Short Introduction to Topics of This Course

1. Complex Event Processing (CEP)
2. Stream Processing (SP)
3. Mining (Association Rule, Graph)
4. Cloud Computing and Map/reduce

Pro-Active Technology vs. Reactive Technology (also known as Complex Event processing or CEP)

Importance

- We have been using “pull” paradigm for a long time (for information)
- We still use it in a number of places
- Is there a better alternative?
- Is so, what is it and how do you quantify it?
- How do we generalize it? Implications?
- Where is this paradigm useful?
Active Technology View (Push)

Push Paradigm

JIT Push

Stream Processing (SP)
Traffic Data
Command Center
Emergency Services
Stream Processing, Mining, Graph analysis …
Video situation analysis …
Notification, Alerts, Rules, …
Transforming Large Amounts of Real-time Data Into Actionable Knowledge
Environment Data
Traffic Data
Battlefield/Surveillance
Network traffic/routing data

What’s a Data Stream?
A continuous data stream: a sequence of data items that are ordered by time or an attribute.

<table>
<thead>
<tr>
<th>past</th>
<th>current</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

• Examples of data streams:
  – Readings from a sensor
  – Readings of stock price
  – Traffic packets over a link (Tc)
  – …
• It is considered continuous as monitoring is done over long periods of time

Where do data streams come from?

• Computer Network Management
  – Traffic data; for a OC192 link, traffic data flow can reach 250Mbytes/sec
  – SNMP data
  – Route table information, BGP table, routing forward table
  – Topology information and so on

• Telecommunication System
  – Call Detail Record information; Hundreds of millions of phone call records from ten millions of customers per day
  – Network management data: alarm message, performance data,…

• Sensor System: sensor readings
  – temperature, lights, others

• Financial System: transaction records.
• Web system, health care and many others.

Database Approach
1. Load all sensor data into databases
2. Define triggers to detect conditions and trigger actions.
So why not use a DBMS?

- Expensive/complicated to set up & maintain
- This cost & complexity must be offset by need
- General-purpose, not suited for special-purpose tasks (e.g. stream, text processing!)
- Not good for applications that need real-time processing

Problems with the Database Approach

1. Time to load all sensor data into a database is large due to the large volume of sensor readings.
2. The old data in the database is not useful since we are only interested in current status.
3. The delay introduced by this approach can be large, which is not acceptable for critical applications
4. A large amount of triggers are not well supported by current database management systems.

Homeland Security

From: http://www.fas.org/spp/starwars/progr...
Summary

Sharma Chakravarthy: DEBS 2008

Database Streams Applications

Current/earlier Approach

Stream Processing Approach

New approach: Data Stream Management System

Challenges of DSMS

- Unbounded memory requirements
  - Blocking operators
  - Synopses and statistic summarizations
  - Sliding windows
- Online processing
  - Need for queues
- Operator/query Modeling
  - Network of queues and servers
  - Queuing theory-based
- Query processing/Optimization
  - Push-based strategies
  - No cost-based model (surprisingly)
  - QoS driven (memory, latency, throughput)
Challenges of DSMS (Contd.)

- Scheduling Strategies
  - Chain scheduling (for memory)
  - Path capacity (for latency, throughput)
  - Hybrid

- Load Shedding
  - To meet QoS Requirements
  - Results in approximate answers

- Admission control
  - If everything else fails
  - Run-time Optimization
  - Based on run-time monitoring
  - Change scheduling and/or activate load shedding

---

Challenges of DSMS (Contd.)

- System Design and Implementation
  - Operator execution
  - Buffer/queue management
  - Scheduling
  - How to monitor output as well as selectivity
  - Run-time optimizer
  - When and how much load to shed
  - Where to shed load?
  - Integrating stream and event processing
    - Synergy between the two
    - Integration issues

---

Outline

- Association rules
  - Traditional Approaches
  - Using Databases and SQL

- Graph mining
  - Main memory approaches
  - Disk-based approaches
  - Database approaches
  - Map/reduce (Distributed approaches)

- Social Network analysis
  - Multiple features
  - Use of multiplexes
  - Communities and Hubs
Motivation

Fraud division, some large telephone company:
"How do we find these guys? There are 10 billion records on 10 million customers in the main database. With all this information we have about our customers and all the calls they make, can't you just ask the database to figure out which lines have been set-up temporarily and exhibited similar calling patterns in the same time periods? The information is in there, I just know it ..."

