CS 744: SUMMARY

Shivaram Venkataraman
Fall 2019
• Midterm 2 on Tuesday

• Poster session Dec 13th, 3-5pm details on Piazza

• Final report Dec 17th

• AEFIS Course feedback form!
### Applications
- Machine Learning
- SQL
- Streaming
- Graph

### Computational Engines

### Scalable Storage Systems

### Resource Management

### Datacenter Architecture

- TensorFlow
- Spark
- Hadoop
- Mesos

Open Compute Project
OUTLINE

Unification vs Specialization

Survey results, Discussion

Big data systems: Looking forward
SPECIALIZATION VS UNIFICATION
GENERALITY: “ONE SIZE FITS ALL” DBMS

1970s
Research prototypes: SystemR and INGRES
Main function: OLTP

From 1990s
Rise of business intelligence workloads
OLAP workloads need to be isolated from OLTP
Solution: Scrape data into data warehouses.
STREAM PROCESSING?

Example: Financial feed processing (Bloomberg, Reuters)
EXAMPLE WORKLOAD

Goals: Maximize message processing throughput on single machine

Scenario: Stock tick is late is if it occurs more than $X$ secs from previous tick

Performance comparison:
- 2.8 GHz, 512 MB memory, single SCSI disk
  - 160,000 messages per second with StreamBase
  - 900 messages per second with DBMS
WHY IS IT SLOW?

DBMS: “Outbound” processing model

1. Insert data
2. Index data, commit transaction
3. Process query, return results

Process after store
WHY IS IT SLOW?

“Inbound” data processing

1. Push inputs into system
2. Process query
3. Return results
4. Optionally store (async)

Only way to do this in DBMS: Triggers
Not performant
**OUTBOUND**

“Pull” records given query
Store data, run any query

**INBOUND**

“Push” records into query
Store queries, pass data through
IS IT JUST STREAMING?

Sensor Networks: TinyDB

Text Search: GFS / MapReduce

Scientific databases: SciDB

Data warehouses
  Column stores, read-oriented vs. write oriented
BIG DATA SYSTEMS

Unified systems

Specialized systems
BENEFITS

Unified systems

Specialized systems
IS IT JUST A CYCLE?
WHERE ARE WE IN THE CYCLE?

PostgreSQL

CIEL

Dryad

2004 - 2011

Spark SQL
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Apache Spark

2011 - 2015

TensorFlow

PyTorch

Apache Flink

2015 - now
BOOTSTRAPPING UNIFIED SYSTEMS?

1. Implement a system/app/functionality that is superior to what is out there
2. Rapidly build an ecosystem providing additional functionalities

Example:

Tensorflow initially target SGD/deep learning
Unifies number of other features
- tf.data supporting map, flat_map etc.
- tf.linalg implementing linear algebra
- tf.sparse for sparse data / shallow models
SURVEY RESULTS
LEARNING OBJECTIVES

At the end of the course you will be able to

- Explain the design and architecture of big data systems
- Compare, contrast and evaluate research papers
- Develop and deploy applications on existing frameworks
- Design, articulate and report new research ideas
DISCUSSION

https://forms.gle/sQFiAKwiQfHEKkPd8
What were some of your goals when you started the course? (Think about the first survey.) Reflect on what part of your goals have been achieved and how.
In the class, we discussed one trend across systems of unification vs. specialization. What are some other trends you have noticed across the papers in the class?
LOOKING FORWARD
NEXT-GENERATION BIG DATA SYSTEMS?

- Workloads
- Data Processing Systems
- Hardware
TRENDS IN WORKLOADS

New functionalities
  Data science / AI
  Robotics

New data sources
  Bio-medical data
  Video streams
  IoT / edge devices

DIVERSITY ?
Fairness in ML?
HOW ROBUST IS YOUR SYSTEM?

Adversarial examples

- ‘Duck’
- \times 0.07
- ‘Horse’

- ‘How are you?’
- \times 0.01
- ‘Open the door’
WHAT CAN SYSTEMS RESEARCH DO?

More than performance?
  Latency, throughput, efficiency
  Ease of use

Some other goals to consider?
  Security, Privacy
  Robustness
  Data bias / ethics
COURSE SUMMARY

