Relational Database for Finite Element Post-processing

Date: 02/03/2016

Hao Liu
and
Mahabub Alam
Outline

- Description of our work and our approach to solve the problem
- Overview the related work and/or papers and explain how our approach is different from them
- Describe what we have done so far, the problems you have faced, and the issues you expect to face in implementing our solution.
- A brief plan on how our plan to address the aforementioned issues
What is Finite Element Analysis (FEA)?

- A numerical method
- Commonly used method for multi-physics problems

Areas of Applications:

- Dynamics: vibration of high-rise buildings, earthquake, etc.
- Thermal analysis: surface heat radiation, thermal stress of a disc, etc.
- Electrical analysis: piezo actuator, electrical signal propagation, etc.
- Biomaterials: human organs and tissues, etc.
- Solid mechanics: beam deflection, a automotive power train, etc.
Question: If we apply a force on a potato, what are the values of the displacements, stresses, and strains at EACH MATERIAL POINT?

To answer this question, we need to solve the following equations:

\[ e_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right) \]

\[ \sigma_{ij} = 2Ge_{ij} + \lambda e_{kk} \delta_{ij} \]

\[ \frac{\partial \sigma_{i1}}{\partial X_1} + \frac{\partial \sigma_{i2}}{\partial X_2} + \frac{\partial \sigma_{i3}}{\partial X_3} + f_i = 0 \]

Solving 15 equations, among which 9 equations are partial differential equations!!

Exact solution: MISSION IMPOSSIBLE !!!

Changed Mission: Approximate Solution by FEA
The Way FEA Works

Discretize the solid

A quarter of Mr. Potato

Mesh of the 3D solid

Element

Node

Solve linear equations

Formulate a set of linear equations with displacements at each node as unknowns

Using a simple function to approximate the displacements in each element
The Way FEA Works

Physical Problem
Simplifying a real engineering problem into a problem that can be solved by FEA

FEA Model
Discretize/mesh the solid, define material properties, apply boundary conditions

FEA Theory
Choose approximate functions, formulate linear equations, and solve equations

Results
Obtain, visualize and explain the results and make your boss happy

Post-processing (Our focus)

Pre-processing

FEA core computations (Most research works)

Our focus
A database design method for finite element analysis (Xingjian, 1992)

- Some support tools and procedures are developed to aid the database design

Supporting Finite Element Analysis with a Relational Database Backend. Part I: There is Life beyond Files (Gerd Heber and Jim Gray, 2005)

- size and integration (globalization)
- environment in support of large scale FEA
- how FEA pipeline uses SQL Server as common data store
- It uses SQL Server and glue solution phases

Supporting Finite Element Analysis with a Relational Database Backend Part II: Database Design and Access

- It discusses database design, and data loading
- It argues that using a database is simpler to implement
- It performs better because it can use data parallelism

Supporting Finite Element Analysis with a Relational Database Backend Part III: OpenDX – Where the Numbers Come Alive

- Showed visualization and data analysis approach
- Capable of visualizing 3-D models
- Their solution visualized in OpenDX (previous IBM Visualization Data Explorer)
Specifically our projects...

Simplified Circular model

Results:
Problem Definition

Project Questions:

I. How can we know which parameter/file is related to the broken element?

II. Write parameters from the corresponding files
Database Management of the Results
Create tables based on increment numbers in each file. (Toad for MySQL)

+ easy programming
+ data consistency
- less efficiency

Create one table for each file (.txt file)

+ less space
+ key data collected
- hard programming
Method 1: Create a Table for Each File
Method 2: MySQL
Main Challenge in Future Work (MySQL for the data mining)

1. Design a suitable relational database
2. Design several queries for desired tuples
   - data file number
   - corresponding input parameters

Open to suggestions/ideas....
Twitter Mining Using R

CS540 – DATABASE MANAGEMENT SYSTEMS
HAFED ALGHAMDI
Overview

- Twitter Mining
- Approach
- Tools
- Progress
What is Twitter Mining?

- The big Idea
  - Analysis over unstructured set of data (text)

- How is that useful?
  - Making decision (iPhone 6s vs Galaxy S6)
  - Future prediction (The Next U.S. President)
  - Products enhancement (Users’ experience)
Approach

- The approach used in this project
  - Searching
  - Extracting
  - Analyzing (A.K.A Sentiment Analysis)
    - Cleanup Tweets (Filtering, GetTerms, Remove Annotations)
    - Manipulate useful information (Frequent Words)
  - Data Visualization
Tools

- Data Analysts
  - Python
  - Hadoop
- In this project
  - R Language (Statistical computing and Visualization)
  - Shiny user-interface (if there is enough time)
Expected Obstacles

- Learning R Language
- Twitter APIs (twitter package)
- Finding appropriate Tools
  - Visualization (wordcloud package)
  - Twitter Connectivity (Rcurl package)
  - Data Manipulation (tm package)
Progress

- Learning R language
- The connection with Twitter APIs
- Search and Extract Tweets
Q&A
Usable Relational Learning

Parisa S. Ataei

Oregon State University

4\textsuperscript{th} Feb. 2016
Relational Learning

$l$: Database instance (background knowledge)

