HPE Vertica & Hadoop
Descripción técnica
Eugenia Moreno
Solution Architect
About Me
The data growth
Expansion of the digital universe

IDC projects that the digital universe will reach 40 ZB by 2020, an amount that exceeds previous forecasts by 5 ZBs.

Machine-generated data is a key driver in the growth of the world’s data – which is projected to increase 15x by 2020 (representing 40% of the digital universe).

2005 2010 2012 2015
0.1ZB 1.2ZB 2.8ZB 8.5ZB

Every 60 seconds

- 98,000+ tweets
- 695,000 status updates
- 11 million instant messages
- 698,445 Google searches
- 168 million+ emails sent
- 1,820 TB of data created
- 217 new mobile web users
Data accessibility today

Typical compromises

- Infrastructure that becomes **unaffordable** at scale
- Analytics power that is accessible to only the **few**
- A **trade-off** between quality of insight & the speed of decisions
- Data is often past its effective **expiration date** to add value
… and data is often chained in siloes
What is needed is a Big Data Platform to harness 100% of the data.
SQL on Hadoop – Why?

Increased Value of Data
Hadoop the basic
Hadoop is a loose collection of related software components.
How does HDFS store data?
Each file is replicated three times

HDFS
Default chunk size: 64MB

File A
HDFS is a distributed filesystem built to never lose data

Files are split into chunks (def: 64MB)
Chunks are replicated (def: 3) across
Chunks are always put on different racks
If a node fails, re-silvering is top priority
Data can be read from any server that has it
How does HDFS store large files?

Store the first copy

HDFS

Default chunk size: 64MB

File A
How does HDFS store large files?

Replicate to other DN

Chunks are always put on different racks

HDFS
Default chunk size: 64MB

File A
How does HDFS store large files?

Replicate to other DNs, default to 3 replicas

If a node fails, re-replicating is top priority
How does HDFS store large files?
Large files are broken into chunks, and each chunk is replicated independently.

File A:
180MB

- 64MB
- 64MB
- 52MB
The NameNode is the lookup service for DN chunks

HDFS

Default chunk size: 64MB

File A?

Go to DN3
MapReduce is the distributed processing system for Hadoop

It’s like Vertica’s Execution Engine

We don’t talk to MapReduce much
Hive is a SQL-ish database build on MapReduce.

Hive

HCatalog

Table1 (a int, b int)
/data/table1/file1.tbl
/data/table1/file2.tbl
/data/table1/file3.tbl

Table2 (a int, b int)
/data/table2/file1.tbl
/data/table2/file1.tbl

HDFS

HN

DN
DN
DN
DN
DN
Many applications just use HCatalog, not Hive

Table1 (a int, b int)
/data/table1/file1.tbl
/data/table1/file2.tbl
/data/table1/file3.tbl

Table2 (a int, b int)
/data/table2/file1.tbl
/data/table2/file2.tbl

Clickstream  Billing  Telemetry
Hadoop is primarily developed by three companies.

Contributions to Apache Hadoop Core, 2011

- **Hortonworks**
- **Yahoo!**
- **Cloudera**
- **IBM**
- **InMobi**
- **LinkedIn**
- **Huawei**
- **Facebook**
- **Other**

**Lines of Code Contributed (%)**

**Patches Contributed (%)**
Hadoop Hype and Future Opportunities
“The fervor for Hadoop disguises its modest adoption and growth rates to date”

“Organizations interested in Hadoop are overwhelmingly at a knowledge-gathering stage, with smaller populations piloting Hadoop for specific analytical workloads or using it in production.”

- Gartner Market Guide for Hadoop Distributions (Jan 2015)

Bottom Line:
- Gartner estimates total market for Hadoop is about 1,400 paying companies (Jan 2015)
- Most have between 10 and 20 nodes
- However, Hadoop appliance adoption has begun and may have a substantial impact in 2015, according to Gartner.
Hadoop-based Analytics has Compromises

Typical Compromises:

- No advanced analytics, Hadoop solutions offer a subset of analytical functions.
- Emerging technologies are sometimes buggy and difficult to implement.
- Optimized file formats, compression, and other optimizations in HP Vertica are not in Hadoop solutions, leading to lower performance.
- Access to human information like voice and video is limited in Hadoop, but available in powerful IDOL engine.
## Technical Requirements Impact Selection

