Graph Analytics on Massively Parallel Processing Databases

Frank McQuillan
Feb 2017
MPP databases effective for graph analytics at scale in the enterprise
Database Engine Popularity

315 systems in ranking, January 2017

<table>
<thead>
<tr>
<th>Rank</th>
<th>DBMS</th>
<th>Database Model</th>
<th>Score 2017</th>
<th>Score 2016</th>
<th>Score 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Oracle +</td>
<td>Relational DBMS</td>
<td>1416.72</td>
<td>+12.32</td>
<td>-79.36</td>
</tr>
<tr>
<td>2.</td>
<td>MySQL +</td>
<td>Relational DBMS</td>
<td>1366.29</td>
<td>-8.12</td>
<td>+67.03</td>
</tr>
<tr>
<td>3.</td>
<td>Microsoft SQL Server</td>
<td>Relational DBMS</td>
<td>1220.95</td>
<td>-5.70</td>
<td>+76.89</td>
</tr>
<tr>
<td>5.</td>
<td>PostgreSQL</td>
<td>Relational DBMS</td>
<td>330.37</td>
<td>+0.35</td>
<td>+47.97</td>
</tr>
<tr>
<td>6.</td>
<td>DB2</td>
<td>Relational DBMS</td>
<td>182.49</td>
<td>-1.85</td>
<td>-13.88</td>
</tr>
<tr>
<td>7.</td>
<td>Cassandra +</td>
<td>Wide column store</td>
<td>136.44</td>
<td>+2.16</td>
<td>+5.49</td>
</tr>
<tr>
<td>8.</td>
<td>Microsoft Access</td>
<td>Relational DBMS</td>
<td>127.45</td>
<td>+2.75</td>
<td>-6.59</td>
</tr>
<tr>
<td>9.</td>
<td>Redis +</td>
<td>Key-value store</td>
<td>118.70</td>
<td>-1.20</td>
<td>+17.54</td>
</tr>
<tr>
<td>10.</td>
<td>SQLite</td>
<td>Relational DBMS</td>
<td>112.38</td>
<td>+1.54</td>
<td>+8.64</td>
</tr>
<tr>
<td>21.</td>
<td>Neo4j +</td>
<td>Graph DBMS</td>
<td>36.26</td>
<td>-0.56</td>
<td>+3.26</td>
</tr>
<tr>
<td>47.</td>
<td>Titan</td>
<td>Graph DBMS</td>
<td>5.50</td>
<td>+0.04</td>
<td>-0.15</td>
</tr>
<tr>
<td>112.</td>
<td>Giraph</td>
<td>Graph DBMS</td>
<td>1.04</td>
<td>+0.00</td>
<td>+0.11</td>
</tr>
<tr>
<td>180.</td>
<td>GraphDB +</td>
<td>Multi-model</td>
<td>0.31</td>
<td>+0.09</td>
<td>+0.21</td>
</tr>
</tbody>
</table>

http://db-engines.com/en/ranking
Graph Engine Trends

http://db-engines.com/en/ranking

21. Neo4j
47. Titan
112. Giraph
180. GraphDB
Introduction to Graphs

• Graphs can be small...
Introduction to Graphs

• ...but many real world graphs are very large
Why Graph Analytics on MPP Databases?

• MPP is built for very large data sets
• Many enterprise use cases combine graph analytics with other techniques
• SQL
  – Most common workload in the enterprise
  – Widely used by analysts and data scientists
  – Ecosystem of business intelligence applications
Why Graph Analytics on MPP Databases?

• Data locality
  – Cost of replicating, moving and transforming data to an external system can be high

• Policy
  – Cost, deployment, oversight, support issues adding a new execution engine
  – Convince the CIO to use a specialized system in production
But...

Can graph analytic processing be efficiently performed on relational data in an MPP database?
Yes!

• Graph analytic processing on Greenplum database using Apache MADlib can solve for a wide range of real world use cases
Apache MADlib (incubating)
Scalable, In-Database Machine Learning

Apache MADlib (incubating): Big Data Machine Learning in SQL for Data Scientists

- Open source, commercially friendly Apache license
- Supports PostgreSQL, Greenplum Database™, and Apache HAWQ (incubating)
- Powerful analytics for big data

- Open source: https://github.com/apache/incubator-madlib
- Downloads and docs: http://madlib.incubator.apache.org/
- Wiki: https://cwiki.apache.org/confluence/display/MADLIB/
MADlib project was initiated in 2011 by EMC/Greenplum architects and Joe Hellerstein from Univ. of California, Berkeley.