<table>
<thead>
<tr>
<th>student</th>
<th>phase</th>
<th>years</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>prequals</td>
<td>2nd</td>
</tr>
<tr>
<td>s2</td>
<td>postquals</td>
<td>3rd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>publication</th>
<th>title</th>
<th>author</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>s1</td>
<td></td>
</tr>
<tr>
<td>t1</td>
<td>p1</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td>s2</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td>p2</td>
<td></td>
</tr>
</tbody>
</table>

professor

<table>
<thead>
<tr>
<th>id</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>associate</td>
</tr>
<tr>
<td>p2</td>
<td>assistant</td>
</tr>
</tbody>
</table>

Schema CSE1

Target relation: advisedBy(stud, prof)

$E$: Training examples advisedBy

<table>
<thead>
<tr>
<th>studid</th>
<th>profid</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>p1</td>
</tr>
<tr>
<td>s2</td>
<td>p2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>studid</th>
<th>profid</th>
</tr>
</thead>
<tbody>
<tr>
<td>s2</td>
<td>p1</td>
</tr>
<tr>
<td>s1</td>
<td>p2</td>
</tr>
</tbody>
</table>

Example:
CSE database*: Information about professors, students, projects, etc.

Relational learning algorithm

advisedBy(X, Y) <- publication(Z, X), publication(Z, Y).

*https://alchemy.cs.washington.edu/data/uw-cse/
Data Representation

CSE database:

student | inPhase | yearsInProgram
---|---|---
| id | id | phase | id | years |
s1 | s1 | prequals | s1 | 2nd |
s2 | s2 | postquals | s2 | 3rd |

professor | hasPosition
---|---
| id | id | position |
p1 | p1 | associate |
p2 | p2 | assistant |

CSE1
- It takes a long time to process queries.
- Schema is hard to understand.

CSE2
- It may introduce NULL values to the database.
Does schema affect learning outcome?!

Learned definitions:

\[ \text{advisedBy}(X,Y) \leftarrow \text{inPhase}(X,\text{"postquals"}), \]
\[ \text{hasPosition}(Y,\text{"associate"}), \text{publication}(Z,X), \]
\[ \text{publication}(Z,Y). \]

\[ \text{advisedBy}(X,Y) \leftarrow \text{false}. \]
Schema Independent Learning Algorithm

$I_{CSE1}$

<table>
<thead>
<tr>
<th>student</th>
<th>inPhase</th>
<th>yearsInProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id</td>
<td>id</td>
</tr>
<tr>
<td>s1</td>
<td>prequals</td>
<td>2nd</td>
</tr>
<tr>
<td>s2</td>
<td>postquals</td>
<td>3rd</td>
</tr>
</tbody>
</table>

professor

<table>
<thead>
<tr>
<th>hasPosition</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>p1</td>
</tr>
<tr>
<td>p2</td>
</tr>
</tbody>
</table>

advisedBy

<table>
<thead>
<tr>
<th>studid</th>
<th>profid</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>p1</td>
</tr>
<tr>
<td>s2</td>
<td>p2</td>
</tr>
</tbody>
</table>

Algorithm $A$ \( \rightarrow \) \( h_{CSE1} \)

$I_{CSE2}$

<table>
<thead>
<tr>
<th>student</th>
<th>professor</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id</td>
</tr>
<tr>
<td>phase</td>
<td>position</td>
</tr>
<tr>
<td>s1</td>
<td>p1</td>
</tr>
<tr>
<td>s2</td>
<td>p2</td>
</tr>
</tbody>
</table>

advisedBy

<table>
<thead>
<tr>
<th>studid</th>
<th>profid</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>p1</td>
</tr>
<tr>
<td>s2</td>
<td>p2</td>
</tr>
<tr>
<td>s1</td>
<td>p2</td>
</tr>
</tbody>
</table>

Algorithm $A$ \( \rightarrow \) \( h_{CSE2} \)
Schema Independent Learning Algorithm

$\mathcal{I}_{CSE1}$

<table>
<thead>
<tr>
<th>student</th>
<th>inPhase</th>
<th>yearsInProgram</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>prequals</td>
<td>2nd</td>
</tr>
<tr>
<td>s2</td>
<td>postquals</td>
<td>3rd</td>
</tr>
</tbody>
</table>

$\mathcal{E}$

<table>
<thead>
<tr>
<th>advisedBy</th>
</tr>
</thead>
<tbody>
<tr>
<td>studid</td>
</tr>
<tr>
<td>s1</td>
</tr>
<tr>
<td>s2</td>
</tr>
</tbody>
</table>

$\mathcal{I}_{CSE2}$

<table>
<thead>
<tr>
<th>student</th>
<th>inPhase</th>
<th>years</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>prequals</td>
<td>2nd</td>
</tr>
<tr>
<td>s2</td>
<td>postquals</td>
<td>3rd</td>
</tr>
</tbody>
</table>

$\mathcal{E}$

<table>
<thead>
<tr>
<th>advisedBy</th>
</tr>
</thead>
<tbody>
<tr>
<td>studid</td>
</tr>
<tr>
<td>s1</td>
</tr>
<tr>
<td>s2</td>
</tr>
</tbody>
</table>

Algorithm A

$h_{CSE1}$

Algorithm A is schema independent

$h_{CSE2}$
Extended ProGolem

### Usable Relational Learning

**4th Feb. 2016**

**Parisa S. Ataei** (Oregon State University)

---

### Algorithm & Implementation

#### Extended ProGolem

```plaintext
student  inPhase  yearsInProgram
id      id      phase      id  years
s1      s1      prequals  s1  2nd
s2      s2      postquals s2  3rd

publication  professor  hasPosition
title  author  id  position
 t1     s1     p1  associate
 t1     p1     p2  assistant
 t2     s2     p2

advisedBy
studid  profid
s1      p1
s2      p2

positive
```

