COMPSCI 590S Systems for Data Science **Fall 2016**

Lecture 5: Databases vs. MapReduce

Lecturer: Emery Berger *Scribe(s): Casey Ray, Jon Karlen*

5.1 Review of MapReduce and Database performance

We reviewed papers from two different camps: From google regarding MapReduce, and those in favor of databases. We discussed the performance of databases, MapReduce, and writing jobs for those two paradigms.

5.1.1 Databases

There are row and column store databases. Databases operate in an ACID fashion (Atomicity, Consistency, Isolation, Durability) via transactions. Prior to SQL(Structured Query Language(was QUEL, which became SEQUEL, then SQL. Sql is a declarative language. Which means, in a manner of speaking, that the programmer states what they want done, and behind the scenes the interpreter determines the step to complete the task(s). An example database with an imperative program instead of a declaritive query.

Income	Name	Age
23000	Brown	29
		..
		..

```
it = db.begin();
count = 0;
for( , it!db.end(); it++)
   {
      if(it.getField("income")>100000){
         count++;
      }
   }
```

A query in a declaritive language will have a much different set of steps. For a query such as an insert, one might use a binary search to locate the correct location to insert a new record based on a given index or set of indices. This index could be represented as a B-tree. The result of these differences is that for tasks commonly done with a declaritive language like SQL, the imperative version of that program is much harder to write in a performative manner. Imperative languages can't hide details like index creation. C# and link use SQL-like code in an imperative language to avoid the problems of imperative code.

There is a standard for SQL, but each vendor has its own dialect of SQL. A SQL query will be of the general form:

```
SELECT...
FROM...
WHERE...
```

An example query would be:

```
SELECT (*)
FROM employee
WHERE employee.income > 100000;
```

When using SQL, one must be very explicit about the contents of the database. To do those, one specifies the content type of each column of each table in each database in what is known as a schema. The schema must be made prior to the creation of the table.

DBMS's don't have the advantage when it comes to semi-structured data. XML is a disaster. It encompasses HTML, but HTML on the web is rarely HTML compliant. Browsers tend to interpret HTML. MR can be a better fit in these cases because it is difficult to put webpages into databases.

5.1.2 Indeces

Indeces can index any field in a DBMS. They can speed up accessing data in tables. Updates need to propagate to indeces, so indeces increase the time it takes to make updates to a database. Problems with DBMSs come from scaling and updating indeces. However, updates are hard with key-value stores. Deadlock can easily occur.

5.1.3 row based vs column based

A database may be row based or column based. This is an implementation detail about the orientation of the data as it is stored on disk. A row based database will store an entire row congiuously on disk. This leads to a performance advantage when pulling an entire record or set of records from disk, as well as being straightforward to implement. This is also an advantage when distributing the database across multiple machines.
A column based database stores an entire column contiguously on disk. When accessing or appending entire rows, this will be slower than a row oriented database. However, there are significantly smaller read costs for certain access patterns (such as find rows where there is some condition). The database can avoid reading significant amounts of data compared to row based databases.
Whether the database is row or column based, it will have a query optimizer (excepet for SQLITE) which will interpret the query, and then optimize the directed steps into the smallest number of steps which it calls a plan, and then executes the plan.

5.1.4 round robin vs hash distributed

Tables can be distributed on different machines in different ways. Round robin distributed tables distribute data evenly without a hash function, while hash distributed tables distribute randomly, and have better load balance.

5.1.5 NoSQL

NoSQL databases (of which MapReduce is one) use imperative languages. They might also be called a key-value store. They implement DHT (Distributed hash tables) with CHORD (consistent hashing). An advantage of this is the ability to scale. However, scaling can cause problems with updating indices.