|  |  |
| --- | --- |
|  | Explain the core changes made in Hadoop 2.x |
|  | -- In case of Hadoop 1.x JobTracker keeps track of resource utilization and job monitoring while in case |
|  | of Hadoop 2.x, Resource Utilization is taken care by Resource Manager and Node Manager, whereas job monitoring |
|  | is taken care by Application Master. |
|  |  |
|  | -- Hadoop 1.x was suited for maximum of 4000 nodes and 40000 tasks while Hadoop 2.x can scale up to 10000 nodes |
|  | and 100000 tasks. |
|  |  |
|  | -- In Hadoop 1.x TaskTracker is configured with static slots. Moreover, a map tasks can not run on reduce slot. So cluster |
|  | utilization is low. While in Hadoop 2.x resources are dynamic and fine-grained.This leads to better cluster utilization. |
|  |  |
|  | -- Hadoop 1.x Supports only MapReduce processing model while Hadoop 2.x supports processing models other than Map Reduce. |
|  |  |
|  |  |
|  |  |
|  |  |
|  | >> Explain the difference between MapReduce 1 and MapReduce 2 / Yarn |
|  |  |
|  | > MapReduce 1 --- |
|  | In a typical Hadoop cluster, racks are interconnected via core switches. Core switches should connect to top-of-rack |
|  | switches Enterprises using Hadoop should consider using 10GbE, bonded Ethernet and redundant top-of-rack switches to |
|  | mitigate risk in the event of failure. A file is broken into 64MB chunks by default and distributed across Data Nodes. |
|  | Each chunk has a default replication factor of 3, meaning there will be 3 copies of the data at any given time. Hadoop |
|  | is "Rack Aware" and HDFS has replicated chunks on nodes on different racks. JobTracker assign tasks to nodes closest to |
|  | the data depending on the location of nodes and helps the NameNode determine the 'closest' chunk to a client during reads. |
|  | The administrator supplies a script which tells Hadoop which rack the node is in, for example: /enterprisedatacenter/rack2. |
|  | Limitations of MapReduce 1.0 – Hadoop can scale up to 4,000 nodes. When it exceeds that limit, it raises unpredictable |
|  | behavior such as cascading failures and serious deterioration of overall cluster. Another issue being multi-tenancy – |
|  | it is impossible to run other frameworks than MapReduce 1.0 on a Hadoop cluster. |
|  |  |
|  |  |
|  |  |
|  | > MapReduce 2 / Yarn --- |
|  | MapReduce 2.0 has two components – |
|  | -- YARN that has cluster resource management capabilities |
|  | -- MapReduce |
|  |  |
|  | In MapReduce 2, the JobTracker is divided into three services: |
|  | ResourceManager a persistent YARN service that receives and runs applications on the cluster. A MapReduce job is an application. |
|  | JobHistoryServer, to provide information about completed jobs Application Master, to manage each MapReduce job and is terminated |
|  | when the job completes. Also, the TaskTracker has been replaced with the NodeManager, a YARN service that manages resources and |
|  | deployment on a node. NodeManager is responsible for launching containers that could either be a map or reduce task. |
|  |  |
|  | This new architecture breaks JobTracker model by allowing a new ResourceManager to manage resource usage across applications, |
|  | with ApplicationMasters taking the responsibility of managing the execution of jobs. This change removes a bottleneck and lets |
|  | Hadoop clusters scale up to larger configurations than 4000 nodes. This architecture also allows simultaneous execution of a |
|  | variety of programming models such as graph processing, iterative processing, machine learning, and general cluster computing, |
|  | including the traditional MapReduce. |