---

**General hypotheses**

- `advisedBy(X, Y) <- student(X), professor(Y), publication(Z, X), publication(Z, Y).`

**Generalization operators**

- Generalize clause over example 1 to cover example 2

**Specific hypotheses**

- `advisedBy(s1, p1)`
- `advisedBy(s2, p2)`

**Bottom clause**

- `advisedBy(X, Y) <- student(X), inPhase(X, "prequals"), yearsInProgram(X, "2nd"), professor(Y), hasPosition(Y, "associate"), publication(Z, X), publication(Z, Y).`
In Memory RDBMS

SQLite
memsql
VOLTDB
Challenges

- Memory size constraint
- Query optimization
- Acceptable performance for various data sets
Results

<table>
<thead>
<tr>
<th>Data set</th>
<th>ProGolem</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSE</td>
<td>0.003ms</td>
<td>0.029ms</td>
</tr>
<tr>
<td>WebKb</td>
<td>0.195ms</td>
<td>0.675ms</td>
</tr>
<tr>
<td>IMDb</td>
<td>208.561ms</td>
<td>1.404ms</td>
</tr>
</tbody>
</table>

Table 1: Runtime comparison of bottom clause construction
Thank You!
Knowledge Explorer

Pooria Azimi
Problem: No Structures
Relationships: Entity Classes

Cabinet Members

- John Kerry
- Hillary Clinton
- Ernest Moniz
- Robert Gates

Presidents

- Obama
- Bush
- Clinton
- Bush Sr.

Parties

- Democratic Party
- Republican Party

Relationships: served in, pred., member of
Relationships: Entity/Class

Cabinet Members
- John Kerry
- Hillary Clinton
- Ernest Moniz
- Robert Gates

Presidents
- Obama
- Bush
- Clinton
- Bush Sr.

Parties
- Democratic Party
- Republican Party

(prev. sec. defense)

party presidents
Knowledge Explorer

- Web app:
  - 3500 JavaScript LOC (front-end SPA)
  - 1200 PHP LOC (back-end API)
- Display relationships and navigate knowledge graph
- Filter results
<table>
<thead>
<tr>
<th>U.S. Presidents</th>
<th>Party Affiliation</th>
<th>Number</th>
<th>Vice President</th>
<th>Years in Office</th>
<th>Age (at inauguration)</th>
<th>Birthplace</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>Democratic</td>
<td>44th</td>
<td>Joe Biden</td>
<td>2009 - 2017, 8 years</td>
<td>54 years 23 weeks</td>
<td>Honolulu, Hawaii</td>
<td>Occidental College</td>
</tr>
<tr>
<td></td>
<td></td>
<td>President</td>
<td></td>
<td></td>
<td></td>
<td>Columbia University</td>
<td>Yale University</td>
</tr>
<tr>
<td>George W Bush</td>
<td>Republican</td>
<td>43rd</td>
<td>Dick Cheney</td>
<td>2001 - 2009, 8 years</td>
<td></td>
<td>New Haven, Connecticut</td>
<td></td>
</tr>
</tbody>
</table>
### Democratic Party

<table>
<thead>
<tr>
<th>Year Founded</th>
<th>1828</th>
<th>Philosophy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>American Liberalism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modern Liberalism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Progressivism</td>
</tr>
<tr>
<td>Years in Existence</td>
<td>187 years</td>
<td>Founder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Andrew Jackson</td>
</tr>
<tr>
<td>Year Founded</td>
<td>1808</td>
<td>Chair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Debbie Wasserman</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schultz</td>
</tr>
<tr>
<td>Website</td>
<td>Democratic Party</td>
<td>Party Name</td>
</tr>
<tr>
<td></td>
<td>({democrats.org})</td>
<td>Democratic</td>
</tr>
</tbody>
</table>
Done

- Basic API
- Display entities
- Display relationships (lists and charts)
- Follow relationships
To-do

• Compare entities?
• Intersect/union lists of entities?
• Interesting relationships?
• Aggregated information?
• Propagate list filterings?
Related tools

- Palantir
- Google’s Knowledge Graph (on the web)
Implement an interpreter for a subset of SQL

→ Encode the abstract syntax as a collection of Haskell data types.
  ◆ The corresponding logic of these data types provides the type system of the SQL language

→ Then focus on the code of the operations themselves.
  ◆ Difficulty: Optimizing the operations
Why write an interpreter?

1. Makes it possible to experiment with extensions to the language.
2. It provides a reference to test other interpreters against.
3. We can implement a version that uses new optimizations and one that doesn’t.
4. Formal proof of properties of the language and any optimizations.
Record types

Our approach uses a **graph structure** for each element. More general than a flat tuple.
Projection Operation

[ [ [ , , ] ] ]

[ [ [ , , ] ] ]

[ [ [ , , ] ] ]
Progress

What we’ve done:

- **syntax** and **state** as Haskell types and data types
- **semantics** as Haskell functions (partially complete)
- example database and expressions

```ghci
ghci> eval microshaftDB (Proj name (Ref employee))
Just [Branch [Leaf "Bitdiddle", Leaf "Ben"],
     , Branch [Leaf "Hacker", Leaf "Alyssa"]
     , Branch [Leaf "Fect", Leaf "Cyd"]
     , ...]
```
Issues and Plans

Issues we’ve faced:

● **Maybe** data type to handle errors complicates semantics
● specifying entries in nested records

Plans to address issues:

● use **Functor**
● use pointer data type and dereference function
Relational Database Based on Natural Language

TEAM: CS Engineers
Members: H.ZHANG, YC.YANG, K.CHEN, K.SHI, X.LEE
Whose Onid is 111?

