Real-time Collaborative Analytics with Vertica Analytic Platform

Stephen Walkauskas,
Architect, Data Management, Vertica
The explosion of data is not news to anyone ...

Every 60 seconds

- 98,000 tweets
- 23,148 + apps downloaded
- 400,710 ads requests
- 208,333 minutes Angry Birds played
- 2000 lyrics played on Tunewiki
Today, data analysis is slow, painful and costly
Imagine a world where you can have a conversation with your data!

Vertica makes this a reality with real time analytics !!!
Introducing Vertica

- SQL Database for Real-time Analytics
- Runs on x86 hardware
- MPP Columnar Architecture – scales to PBs!
- Reduced footprint via Advanced Compression
- Extensible analytics capabilities
- Easy to setup and use
- Elastic - grow/shrink as needed
- Extensive Ecosystem of analytic tools
Proven by 2500+ Customers Worldwide

► Promotional Testing
► Claims Analyses
► Patient Records Analyses
► Clinical data Analyses
► Fraud Monitoring
► Financial tracking
► Tick data back-testing

► Behavior Analytics
► Click Stream Analyses
► Network Analyses
► Customer Analytics
► Compliance Testing
► Loyalty Analysis
► Campaign Management
5 Building Blocks for Collaborative Analytics

Performance that enables Interactive and Iterative Q&A with the Data

Extensible + ability to share (tools, views, code and data)

Ability to record and replay Analyst “thought-process”

Sand-boxing data to enable ad-hoc and intense experimentation

Ability to dynamically access a variety of data sources

Vertica can be an excellent platform for collaboration!
Performance that enables Interactive and Iterative Q&A with the Data

“Vertica opened doors to analyses that otherwise were either too time-intensive or impossible. A larger team of business managers now have faster, easier access to more information. That knowledge is invaluable in an aggressively competitive market like ours.”

-Brian Harvell, Executive director for network operations, Comcast
Column Store Innovation: Efficiency
Column Store – Column-Based Disk I/O

Typical FinServ price per stock for 1 day

Column Store - Reads 3 columns

Row Store - Reads all columns

SELECT AVG(price) FROM tickstore WHERE symbol = 'AAPL' AND date = '5/06/09'
# Column Store – Sort and Encode for Speed

<table>
<thead>
<tr>
<th>Student_ID</th>
<th>Name</th>
<th>Gender</th>
<th>Class</th>
<th>Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1256678</td>
<td>Cappiello, Emilia</td>
<td>F</td>
<td>Sophomore</td>
<td>62</td>
<td>D</td>
</tr>
<tr>
<td>1254038</td>
<td>Dalal, Alana</td>
<td>F</td>
<td>Senior</td>
<td>92</td>
<td>A</td>
</tr>
<tr>
<td>1278858</td>
<td>Orner, Katy</td>
<td>F</td>
<td>Junior</td>
<td>76</td>
<td>C</td>
</tr>
<tr>
<td>1230807</td>
<td>Frigo, Avis</td>
<td>M</td>
<td>Senior</td>
<td>64</td>
<td>D</td>
</tr>
<tr>
<td>1210466</td>
<td>Stober, Saundra</td>
<td>F</td>
<td>Junior</td>
<td>90</td>
<td>A</td>
</tr>
<tr>
<td>1249290</td>
<td>Borba, Milagros</td>
<td>F</td>
<td>Freshman</td>
<td>96</td>
<td>A</td>
</tr>
<tr>
<td>1244262</td>
<td>Sosnowski, Hillary</td>
<td>F</td>
<td>Junior</td>
<td>68</td>
<td>D</td>
</tr>
<tr>
<td>1252490</td>
<td>Nibert, Emilia</td>
<td>F</td>
<td>Sophomore</td>
<td>59</td>
<td>F</td>
</tr>
<tr>
<td>1267170</td>
<td>Popovic, Tanisha</td>
<td>F</td>
<td>Freshman</td>
<td>95</td>
<td>A</td>
</tr>
<tr>
<td>1248100</td>
<td>Schreckengost, Max</td>
<td>M</td>
<td>Senior</td>
<td>76</td>
<td>C</td>
</tr>
<tr>
<td>1243483</td>
<td>Porcelli, Darren</td>
<td>M</td>
<td>Junior</td>
<td>67</td>
<td>D</td>
</tr>
<tr>
<td>1230382</td>
<td>Sinko, Erik</td>
<td>M</td>
<td>Freshman</td>
<td>91</td>
<td>A</td>
</tr>
<tr>
<td>1240224</td>
<td>Tarvin, Julio</td>
<td>M</td>
<td>Sophomore</td>
<td>85</td>
<td>B</td>
</tr>
<tr>
<td>1222781</td>
<td>Lessig, Elnora</td>
<td>F</td>
<td>Junior</td>
<td>63</td>
<td>D</td>
</tr>
<tr>
<td>1231806</td>
<td>Thon, Max</td>
<td>M</td>
<td>Sophomore</td>
<td>82</td>
<td>B</td>
</tr>
<tr>
<td>1246648</td>
<td>Trembley, Allyson</td>
<td>F</td>
<td>Junior</td>
<td>100</td>
<td>A</td>
</tr>
</tbody>
</table>
## Column Store – Sort and Encode for Speed

