Summary Section 2.4 Extensions to MapReduce

This section introduces extensions and modifications of Hadoop MapReduce system. The most popular systems include UC Berkeley’s Spark., Google’s TensorFlow, and a graph model of data, Google’s Pregel.

## Workflow System

These extensions are all use a “Workflow” architecture. They share three major characteristics with MapReduce.

1. Build on a distributed file system.
2. Manage tremendous tasks, whereas only need to write small number of functions.
3. Handle failures occur during execution without restart job all over.

MapReduce paradigm consists of a simple two step structure, Map and Reduce to solve most every problem relates to massive data processing.

But it has a few drawbacks that need to create multiple steps multiple jobs, which will require more time consuming.

Workflow systems improved MapReduce by using an acyclic graph to deal with any collection of functions.

Diagram

Description automatically generated

## Spark

Spark keeps many of Hadoop’s MapReduce's benefits, while improve many MapReduce drawbacks as follows:

* Performance

Spark utilise RAM to process data in memory, while MapReduce persists data back to the disk after each Map-Reduce task. This allows Spark save communication time. Therefore, Spark could drastically speed up large scale of big data tasks.

Also, Spark works well for smaller data sets that can all fit into a server's RAM.

With workflow architecture, Spark breaks down large dataset and process them in parallel.

*Iterative computations are used to pass over the same data many times.*

* Ease of use

Spark is easier to program and therefore it has a faster learning curve than MapReduce.

Spark provide pre-built APIs for Java, Scala, Python, and R, etc. It is easy to write user-defined functions for different developers. Whereas MapReduce is written in Java.

*Spark includes a core data processing engine, as well as libraries for SQL, machine learning, and stream processing.*

* Compatibility

It’s compatible with all of Hadoop’s data sources and file formats,

* Data processing

Hadoop MapReduce is great for batch processing. Whereas, Spark can do much more.

Spark can do real-time processing due to its high performance. Spark is capable to process graphs and deal with machine learning tasks. Spark offers a "one size fits all" platform that you can use rather than splitting tasks across different platforms.

* Security
* Cost

Spark should be more cost-effective. Spark requires less hardware to perform the same tasks much faster, especially on the cloud where compute power is paid per use.

## TensorFlow

TensorFlow is famous for machine-learning applications. In fact, it is a workflow architecture as well.