Scalable Data Analysis (CIS 602-02)

MapReduce

Dr. David Koop
Scaling Up

PC

[Haeberlen and Ives, 2015]
Scaling Up

PC  Server

[Haeberlen and Ives, 2015]
Scaling Up

[PC] [Server] [Cluster]

[Haeberlen and Ives, 2015]
Scaling Up

PC  Server  Cluster  Data center

[Haeberlen and Ives, 2015]
Scaling Up

PC  Server  Cluster  Data center  Network of data centers

[Haeberlen and Ives, 2015]
Scale Problems

1. Difficult to dimension
   - Load can vary considerably
   - Waste resources of lose customers

2. Expensive
   - Hardware costs
   - Personnel costs
   - Maintenance costs

3. Difficult to scale
   - scaling up (new machines, new buildings)
   - scaling down (energy, fixed costs)

[Haeberlen and Ives, 2015]
Power Plant to Cloud Analogy

Power source directly connected

[Power source]

Power source

Network

Meter

Customer

[Haeberlen and Ives, 2015]
Everything as a Service

- Software as a service (SaaS) [Restaurant]
- Platform as a service (PaaS) [Take-out food]
- Infrastructure as a service (IaaS) [Grocery]

[Haeberlen and Ives, 2015]
Virtualization

- Hypervisor controls access to resources
- Flexibility for provider
- Secure and isolated
- Performance may be hard to predict

[Haeberlen and Ives, 2015]
Cloud Challenges and Opportunities

- Availability: What happens to my business if there is an outage in the cloud?
- Data lock-in: How do I move my data from one cloud to another?
- Data confidentiality and auditability: How do I make sure that the cloud doesn't leak my confidential data?
- Data transfer bottlenecks: How do I copy large amounts of data from/to the cloud?
- Performance unpredictability: VMs sharing the same disk

[Haeberlen and Ives, 2015]
Cloud Challenges and Opportunities

- Scalable storage: Cloud model (short-term usage, no up-front cost, infinite capacity on demand) does not fit persistent storage well
- Bugs in large distributed systems: Many errors cannot be reproduced in smaller configs
- Scaling quickly: Dealing with boot times and idle power
- Reputation fate sharing: One customer's bad behavior can affect the reputation of others using the same cloud (e.g. spam, FBI raids)
- Software licensing: Licenses tied to computers, how to scale?
HPC, HTC, and MTC

- High-Performance Computing (HPC): execute a single job quickly
- High-Throughput Computing (HTC): execute many jobs over a long period of time
- Many-Task Computing (MTC): many tasks but over shorter times

- HPC -> tightly-coupled and HTC -> loosely-coupled
  - HTC jobs can be run independently, interconnects aren't as important as with HPC
- MTC as a bridge between HPC and HTC
Many-Task Computing

- Data Analysis, Mining
- Big Data and Many Tasks
- Many Loosely Coupled Tasks
- Heroic MPI Tasks

Input Data Size

- Hi
- Med
- Low

Number of Tasks

- 1
- 1K
- 1M

[Raicu et al., 2008]
Many-Task Computing

- Single task, modest data
  - MPI, etc...

- Many Tasks
  - DAGMan+Pegasus
  - Karajan+Swift+Falkon

- Much Data
  - MapReduce/Hadoop
  - Dryad

- Complex Tasks, Much Data
  - Dryad, Pig, Sawzall
  - Swift+Falkon (using data diffusion)

[Raicu et al., 2008]
Clouds vs. the Rest

- Clouds are not as efficient as existing scientific computing resources (e.g. grids, supercomputers, clusters)
- Clouds are reasonably cheap and work well for short deadlines
- Funding a cluster versus funding virtual computing time
- Considerations for the cloud:
  - Setup time
  - Different virtual machines consuming resources
Projects

- Remember, for dataset analyses, you want to be able to answer questions with the data and/or the outputs
- Questions?
Assignment 3

• How many of you have used Amazon Web Services (EC2)?
• Register for student account:
  - https://aws.amazon.com/education/awseducate/
• Assignment Idea: Spark analysis via Python interface on EC2
How to program for clouds

• If you have access to a cluster or cloud resources, how do you utilize these resources?

• Software working at a low level to allow data to be passed back and forth
  - Libraries like Message Passing Interface (MPI)
  - How to handle hardware failures?
  - How to distribute the data?
  - How to schedule nodes?

• Software working at a higher abstraction that deals with low-level issues
  - MapReduce, Spark, TensorFlow, etc.
MapReduce: Simplified Data Processing on Large Clusters

J. Dean and S. Ghemawat
MapReduce Overview

- Pre-loaded local input data
  - Mapping process

- Intermediate data from mappers
  - Values exchanged by shuffle process

- Reducing process generates outputs
  - Outputs stored locally

[Yahoo! Hadoop Tutorial]
Word Count Example

The overall MapReduce word count process

Input

Splitting

Mapping

Shuffling

Reducing

Final result

Deer Bear River
Car Car River
Deer Car Bear

Deer, 1
Bear, 1
River, 1

Car, 1
Car, 1
River, 1

Deer, 1
Bear, 1

Bear, 2

Bear, 1

Car, 3

Car, 1
Car, 1
Car, 1

Deer, 1
Deer, 1

Deer, 2

River, 1
River, 1

River, 2

Bear, 2
Car, 3
Deer, 2
River, 2
Parallel Databases vs. Hadoop (MapReduce)

J. Freire
Next Class

• MapReduce restricts the type of programs so that more can be automated
• What about multi-pass algorithms?
• Spark: extend a programming language with a distributed dataset