<table>
<thead>
<tr>
<th>Gender</th>
<th>Class</th>
<th>Grade</th>
<th>Score</th>
<th>Name</th>
<th>Student_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Sophomore</td>
<td>D</td>
<td>62</td>
<td>Cappiello, Emilia</td>
<td>1256678</td>
</tr>
<tr>
<td>F</td>
<td>Senior</td>
<td>A</td>
<td>92</td>
<td>Dalal, Alana</td>
<td>1254038</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>C</td>
<td>76</td>
<td>Orner, Katy</td>
<td>1278858</td>
</tr>
<tr>
<td>M</td>
<td>Senior</td>
<td>D</td>
<td>64</td>
<td>Frigo, Avis</td>
<td>1230807</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>A</td>
<td>90</td>
<td>Stober, Saundra</td>
<td>1210466</td>
</tr>
<tr>
<td>F</td>
<td>Freshman</td>
<td>A</td>
<td>96</td>
<td>Borba, Milagros</td>
<td>1249290</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>D</td>
<td>68</td>
<td>Sosnowski, Hillary</td>
<td>1244262</td>
</tr>
<tr>
<td>F</td>
<td>Sophomore</td>
<td>F</td>
<td>59</td>
<td>Nibert, Emilia</td>
<td>1252490</td>
</tr>
<tr>
<td>F</td>
<td>Freshman</td>
<td>A</td>
<td>95</td>
<td>Popovic, Tanisha</td>
<td>1267170</td>
</tr>
<tr>
<td>M</td>
<td>Senior</td>
<td>C</td>
<td>76</td>
<td>Schreckengost, Max</td>
<td>1248100</td>
</tr>
<tr>
<td>M</td>
<td>Junior</td>
<td>D</td>
<td>67</td>
<td>Porcelli, Darren</td>
<td>1243483</td>
</tr>
<tr>
<td>M</td>
<td>Freshman</td>
<td>A</td>
<td>91</td>
<td>Sinko, Erik</td>
<td>1230382</td>
</tr>
<tr>
<td>M</td>
<td>Sophomore</td>
<td>B</td>
<td>85</td>
<td>Tarvin, Julio</td>
<td>1240224</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>D</td>
<td>63</td>
<td>Lessig, Elnora</td>
<td>1222781</td>
</tr>
<tr>
<td>M</td>
<td>Sophomore</td>
<td>B</td>
<td>82</td>
<td>Thon, Max</td>
<td>1231806</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>A</td>
<td>100</td>
<td>Trembley, Allyson</td>
<td>1246648</td>
</tr>
</tbody>
</table>

**Columns used in predicates**

**Correlated values “indexed” by preceding column**
### Column Store – Sort and Encode for Speed

#### Table

<table>
<thead>
<tr>
<th>Gender</th>
<th>Class</th>
<th>Grade</th>
<th>Score</th>
<th>Name</th>
<th>Student_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Freshman</td>
<td>A</td>
<td>95</td>
<td>Popovic, Tanisha</td>
<td>1267170</td>
</tr>
<tr>
<td>F</td>
<td>Freshman</td>
<td>A</td>
<td>96</td>
<td>Borba, Milagros</td>
<td>1249290</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>A</td>
<td>90</td>
<td>Stober, Saundra</td>
<td>1210466</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>A</td>
<td>100</td>
<td>Trembley, Allyson</td>
<td>1246648</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>C</td>
<td>76</td>
<td>Orner, Katy</td>
<td>1278858</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>D</td>
<td>63</td>
<td>Lessig, Elnora</td>
<td>1222781</td>
</tr>
<tr>
<td>F</td>
<td>Junior</td>
<td>D</td>
<td>68</td>
<td>Sosnowski, Hillary</td>
<td>1244262</td>
</tr>
<tr>
<td>F</td>
<td>Senior</td>
<td>A</td>
<td>92</td>
<td>Dalal, Alana</td>
<td>1254038</td>
</tr>
<tr>
<td>F</td>
<td>Sophomore</td>
<td>D</td>
<td>62</td>
<td>Cappiello, Emilia</td>
<td>1256678</td>
</tr>
<tr>
<td>F</td>
<td>Sophomore</td>
<td>F</td>
<td>59</td>
<td>Nibert, Emilia</td>
<td>1252490</td>
</tr>
<tr>
<td>F</td>
<td>Sophomore</td>
<td>F</td>
<td>91</td>
<td>Sinko, Erik</td>
<td>1230382</td>
</tr>
<tr>
<td>M</td>
<td>Freshman</td>
<td>A</td>
<td>67</td>
<td>Porcelli, Darren</td>
<td>1243483</td>
</tr>
<tr>
<td>M</td>
<td>Junior</td>
<td>D</td>
<td>82</td>
<td>Thon, Max</td>
<td>1231806</td>
</tr>
<tr>
<td>M</td>
<td>Sophomore</td>
<td>B</td>
<td>85</td>
<td>Tarvin, Julio</td>
<td>1240224</td>
</tr>
<tr>
<td>M</td>
<td>Sophomore</td>
<td>B</td>
<td>76</td>
<td>Schreckengost, Max</td>
<td>1248100</td>
</tr>
<tr>
<td>M</td>
<td>Senior</td>
<td>C</td>
<td>64</td>
<td>Frigo, Avis</td>
<td>1230807</td>
</tr>
<tr>
<td>M</td>
<td>Senior</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Notes

