DATA INTENSIVE RESEARCH AT PNNL

JOHN FEO
CENTER FOR ADAPTIVE SUPERCOMPUTING SOFTWARE

HPC USER FORUM
APRIL, 2012
The landscape

- We lots of data
- We have big machines
- We have lots of problems

**SO WHAT’S THE PROBLEM ???**

Analysts do not have the tools to specify their problem !!!

- SQL ? SPARQL ?
- JAVA ?
- C ?
Knowledge discovery

- Construct relationships and extract critical information in a timely manner

- Mixed data
- Unknown workflows
- Difficult search and optimization problems over complex data types
Complex query workshop

▶ Goals

- Develop a set of abstract graph query patterns
- Instantiable against a set of large triple data stores
- To produce compelling standard queries

▶ Process

- Identify real semantic graph data in many domains (e.g. bionetworks, social networks, e-science, government)
- Define challenging queries representative of graph search patterns common in use cases in those domains
- Identify a set of domain-independent, mathematically abstract search patterns for which the domain-specific queries are instantiations

Given a set of proposals, investigators, and authors of papers on relevant topics, find a subset of suitable authors without conflicts to review each proposal.

A suitable author:

1. has written on a related topic of the proposal (distance measure)
2. has no conflict with any investigator of the proposal
3. resides in North America

A conflict occurs if:

1. The author is a submitter of another proposal
2. The author is married to an investigator of the proposal
3. The author has co-authored a paper with an investigator of the proposal within the last 48 months
What makes it a difficult query?

- The presence of negation
- Inference
  - Geographic partonomy
  - Query composition
  - Topic hierarchy
- Recursion (sub-awards)
- Aggregation (discard papers with more than 12 authors)
- Disjuction
- Directed and undirected links
- Quantitative temporal attributes and arithmetic expressions
- Units of measure (months)
PARTY PROBLEM

You're throwing a party for your friends, but since your friends may not all know each other, you will invite friends of friends such that every one will know at least one person (besides you) at the party. Not to make the party too large or too expensive, you wish to minimize the number of guests and the amount of food consumed.

- Facebook Hacker’s Cup Challenge 2012
Facebook

OUR MISSION
To make the world more open and connected.

- 845 million monthly active users
- 2.7 billion Likes & Comments per day
- 250 million photos uploaded per day
- 100 billion friendships
Assume Facebook was an RDB

- F = \textbf{select}(\text{FRIENDS, JOHN})
- FF = \textbf{select}(\text{FRIENDS, F})
- W = \textbf{select}(\text{FOOD\_CONSUMED, \{F, FF\}})

Create a graph \textbf{G} with nodes from \textbf{F} and \textbf{FF} and undirected weighted edges such that
  - An edge exists between friends
  - The weight of the edge is the sum of the food consumed by the friends

Compute the \textbf{Steiner Tree} of \textbf{G} with terminal nodes \textbf{F}
Let $G = (V, E)$ be an undirected graph with weighted edges and let $R \subseteq V$, determine the least cost connected subgraph spanning $R$. Vertices in $R$ are called terminal nodes and those in $V - R$ are called Steiner vertices.

Note there may be no solution to the Party Problem if the graph of your friends and their friends is disconnected.

- How many levels of friends do you select??
- Compute Steiner Tree on the whole Facebook social graph with terminal nodes $F$
Components of modern search

- **User Interface**
  - Problem specification
  - Data processing
  - Visualization

- **Search / Query**
  - Query Optimization
  - On-the-fly Inferencing
  - Self-describing task management

- **Data Storage & Manipulation**
  - Data Ingestion
  - Dictionary Encoding
  - Materialized Inference

- **Analysis**
  - Statistical analysis
  - Relational methods
  - Graph methods
  - Optimization methods

Result Set
Problem design environment

- X := OpenAccessTo(uri.bioData);
- Y := OpenAccessTo(uri.relationshipExtractor);
- R := AskQuery(ThisQuery, X);
- Move(R, Y); //move query results to Y
- Launch(InSpire, params. R);
- E := Extract(RelExtractor, R, relType);
- G := OpenAccessTo(uri.XMT);
- Move(E, XMT);
- C := Launch(CommunityDetection, E);
An architecture for search
Use the best system

- Relational database server
- MapReduce cluster
- Large graph engine
WE NEED TO TAKE THE PROBLEM SPECIFICATION LAYER SERIOUSLY !!!