SQL

```
SELECT Name
FROM Table_Student
WHERE Onid=111
```
Past searches

◆ Lunar (Woods, 1971):
  • Interaction between user’s query and two databases for rock samples.
◆ Paper (Tsukamoto, 2007)
  • Semi-automatic interface of natural language and SQL.
◆ Quepy (Machinalis, 2012)
  • A python framework for transforming natural languages to SPARQL and MQL.
NL to SQL Algorithm

NL to Keyword

SQL to Keyword
Example: student (name, onid)

Input NL: What is xiangli's onid?

Keyword: xiangli, onid

Attribute: name, onid

Table: student

SQL:

```
SELECT onid
FROM student
WHERE name=xiangli
```
Interface of NP to SQL

➢ Database: MySQL

➢ Language: Java

➢ IDE: Eclipse
## Table Content

<table>
<thead>
<tr>
<th>Onid</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>932504512</td>
<td>KaiShi</td>
</tr>
<tr>
<td>932514926</td>
<td>XiangLi</td>
</tr>
<tr>
<td>932532857</td>
<td>YiChiaoYang</td>
</tr>
<tr>
<td>932545938</td>
<td>HeZhang</td>
</tr>
<tr>
<td>932594212</td>
<td>KunChen</td>
</tr>
</tbody>
</table>
Basic Query Panel
Different NLs Get the Same Result
Different NLs Get the Same Result
Conclusion

- What we have done:
  - Algorithm
  - Interface

- Highlights:
  - Easy to use database
  - Extendable (Example: Adding voice module)
Future Work

🌟 Algorithm Part

- Optimization: We will focus on the efficiency of translation between NL and SQL.

🌟 Implement Part

- Select: we will test our algorithm in select part for “<”, “>”. For example, to look up whose grades between 60 to 90.
- Delete: we will implement delete function to prove our algorithm.
Reference


Thanks for your attention!

1 Minutes Q&A
CS540 : Github 2 RDF (G2R).
Midterm project status.

Umme Ayda Mannan
February 4, 2016
Github 2 RDF (G2R)

**Problem:**
Current tools like bitergia, openhub don’t provide answers to high level questions like
“What proportion of developers follow the patch review process mandated by the project?”

**Why is this important:**
• Important to researchers: Is FOSS actually meritocratic?
• Important to project management: Tracking performance.

**Goal:**
Answer such queries and visualize them in meaningful way.
Related work

- Bitergia, Openhub, Github answers queries related to either code base or mailing list, **not both**
- Black Duck Open Hub: [https://www.openhub.net/](https://www.openhub.net/)
Non trivial problem

• Merging multiple disparate sources (Git repositories, Mailing list).
• Large volume of data.
• Messy data requiring a lot of refining.
• Generate training data for NB classifier.
# My approach

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Download project information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Data refining and processing</td>
</tr>
<tr>
<td>Step 3</td>
<td>Store data in sql database</td>
</tr>
<tr>
<td>Step 4</td>
<td>Convert sql database into RDF</td>
</tr>
<tr>
<td>Step 5</td>
<td>Visualization of the queries</td>
</tr>
</tbody>
</table>
Project Status

**Step 1:** Download project related information from
- Github.
- Mailing list.
- Issue Tracking system.

**Step 2:** Refine data
- Match email address to person using Fuzzy logic.
- Classify commits into three categories: bug fix, new feature and other using NB classifier.
Project Status

Step 3
• Store information in a sql database.

Step 4
• Convert sql database into RDF: using D2RQ tool.
• Answer a set of queries which are related to projects and developers.
## Project Status: Quick look

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Download project information</td>
<td>Done</td>
</tr>
<tr>
<td>Step 2</td>
<td>Refining data</td>
<td>Done</td>
</tr>
<tr>
<td>Step 3</td>
<td>Store data in sql database</td>
<td>Done</td>
</tr>
<tr>
<td>Step 4</td>
<td>Convert sql database into RDF</td>
<td>Done</td>
</tr>
<tr>
<td>Step 5</td>
<td>Visualization of the queries</td>
<td>On going</td>
</tr>
</tbody>
</table>
Issues faced

• Merging multiple disparate sources (Git repositories, Mailing list or Bug repository)

• Large volume of data
  • From 2011 to 2013 in lkml 489,233 emails.
  • From 2011 to 2013 in Linux kernel 179,678 commits.

• Refining data
  • Developers use multiple emails and git accounts.
  • Need to match them using fuzzing.

• Classifying the commits.
Future Plan

• Finalize queries to visualize.
  • “What proportion of developers follow the patch review process mandated by the project?”
  • “What number of un-reviewed patches make it into the code base?”