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Vertica</th>
<th>Hadoop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low latency interactive analytics</td>
<td>✓</td>
<td>□</td>
</tr>
<tr>
<td>ANSI SQL Compliance</td>
<td>✓</td>
<td>□</td>
</tr>
<tr>
<td>ACID compliance</td>
<td>✓</td>
<td>□</td>
</tr>
<tr>
<td>Preprocessing of unstructured data</td>
<td>w/IDOL</td>
<td>✓</td>
</tr>
<tr>
<td>High Quality Structured Data</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Polyglot Persistence (many languages performing analytics on many data stores/formats)</td>
<td>✓</td>
<td>□</td>
</tr>
<tr>
<td>Ungoverned Sandbox</td>
<td>□</td>
<td>✓</td>
</tr>
<tr>
<td>Data Lake and Discovery</td>
<td>□</td>
<td>✓</td>
</tr>
<tr>
<td>Extensive Security and Compliance</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Online Transactional Processing (OLTP)</td>
<td>□</td>
<td>✓</td>
</tr>
</tbody>
</table>
The Hadoop Experiment
How companies are sandboxing big data analytics with Hadoop

Data is Dumped into Hadoop
Low Cost Storage, Unclear Analytical Objectives, Hadoop is Cool

Discovery Analytics
Can’t run all their queries, Buggy, Not Ecosystem Friendly, Not ACID compliant

Seek like-priced alternatives to Meet SLAs
HP Vertica for SQL on Hadoop, Vertica with Flex Zone

Big Data technology gap
Building a Smarter Data Lake with HP Vertica

- SQL on Hadoop functionality offers the only full-featured query engine on Hadoop
  - Same Core Engine
  - Hadoop Distribution Agnostic
  - Enterprise-ready Solution
  - World-class Enterprise Support and Services
  - Open platform
  - Ready for Haven
- Economics that work
One Query Engine to Serve it all
Store Data in HP Vertica or any Hadoop Distribution

- **Query data in place in Hadoop Formats**
- **Co-Locate and leverage existing Hadoop infrastructure**
- **HP Vertica performance on lower-cost infrastructure**
- **Single query engine across diverse formats and infrastructure**

<table>
<thead>
<tr>
<th>Query Engine</th>
<th>HP Vertica ANSI SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Vertica Optimized (ROS, Flex Tables)</td>
</tr>
<tr>
<td>File System</td>
<td>Vertica (EXT4)</td>
</tr>
</tbody>
</table>
Vertica SQL on Hadoop
HP Vertica for SQL on Hadoop

Query Data, no matter where it is

- Install HP Vertica directly on your Hadoop infrastructure
- ORC, Parquet, Avro, Vertica ROS and JSON supported
- Full-functionality ANSI SQL
- 100% of TPC-DS queries
- No helper node or single point of failure
- Competitive price point
HP Vertica for SQL on Hadoop (Architecture)

• Same Vertica MPP Columnar Architecture
• Base ANSI SQL
• Co-Located with Hadoop
• Data Query Across parquet, ORC, JSON, and many other format
• Hadoop Agnostic
VSQLH makes Vertica the power tool for your Hadoop data
HP VERTICA SQL on Hadoop
How do we Access?

- HCatalog Connector
- HDFS Connector
- ORC & Parquet Reader
- Storage Location Access
Flat files in HDFS can be accessed via the HDFS Connector

CREATE EXTERNAL TABLE shipping_dimension
    (shipping_key integer,
     ship_type char(30),
     ship_mode char(10),
     ship_carrier char(20))
AS COPY FROM SOURCE
Hdfs(url='http://n01:50070/webhdfs/v1/posts/*');
HDFS Files as External Table in Vertica

```sql
create external table shipping_dimension_hdfs_ext
    (shipping_key integer,
     ship_type char(30),
     ship_mode char(10),
     ship_carrier char(20))
    CREATE TABLE
    select shipping_key from shipping_dimension_hdfs_ext limit 2;
    shipping_key
    -----------
    1
    2
    (2 rows)
```

HDFS(url='http://172.16.27.6:50070/webhdfs/v1/user/dbadmin/shippingDimension/*1):
HDFS Files bulk loaded into Vertica

```
[root@pocg8-26-node1 kerberos_Examples] $ hdfs dfs -ls /user/dbadmin/shippingDimension/
Found 1 items
-rw-r--r-- 3 dbadmin verticadba 270B 2016-01-21 16:36 /user/dbadmin/shippingDimension/Shipping_Dimension.tbl

[root@pocg8-26-node1 kerberos_Examples] $
```

```
[dbadmin@pocg8-26-node1 kerberos_Examples]$ vsql -aef vertica_hdfs_real.sql
drop table shipping_dimension_hdfs_local;
vsq1:vertica_hdfs_real.sql:4: ROLLBACK 4876: Table "shipping_dimension_hdfs_local" does not exist
create table shipping_dimension_hdfs_local 
( 
    shipping_key integer,
    ship_type char(30),
    ship_mode char(10),
    ship_carrier char(20)
);
CREATE TABLE
COPY shipping_dimension_hdfs_local
source HDFS(url='http://172.16.27.6:50070/webhdfs/v1/user/dbadmin/shippingDimension/*')
Rows Loaded
-------------
100
(1 row)

select shipping_key from shipping_dimension_hdfs_local limit 2;
shipping_key
-------------
5
8
(2 rows)

[dbadmin@pocg8-26-node1 kerberos_Examples]$ 
```
Structured data in HCatalog can be accessed via the HCatalog Connector

