On Cloud Computational Models and the Heterogeneity Challenge

Raouf Boutaba

D. Cheriton School of Computer Science
University of Waterloo

WCU IT Convergence Engineering Division
POSTECH

FOME, December 13, 2011

Outline

- Introduction to MapReduce and Hadoop
- Heterogeneity of Production MapReduce clusters
- Research Challenges
- Conclusion

Data Intensive Computation in the Cloud

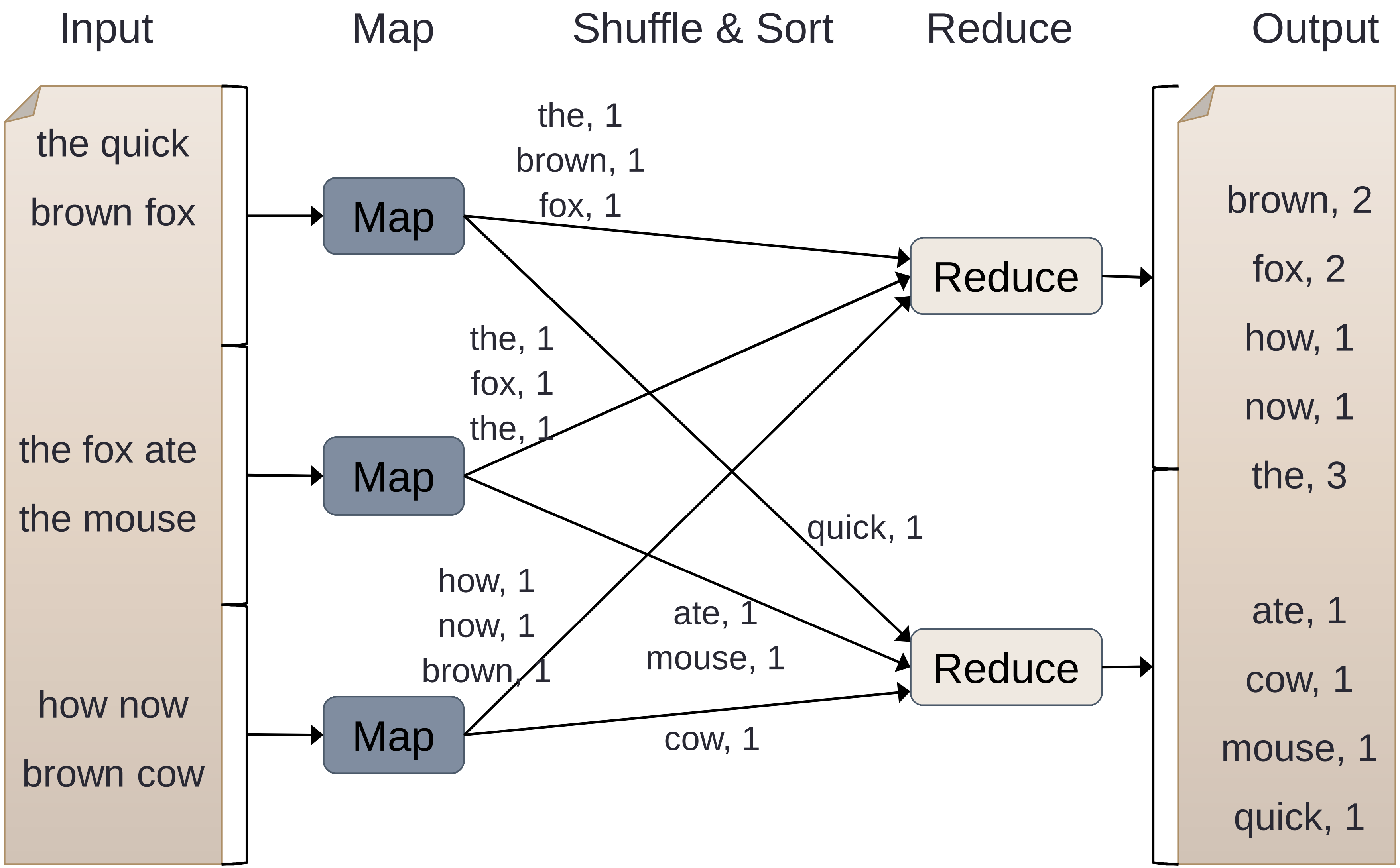
- Huge volume of data
  - Google (2008): 20PB data per day
  - Facebook (2010): 36 PB of stored data, processing 80-90TB per day
  - Yahoo (2010): 170 PB data stored spread across the globe
    Processing 3 PB per day

