Data Management in the Cloud:
Limitations and Opportunities

Annies Ductan

Discussion Outline:

- **Introduction**
  - Overview
  - Vision of Cloud Computing
- **Managing Data in The Cloud**
  - Cloud Characteristics
  - Data Management Applications
    - Transactional
    - Analytical
- **Analyzing Data in The Cloud**
  - Ideal Properties and Features
  - MapReduce-like Software
  - Shared-Nothing Parallel Databases
  - Hybrid Solution
- **Summary**

Overview:

- *There is a need for research and engineering work to be done in creating a hybrid MapReduce/Parallel Database System*

- **Three Main Topics Discussed:**
  - Features a DBMS should implement for large scale data analysis
  - Available options through open source and commercial database
  - The need for new DBMS
    - Design specifically for cloud environments

Vision:

- **Present:**
  - Moving computer processing, storage, and software delivery away from the desktop and local servers
- **Future:**
  - Into next generation data centers hosted by large infrastructure companies such as:
    - *Amazon*
    - *Google*
    - *Yahoo*
    - *Microsoft*
    - *Sun*
- **Benefits:**
  - Corporations are free from large IT capital investments
  - Enables corporations with extremely powerful computing resources over the network

Vision:

- **Advantages:**
  - Ability for elastic computing
  - Resources are available for growth spikes
  - Low cost
    - Pay only for what is use:
      - Example: *Amazon's EC2* has computing resources for virtual private server instances:
        - Small
        - Large
        - Extra-large
- **Disadvantages:**
  - Privacy violation possible
  - Hosting company has ability to access customers data without permission

Managing Data In the Cloud:

- **Amazon EC2 Instance Types**

Available Instance Types

- **Small Instance** – default
  1.7 GB memory
  1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)
  160 GB instance storage
  32-bit or 64-bit platform
  I/O Performance: Moderate
  API name: m1.small

- **Medium Instance**
  3.75 GB memory
  2 EC2 Compute Unit (1 virtual core with 2 EC2 Compute Unit)
  410 GB instance storage
  32-bit or 64-bit platform
  I/O Performance: Moderate
  API name: m1.medium

- **Large Instance**
  7.5 GB memory
  4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)
  850 GB instance storage
  64-bit platform
  I/O Performance: High
  API name: m1.large

- **Extra Large Instance**
  15 GB memory
  8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
  1,690 GB instance storage
  64-bit platform
  I/O Performance: High
  API name: m1.xlarge

Managing Data In the Cloud:

Three characteristics of cloud:

- **Computing power is elastic if the workload is parallelizable:**
  - Obtaining additional resources by allocating additional server instances to a task
  - Suitable for applications designed to run on shared-nothing architecture
    - Only useful if the application is able to take advantage of the additional server instance by offloading some of its required work to the new instances running in parallel with the old instances
- **Data is stored at an untrusted host:**
  - US Patriot Act allows the government to demand access to the data stored on any computer
  - Most cloud computing vendors give the customer little control over where data is stored
    - Example:
      - *Amazon S3* only allows a customer to choose between US and EU data storage
    - Customer have little choice unless the data is encrypted using a key not located at the host
- **Data is replicated across large geographic distances:**
  - Allows cloud computing providers have to provide high levels of fault tolerance by replicating data across large geographic distances

Managing Data In the Cloud:

- Data Management Market :
  - Composed of two large components:
    - ***Transactional Data Management***
      - Databases that back:
        - Banking
        - Airline Reservation
        - Online e-commerce
        - Supply chain management
    - ***Analytical Data Management***
      - Applications querying a data store for:
        - Business planning
        - Problem Solving
        - Decision Support

Managing Data In the Cloud:

Transactional vs. Analytical

- ***Characteristics of Transactional Data Management:***
  - Shared-nothing architecture
    - Typically not used
  - ACID guarantees
    - Hard to maintain
  - Security
    - Risk involved with untrusted host
- ***Characteristics of Analytical Data Management:***
  - Shared-nothing architecture
    - Good Match
  - ACID guarantees
    - Typically not needed
  - Security
    - Sensitive data not used in analysis

Managing Data In the Cloud:

- **Characteristics of Transactional Data Management**:
  - Rely on ACID guarantees of databases
  - Tend to be write-intensive
- Not likely to be stored in the cloud:
  - Do not typically use a shared-nothing architecture
  - Difficulties in maintain ACID guarantees replicating data over large geographic distances
  - Risks storing transactional data in an untrusted host

Managing Data In the Cloud:

- Transactional Data Management:

Shared-nothing Architecture *know for scalability:*

Not relevant to Transactional Data Management:

- Majority of deployments are less than 1TB in size
- Data is partitioned across sites
- Transactions can not be restricted to accessing data from a single site

What this means for cloud computing:

- Complex distributed locking and commit protocols, and in shipping data over a network
- Increased latency and potential network bandwidth bottlenecks

- Transactional Database Providers:
  - ***Oracle***
    - Implemented the Oracle Database Machine, uses a shared-nothing architecture at the storage layer designed only to be used for data warehouses
  - ***IBM DB2***
    - Created an add-on feature Database Partitioning Feature(DPF) to Flagship product designed to scale analytical application running on data warehouses
  - ***Microsoft SQL Server*** (No shared-nothing architecture)
  - ***Sybase*** (No shared-nothing architecture)

Managing Data In the Cloud:

- Transactional Data Management:

Maintain ACID guarantees

- Distributed computer systems must choose between:
  - Consistency
  - Availability
  - Partition tolerance

- When replicating data over a wide area it leaves consistency and availability to choose from compromising part of the ACID to yield a reasonable system availability

  ~~*Consistency*~~

  Availability

Managing Data In the Cloud:

- Transactional Data Management:

Security

- Type of information gathered from TDMS
  - Operational data needed to power mission-critical business processes such as:
    - Customer data
    - Credit card numbers

- Because of this they conclude that TDM applications are not well suited for cloud deployment

- Companies attempt to implement TDM products for running in Amazon's cloud:
  - EnterpriseDB's
  - Postgres Plus Advanced Server
  - Oracle

Managing Data In the Cloud:

- **Characteristics of Analytical Data Management**:
  - Shared-nothing architecture is a good match for analytical data management
  - ACID guarantees are typically not needed
  - Sensitive data can often be left out of the analysis

Managing Data In the Cloud:

- Analytical Data Management:

Shared-nothing architecture

- Primary driver is the increasing amount of data involved in data analysis workloads consisting of:
  - Large scans
  - Multidimensional aggregations
  - Star schema joins
  - Making it easy to parallelize across nodes in a shared-nothing network

- Infrequent writes eliminates the need for complex distributed locking and commit protocols.

Managing Data In the Cloud:

- Analytical Data Management:

ACID guarantees

- Makes ***atomicity, consistency,*** and ***isolation*** of ACID easy to obtain due to:
  - Infrequent writes in the analytical database workloads
  - The fact that data can be performed on a recent snapshot of the data and not the most recent data.

Managing Data In the Cloud:

- Analytical Data Management:

Security

- Sensitive data can be left out of the analysis

- Less granular data can be analyzed instead of the lowest level of data

  - Three options for highly sensitive data:

    - Leave it out of the analytical data store

    - Include it after anonymization function

    - Include after encrypting

Data Analysis in the Cloud:

- **Ideal Properties and Features in cloud DBMS:**
  - Efficiency
  - Fault Tolerance
  - Ability to run in a heterogeneous environment
  - Ability to operate on encrypted data
  - Ability to interface with business intelligence products

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **MapReduce-like software**
  - **Fault Tolerance:**
    - Designed with fault tolerance as a high priority.
    - Data analysis job is divided into many small tasks
    - Tasks assigned to a failed machine are transparently reassigned to another machine.
- MapReduce experiment:

  Killing 200 out of 1746 worker processes involved in a MapReduce job resulted in only a 5% degradation in query performance