UrbanDictionary.com:

**mad** (adj.): an adjective used to enhance a noun.

1- dude, you got skills.
2- dude, you got mad skills.
**Generalized Linear Models**
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression
- Ordinal Regression
- Cox Proportional Hazards Regression
- Elastic Net Regularization
- Robust Variance (Huber-White), Clustered Variance, Marginal Effects

**Matrix Factorization**
- Singular Value Decomposition (SVD)
- Low Rank

**Linear Systems**
- Sparse and Dense Solvers
- Linear Algebra

**Other Machine Learning Algorithms**
- Principal Component Analysis (PCA)
- Association Rules (Apriori)
- Topic Modeling (Parallel LDA)
- Decision Trees
- Random Forest
- Conditional Random Field (CRF)
- Clustering (K-means)
- Cross Validation
- Naïve Bayes
- Support Vector Machines (SVM)
- Prediction Metrics
- K-Nearest Neighbors

**Time Series**
- ARIMA

**Path Functions**
- Operations on Pattern Matches

**Descriptive Statistics**
- Sketch-Based Estimators
  - CountMin (Cormode-Muth.)
  - FM (Flajolet-Martin)
  - MFV (Most Frequent Values)
- Correlation and Covariance
- Summary

**Inferential Statistics**
- Hypothesis Tests

**Utility Modules**
- Array and Matrix Operations
- Sparse Vectors
- Random Sampling
- Probability Functions
- Data Preparation
- PMML Export
- Conjugate Gradient
- Stemming
- Sessionization
- Pivot

**New in v1.10, more to come**
Example Usage

Train a model

```
SELECT madlib.linregr_train('houses',
  'houses_out',
  'price',
  'ARRAY[1, tax, bath, size]',
  'bedroom'
)

```  
--- Input table
--- Output table
--- Variable to predict
--- Features in data
--- Group data to create
--- multiple models

Predict for new data

```
SELECT houses.*,
  madlib.linregr_predict(ARRAY[1, tax, bath, size],
  model.coef)as predict
FROM houses_test, houses_out as model;

```  
--- Use same features
--- Combine test data
--- and model table
Greenplum Database

Master Servers
Query planning & dispatch

Network Interconnect

Segment Servers
Query processing & data storage

External Sources
Load, streaming, etc.

SQL

Massively Parallel Processing
MADlib on Greenplum

- **Master Servers**
  - Query planning & dispatch

- **Segment Servers**
  - Query processing & data storage

- **Network Interconnect**

- **External Sources**
  - Load, streaming, etc.

- **In-Database Functions**
  - Machine learning & statistics & math & utilities

- **Massively Parallel Processing**

- **Input validation & pre-processing**

- **SQL**, **R**, **Python**

- **MADlib**

- **Pivotal**

- **Load, streaming, etc.**
Graph Representation in MADlib

Directed graph (digraph)

Vertex or node

Edge

Edge weight (can be negative)
Graph Representation in MADlib

### Vertex Table

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Vertex Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
</tr>
</tbody>
</table>

### Edge Table

<table>
<thead>
<tr>
<th>Source Vertex</th>
<th>Dest Vertex</th>
<th>Edge Weight</th>
<th>Edge Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1.0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5.0</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3.0</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>8.0</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3.0</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.0</td>
<td>...</td>
</tr>
</tbody>
</table>
Single Source Shortest Path

- Given a graph and a source vertex, find a path to every vertex such that the sum of the weights of its constituent edges is minimized.

