MapReduce

1. Original computation:

Input data is large and computations have to be distributed across hundreds or thousands of machine.

1. Programming Model:

Input pair->MapFunctions->intermediate key/value pair

Same intermediate key/value pairs->ReduceFunctions->output files

1. Process:
2. Splits files into M pieces of typically 16-64MB per piece
3. M map tasks and R reduce tasks to assign, master pick and assign
4. Worker first parse input data and extracts the pairs, then passes them to the map functions and produce the intermediate key/value pair.
5. Periodically, buffer pairs are written into the local disk, 分成R区域并且通过partitioning functions将这些partition的地址再传到master中让它来分配，通过key值分区。
6. Reduce进行数据的读入，并且将其key值进行排序。
7. Reduce worker将相同的intermediate key值传入对应的reduce function里然后每个partition有一个final output file
8. Semantics in the presence of failure rely on atomic commits of map and reduce task outputs to achieve.
9. Backup Tasks:

在快要结束的时候， master同时将最后的这些任务在backup上和正常的内容上同时运行，哪一个先结束就代表着整个过程都结束。

1. Fault Tolerance:
2. Worker Failure:

Periodically master will ping each worker to make sure they’re working. So if a worker stops working:

1’ 将完成的map在那台机子上的也全部重新做一遍，因为存的是local，那么那个worker以及那个盘全坏掉了所以就不能access

2’ 而如果是reduce完成的，而这台reduce坏掉了，那么不用重新做，因为那样的话已经存在local里了，所以就不用了

3’ 如果A坏了，然后用B去execute，那么，所有没有读取A数据的都被告知应从此刻开始从B中读取数据

1. Master Failure:

Periodically write checkpoints of the data in itself.

1. Combiner Function:

Executed on map machines, output of it is written to the output file intermediate and will be sent to a reduce task.

1. Side-effects:

定时write to temporary file and atomically renames this file once fully generated.

1. Skipping Bad Records:

Each worker process installs a signal handler that catches segmentation violations and bus errors.