- **Columns used in predicates**
- **Correlated values “indexed” by preceding column**

---

© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice.
Column Store – Sort and Encode for Speed

Example query: select avg( Score ) from example where Class = ‘Junior’ and Gender = ‘F’ and Grade = 'A'
Column Store – Column Based Encoding

Compression Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Compression Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clickstream</td>
<td>10:1</td>
</tr>
<tr>
<td>Audit</td>
<td>10:1</td>
</tr>
<tr>
<td>Trading</td>
<td>5:1</td>
</tr>
<tr>
<td>SNMP</td>
<td>20:1</td>
</tr>
<tr>
<td>Network Logs</td>
<td>60:1</td>
</tr>
<tr>
<td>Marketing</td>
<td>20:1</td>
</tr>
<tr>
<td>Consumer</td>
<td>30:1</td>
</tr>
<tr>
<td>CDR</td>
<td>8:1</td>
</tr>
</tbody>
</table>

0% 20% 40% 60% 80% 100%

Encoded Data  Raw Data
Real-Time Loading and Querying

Write-Optimized Store – WOS
Read-Optimized Store – ROS
Tuple Mover – TM

Batched Write

Write

Read

WOS

ROS

Batched Write
Shared-Nothing, Scale-Out Architecture

Massively Parallel Processing (MPP)

100% peer-to-peer
No specialized nodes
Can query & load to any node
Linear scalability
Revolutionary High Availability

RAID-like functionality within DB
Smart K-safety
Always-on loads & queries
Extensible + Ability to Share

A rich analytic platform with a large set of built-in analytics

Extensibility to develop custom algorithms

Share tools, views, code and data

Ability for users to run analytics defined by someone else via a standard tool
Vertica has a Rich Analytics Platform

**SQL**
- Window functions
- Graph
- Monte Carlo
- Statistical
- Geospatial

**Extended SQL**
- Sessionization
- Time series
- Pattern matching
- Event series joins

**SDKs**
- C++
- R
- More to come

Check out: http://www.vertica.com/2011/10/05/being-green-with-data-exhaust
Vertica Analytics Platform SDK

A framework for User-defined Extensions

Languages: C/C++ and R
Simple: concise APIs and examples accelerate deployment
Flexible: operate on Structured and Unstructured data sets, fenced Option for Security
Efficient: In-process, fully parallel

User Community: Github.com

Check out: https://github.com/vertica/Vertica-Extension-Packages
Analytic Extensions in R

What is R?

• Open source language for statistical computing
• Wide range of packages available for advanced data mining and statistical analysis

Advantages of running R from inside Vertica

• Vertica automatically parallelizes the execution of user defined R code
• Optimized data transfer between Vertica and R
• Enable ‘R’ users to benefit from Vertica’s scalable MPP platform
• Enable ‘SQL’ users to benefit from advanced analytics with R
R Analytics Use Cases

Data-Mining Algorithms
K-Means Clustering – Segment customers based on geography, usage patterns, etc
Page Rank – Identify the influencers among my customers / users
K-Nearest Neighbors Classification
Naïve Bayes Classification
Classification and Regression Trees

In-Database Scoring
Financial Services: What is the probability of default for each mortgage in our portfolio?
Sensor data: What is the probability of failure for each of my in-home devices?
Health care: What is the probability that this medical insurance claim is fraudulent?
Record and replay Analyst “thought-process”

Data Collector
Comprehensive tracking of what the database is doing
Automatic and low over-head collection
Includes query logging, performance profiling and so on
Easy SQL access to retrieve data back
Privilege model for sharing / protecting access to activity
Customizable Retention Policy
Sand-boxing data to enable ad-hoc and intense experimentation

Efficient snapshot objects, with COW semantics

**Export from one Vertica Cluster to another**

SQL command to transfer data subset to another cluster

Source and Target cluster can be different in size, config and physical design

Optimized protocol for data transfer, keeping data compressed when possible

Ability to export to a cluster on EC2
Ability to dynamically access a variety of data sources

**User-defined Loads & External Tables**

UDL is an extensible adapter API to load data from any source, in any format.

External Tables provides ability to make loads “dynamic” i.e. at query time.

Connectors available to data sources such as HDFS, Other Databases, etc.
Want a conversation with your data? Evaluate Vertica!

**Enterprise Edition**
- Free 30 day evaluation

**Community Edition**
- Free Download 1TB, 3 nodes

Check Out Vertica Extensions on Github!
- [https://github.com/vertica/Vertica-Extension-Packages](https://github.com/vertica/Vertica-Extension-Packages)

And especially for Baseball fans!