Image from https://en.wikipedia.org/wiki/Shortest_path_problem

Shortest path (A, C, E, D, F) between vertices A and F in the weighted directed graph.
Single Source Shortest Path

• Use cases
  – Vehicle routing/navigation
  – Degrees of separation in a social network
  – Min-delay path in a telecommunications network
  – Plant and facility layout
  – VLSI design
SSSP Performance on Greenplum Database

SSSP, 100 edges per vertex, Greenplum 4.3.10

Time (s)

100 200 300 400 500

Num vertices (X 1000)

Bellman-Ford algorithm
O(VE) worst case but not common

Greenplum cluster:
- 1 master
- 4 segment hosts with 6 segments per host

50M edges
Single Source Shortest Path in MADlib

**SSSP**

```sql
graph_sssp( vertex_table, -- vertex table
    vertex_id, -- col in vertex table containing vertex IDs
    edge_table, -- edge table
    edge_args, -- source, dest and edge weights col in the edge table
    source_vertex, -- source vertex for the algorithm to start
    sssp_table -- output table of SSSP for all dest vertices
);
```

**Path retrieval**

```sql
graph_sssp_get_path( sssp_table, -- sssp table
    dest_vertex -- dest of the path of interest
);
```
Implementation Considerations

• Relationships
  – Not a 1st class citizen in relational databases (unlike certain graph databases)
  – **JOIN** operations are compute and memory intensive so want to minimize

• Table scans
  – Depth first search involves more table scans (expensive) than breadth first search
  – Greedy algorithms that do not take advantage of query optimizer will be slower
Implementation Considerations

• Database limits
  – PostgreSQL limits maximum field size to 1GB
## MADlib Graph Roadmap (Near Term)*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Uses</th>
</tr>
</thead>
</table>
| All pairs shortest path (APSP)   | ● $O(V^3)$ Floyd-Warshall  
● Betweenness and closeness centrality measures to identify influencers  
● Graph diameter                |
| Page rank                        | ● Identify importance of vertices                                                |
| Connected components             | ● Clustering common components  
● Measure of resilience in network flow problems                                 |
| Graph cut                        | ● Partition a graph into two disjoint subsets                                    |

*Subject to community interest and contribution, and subject to change at any time without notice.*
Cybersecurity Example
Lateral Movement Detection
Defending the perimeter no longer enough

No 100%, fool-proof way to keep bad actors out

Some threats come from within

The idea of a perimeter becoming obsolete with mobile, cloud, IoT

Need better methods for threat detection inside the network
A handful of users are targeted by two phishing attacks: one user opens Zero day payload (CVE-02011-0609). The user machine is accessed remotely by Poison Ivy tool. Attacker elevates access to important user, service and admin accounts, and specific systems. Data is acquired from target servers and staged for exfiltration. Data is exfiltrated via encrypted files over ftp to external, compromised machine at a hosting provider.
Lateral Movement Detection

**What:** Identify anomalous user-level access to hosts

**How:** Look at people & machines
- Users (user behavior models)
- Network, servers (user peer models)

**Scenarios:**
- Network reconnaissance from remote adversary on hijacked device
- Ill-intentioned activities by legitimate employee
- Access policy abuse

**Business values:**
- Immediate security alert generation
- Enhanced SIEM alert queue prioritization
- Focused monitoring
- Future integration with other analytic models for 360° attack view
Lateral Movement Detection (LMD) – Flow Diagram

Logs
- Active Directory Activity
- Active Directory Metadata
- Server Information
- LDAP Activity

Data Store
- Semi-structured
- Structured

External Tables

Graph Model

Regression Model

Cluster Model

Recommendation System

User Behavioral Model

Anomalous Users

Greenplum
Regression-Based Model

Model to identify users with unusual variation in the number of servers accessed over time
Build a regression model for each user \((Y = aX + b)\)
No. of servers accessed each week \((Y)\) ~ Week Index \((X)\)
Find the slope of the regression line for each user \((a)\)
Identify users who have a high positive or negative slope to find users with unusual activity
User Behavior Models (UBM)

Build historical behavioral profile for each user based on following features:

- Servers accessed
- IP addresses logged in from
- Geographical information of login

Models stress individual user/job log-in frequency

Multiple **Feature Generations** reduce false alarms:

- Aggregate servers to respective server group
- Incorporate server criticality
- Assign more weight to less popular servers and IP addresses
- E.g. print servers are low-weighted
- Use recommendation engine to suggest servers to users based on job roles and peers
Using historical windows events data to build graphs* of typical user behavior
- Which machines does the user log into?
- Which machines does the user log in from?
- How often?
- In which order?

Ask if this behavior is typical
- Is it typical for this user?
- Is it typical for someone in a particular department?
- Is this typical for someone in the user’s job role?