• Implement visualization of queries.
  • D3sparql/sgvizler for visualization.
Thank you
<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Continent</th>
<th>Region</th>
<th>HeadOfState</th>
<th>SurfaceArea</th>
<th>IndepYear</th>
<th>Population</th>
<th>LifeExpectancy</th>
<th>GNP</th>
<th>GNPOld</th>
<th>LocalName</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABW</td>
<td>Aruba</td>
<td>North America</td>
<td>Caribbean</td>
<td></td>
<td>193.00</td>
<td>1986</td>
<td>103000</td>
<td>73.4</td>
<td>828.00</td>
<td>793.00</td>
<td>Aruba</td>
</tr>
<tr>
<td>AFG</td>
<td>Afghanistan</td>
<td>Asia</td>
<td>Southern and Central Asia</td>
<td>Mohammad Omar</td>
<td>652909.00</td>
<td>1919</td>
<td>22270000</td>
<td>45.9</td>
<td>5976.00</td>
<td>NULL</td>
<td>Afghani/Afghanistan</td>
</tr>
<tr>
<td>AGO</td>
<td>Angola</td>
<td>Africa</td>
<td>Central Africa</td>
<td>Jose Eduardo dos Santos</td>
<td>1246700.00</td>
<td>1975</td>
<td>12878000</td>
<td>38.3</td>
<td>6648.00</td>
<td>7984.00</td>
<td>Angola</td>
</tr>
<tr>
<td>AIA</td>
<td>Anguilla</td>
<td>North America</td>
<td>Caribbean</td>
<td></td>
<td>96.00</td>
<td>1967</td>
<td>8000</td>
<td>76.1</td>
<td>63.20</td>
<td>NULL</td>
<td>Anguilla</td>
</tr>
<tr>
<td>ALB</td>
<td>Albania</td>
<td>Europe</td>
<td>Southern Europe</td>
<td></td>
<td>28745.00</td>
<td>1912</td>
<td>3401200</td>
<td>71.6</td>
<td>3205.00</td>
<td>2500.00</td>
<td>ShqipAnera</td>
</tr>
<tr>
<td>AND</td>
<td>Andorra</td>
<td>Europe</td>
<td>Southern Europe</td>
<td></td>
<td>486.00</td>
<td>1773</td>
<td>78000</td>
<td>83.5</td>
<td>1630.00</td>
<td>NULL</td>
<td>Andorra</td>
</tr>
<tr>
<td>AT</td>
<td>[Parliamentary Co-principality]</td>
<td>North America</td>
<td>Caribbean</td>
<td></td>
<td>300.00</td>
<td>NULL</td>
<td>217000</td>
<td>74.7</td>
<td>1941.00</td>
<td>NULL</td>
<td>Nederlandse Antillen</td>
</tr>
<tr>
<td>ARE</td>
<td>United Arab Emirates</td>
<td>Asia</td>
<td>Middle East</td>
<td></td>
<td>83600.00</td>
<td>1971</td>
<td>2441000</td>
<td>74.1</td>
<td>37986.00</td>
<td>36846.00</td>
<td>[Al-Imarat al-Arabiya al-Muttaţa]</td>
</tr>
<tr>
<td>AT</td>
<td>[Parliamentary Co-principality]</td>
<td>South America</td>
<td>Amazonia</td>
<td></td>
<td>2780400.00</td>
<td>1816</td>
<td>3703200</td>
<td>75.5</td>
<td>34023.00</td>
<td>32331.00</td>
<td>Argentina</td>
</tr>
<tr>
<td>ARM</td>
<td>Armenia</td>
<td>Asia</td>
<td>Central Asia</td>
<td>Robert Kotajian</td>
<td>29800.00</td>
<td>1991</td>
<td>3520000</td>
<td>64.4</td>
<td>1813.00</td>
<td>1627.00</td>
<td>Hejazistan</td>
</tr>
<tr>
<td>ASM</td>
<td>American Samoa</td>
<td>Oceania</td>
<td>Polynesia</td>
<td>George W. Bush</td>
<td>195.00</td>
<td>NULL</td>
<td>68000</td>
<td>75.1</td>
<td>334.00</td>
<td>NULL</td>
<td>Amerika Samoa</td>
</tr>
<tr>
<td>AT</td>
<td>[Parliamentary Co-principality]</td>
<td>Antarctica</td>
<td></td>
<td></td>
<td>13120000.00</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>Terres australes franãaises</td>
</tr>
<tr>
<td>AT</td>
<td>Co-administrated</td>
<td>South America</td>
<td>Amazonia</td>
<td></td>
<td>7780.00</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>Terres australes franãaises</td>
</tr>
<tr>
<td>ATF</td>
<td>French Southern territories</td>
<td>Antarctica</td>
<td>Antarctic</td>
<td></td>
<td>13120000.00</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>0</td>
<td>NULL</td>
<td>Terres australes franãaises</td>
</tr>
<tr>
<td>ATG</td>
<td>Antigua and Barbuda</td>
<td>North America</td>
<td>Caribbean</td>
<td>Jacques Chirac</td>
<td>442.00</td>
<td>NULL</td>
<td>68000</td>
<td>70.5</td>
<td>612.00</td>
<td>584.00</td>
<td>Antigua and Barbuda</td>
</tr>
<tr>
<td>AUS</td>
<td>Australia</td>
<td>Oceania</td>
<td>Australia and New Zealand</td>
<td></td>
<td>7741220.00</td>
<td>1901</td>
<td>18886000</td>
<td>79.8</td>
<td>351182.00</td>
<td>392911.00</td>
<td>Australia</td>
</tr>
<tr>
<td>AUT</td>
<td>Austria</td>
<td>Europe</td>
<td>Western Europe</td>
<td>Thomas Kristl</td>
<td>438559.00</td>
<td>1804</td>
<td>8018000</td>
<td>77.7</td>
<td>211860.00</td>
<td>206025.00</td>
<td>Oesterreich</td>
</tr>
<tr>
<td>AZE</td>
<td>Azerbaijan</td>
<td>Europe</td>
<td>Western Europe</td>
<td>Heydar Aliyev</td>
<td>86000.00</td>
<td>1918</td>
<td>7734000</td>
<td>62.9</td>
<td>4127.00</td>
<td>4000.00</td>
<td>AzErbaycan</td>
</tr>
<tr>
<td>BDI</td>
<td>Burundi</td>
<td>Africa</td>
<td>Eastern Africa</td>
<td>Pierre Buyoya</td>
<td>27834.00</td>
<td>1962</td>
<td>6695000</td>
<td>46.2</td>
<td>903.00</td>
<td>982.00</td>
<td>Burundi/Uburundi</td>
</tr>
<tr>
<td>BEL</td>
<td>Belgium</td>
<td>Europe</td>
<td>Western Europe</td>
<td>Pierre Buyoya</td>
<td>30518.00</td>
<td>1830</td>
<td>10390000</td>
<td>77.8</td>
<td>249704.00</td>
<td>243948.00</td>
<td>Belgique/Belgique</td>
</tr>
<tr>
<td>BEN</td>
<td>Benin</td>
<td>Africa</td>
<td>Western Africa</td>
<td>Mathieu Kaboré</td>
<td>112622.00</td>
<td>1960</td>
<td>6097000</td>
<td>50.2</td>
<td>2357.00</td>
<td>2141.00</td>
<td>BAmunin</td>
</tr>
</tbody>
</table>
Visualizing Database Using D3