CREATE HCATALOG SCHEMA hive WITH
    HOSTNAME='hcat.vertica.com'
    HCATALOG_DB='hive'
    HCATALOG_USER='hive_user';
Structured data in HCatalog can be accessed via the HCatalog Connector

```sql
SELECT
  distinct ship_type,
  ship_mode,
  ship_carrier
FROM hcat.shipping_dimension
WHERE shipping_key >= 10627
GROUP BY ship_mode,
  ship_carrier;
```
Access Hive: Show Tables

hive -e "show tables;"

Logging initialized using configuration in file:/etc/hive/conf/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.7.0.3010/hadoop/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Slf4jLoggerFactory]
SLF4J: Found binding in [jar:file:/usr/hdp/2.7.0.3010/hadoop/lib/hive-jdbc-0.14.0.2.7.0.3010-standalone.jar!/org/slf4j/impl/Slf4jLoggerFactory]
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
OK

shipping_dimension
shipping_dimensionorc

Time taken: 1.735 seconds, Fetched: 2 row(s)

Access Hive: Describe Tables

```sql
-- Access Hive: Describe Tables

-- Hive - describe extended shipping_dimension:

10/03/22 12:48:27 WARN hive.conf.HiveConf: HiveConf of name hive.optimize.mapjoin.reduce does not exist
10/03/22 12:48:27 WARN hive.conf.HiveConf: HiveConf of name hive.heapsize does not exist
10/03/22 12:48:27 WARN hive.conf.HiveConf: HiveConf of name hive.server2.enable impersonation does not exist

Logging initialized using configuration in file:/etc/hive/conf/hive-log4j.properties

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/usr/hdp/2.8.0-3159/hadoop/lib/SLF4J-Log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]


SLF4J: Actual binding is of type [org.slf4j.simpleLoggerFactory]

Detailed Table Information


```

Hewlett Packard Enterprise
Access Hive/HCatalog with HCatalog Connector

```sql
[dbadmin@pocg8-26-node1 kerberos_Examples]$ vsql -aef vertica_hcatalog.sql
drop schema hcat;

DROP SCHEMA
create hcatalog schema hcat with hostname='172.16.27.7' hcatalog_schema='default';

CREATE SCHEMA
select shipping_key from hcat.shipping_dimension limit 2;

<table>
<thead>
<tr>
<th>shipping_key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

(2 rows)

[dbadmin@pocg8-26-node1 kerberos_Examples]$  
```
Benefit for ORC similarly to Vertica’s native format

**ORC format**

- Tables stored in column-major blocks of rows (stripes)
- Metadata stored in file footer, stripe footer and index
- Full data type support
  - primitive SQL types
  - complex Hive types
- Encoding (RLE, delta, dictionary)
- Compression (zlib, snappy)
Read ORC conveniently and fast

Built-in ORC reader

• No UDL setup needed – ORC is natively supported by Vertica

• Create and query ORC tables
  - Stream as external tables
    CREATE EXTERNAL TABLE t(...) AS COPY FROM "..." ORC;

• Load into Vertica tables
  CREATE TABLE t(...) ;
  COPY t FROM "..." ORC;
ORC Files as External Table in Vertica

```
[dbadmin@pocg8-26-node1 kerberos_Examples]$ vsql -aef vertica_orc_ext.sql
DROP TABLE shipping_dimension_orc_ext;

CREATE EXTERNAL TABLE shipping_dimension_orc_ext
(
    shipping_key integer,
    ship_type char(30),
    ship_mode char(10),
    ship_carrier char(20)
) AS COPY FROM 'webhdfs://172.16.27.6:50070/user/dbadmin/shippingDimension_orc/000000_0' ON ANY NODE ORC;

SELECT shipping_key FROM shipping_dimension_orc_ext LIMIT 2;
```

```
<table>
<thead>
<tr>
<th>shipping_key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
(2 rows)
```

