EXC Notes
S0936300

November 9, 2013

1 MapReduce

Map reduce is a programming model for processing large data sets with a parallel, distributed algorithm cluster. It is inspired by map and reduce functions used in functional programming although the functionality differs in that it is not soley used for the ability to map and reduce, but due to scalability and fault-tolerance achieved for a variety of applications by optimizing the execution engine once [1][2].

Properties:

1. Assumes large numbers of cheap, commodity machines
2. Failure is part of life
3. Tailored for dealing with Big Data
4. Simple
5. Scales well

Who uses it?

1. Google, Facebook, Twitter
2. IBM
3. Amazong Web Services
4. Edinburgh
5. Many small start-ups

MR has generated a lot of interest. It solve all scaling problems, google use it (so it must be great!), start-ups love it and they generate a lot of chatter in the Tech press(big companies use DBs and they don't talk about them). Who needs complicated, expensive DB's anyway?

Map reduce is composed of the Map() function which performs filtering and sorting as well as the Reduce() function which performs a summary operation(such as counting the number of students in each queue).

1. Map(): Master node takes input, divides into smaller sub-problems and distributes these to the workers. Workers can sub-divide again. Once a worker has it's answer, it passes this to the master node.

2. Reduce(): Master node then collects answers to all the sub-problems and combines to form the output.

1.1 Critique

Considerations as to whether MR can replace parallel databases. Parallel databases have been in development for over 20 years, they are robust, fast, scalable and based upon declarative data models.

MR is not really suited for low-latency problems as it's batch nature and lack of real-time guarentees means you shouldn't use it for front-end tasks.

MR is not a good fit for problems which need global state information. Many Machine Learning algorithms require maintenance of centralised information and this implies a single task.

Which application classes might MR be a better choice than a P-DB?

- Extract-transform-load problems
- Complex analytics
- Semi-structured data(no single scheme for the data, I.e logs from multiple sources)
- Quick and dirty analyses

Results indicated

- For a range of core tasks, P-DB was faster than Hadoop. P-DBS are flexible enough to deal with semi-structured data (unclear whether this is implementation specific)

- Hadoop was criticised as being too low-level
- Hadoop was easier for quick-and-dirty tasks.
  - Writing MR jobs can be easier than complex SQL queries.
  - Non-specialists can quickly write MR jobs
- Hadoop is a lot cheaper

2 Other concepts

2.1 BigTable

BigTable is a form of Database.

3 Definitions

Definition: Cluster

Large number of nodes(computers) that are on the same local network and use the same hardware

Definition: Moore's law

Computing power doubles every 18 months.

Definition: Kryder's law

Storage is growing even faster than Moore's law

Definition: Cell

A grouping of servers, admins, users and clients

4 Questions

Question: Do we use Hadoop with MapReduce?

MapReduce libraries have been written in many programming languages with different levels of optimization. A popular open-source implementation is Apache Hadoop [1]

References

[1] Wikipedia

[2] Re-write and remove detailed garble when know more on the topic.