Problem

- “Find-similar” problem just described is hard
  - e.g., “What products need to be improved?”
  - e.g., “Which books won't be checked out and can be taken off the shelves?”
- Why?
  - Massive amounts of data
  - More and more online data stores (e.g., Web, corporate databases, etc.)
- No easy way to describe what to look for
- Traditional, interactive approaches fail
  - Size of data, different purposes

Data Mining

- Data Mining (DM) is part of the knowledge discovery process carried out to extract valid patterns and relationships in very large data sets
  - Usually don't know what to look for, like a “voyage into the unknown”
- Regarded as unsupervised learning from basic facts (axioms) and data
- Roots in AI and statistics
  - Uses techniques from machine learning, pattern recognition, statistics, database, visualization, etc.

Data Mining has come about due to

- Convergence of multiple technologies
Data Mining

- Causality and correlation
- The above two are different!
- Which one does mining try to identify?
- Show crazy-correlations website!

DM Vs. Machine learning

- ML methods form the core of DM
- Amount of data makes a (big) difference
  - accessing examples can be a problem
  - missing values and incomplete data
- DM has more modest goals: automating the tedious discovery tasks

Cloud Computing, Big Data, and Map/Reduce (CC & M/R)

- Cloud Computing makes computer infrastructure and services available "on-need" or "on-demand" basis.
- The computing infrastructure could include hard disks, development platform, database, computing power, or complete software applications.
- To access these resources from the cloud vendors, organizations do not need to make any large scale capital expenditures.
What is cloud computing

- Organizations need to "pay per use". That is, organizations need to pay only as much for the computing infrastructure as they use.
- The billing model of cloud computing is similar to the electricity payment that we do on the basis of usage.
- Terminology
  - Vendor: cc service provider
  - Organization: cc user

Cloud computing: components

- To get cloud computing to work, you need:
  - Thin clients (or clients with a thick-thin switch)
  - Grid computing: link disparate computers to form one large infrastructure, harnessing unused resources (we will see subtle differences later)
  - Utility computing: paying for what you use on shared servers like you pay for a public utility (such as electricity, gas, and so on).
  - On-demand resource provisioning: not static provisioning, no need to indicate resource requirements ahead

Example

- Consider the official Wimbledon site. The site gets extremely high traffic in the two (to three) weeks when the championship happens. For this two weeks period, this site will have high server usage.
- For rest of the year the site will have low traffic and hence most of the resources will be idle
- Spare capacity need to be maintained or leased from somewhere!
- Internet-scale elasticity

Problems with earlier approaches

- Under-utilized server resources waste computing power and energy (and cost money)
- Over-utilized servers cause interruption or degradation of service levels
- Resource demands are increasingly of highly dynamic nature and Internet-scale
- On-demand resources are a means for faster time-to-market, and cost-effective innovation processes
Characteristics of Big Data

- **4V:** Volume, Velocity, Variety, Veracity

Big Data Analytics

- **Definition:** A process of inspecting, cleaning, transforming, and modeling big data with the goal of discovering useful information, suggesting conclusions, and supporting decision making.

- **Connection to data mining**
  - Analytics include both data analysis (mining) and communication (guide decision making).
  - Analytics is not so much concerned with individual analyses or analysis steps, but with the entire methodology.

Data Center as a computer

- **Claim:** There are dramatic differences between developing software for millions to use as a service versus distributing software for millions to run on their PCs.
  - Availability, dependability
  - Bandwidth (with low latency) to service a large number of users
  - Innovation is fast as the software is in their control!

- This has led to distributed data centers.

Data Center as a computer

- **What are the useful programming abstractions for such a large system?**
- **How do thousands of computers behave differently from a small system?**
- **What must you do differently to run an abstraction on thousands of computers?**
- **Google proposed a two-phase primitive:**
  - Phase 1: Maps a user supplied function onto thousands of computers
  - Phase 2: Reduces the returned values from all those instances into a set of results
Data Center as a computer

- Map/Reduce was born
- Runs on heterogeneous computers (even different generations)
- Runs on heterogeneous OSs
- Scheduler is dynamic and accommodates above (as compared to batch schedulers of Grid computing)
- Failures are handled transparently!
- Hadoop sorted 1 TB in 209 seconds (2009)!!
- Google regenerated its index using Map/Reduce
- Fairly easy to program, easy to understand!

Map/Reduce

- Is a programming paradigm
- Is a parallel programming paradigm
- Is derived from functional programming (specification vs. procedural programming)
- Many a times the question asked is:
  - Is there a difference between Map/Reduce and traditional parallel programming?

What is Map/Reduce?

- Data-parallel programming model for clusters of commodity machines
- Pioneered by Google
  - Processes 20 PB of data per day
- Popularized by open-source Hadoop project
  - Used by Yahoo!, Facebook, Amazon, …

MapReduce

- Input: a set of key/value pairs (generated from input data)
- User supplies two functions:
  - map(k,v) → list of (k1,list(v1))
  - reduce(k1, list(v1)) → list of (k2, v2)
- Sort Group (k’, v’) by k’
- Reduce(k’, v[]) → v’