- Very large # of service requests requiring fast response
  - Thousands of servers used
  - Google: 200+ clusters, hundreds of thousands computers
  - Facebook: 2000+ computers
  - Yahoo: 34000+ computers

New Programming Model

- To support large-scale data-intensive computation in a timely manner
- MapReduce
  - Introduced by Google
  - Support distributed computing on large data sets on clusters of computers
  - Several implementations: Google, Oracle, Hadoop...
- Benefits of MapReduce
  - Highly scalable
  - Built-in fault tolerance

MapReduce - Word Count Example

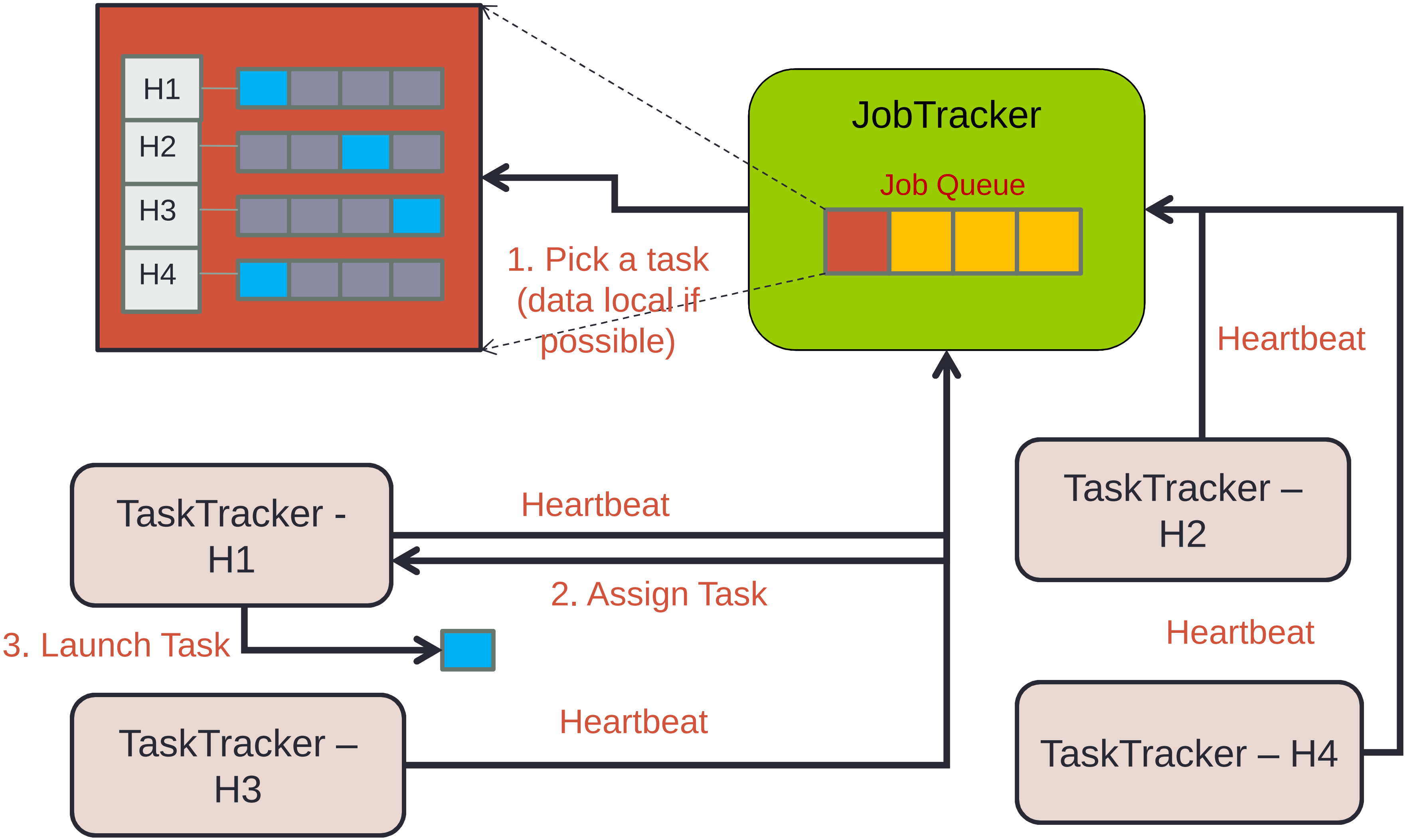


Apache Hadoop

- An open source implementation of MapReduce
- Enables applications to work with thousands of nodes and petabytes of data
- Includes a range of subprojects