Team members:
Akshay Salvi
Bhargav Pandya
Rahul Patel
Shivani Wanjara
Problem

- Everyone likes to see attractive and appealing visuals.
- Even though database queries have been used in many of application, there are only a limited number of ways to represent results, most common being a result table.
- Will this be a pleasure to the eyes?
Approach

Represent the tables from the database that are

1. Easy-to-read
2. Attractive
3. Visually pleasing
Data Driven Documents (D3)

- Javascript Library
- Manipulates DOM objects using data.
- Customized visualizations.
- Extremely fast.
- Supports large datasets.
- Dynamic and interactive behavior.
- Works directly with SVG and is generic.
Work So Far

- MySQL Database.
- Geographical database like MONDIAL.
- Tables like Country, State, City, Boundaries, Water bodies etc.
- SELECTion and Aggregation Queries
Difficulties Faced

- Choosing the correct type of Visualisation.
Thank You.

Questions?
Data Cleansing and Integration

Neil Parmar
Cleansing

- **ETL**
  - Responsible for pulling data out from sources.
  - Cleaning data.
  - Placing data into the data repository.

- **E-LT**
  - Cost-effective data transformation.
  - ODI
Cleansing Importance

- Increase in ETL complexity
- Alienating data cleansing process to reduce ETL workload.
Google Refine

- Messy data and inconsistency
- Edit/Undo/Redo options
- Security
- Any particular data format
Recent Work...

• Process of Data Cleaning.

• Architecture

• Possibility to add other open refine extensions
Student Recruitment Application

CS 540 DATABASE MANAGEMENT SYSTEMS
Team Members

Swathi Sri Vishnu Priya Rayala – rayalas@oregonstate.edu
Jake Joseph – josepjak@oregonstate.edu
Prathveer Rai – raip@oregonstate.edu
Introduction

- An interface where students can collaborate on projects and recruiters can search for potential candidates.

- Students can choose to initiate a new project idea or collaborate with other students on existing projects.

- Student contributions are quantified and can be statistically visualized using Google charts.

- Based on this visualization, the expertise of the student in different aspects of the project such as development, testing and documentation can be understood.
A search interface is provided in the application.

Students can search for projects, skills/languages they wish to work with.

Recruiters can search for potential students based on different criteria.
**Sample Screen shots of Application**

**Students FOSS Project**

### Notifications

<table>
<thead>
<tr>
<th>Idea</th>
<th>Contribution</th>
<th>Activity</th>
<th>Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panic Button</td>
<td>Came up with the Idea</td>
<td>2016-01-17</td>
<td>Vishnu</td>
</tr>
<tr>
<td>Student FOSS Launcher</td>
<td>Worked on Testing</td>
<td>2016-01-17</td>
<td>Vishnu</td>
</tr>
<tr>
<td>Student FOSS Launcher</td>
<td>Worked on editing the developing the FOSS.</td>
<td>2016-01-17</td>
<td>Vishnu</td>
</tr>
<tr>
<td>Student FOSS Launcher</td>
<td>Came up with the Idea</td>
<td>2015-09-16</td>
<td>Vishnu</td>
</tr>
<tr>
<td>Student FOSS Launcher</td>
<td>Worked on Documenting the Features</td>
<td>2016-01-16</td>
<td>Vishnu</td>
</tr>
</tbody>
</table>

### Profile Overview

<table>
<thead>
<tr>
<th>Type</th>
<th>Points Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiator</td>
<td>23.6%</td>
</tr>
<tr>
<td>Document...</td>
<td>20.1%</td>
</tr>
<tr>
<td>Design</td>
<td>19.0%</td>
</tr>
<tr>
<td>Developm...</td>
<td>15.3%</td>
</tr>
<tr>
<td>Testing</td>
<td>23.6%</td>
</tr>
</tbody>
</table>

