Transcript name: Hadoop architecture – Topology/rack awareness

English

Hadoop has awareness of the topology of the network. This allows it to optimize where it sends the computations to be applied to the data. Placing the work as close as possible to the data it operates on maximizes the bandwidth available for reading the data. In the diagram, the data we wish to apply processing to is block B1, the light blue rectangle on node n1 on rack 1.

When deciding which TaskTracker should receive a MapTask that reads data from B1, the best option is to choose the TaskTracker that runs on the same node as the data.

If we can't place the computation on the same node, our next best option is to place it on a node in the same rack as the data.

The worst case that Hadoop currently supports is when the computation must be done from a node in a different rack than the data. When rack-awareness is configured for your cluster, Hadoop will always try to run the task on the TaskTracker node with the highest bandwidth access to the data.

Let us walk through an example of how a file gets written to HDFS.

First, the client submits a "create" request to the NameNode. The NameNode checks that the file does not already exist and the client has permission to write the file.

If that succeeds, the NameNode determines the DataNode to write the first block to. If the client is running on a DataNode, it will try to place it there. Otherwise, it chooses at random.

By default, data is replicated to two other places in the cluster. A pipeline is built between the three DataNodes that make up the pipeline. The second DataNode is a randomly chosen node on a rack other than that of the first replica of the block. This is to increase redundancy.

The final replica is placed on a random node within the same rack as the second replica. The data is piped from the second DataNode to the third.

To ensure the write was successful before continuing, acknowledgment packets are sent back from the third DataNode to the second,

From the second DataNode to the first

And from the first DataNode to the client

This process occurs for each of the blocks that make up the file, in this case, the second

and the third block. Notice that, for every block, there is a replica on at least two racks.

When the client is done writing to the DataNode pipeline and has received acknowledgements, it tells the NameNode that it is complete. The NameNode will check that the blocks are at least minimally replicated before responding.

This concludes this presentation. Thank you for watching.