  - **HDFS**: A distributed file system that provides high throughput access to application data
  - **MapReduce**: A software framework for distributed processing of large data sets on compute clusters

http://hadoop.apache.org/

Apache Hadoop

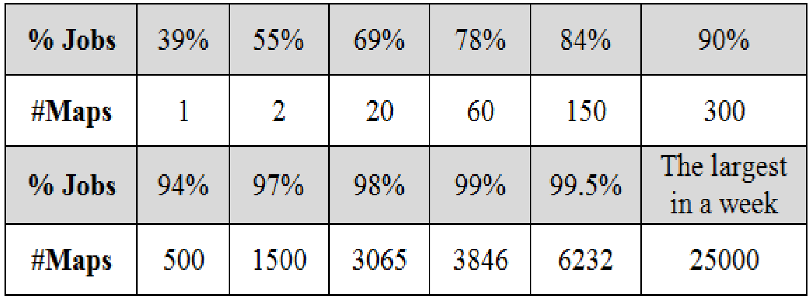


Outline

- Introduction to MapReduce and Hadoop
- Heterogeneity of Production MapReduce clusters
- Research Challenges
- Conclusion

Heterogenous Jobs Sizes

TABLE II: CDF OF NUMBER OF MAP TASKS IN A HADOOP CLUSTER AT FACEBOOK



% Jobs	39%	55%	69%	78%	84%	90%
#Maps	1	2	20	60	150	300
% Jobs	94%	97%	98%	99%	99.5%	The largest in a week
#Maps	500	1500	3065	3846	6232	25000

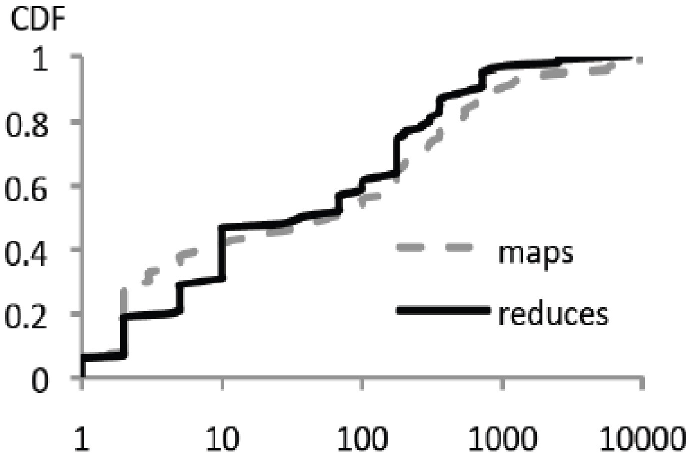


Figure 1. CDF of number of map and reduce tasks in a Hadoop cluster at Internet Company [4]

