SCALABLE GRAPH MINING WITH THE VERTICA ANALYTIC PLATFORM

Shilpa Lawande
Lakshmikant Shrinivas
It’s Labor Day!!
Flashback to 2 years ago …
Sep 2009 … a gaming company was just “trying out” Vertica!

Today… Zynga has one of the largest Vertica DBs!
What Zynga says about Vertica today …

“With over 150 million players, 3TB of new data a day and 230 nodes spread across two clusters Zynga's columnar data warehouse from Vertica is no analytical windup toy.

Ken Rudin, VP of Analytics, Zynga

Vertica is an awesome platform for GRAPH analysis!!
How soon do players stop playing a game?
Often as high as 50%

How many players does one player draw into the game?
If 100 Mafia Wars users are likely to get 5 of their friends to join, VC = 1.05

Lifetime revenue per user

Understanding Social Games
Raising the Viral Coefficient of a Game

• Naïve techniques:
  – Advertise in-game actions to friends
  – Send gifts to players if they gift others, etc.

• Not all players are equal
  – Active v/s Passive players
  – Within active group, gifts are welcome!
  – Within passive group, they are considered wall spam!

• Use **Social Graph analysis** to guide in-game interactions of players!
Typical Analytics Workflow in Online Games

- Load real-time game event data every 5 minutes
- Social Graph of Influencers is calculated every night
- Models resulting fed back into the game for the next day’s games
Graph Problem: Finding the Influencers!

- Social Network Analysis has been studied since the 50’s!!
- Model social relationships in terms of graph/network theory
- Graph Centrality Measures and other metrics
Simulated data from a social game

Scale similar to popular social games
- 20M daily active users, 90M monthly active users
- Monthly active users graph has about 405M edges (“active friends”)
- Most common number of active friends is 3-5
- ~1 TB of raw data
- Hardware: 4 nodes (12 cores, 96GB RAM each)

Demo: Product Placement in a Social Game
Demo: Methodology

Part 1: Identify influential users in graph
- Find approximate 4-core (Note: graph doesn't have a true 4-core)

Part 2: Measure “influence” to guide product placement
- Pick 2 sets of candidates, one set from approx. 4-core, and second randomly
- 4-core users will be gifted (virtual) Coca Cola
- Random users will be gifted (virtual) Pepsi
- Simulate user interactions over a 1-week period
- Compare penetration of Coke vs. Pepsi
Challenge: Visualizing k-cores

No tool we tried could visualize 90M node, 405M edge graph

Generated pictures with smaller graph (10K nodes, 45K edges)

Highlights some of the challenges in dealing with huge graphs
Comparing different approaches

- Bad ideas
  - Brute force CONNECT BY or Many-way Self-Joins don’t scale
  - Custom data structures – limited to in-memory sizes

- Our approach
  - Iteration over relational tables using MPP operations
  - Efficient join execution using MERGE joins on sorted data
  - Reduce data processed with each iteration
A Vertica primer: The Four C’s

Columnar storage and execution
Achieve best data query performance with unique Vertica column store

Clustering
Linear scaling by adding more resources on the fly

Compression
Store more data, provide more views, use less hardware

Continuous performance
Query and load 24x7 with zero administration
Column Store - Reads 3 columns

Row Store - Reads all columns

SELECT AVG(price) FROM tickstore WHERE symbol = ‘AAPL’ AND date = ‘5/06/09’
Multiple Sorted Projections of Data

Different projections may be optimized for different queries
Optimizer is fully aware of sorted and compressed data
Shared-Nothing, MPP Architecture

100% peer-to-peer
- No specialized nodes
- Can query & load to any node

Commodity x86 Hardware
Compression

- 50% – 90% reduction in data size
- Over a dozen compression & encoding types
- Able to operate efficiently on encoded data
- Minimizes hardware required and reduces capital cost
- Up to ~1PB DB in a single rack
Continuous Loading While Querying

- Light-weight Transaction Model
  - No logging
  - Snapshot isolation
- Hybrid in-memory / disk load model
  - Very Fast Trickle Loads enables Real-time Analytics!
- Built-in high availability
No surprise Vertica excels at Graph problems?