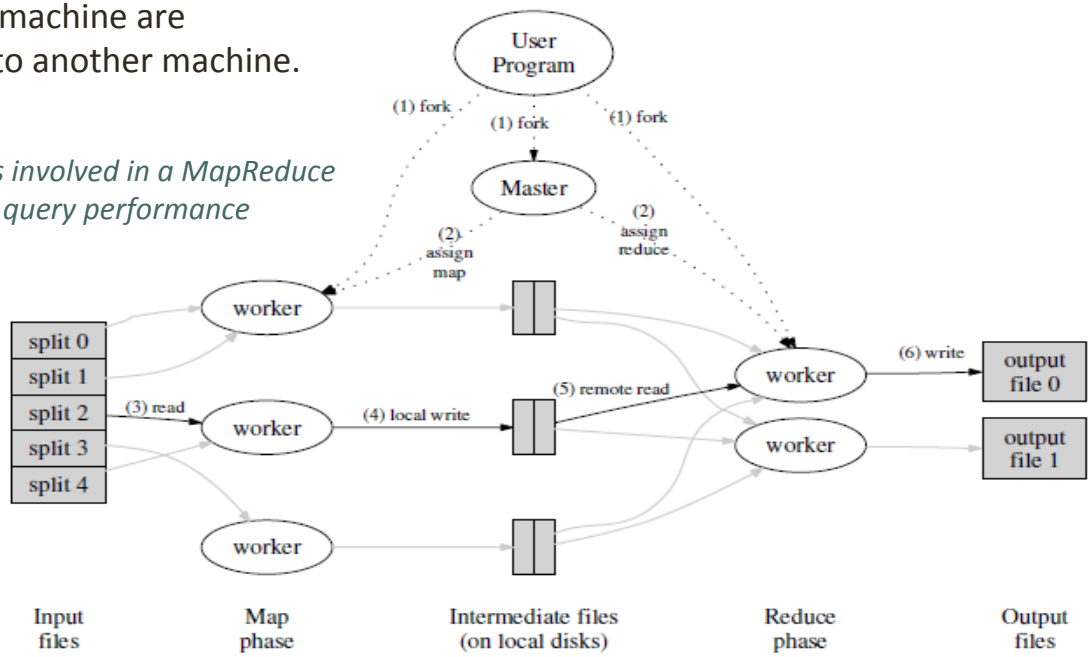

Figure 1: Execution overview

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **MapReduce-like software**

- **Ability to run in a heterogeneous environment**
  - Towards the end of a MapReduce job tasks that are still in progress get redundantly executed on other machines

  - Task are marked as completed as soon as the primary or the backup execution has completed:
    - Limits the effect *"straggler"* machines can have on total query time
    - Backup executions assigned to these machines will complete first

  - MapReduce Experiment:
    - It was shown that backup task execution improves query performance by 44% by alleviating the affects caused by slower machines

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **MapReduce-like software**

- **Figure(b)** Show an execution of the sort program with backup tasks disabled.
- *Results: Execution flow is similar **Figure (a)** except that there is a very long tail where write activity barely occurs.*
- After 960 seconds, all except 5 of the reduce tasks are completed.
- The last few stragglers don't finish until 300 seconds later
- The entire computation takes 1283 seconds, an increase of 44% in elapsed time