Graph models are sensitive to direction, order, and frequency

• 4th Apache MADlib (incubating) release Feb 2017
• Project is moving toward top level status

You are welcome to join us!!!
MPP databases effective for graph analytics at scale in the enterprise
References

[1] The case against specialized graph analytics engines

[2] MADlib papers
https://www2.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-38.pdf


Apache MADlib Resources

- **Web site**

- **Wiki**

- **User docs**

- **Technical docs**
  - [http://madlib.incubator.apache.org/design.pdf](http://madlib.incubator.apache.org/design.pdf)

- **Pivotal commercial site**
  - [http://pivotal.io/madlib](http://pivotal.io/madlib)

- **Mailing lists and JIRAs**
  - [https://mail-archives.apache.org/mod_mbox/incubator-madlib-dev/](https://mail-archives.apache.org/mod_mbox/incubator-madlib-dev/)
  - [http://mail-archives.apache.org/mod_mbox/incubator-madlib-user/](http://mail-archives.apache.org/mod_mbox/incubator-madlib-user/)
  - [https://issues.apache.org/jira/browse/MADLIB](https://issues.apache.org/jira/browse/MADLIB)

- **PivotalR**
  - [https://cran.r-project.org/web/packages/PivotalR/index.html](https://cran.r-project.org/web/packages/PivotalR/index.html)

- **Github**
  - [https://github.com/apache/incubator-madlib](https://github.com/apache/incubator-madlib)
  - [https://github.com/pivotalsoftware/PivotalR](https://github.com/pivotalsoftware/PivotalR)
Thank you!
Backup Slides
MADlib Execution Flow

Client → Database Server → Master → Segment 1 → Segment 2 → Segment n

SQL → String → Stored Procedure → Result Set → Aggregation

psql
Iterative Model Execution

Stored Procedure for Model

```plaintext
model = init(...)
WHILE model not converged
   model =
      SELECT
         model.aggregation(...)
      FROM
         data table
   ENDWHILE
```

Transition Function
Operates on tuples or mini-batches to update transition state (model)

1. Transition Function
2. Merge Function
   Combines transition states
3. Final Function
   Transforms transition state into output value
MADlib Architecture

User Interface

High-Level Iteration Layer
(iteration controller)

Functions for Inner Loops
(implements ML logic)

Low-level Abstraction Layer
(array operations,
C++ to DB type-bridge, …)

C API
(Greenplum, PostgreSQL, HAWQ)

RDBMS
Built-in Functions

Python

C++

SQL
Greenplum Database

**System Access**
- **Client Access**
  - PSQL, ODBC, JDBC
- **Bulk Load/Unload**
  - GPLoad, GPFdist, External Tables, GPHDFS
- **Admin Tools**
  - GP Perfmon, GP Support
- **3rd Party Tools**
  - Compatible with Industry Standard BI & ETL Tools

**Data Processing**
- **SQL Standard Compliance**
- **Workload Management**
  - Resource Queues
  - GP Workload Manager
- **Big Data Query Processing**
  - GPORCA Optimizer
  - MPP Query Execution
- **In-Database Programming Languages**
  - PL/pgSQL, PL/Python, PL/R, PL/Perl, PL/Java, PL/C
- **In-Database Analytics & Extensions**
  - MADlib, PostGIS, PGCrypto

**Data Storage**
- **Fully Acid Compliant Transactional Database**
- **Polymorphic Storage**
  - HEAP, Append Only, Columnar, External, Compression
- **Multi-Version Concurrency Control (MVCC)**
- **Indexes**
  - B-Tree, Bitmap, GiST
Pivotal Query Optimizer

- Applies broad set of optimization strategies at once
  - Considers many more plan alternatives
  - Optimizes a wider range of queries
  - Optimizes memory usage
- Significant improvements for demanding queries

Turns a SQL query into an execution plan
Cost of Cybercrime on the Rise

- Cybercrime costs average US enterprise $17m per year*
- Cost grew at 15% CAGR over last three years
- Any given cybercrime can cost significantly more
- Target’s 2014 hack cost company approximately $162m
- Costs not just financial, also reputational

*Source: 2016 Cost of Cyber Crime Study & the Risk of Business Innovation, Ponemon Institute