Most jobs are small, a few jobs are very large

Bimodal Distribution of Job Lengths

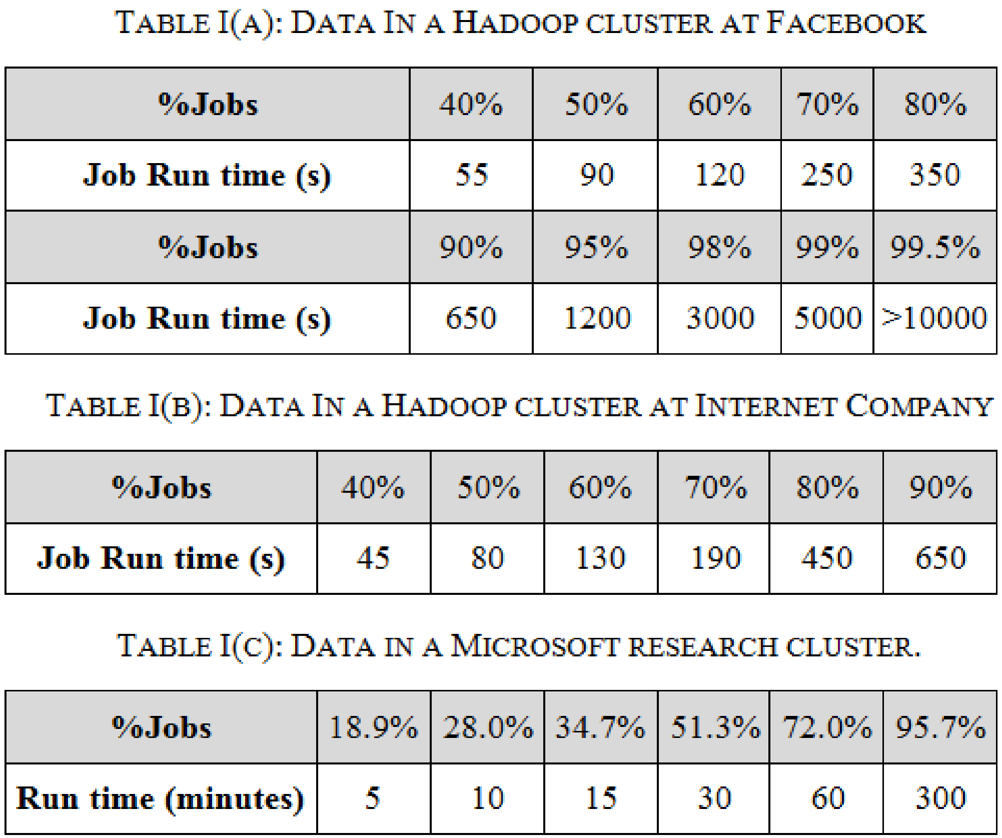


TABLE I(A): DATA IN A HADOOP CLUSTER AT FACEBOOK

%Jobs	40%	50%	60%	70%	80%
Job Run time (s)	55	90	120	250	350
%Jobs	90%	95%	98%	99%	99.5%
Job Run time (s)	650	1200	3000	5000	>10000

TABLE I(B): DATA IN A HADOOP CLUSTER AT INTERNET COMPANY

%Jobs	40%	50%	60%	70%	80%	90%
Job Run time (s)	45	80	130	190	450	650

TABLE I(C): DATA IN A MICROSOFT RESEARCH CLUSTER.

%Jobs	18.9%	28.0%	34.7%	51.3%	72.0%	95.7%
Run time (minutes)	5	10	15	30	60	300

Most jobs are short, a small fraction of very long jobs

Fluctuating Job Arrival Rates

- Arrival rate of MapReduce jobs is also highly variable from time to time

- Inter-arrival time exhibits an on-off pattern according to the time of the day
  - During daytime job arrival rate can be quite intense, as around 40% inter-job arrival time is less than 10s.
  - At night time, job arrival intervals can be very long

The arrival rate of slot requests can be very spiky

Heterogeneous Resource Requirements/Hardware

- Resource Requirements are heterogeneous
    - Varying requirements in terms of CPU, Memory, I/O, Bandwidth
- Performance objectives are heterogeneous
    - Production jobs vs. Non-production jobs
- Hardware is heterogeneous
    - Often multiple generations of server/ networking equipment in a cluster
    - To leverage previous investment