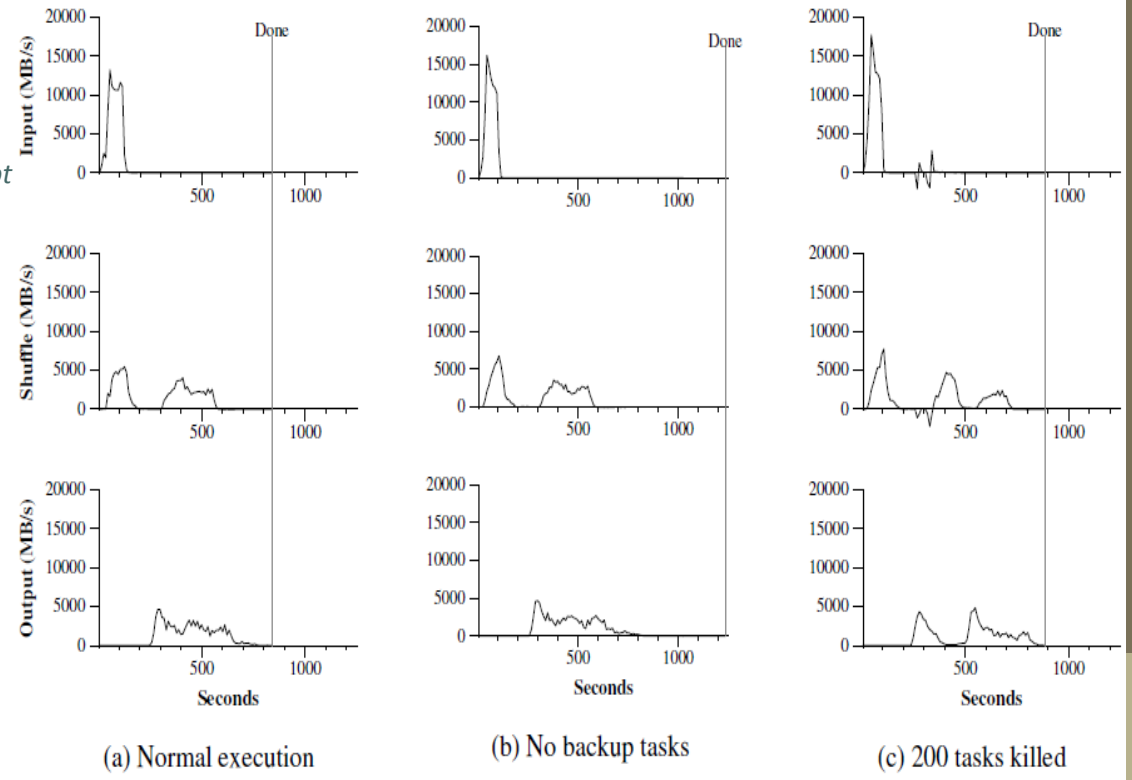

Figure 3: Data transfer rates over time for different executions of the sort program

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **MapReduce-like software**
  - **Ability to operate on encrypted data**
    - No ability to operate on encrypted data
    - Needs to be provided by user-defined code
  - **Ability to interface with business intelligence products**
    - Does not easily interface with existing business intelligence products
    - MapReduce was not intended to be a database system
      - Not SQL compliant

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **MapReduce-like software**
- The ***Grep*** program:
- Scans through 10^{10} 100-byte records searching for rare three-character patterns
  - Patterns occur in 92,337 records
- The input is split into 64MB pieces (M = 15000)
- The entire output is placed in one file (R = 1)

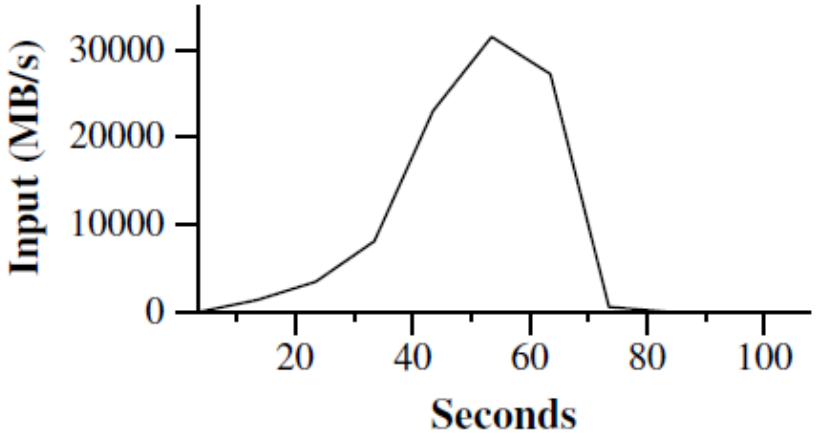

Figure 2: Data transfer rate over time

Data Analysis in the Cloud:

Available options for open source and commercial database:
MapReduce-like software

- **Efficiency**

  MapReduce experiment:

  - *A rare string is searched for inside a 1TB dataset*
  - *1TB of data is read off 3600 disks in the cluster (in parallel) where a pattern search is performed*
  - *The entire Grep query took 150 seconds to complete*
  - *Dividing 1TB of data by the 3600 disks and 150 seconds to run the query:*
    - *Resulted in an **average throughput of less than 2 MB/s/disk***
  - *At peak performance MapReduce was reading data at 32GB/s which is less than 10MB/s/disk.*
  - *Conclusion:*

    There is a need for improvement:
    - *Given the long start-up time to get to peak performance, and that peak performance is **four to six times slower** than how fast disks in the cluster could actually be read*

