

Big Data MapReduce

MapReduce

❖ Hadoop

- Hadoop is a Reliable Shared Storage and Analysis System

- Hadoop = HDFS + MapReduce + α
 - HDFS provides Data Storage
 - HDFS: Hadoop Distributed FileSystem
 - MapReduce provides Data Analysis
 - MapReduce = Map Function + Reduce Function

MapReduce

❖ Scaling Out

- Scaling out is done by the DFS (Distributed FileSystem), where the data is divided and stored in distributed computers & servers
- Hadoop uses HDFS to move the MapReduce computation to several distributed computing machines that will process a part of the divided data assigned

MapReduce

❖ Jobs

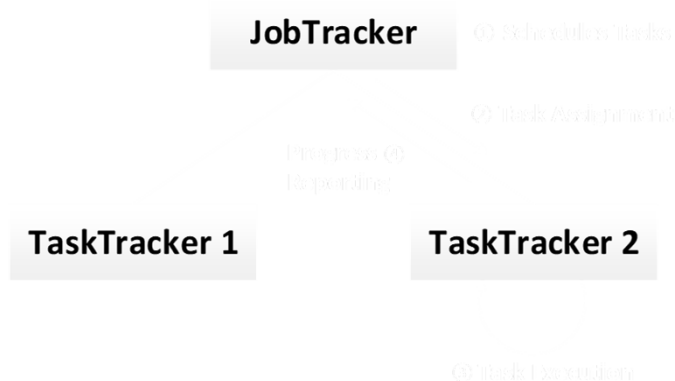
- MapReduce *job* is a unit of work that needs to be executed
- Job types: Data input, MapReduce program, Configuration Information, etc.
- Job is executed by dividing it into one of two types of tasks
 - *Map* Task
 - *Reduce* Task

MapReduce

❖ Node types for Job execution

- Job execution is controlled by 2 types of nodes
 - *JobTracker*
 - *TaskTracker*
- JobTracker coordinates all jobs
- JobTracker schedules all tasks and assigns the tasks to TaskTrackers

MapReduce



- TaskTracker will execute its assigned task
- TaskTracker will send a progress reports to the JobTracker
- JobTracker will keep a record of the progress of all jobs executed

MapReduce

❖ Data flow

- Hadoop divides the input into *input splits* (or *splits*) suitable for the MapReduce job
- *Split* has a fixed-size
- *Split* size is commonly matched to the size of a HDFS block (64 MB) for maximum processing efficiency

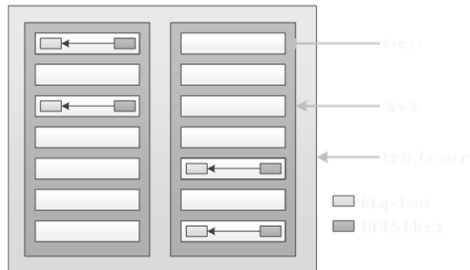
MapReduce

❖ Data flow

- Map Task is created for each split
- Map Task executes the map function for all records within the split
- Hadoop commonly executes the Map Task on the node where the input data resides

MapReduce

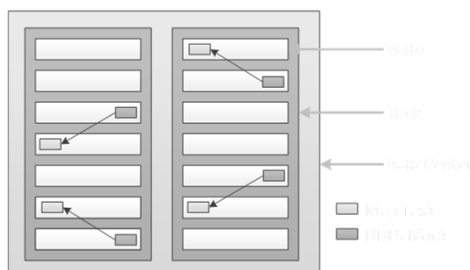
❖ Data flow



- *Data-Local Map Task*
- *Data locality optimization*
does not need to use the cluster network
- *Data-local flow process shows why the*
Optimal Split Size = 64 MB HDFS Block Size

MapReduce

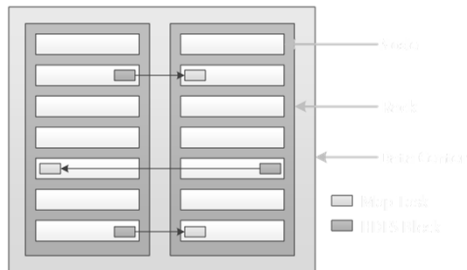
❖ Data flow



- *Rack-Local Map Task*
- A node hosting the HDFS block replicas for a map task's input split could be running other map tasks
- Job Scheduler will look for a free map slot on a node in the same rack as one of the blocks