**Total Points : 635**
Sample Screen shots of Application

You searched for java

Search Results

<table>
<thead>
<tr>
<th>Student</th>
<th>Contact</th>
<th>Projects</th>
<th>Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vishnu</td>
<td><a href="mailto:rayalas@oregonstate.edu">rayalas@oregonstate.edu</a></td>
<td>Student FOSS Launcher, Panic Button</td>
<td>C, C++, DB2, Java, MVC Framework, MySQL, PHP, .NET, html, pptx</td>
</tr>
<tr>
<td>Lakshman</td>
<td><a href="mailto:lakshman@oregonstate.edu">lakshman@oregonstate.edu</a></td>
<td></td>
<td>Database, Java, Data Warehousing</td>
</tr>
</tbody>
</table>
An example of the issue of access path independence is as follows:

- Suppose we have two tables Employee and Manager.
- We decide to merge these two tables into one, which is Department.
- Then the same Query statements used to query the initial two tables (Employee and Manager) will fail after this change, as these tables do not exist anymore.

Hence we can see here that a change in the conceptual level through deleting the initial tables (Employee and Manager) does affect the application accessing the database.

The aspect of providing logical independence is a tough issue.
To overcome this issue, we propose that creating views of previous or changed logical relations so that their information remains.

Hence to the example issue specified above, we can

- Create Views of the essentially deleted relations, Manager and Employee.
- Now the same Query statements used to query those tables will still work.
- Any changes to attributes in the new relation, in this case Department can be analyzed and accordingly set inside the newly created views (Employee and Manager) of the old relations, thereby saving the access path to the queries that originally accessed these tables.
- SQL allows update operations its created view instances.

Hence, allowing logical data independence to exist in the matter of access paths
Thank you!!

Questions/Comments?
Midterm Presentation for
CS 540 DBMS
DR. ARASH TERMEHCHY
Data Mining in and for Computer Networks

OMKAR THAKUR (CS)
ARSALAN KHAN BANGASH (ECE)
What is Data Mining?

- It is the practice of examining large databases in order to generate new information.
- It is used by ecommerce sites,
Evolution of Networks

Speed
- The speed of internet in 90s when it started was 28.8 kbps.
- Gradually it became 56 kbps.....then 128 kbps at the end of 90s
- Now common speeds are 1500 Kbps or 1.5 Mbps
The data on the internet has also increased almost 98% from last 5 years.

It is set to increase 10 fold to 44 zettabytes in 2020.
Challenges in Network Mining

- Keeping a network under control and capturing and analyzing its data is the key challenge in today’s era of internet.
- And this challenge is going on increasing day by day due to increasing in the amounts of data and internet.
Current situation

On-line monitoring and analysis

Capture → On-line Processing → Dump results

Off-line analysis

Off-line Processing → Dump results

Disk → Disk
In order for traffic capture and analysis, the time to receive a minimum size Ethernet frame at 10 Gbps is less than 70ns.

Therefore, a few hundred clock cycles are left to multi Ghz processor for handling a captured packet.

When many datasets are integrated then it will be a huge problem. Therefore there are delays in the data mining in such a network.
- Multiprocessor machines that concurrently process multiple packets

- Data Transport Middleware UDT-Protocol

- Data Mining Middleware
Multiprocessor Machines

- Deployment of multiprocessor machines that concurrently process multiple packets

- Therefore due to this, the time availability to handle a single packet increases.

- DISADVANTAGE:
  - multiprocessor systems are costly and power consuming

- Fig # of ur choice
Our Planning for solution

- Analyzing different systems and thorough reading of research papers.
- Comparing results of surveys and research papers
- Implementing the best feasible solutions
Data Transport Middleware UDT-Protocol

- An application level data transport protocol
- UDT acts like UDP but has its own reliability control and congestion control mechanism
- UDT supports multiple congestion control algorithms
- End users will use UDT Gateway to transfer files
- It works as TCP, web application or HTTP but data resides on a data server connected to a UDT gateway machine
Data Mining Middleware

- Data Transport Middleware will not enable wide-area data intensive applications
- DMM used to scale the high volume data flow
- DMM is supported by streaming model for processing data
- 2 steps on algorithm
  a) Data examined only once
  b) Fixed amount of storage available
Data mining algorithms are computed using Histograms.

Binary partition dynamic histogram algorithm is designed to scale high volume data flows.

BESH – Best Effort streaming Histogram

Algorithm has bucket each bucket covers a scope of multiple values rather than single values.
Thank You
Visualize RDF data management system based on JENA

Jingyuan Xu, Hao Wang, Xinran Peng, Kui Wang, Changshuang Kou
Problem: Help users using RDF operations

Importance: Simplify the old data format
Three Key Point

- Data converter
- Database operation
- Data Visualization
Related work

- Data converter
  - RDF Extension
  - RDF123
  - csv2rdf4lod
  - Tarql
Related work

- Data visualization
- GraphVizdb
Related work

- Database operations
  - We apply a graphical interface and users can do the database operations without using terminal or configuration environment
Users have CSV files, then.

1. CSV to **N-Triples** (Jena)
   1. We will use an jena package--propertytable

2. **riotcmdx**

   A command line tool for direct and scalable transforming from CSV to the formatted RDF syntax (i.e. N-Triples),
What we want?

