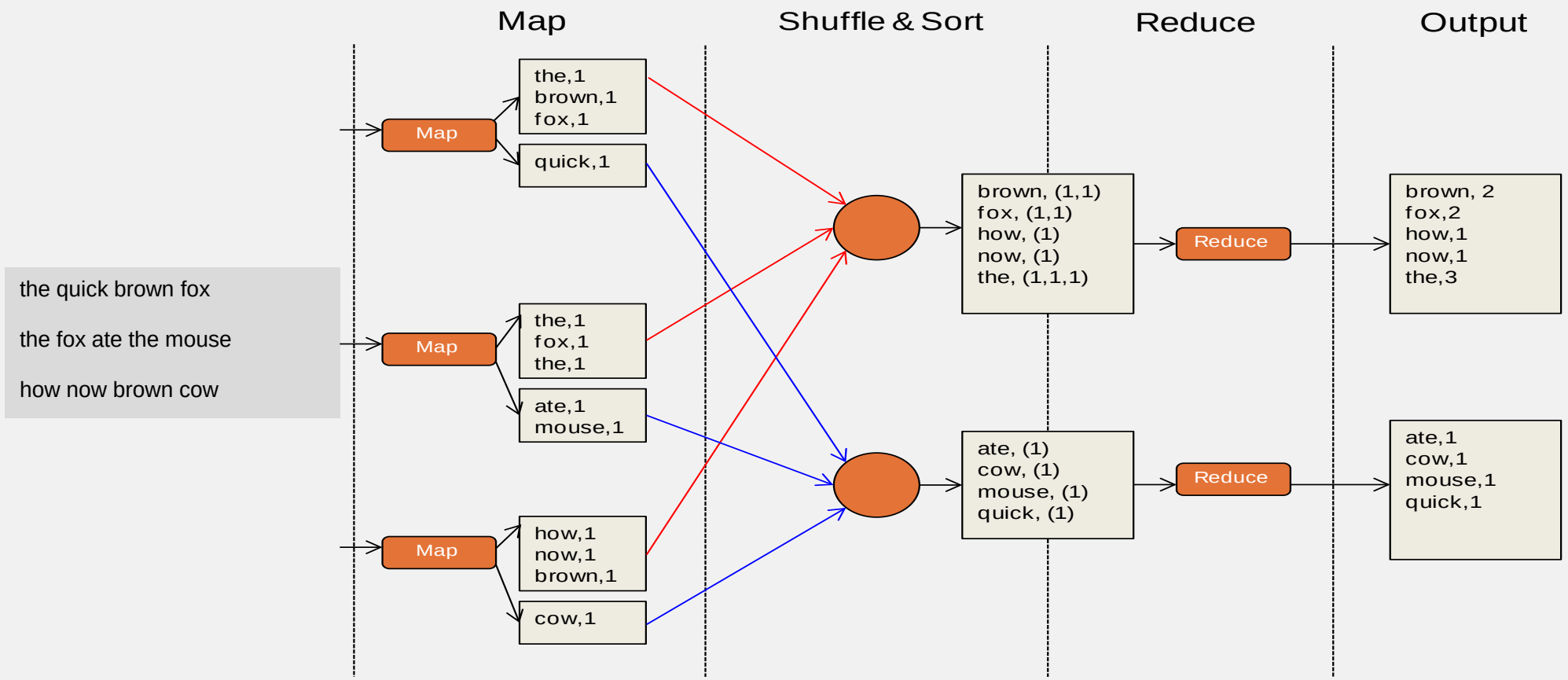
CLOUD COMPUTING APPLICATIONS

YARN: MapReduce on YARN

Roy Campbell & Reza Farivar

MapReduce


Map
Shuffle & Sort
Reduce
Output
the quick brown fox
the fox ate the mouse
how now brown cow
Map
Map
Map
the,1
brown,1
fox,1
quick,1
the,1
fox,1
the,1
ate,1
mouse,1
how,1
now,1
brown,1
cow,1
brown, (1,1)
fox, (1,1)
how, (1)
now, (1)
the, (1,1,1)
ate, (1)
cow, (1)
mouse, (1)
quick, (1)
Reduce
Reduce
brown, 2
fox,2
how,1
now,1
the,3
ate,1
cow,1
mouse,1
quick,1

MapReduce on YARN

- MapReduce AM determines number of map and reduce tasks
  - Split metainfo file indicates number of map tasks based on number of splits
  - Job config determines number of reducers
- AM schedules when to request containers for map and reduce tasks
  - Split metainfo file has data locality for each map task
  - Reducers have no locality
  - Uses headroom provided by RM to avoid livelocks where reducers consume all available resources but more maps need to run

MapReduce on YARN

- Tasks connect back to AM upon startup via TaskUmbilicalProtocol
  - Report progress, liveliness
  - AM kills tasks that do not report progress in a timely manner
  - AM provides reducers with shuffle data locations
  - Reducers notify AM of shuffle fetch failures; AM relaunches map tasks if necessary

MapReduce on YARN

- Shuffle provided as a plugin service to NodeManagers
  - Shuffle port configurable, passed to reducers via AM
- AM responsible for job history
  - Job history events written to a file as job progresses
  - Copied to a drop location in HDFS when job completes
  - Used to provide recovery when AM crashes and is retried by RM
- MapReduce AM provides client interface
  - Report job and tasks status
  - Kill job or task attempts
  - Web app and services
  - Client can redirect to job history server if application has completed