MapReduce

❖ Data flow



- *Off-Rack Map Task*
- Needed when the Job Scheduler cannot perform *data-local or rack-local map tasks*
- Uses inter-rack network transfer

MapReduce

❖ Map

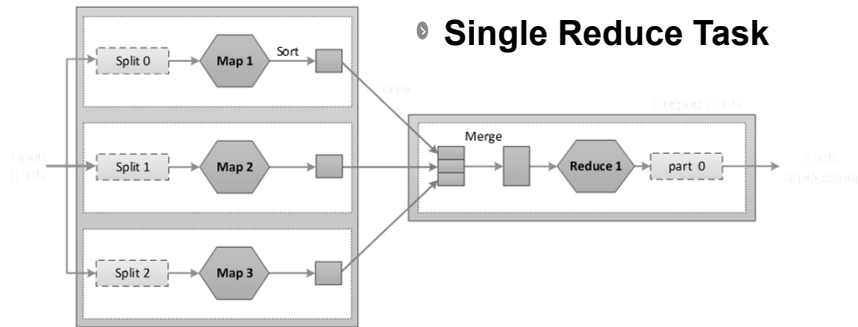
- Map task will write its output to the local disk
- Map task output is not the final output, it is only the intermediate output

❖ Reduce

- Map task output is processed by Reduce Tasks to produce the final output
- Reduce Task output is stored in HDFS
 - For a completed job, the Map Task output can be discarded

MapReduce

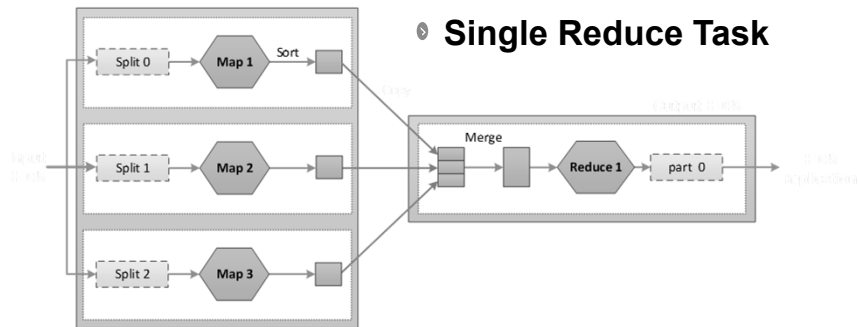
❖ Single Reduce Task



- Node includes Split, Map, Sort, and Output unit
- Light blue arrows show data transfers in a node
- Orange arrows show data transfers between nodes

MapReduce

❖ Single Reduce Task



- Number of reduce tasks is specified independently, and is not based on the size of the input

MapReduce

❖ **Combiner Function**

- User specified function to run on the Map output → Forms the input to the Reduce function
- Specifically designed to minimize the data transferred between Map Tasks and Reduce Tasks
- Solves the problem of limited network speed on the cluster and helps to reduce the time in completing MapReduce jobs

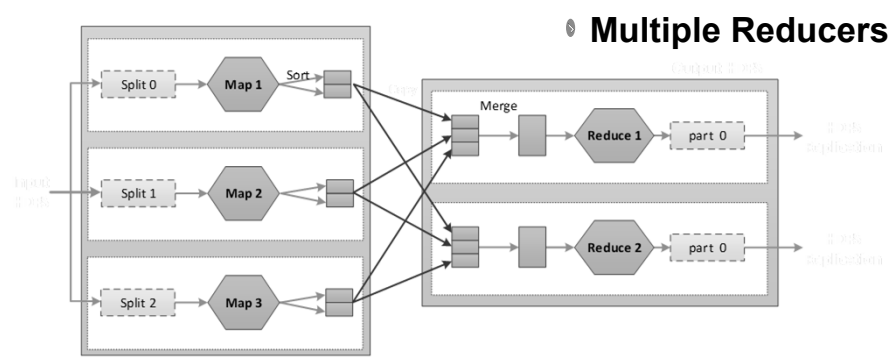
MapReduce

❖ **Multiple Reducer**

- Map tasks *partition* their output, each creating one partition for each reduce task
- Each partition may use many keys and key associated values
- All records for a key are kept in a single partition