2. N-Triples to RDF/XML

org.apache.jena.rdf.model
model.read & model.write

- <rdf:RDF>
  - <rdf:Description>
    <j.1:address>13th</j.1:address>
    <j.1:phone rdf:datatype="http://www.w3.org/2001/XMLSchema#double">463256533</j.1:phone>
    <j.1:line rdf:datatype="http://www.w3.org/2001/XMLSchema#double">45</j.1:line>
    <j.1:row rdf:datatype="http://www.w3.org/2001/XMLSchema#double">23</j.1:row>
    <j.1:name>d</j.1:name>
    <j.0:row rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">2</j.0:row>
  </rdf:Description>
  - <rdf:Description>
    <j.1:address>7th</j.1:address>
    <j.1:phone rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4563234</j.1:phone>
    <j.1:line rdf:datatype="http://www.w3.org/2001/XMLSchema#double">11</j.1:line>
    <j.1:row rdf:datatype="http://www.w3.org/2001/XMLSchema#double">12</j.1:row>
    <j.1:name>s</j.1:name>
    <j.0:row rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1</j.0:row>
  </rdf:Description>
</rdf:RDF>
Further work

- Standard (File stream) or Extension (N-Triple)?
- Implement RDF operations & visualization w/ Jena
- More usable
References

- [CSV](https://www.w3.org/wiki/ConverterToRdf#CSV_Comma-Separated_Values)
- [GraphViz DB](http://83.212.105.129:8080/graphVizdb/)
Thank you
Online Video Sharing System

Marvel Team:

Luyao Zhang
Ying Dai
Zhen Tang
Ruizhi Guo
Punyapich Limsuwan

Oregon State University
College of Engineering
Proposed Ideas

Online Video Sharing System

Make profit

Advertisements

View
Click
Publish

View
Search
Upload
Rate
Subscribe

Play together

Registered Users

Part of Income

Partners

Published Videos

Recommend

Publish high quality videos

Main Goals:
1. Increase revenue of the website from advertisement
2. Improve user experience through video recommendation
Related Works

Algorithms for advertisement broadcasting and video recommendation

**Amazon: Item-to-Item Collaborative Filtering**
- Item-to-Item Collaborative Filtering: Compare the similar products instead of customers

**YouTube: Social Video Recommendation System**
- Keep track of the video watched on the web by users belonging to a social group using a proxy server.
- Using these watch history to recommend users in group

**Cookie-based Advertising**
- Track users’ internet usage behavior and analyze what they may be interested in

**Content and user-oblivious video-recommendation algorithm**
- Algorithm design based on watching sequence (length & frequency)
## Research Plans

<table>
<thead>
<tr>
<th>Collect data &amp; Build the database</th>
<th>Implement SQL queries on the database</th>
<th>Apply machine learning to build models for recommendation &amp; ad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Find the videos with the lowest average rating</td>
<td>Most recent viewed</td>
</tr>
<tr>
<td>Videos</td>
<td>Find recently registered users who have uploaded at least one video</td>
<td>Users’ favorite</td>
</tr>
<tr>
<td>Advertisements</td>
<td>Find poorly targeted advertisement which has a lower click-to-view percentage</td>
<td>High ratings</td>
</tr>
<tr>
<td>Views</td>
<td>Find partners who make the most money from uploaded videos</td>
<td>Advertisement targets</td>
</tr>
<tr>
<td>Partners</td>
<td></td>
<td>Advertisement types</td>
</tr>
</tbody>
</table>
**Progress**

**Platform:** MySQL database management system

**Data collection:** from YouTube and other online video websites

### Users

<table>
<thead>
<tr>
<th>Username</th>
<th>Password</th>
<th>Name</th>
<th>Email</th>
<th>Date Registered</th>
<th>Favorite Category of Videos</th>
</tr>
</thead>
</table>

### Videos

<table>
<thead>
<tr>
<th>Video Id</th>
<th>Title</th>
<th>Uploaded User</th>
<th>Uploaded Time</th>
<th>Category</th>
</tr>
</thead>
</table>

### Advertisements

<table>
<thead>
<tr>
<th>Ad Id</th>
<th>Advertiser</th>
<th>Price</th>
<th>Views Wanted</th>
</tr>
</thead>
</table>

### Ad Target Categories

<table>
<thead>
<tr>
<th>Ad Id</th>
<th>Category</th>
</tr>
</thead>
</table>

### Views

<table>
<thead>
<tr>
<th>Username</th>
<th>Video Id</th>
<th>View Time</th>
<th>Rating</th>
<th>Category</th>
<th>Ad Shown</th>
<th>Ad Clicked</th>
</tr>
</thead>
</table>

### Partners

<table>
<thead>
<tr>
<th>Username</th>
<th>Total Click</th>
<th>Share Percentage</th>
</tr>
</thead>
</table>
Difficulties & Future Works

Data collection
Difficulties: easy to get video information, hard to acquire other data
Eg: users’ information, ad price, partner’s owning…
Solution: make up some users data and advisements data

SQL queries implementation
Difficulties: hard to simulate online website data updating
Eg: view videos, rate, click ad, subscribe to a channel…
Solution: prepare and directly load these data into database, only update them for demo purpose

Advertisement and recommendation modeling
Difficulties: unable to evaluate the performance of built models/used algorithms for advertisement broadcasting and videos recommendation
No standard training/testing data for comparison and validation
Solution: design a special data set for training and testing
Thank you!

Q & A