Introduction to Data Science

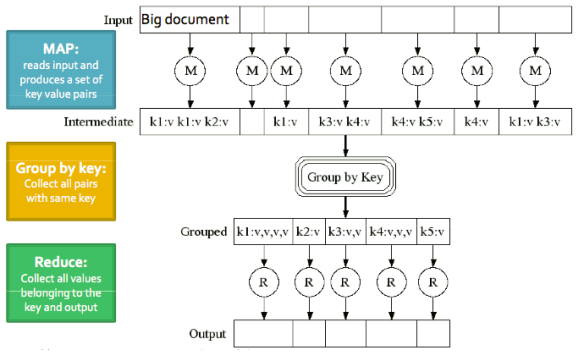
CS 5665
Utah State University
Department of Computer Science
Instructor: Prof. Kyumin Lee

MapReduce

What do you do?

- Define two functions:
  - map (k, v) → <k', v'>*
  - reduce (k', <v'>) → <k', v''>*
- All v' with the same k' are reduced together and processed in v' order

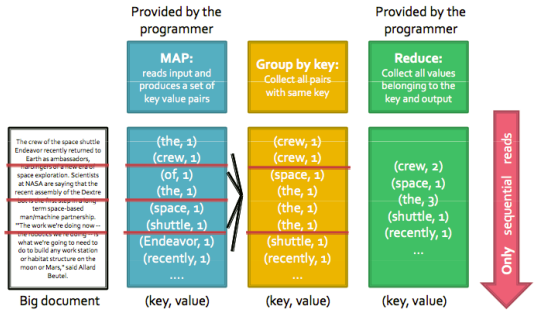
MapReduce: A diagram



MapReduce: Word counting

- Program specifies two primary methods
  - Map(k,v) →<k', v'>*
  - Reduce(k',<v'>*) →<k', v''>*
- Ex: Two documents - (d1, "the crew") and (d2, "of the")
  - Map(d1, "the crew") => [(the, 1), (crew, 1)]
  - Map(d2, "of the") => [(of, 1), (the, 1)]
  - MapReduce runs its grouper module and calls reduce for every key
  - Reduce (the, [1,1]) => (the, 2)
  - Reduce (crew, [1]) => (crew, 1)
  - Reduce (of, [1]) => (of, 1)

MapReduce: Word counting



Coordination

- Master data structures
  - Task status: (idle, in-progress, completed)
  - Idle tasks get scheduled as workers become available
  - When a map task completes, it sends the master the location and sizes of its R intermediate files, one for each reducer
  - Master pushes this info to reducers
- Master pings workers periodically to detect failures

Failures

- Map worker failure
  - Map tasks completed or in-progress at worker are reset to idle
  - Reduce workers are notified when task is rescheduled on another worker
- Reduce worker failure
  - Only in-progress tasks are reset to idle
- Master failure
  - MapReduce task is aborted and client is notified

Example 1: Host size

- Suppose we have a large web corpus
- Let's look at the metadata file
  - Lines of the form (URL, size, date, …)
- For each host, find the total number of bytes
  - i.e., the sum of the page sizes for all URLs from that host
- Solution:
  - Map(line_no, (URL, size, date, …)) => [(URL_host, size)]
  - Reduce(URL_host, [size1, size2,….]) => (URL_host, sum([size1, size2,….]))