* input key/value pairs produce output key value pairs
* Map
  + takes an input pair and produces set of intermediate key/value pairs
* Reduce
  + intermediate key I and set of values for that key
  + merges together values
* Word counter example
  + map takes a document and for each word associates it with a count (1)
  + reduce takes a word as key and sums up all counts
* Execution
  + M splits (M= number of map jobs)
  + Partition intermediate key space into R pieces (R= number of reduce jobs)
    - partition function ex: hash(key) mod R
  + master assigns workers tasks
  + Map tasks
    - passes each key,value pair in input split to map function
    - output k/v of map written to local disk, partitioned into R regions
    - locations of the k/v pairs passed to master
  + Reduce tasks
    - uses remote procedure calls to read buffered data from local disks of map workers
    - sorts all intermediate keys so same keys are grouped together
    - Pass each unique key and set of values to reduce
    - output of reduce function appended to final output file for task
  + Master
    - stores state for each task
    - pings every worker periodically
    - for failed workers
      * any ongoing map/reduce task reset back to idle
      * completed map tasks re-executed on failure
      * reduce task output stored in global file system so no need to re-execute
    - abort if master fails
* atomic commits of map and reduce task outputs
  + - * each in progress task writes output to private temp files
      * once map task completes, worker sends a message to master with R file names (1 per reduce task)
      * once reduce task completes, reduce worker atomically renames temp output file to final output file
      * if multiple reduce tasks complete at same time, automic rename operation by the file system makes only 1 of the files the real one
* GFS
  + if done on GFS, master tries to schedule map tasks on machines that contains replica of corresponding input data
* Backup tasks
  + When mapreduce operation close to completion, master schedules backup executions on remaining in-progress tasks
* within given partition, intermediate k/v pairs processed in increasing key order
* Combiner function
  + executed on each machine that performs a map task
  + basically same come as a reduce function, but put in the intermediate files that are inputs to reduce tasks