Work distribution challenges

Dom Jina

17 Jan 2024

Problems with parallelism

count nums of each word in docs

- Seriel solution: create counter and update as we scan through
- Process both docs at same time and add counters together as final step
- Additional time merging results from pipeline
- If on different machines, we need to include network transfer time
- As architecture becomes more complex, we need orchestration

If a machine fails mid process, we need to reallocate the task.
This involves getting another copy of the data that the process was using.
The restart the task.

What happens if the master fails?
What happens if we lose track of one of the machines? I.e. network error

What we need to consider when creating parallel architecure

- Load balancing
- Overheads
- orchestration

Map reduce paradigm/framework

Basic ideas:
Map(Key1, Value1) → List[<Key2, Value2>]
Reduce(Key2, List[Value2]) → List<Value2>

Map a word to a list of documents that have that word - reverse index

Map example: Get words and count
Reduce example: sum the words

Hadoop: open source version of Map & Reduce made by yahoo

Hadoop is an implementation fo the map reduce in java along with a distributed file storage called HDFS
HDFS - Hadoop distributed file system

Hadoop compute:

- Client - Software that the client uses to submit work to the hadoop cluster
- Job Tracker - Recieves work and uses the name node to identify where data is located. Assigns work to task trackers
- Task Tracker - Executes map and reduce tasks

Hadoop HDFS

- Name node - tracks location of all data held on the HDFS
- Data Node - Handles storage access of the local files on the machine

hadoop versioning: W3-01