Large scale data analysis has changed the world
COURSE SUMMARY

Applications

- Machine Learning
- SQL
- Streaming
- Graph

Computational Engines

Scalable Storage Systems

Resource Management

Your System Here?
TRENDS IN HARDWARE

- Intel Core i7 4 cores 4.2 GHz (Boost to 4.5 GHz)
- Intel Core i7 4 cores 4.0 GHz (Boost to 4.2 GHz)
- Intel Core i7 4 cores 3.9 GHz (Boost to 3.6 GHz)
- Intel Xeon 4 cores 3.7 GHz (Boost to 4.1 GHz)
- Intel Xeon 4 cores 3.6 GHz (Boost to 4.0 GHz)
- Intel Core i7 4 cores 3.4 GHz (Boost to 3.8 GHz)
- Intel Xeon 4 cores 3.3 GHz (Boost to 3.6 GHz)
- Intel Core i7 Extreme 4 cores 3.2 GHz (Boost to 3.5 GHz)
- Intel Core 2 Extreme 2 cores, 2.9 GHz
- Intel Core Duo Extreme 2 cores, 3.0 GHz
- Intel Core i7 2 cores, 2.5 GHz
- IBM Power4, 1.3 GHz
- IBM POWERStation 100, 150 MHz
- Digital PowerStation 5/300, 300 MHz
- Digital AlphaStation 4/266, 266 MHz
- Digital AlphaStation 5/500, 500 MHz
- Digital AlphaServer 5/600, 600 MHz
- Digital AlphaStation 5/600, 600 MHz
- Digital AlphaStation 4/266, 266 MHz
- Digital AlphaStation 5/300, 300 MHz
- MIPS M/120, 16.7 MHz
- Sun/4/260, 16.7 MHz
- AX-11/780, 5 MHz

- Intel D850EMVR motherboard (3.06 GHz, Pentium 4 processor with Hyper-Threading Technology)
- IBM RS/6000/540, 30 MHz
- MIPS M2000, 25 MHz

Performance (vs. VAX-11/780)

- 23%/year
- 12%/year
- 3.5%/year
- 52%/year
- 25%/year
TRENDS IN HARDWARE

Domain specific hardware
TPUs, FPGAs, ASICs
Power-efficiency

Heterogeneous I/O
Infiniband, 100Gbps Ethernet
SSD, NVM

SPECIALIZATION!
<table>
<thead>
<tr>
<th>Machine</th>
<th>Memory (GB)</th>
<th>Compute Units (ECU)</th>
<th>Local Storage (GB)</th>
<th>Cost / hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.nano</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>$0.0058</td>
</tr>
<tr>
<td>r5d.24xlarge</td>
<td>244 768</td>
<td>104–96</td>
<td>4x900 NVMe</td>
<td>$6.912</td>
</tr>
<tr>
<td>x1.32xlarge</td>
<td>2 TB</td>
<td>4 * Xeon E7</td>
<td>3.4 TB (SSD)</td>
<td>$13.338</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>488 GB</td>
<td>8 Nvidia Tesla V100 GPUs</td>
<td>0</td>
<td>$24.48</td>
</tr>
</tbody>
</table>
OTHER DIFFERENCES

SQL Query API

Standard syntax insufficient for streaming
Need for windows at lowest levels

Server-client arch

Typical DBMS has single server many clients
Untrusted clients → separate processes
Embedded streaming: single address space with C++ UDFs
OTHER DIFFERENCES

Availability
Stream processors need fast recovery
Process-pairs instead of undo/redo logging
Can tolerate loss of tuples?

Synchronizing State
Read-write semaphores between operators
No need for ACID transactions
WHAT CHANGED FOR OLTP?

Memory capacity
- Enables push based architecture
- Reduces time per OLTP query, etc.

Cluster computing
- Transaction mechanisms
- Failure recovery (recover from active replica)
IN-MEMORY DBMS (HSTORE/VOLT-DB)

Support (arbitrary) stored procedures running in RDBMS process space
- Context switch adds to latency, this is faster.
- Consider entire workload when optimizing query and data placement.

Eliminate redo logs: recover by replication (cannot eliminate undo)

Support different data structures and algorithms
- E.g., partition data to avoid multi-threaded access, transaction differently.
Context of DBMS – No one system fits all the workloads

Main areas of differences

- Storing data vs. Storing queries
- Query API
- Server/client architecture
- Availability
- Synchronization
Impact of Big Data Systems?

When Algorithms Discriminate

Claire Cain Miller @clairecm July 9, 2015

The online world is shaped by forces beyond our control, determining the stories we read on Facebook, the people we meet on OkCupid and the search results we see on Google. Big data is used to make decisions about health care, employment, housing, education and policing.

But can computer programs be discriminatory?

There is a widespread belief that software and algorithms that rely on data are objective. But software is not free of human influence. Algorithms are written and maintained by people, and machine learning algorithms adjust what they do based on people's behavior. As a result, say researchers in computer science, ethics and law, algorithms can reinforce human prejudices.

Google's online advertising system, for instance, showed an ad for high-income jobs to men much more often than it showed the ad to women, a new study by Carnegie Mellon University researchers found.

Research from Harvard University found that ads for arrest records were significantly more likely to show up on searches for distinctively black names or a historically black fraternity. The Federal Trade Commission said advertisers are able to target people who live in low-income neighborhoods.

Fairness in ML?

RECENT COMMENTS

Tom July 10, 2015

Discrimination against women persists in other ways. Take the obituary column of the NYT - on a good week, you will find obits for perhaps...

SierramanCA July 10, 2015

"There is a widespread belief that software and algorithms that rely on data are objective." says Ms. Miller. Well, Ms. Miller, two things...

Dalgleish July 10, 2015

Algorithms are written by people. People are biased, not objective. Daniel Kahneman et al. have proven this.

SEE ALL COMMENTS
BIG DATA SYSTEMS

What are some examples of unified systems

Benefits
UNIFICATION  SPECIALIZATION