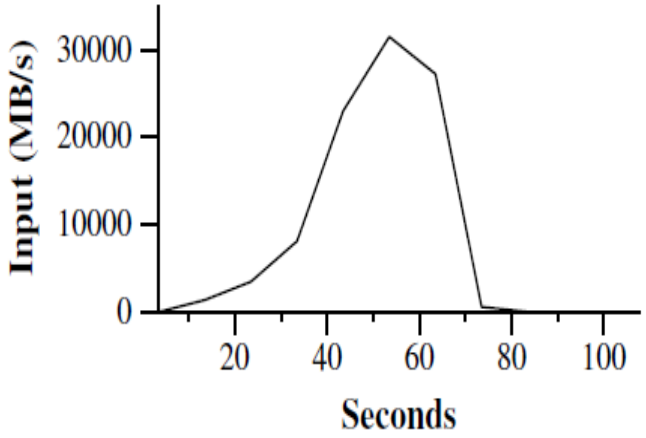

Figure 2: Data transfer rate over time

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **Shared-Nothing Parallel Databases**
  - **Ability to interface with business intelligence products**
    - Not designed to work on top of databases
  - **Efficiency**
    - Improve query performance by implementing:
      - Indexes
      - Materialized views
      - Compressions
  - **Fault Tolerance**
    - Most parallel database systems restart a query upon a failure
    - Designed for environments where queries take no more than a few hours and run on a few hundred machines
    - Failures are rare so an occasional query restart is not problematic
      - Cloud environment offer:
        - Machines are cheaper
        - Less reliable
        - Less powerful
        - Failures more common

Data Analysis in the Cloud:

Available options for open source and commercial database:

- **Shared-Nothing Parallel Databases**
  - **Ability to run in a heterogeneous environment**
    - Designed to run on homogeneous equipment
    - Susceptible to degraded performance:
      - If a small subset of nodes in the parallel cluster are performing poorly
  - **Ability to operate on encrypted data**
    - Not implement
    - Encryption needs to be hand-code to support user defined functions

Data Analysis in the Cloud:

- **A Call For A Hybrid Solution:**
  - *Map-Reduce-like* soft and *Parallel databases* are ideal solutions for data analysis in the cloud
  - Each provide certain features that meets the five recommendation solution for impacting cloud market:
    - Hybrid solution combining:
      - Fault tolerance
      - Heterogeneous cluster
      - Ease of use capabilities

MapReduce + Parallel Database Systems = efficiency, performance, tool plugability

Data Analysis in the Cloud:

- **A Call For A Hybrid Solution:**
  - Current Solutions :
    - Aim to integrate declarative queries from the database community into MapReduce-like software
    - Provideds greater data independence, code reusability, and automatic query optimization
      - ***Pig*** project (Yahoo)
      - ***SCOPE*** project (Microsoft)
      - ***Greenplum*** and ***Aster Data***
        - Added the ability to write MapReduce functions over data stored in their parallel database products

Data Analysis in the Cloud:

- **A Call For A Hybrid Solution:**

There remains a need for a hybrid solution at the systems level in addition to at the language level:

- Future research:
  - How to combine the advantages of MapReduce-like software with the efficiency and shared-work advantages that come with loading data and creating performance enhancing data structures?
    - Proposal:
      - Incremental algorithms
      - Allows data to be initially read directly off the file system
        - Each time data is accessed progress is made towards the many activities surrounding a DBMS load
          - Compression
          - Index
          - Materialized view creation

Data Analysis in the Cloud:

- **A Call For A Hybrid Solution:**

There remains a need for a hybrid solution at the systems level in addition to at the language level:

- Future research:
  - How to balance the tradeoffs between fault tolerance and performance?
    - Through fault tolerance maximization by carefully check pointing intermediate results at the cost of performance
    - Providing a system that can adjust its levels of fault tolerance on the fly given an observed failure rate.

Summary:

- The advantages and disadvantages of deploying database systems in the cloud
- *Read-mostly* analytical database management applications are best suited for cloud deployment, than transactional data management applications
- Large scale data analysis is not suited for cloud deployment and there is a need for a new design of DBMS to handle this type of analysis
- The need for hybrid solutions at the systems level that combines the features of MapReduce and Parallel Database Systems

Questions?