MapReduce

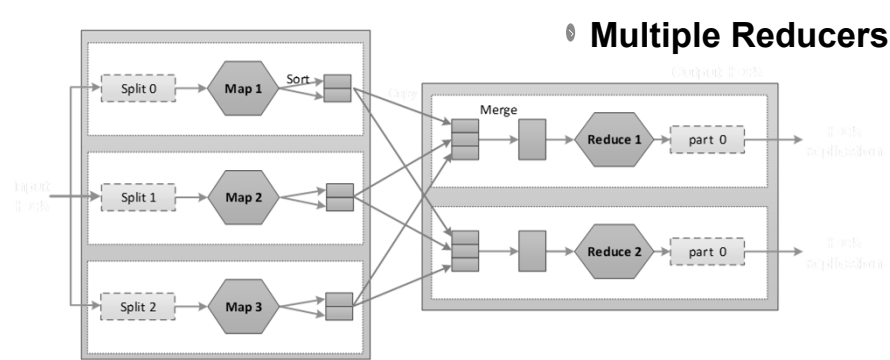
❖ Multiple Reducers



- Shuffle process is used in the data flow between the Map tasks and Reduce tasks

MapReduce

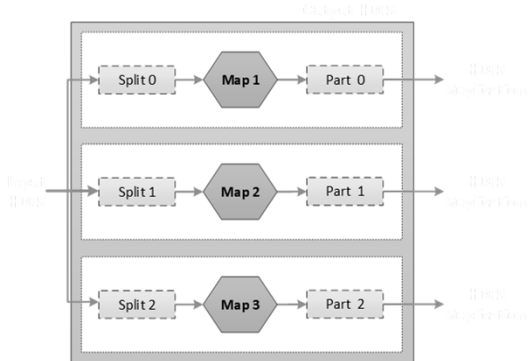
❖ Multiple Reducers



- Shuffle process is used in the data flow between the Map tasks and Reduce tasks

MapReduce

❖ Zero Reducer



- Zero reducer uses no shuffle process
- Applied when all of the processing can be carried out in parallel Map tasks

Big Data References

References

- V. Mayer-Schönberger, and K. Cukier, Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt, 2013.
- T. White, Hadoop: The Definitive Guide. O'Reilly Media, 2012.
- J. Venner, Pro Hadoop. Apress, 2009.
- S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, "Big Data, Analytics and the Path From Insights to Value," MIT Sloan Management Review, vol. 52, no. 2, Winter 2011.
- B. Randal, R. H. Katz, and E. D. Lazowska, "Big-data Computing: Creating revolutionary breakthroughs in commerce, science and society," Computing Community Consortium, pp. 1-15, Dec. 2008.
- G. Linden, B. Smith, and J. York. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan/Feb. 2003.

References

- J. R. Galbraith, "Organizational Design Challenges Resulting From Big Data," Journal of Organization Design, vol. 3, no. 1, pp. 2-13, Apr. 2014.
- S. Sagirolu and D. Sinanc, "Big data: A review," Proc. IEEE International Conference on Collaboration Technologies and Systems, pp. 42-47, May 2013.
- M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," Mobile Networks and Applications, vol. 19, no. 2, pp. 171-209, Jan. 2014.
- X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data Mining with Big Data," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, Jan. 2014.
- Z. Zheng, J. Zhu, and M. R. Lyu, "Service-Generated Big Data and Big Data-as-a-Service: An Overview," Proc. IEEE International Congress on Big Data, pp. 403-410, Jun/Jul. 2013.

References

- I. Palit and C.K. Reddy, “Scalable and Parallel Boosting with MapReduce,” IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 10, pp. 1904-1916, 2012.
- M.-Y Choi, E.-A. Cho, D.-H. Park, C.-J Moon, and D.-K. Baik, “A Database Synchronization Algorithm for Mobile Devices,” IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 392-398, May 2010.
- IBM, What is big data?, <http://www.ibm.com/software/data/bigdata/what-is-big-data.html> [Accessed June 1, 2015]
- Hadoop Apache, <http://hadoop.apache.org>
- Wikipedia, <http://www.wikipedia.org>

Image sources

- Walmart Logo, By Walmart [Public domain], via Wikimedia Commons
- Amazon Logo, By Balajimuthazhagan (Own work) [CC BY-SA 3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>)], via Wikimedia Commons