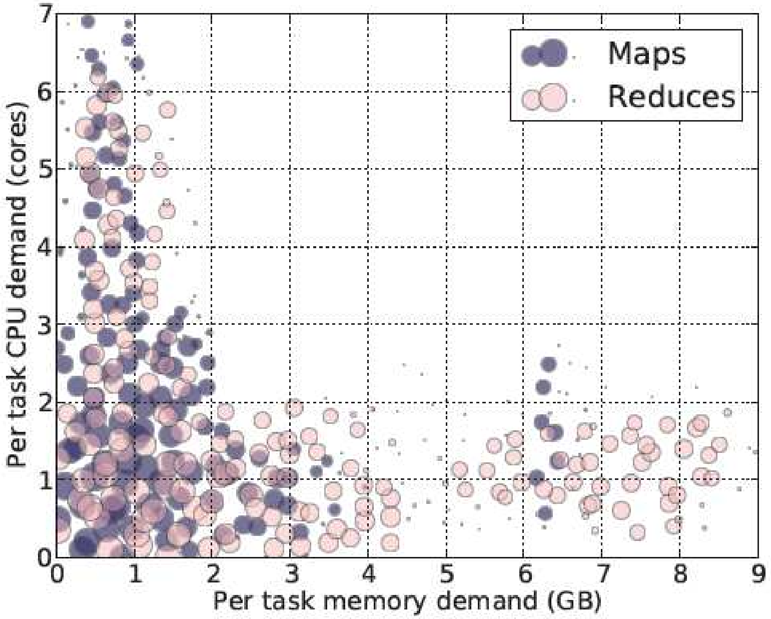


Figure 1: CPU and memory demands of tasks in a 2000-node Hadoop cluster at Facebook over one month (October 2010). Each bubble's size is logarithmic in the number of tasks in its region.

Outline

- Introduction to MapReduce and Hadoop
- Heterogeneity of Production MapReduce clusters
- Research Challenges
  - Job Scheduling
  - Data and Task Placement
  - Resource Sharing
  - Performance-aware resource allocation
- Conclusion

Job Scheduling

- Need to carefully design scheduling algorithms to assign tasks to machines
  - With consideration to both efficiency and fairness (tradeoff?)
    - Efficiency: higher utilization
    - Fairness: large jobs should not monopolize the cluster; low latency for small jobs
  - Account for various sources of performance bottlenecks (resource, network, location, etc.) due to heterogeneity of workload/resources
  - Preemption commonly used to give priority to production tasks
    - Can be unexpectedly high under heterogeneous workload
  - Handle outliers to reduce job response time.

Task and Data Placement

- Data and communication locality can substantially reduce job completion time and traffic in data centers
  - Typically achieved through careful placement of data and tasks
- Two optimization problems
  - **Data Placement**: Which physical machine should be used to store each data block
  - **VM Placement**: Where should the VMs be provisioned to process these blocks
  - Often related as there is often a fixed set of jobs that process each data set
- Data replication based on popularity; data placement avoiding machine hotspot (co-location of popular blocks); data locality based task placement; task migration

Resource Sharing

- Resource sharing concerns the division of resources among collocated tasks
  - 4 types of resources: CPU, memory, disk I/O and network
- Current approach: slot based resource allocation
  - Physical resources on each machine divided into multiple identical slots
  - Each task is assigned a single slot
- Limitations:
  - Tasks have (1) heterogeneous resource demand and (2) Differing performance objectives
  - CPU intensive tasks want more CPU, whereas I/O intensive tasks want more disk I/O
    - Slot-based resource allocation is sub-optimal
- Optimization of bandwidth allocation for each stage of MapReduce
  - Broadcast, Shuffle, Incast

Performance-Aware Resource Allocation

- MapReduce jobs have differing performance objectives in terms of competition time and throughput
- Currently Hadoop does not provide mechanisms to guarantee completion time
  - Job priority in Hadoop only specifies the relative weight of each job
- Designing an SLO-aware MapReduce resource allocation is challenging
  - Need a performance model for MapReduce jobs
    - Need to consider resource requirements, machine capacity and capability, location of input data, failure rate, and dynamic network condition
  - Uses of performance model: Estimating completion time and cost of a given job; determine the number of map and reduce tasks to meet deadline constraints; finding appropriate resource allocation to satisfy job completion time.

Conclusion

- Data-intensive computations are becoming a major application of Cloud computing
- This talk
  - An analysis of the heterogeneous characteristics in production clusters
  - Research challenges introduced by workload and resource heterogeneity
  - (*in the paper*) A survey of representative work on each of these challenges
- There is much more to be done
  - Most of the existing work was carried out in the last 3 years
  - There is still much room for improvements and innovations