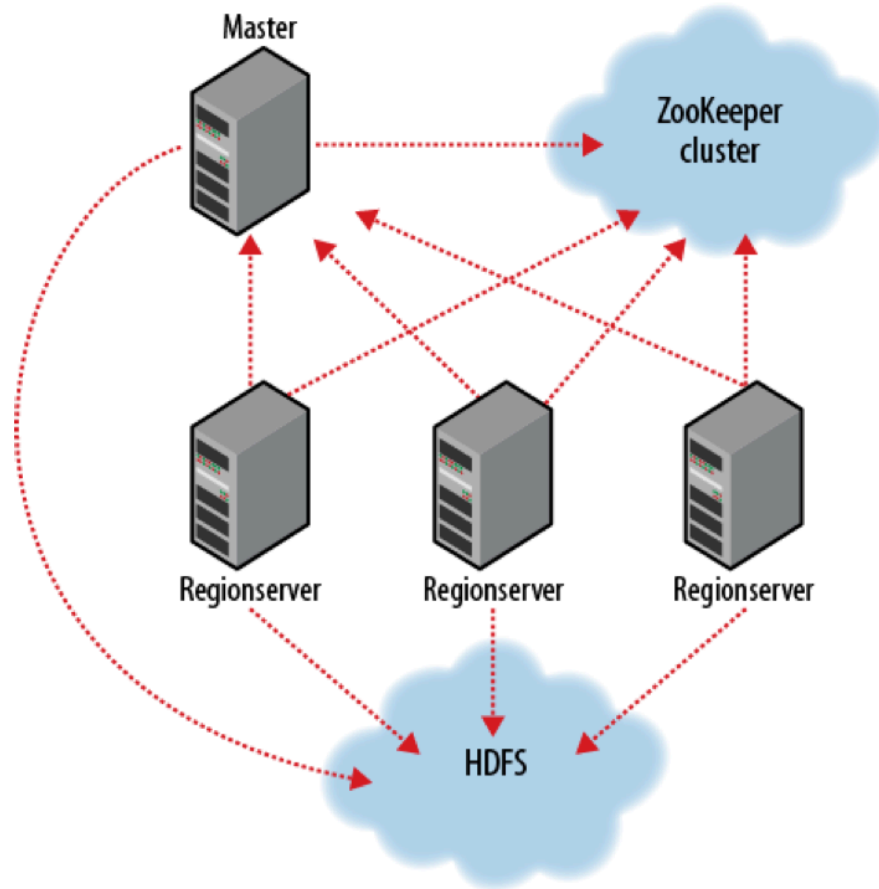
- Table cells are the unit of access
 - Content is uninterpreted array of bytes
 - A cell versioned (can have multiple versions)
- Table cell is accessed by
 - Row, column, and version (often a timestamp)
- Columns are grouped into families

Column-oriented Database

- New column family members can be added
- Column family members are stored together
- For best performance, family members should be accessed together
- Rows can be subset into regions

Hbase

Figure 13-1 from Hadoop The Definitive Guide



ZooKeeper

- Messaging and synchronization in a distributed environment
 - Distributed queues, locks
 - Leader election among a group of peers
- High availability (tolerates failures)
- Loosely coupled interactions
 - Rendezvous mechanism

Pig

- Higher level data structures and operations
 - Higher level than Java code for map-reduce job
- Language: Pig Latin
 - Operations and transformations on data
 - Pig converts these to map-reduce jobs for you
- Think of it as a query language for data in HDFS

Pig Examples

Filtering

SQL

The filter pattern is synonymous to using the WHERE clause in a SELECT * statement. The records stay the same, but some are simply filtered out. For example:

```
SELECT * FROM table WHERE value < 3;
```

Pig

The FILTER keyword.

```
b = FILTER a BY value < 3;
```

Pig Examples

Top 10

SQL

In a traditional and small SQL database, ordering may not be a big deal. In this case, you would retrieve data ordered by the criterion for which you want the top ten, then take a limit. You could follow this same approach in MapReduce, but as you will find out in later patterns, sorting is an expensive operation.

```
SELECT * FROM table ORDER BY col4 DESC LIMIT 10;
```

Pig

Pig will have issues performing this query in any sort of optimal way. The most straightforward pattern is to mirror the SQL query, but the ordering is expensive just to find a few records. This is a situation in which you'll find major gains in using Java MapReduce instead of Pig.

```
B = ORDER A BY col4 DESC;
```

```
C = LIMIT B 10;
```

Pig Examples

Distinct

SQL

SELECT DISTINCT performs this operation for us in SQL.

```
SELECT DISTINCT * FROM table;
```

Pig

The DISTINCT operation.

```
b = DISTINCT a;
```

Pig Examples

Binning

Pig

The SPLIT operation in Pig implements this pattern.

```
SPLIT data INTO  
  eights IF col1 == 8,  
  bigs IF col1 > 8,  
  smalls IF (col1 < 8 AND col1 > 0);
```

Pig Examples

Sorting

SQL

Ordering in SQL is pretty easy!

```
SELECT * FROM data ORDER BY col1;
```

Pig

Ordering in Pig is syntactically pretty easy, but it's a very expensive operation. Behind the scenes, it will run a multi-stage MapReduce job to first find the partitions, and then perform the actual sort.

```
c = ORDER b BY col1;
```

Pig Examples

Sorting

Pig

Pig has native support for a replicated join through a simple modification to the standard join operation syntax. Only inner and left outer joins are supported for replicated joins, for the same reasons we couldn't do it above. The order of the data sets in the line of code matters because all but the first data sets listed are stored in-memory.

```
huge = LOAD 'huge_data' AS (h1,h2);
smallest = LOAD 'smallest_data' AS (ss1,ss2);
small = LOAD 'small_data' AS (s1,s2);
A = JOIN huge BY h1, small BY s1, smallest BY ss1 USING 'replicated';
```


Impala

- Interactive SQL for data in HDFS, HBase
- SQL processing engine
 - Parallel execution
 - Horizontal scaling
- Runs on each data node
 - Direct access to HDFS, HBase (no map-reduce)

Hive

- Data warehouse on top of Hadoop
- SQL for access
 - Hive converts a query into a map-reduce steps
- Various Hive clients are available
 - JDBC, ODBC, Thrift, ...