- Data available in multiple sort orders – many algorithms benefit from sorting graph by both source or destination nodes
- Sparse graphs encodes really well using RLE
- Merge Joins are very efficient (no sort required)
- Compression / encoding minimizes data sent over network during distributed joins
Demo: Computing approximate 4-core on Vertica: SQL+Iteration

Initialize:
create table ncore1(src int, tgt int);
create table ncore2(src int, tgt int);
create table lessn (src int, count int);
create table lessn2 (src int, count int);

Iteration:
truncate table lessn;
insert into lessn select src, count(tgt) from ncore1 group by src having count(tgt) < 4;

truncate table ncore2;
insert into ncore2 select * from ncore1 where src not in (select src from lessn) and tgt not in (select src from lessn);

truncate table lessn2;
insert into lessn2 select src, count(tgt) from ncore2 group by src having count(tgt) < 4;

truncate table ncore1;
insert into ncore1 select * from ncore2 where src not in (select src from lessn2) and tgt not in (select src from lessn2);

Final:
select src, count(tgt) from ncore1 group by src order by count(tgt) desc limit 10;
Demo: Part 1 Results

- **Data Size:**
  - Raw: 1.1 TB
  - Compressed (2-way replication): 366GB

- **Total Time to compute 4-core:**
  - 5m, 30s (+/- 15s)

- **Breakdown of Timings**
  - Creating Active Friends Graph (90M users, 405 edges): ~1m
  - Initialization: ~20s
  - Average Time per Iteration: ~30s
  - Counting # of users in 4-core: a few ms
  - Number of iterations: 8
  - Number of users after 8 iterations: 34,033
Part 1: Identify influential users in graph

- Find approximate 4-core (Note: graph doesn't have a true 4-core)

Part 2: Measure “influence” to guide product placement

- Pick 2 sets of candidates, one set from approx. 4-core, and second randomly
- 4-core users will be gifted (virtual) Coca Cola
- Random users will be gifted (virtual) Pepsi
- Simulate user interactions over a 1-week period
- Compare penetration of Coke vs. Pepsi
Simulate user activity and measure effects

- Treat problem like infection spreading
  - Probability that user buys a (virtual) soda after interacting with a friend who has it is 5%

- Compare penetration of Coke vs. Pepsi
  - Continuously load data into Vertica in 15 second batches
  - Compare number of users that bought Coke vs. number that bought Pepsi
  - Notice how the penetration of the two products changes over time
Demo: Part 2 Results & Summary

Metrics:
- Load rate on 4 nodes with 2x replication: 40GB / min
- Translates to **20TB/hr** on full (32-node) HP/Vertica Appliance rack
- Average Query Time: 0.3s (Witness the magic of encoding!)

Simple model, but shows that SQL can be a powerful tool for graph analysis

- Derive value from graphs that are too large to visualize
- Useful metrics can be computed in a matter of minutes
Social networks are everywhere. Influence is a powerful tool.

- Telecom – “friends and family”
- Online rating systems
- Online communities and User groups

Gartner: Social CRM $1B market in 2012
Graph Problems are everywhere!

- **Business**: Marketing analytics
- **Problem**: De-duping household data from multiple sources to identify unique individuals, households and residences

- **Nodes**: Data sources such as Trans Union, Experian records
- **Edges**: Person A is same as B
  - USPS National Change of Address records
  - Firstname / Lastname spellings, sounds
  - Same social sec#

Hmmm… This doesn’t look like a database problem, does it?
The solution

• This is a well-known graph problem called “Set Union Find”

• Iterative solution
  – In each iteration, the graph gets progressively smaller
  – Typically converges in small number of iterations

• SQL representation consists of simple Merge Joins and Group Bys

• Very efficient in Vertica because of MPP joins on sorted projections
Counting Triangles: SQL v No-SQL 😊?

- Clustering Coefficient of Node
  - How tightly knit is the network?
  - Applications in spam detection, etc.

- Map-Reduce is widely believed to be the only solution

- Ran a simple 2-pass algorithm on graph of 90M edges
  - Hadoop: 100+ lines of Java
  - Pig: 10 lines
  - Vertica: 3 DDL + 1 SELECT w 3-way join

- Out-of-the-box Vertica ~10x faster
- With some tuning ~20-30x faster
- Space usage: 200x smaller
- Best case Vertica timing: 97s

Social network analysis is a powerful tool!

Databases can be very good at graph problems!

Got SQL Graph Algorithms?

Food for thought …
References and Other Info …

- www.vertica.com
- http://www.faculty.ucr.edu/~hanneman/nettext/C8_E Embedding.html

Write to us:
shilpa@vertica.com
lshrinivas@vertica.com
Twitter: @slawande
Try Vertica:
http://www.vertica.com/evaluate/
Jobs: resumes@vertica.com
Have a great weekend!