[dbadmin@pocg8-26-node1 kerberos_Examples]$
ORC Files Bulk Loaded into Vertica

```sql
[dbadmin@pcg8-26-node1 kerberos_Examples]$ vsq1 -aef vertica_orc_local.sql
drop table shipping_dimension_orc_local;
vsq1:vertica_orc_local.sql:1: ROLLBACK: 4876: Table "shipping_dimension_orc_local" does not exist
create table shipping_dimension_orc_local
()
);
CREATE TABLE
copy shipping_dimension_orc_local from 'webhdfs://172.16.27.6:50070/user/dbadmin/shippingDimension_orc/000000_0' ON ANY NODE ORC;

Rows Loaded
---------
100

select shipping_key from shipping_dimension_orc_local limit 2;
shipping_key
---------
6
12
```
Vertica Proprietary Format in HDFS

CREATE LOCATION 'webhdfs://server/user/asmith/data'
  ALL NODES
  SHARED
  USAGE 'data'
  LABEL 'hdfs';

SELECT set_object_storage_policy('mydb', 'hdfs');
Vertica ROS Files

```sql
[dbadmin@pocgt-16 node1 terbers_examples]$ hdfs dfs -ls /user/dbadmin/vertica
Found 4 items
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:23 /user/dbadmin/vertica/v_br_node0001
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:23 /user/dbadmin/vertica/v_br_node0002
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:23 /user/dbadmin/vertica/v_br_node0003
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:23 /user/dbadmin/vertica/v_br_node0004
[dbadmin@pocgt-16 node1 terbers_examples]$ hdfs dfs -ls /user/dbadmin/vertica/v_br_node0001
Found 2 items
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:22 /user/dbadmin/vertica/v_br_node0001/099
-dwrxr-x--- dbadmin verticadb 0 2016-01-21 23:22 /user/dbadmin/vertica/v_br_node0001/119
[dbadmin@pocgt-16 node1 terbers_examples]$ create location 'webhdfs://172.16.27.6:50070/user/dbadmin/vertica' all nodes shared usage 'data' label 'hdfs';
for supported HDFS versions.
CREATE LOCATION
drop table shipping_dimension;
vsq1:vertica_ros_hdfs.sql:3: ROLLBACK 4870: Table "shipping_dimension" does not exist.
create table shipping_dimension
           (shipping_key integer,
           ship_type char(10),
           ship_mode char(10),
           ship_carrier char(20));

CREATE TABLE
select set object_storage_policy('shipping_dimension','hdfs');
reset.object_storage_policy;

------------------------
Object storage policy set.
(1 row)

copy shipping_dimension from '/opt/vertica/examples/VMart_Schema/Shipping_Dimension.tbl' direct;
  Rows Loaded
------------------------
100
(1 row)

select shipping_key from shipping_dimension limit 2;
  shipping_key
  ------------------------
     1
     9
(2 rows)
```

---

Hewlett Packard Enterprise
## SQL on Hadoop – Overall connectors comparison

<table>
<thead>
<tr>
<th>Connector</th>
<th>Integration</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORCFile Reader</td>
<td>Allows you to directly query ORC files and take advantage of their optimizations.</td>
<td>High</td>
</tr>
<tr>
<td>ParquetFile Reader (coming soon, not discussed in slides)</td>
<td>Allows you to directly query Parquet files and take advantage of their optimizations.</td>
<td>High</td>
</tr>
<tr>
<td>HDFS Connector</td>
<td>Allows users to load structured data from the Hadoop Distributed Filesystem (HDFS) and create an external table based on structured data stored in HDFS.</td>
<td>High</td>
</tr>
<tr>
<td>ROS Containers on HDFS</td>
<td>Allows users to store Vertica formatted data on HDFS.</td>
<td>High</td>
</tr>
<tr>
<td>Hcatalog Connector</td>
<td>Allows users to query data stored in Hive using the HP Vertica native SQL syntax.</td>
<td>Medium/Low</td>
</tr>
<tr>
<td>MapReduce Connector</td>
<td>Allows users to create a MapReduce job or Pig Script that can read data in Hadoop.</td>
<td>Medium/Low</td>
</tr>
</tbody>
</table>
Vertica can access Kerberized Hadoop clusters

• Support for Kerberos
  – Session- and process-wide ticket reuse
  – Multiple concurrent queries
  – Transparent on a Kerberized Vertica cluster
TPC-DS Adhoc Queries

- Competition only runs 14 of 31 TPC-DS Queries
- Off the 14, VSQLH is Comparable or Better in 9
• Competition only runs 21 of 46 TPC-DS Queries
• Off the 21, VSQLH is Comparable or Better in 16
• Competition only runs 14 of 22 TPC-DS Queries
• Off the 14, VSQLH is Comparable or Better in 10
~ ORC Equivalent/Better
~ ROS 5xs Faster
Contribute to VSQLH

Open-source VSQLH components

• ORC C++ API
  – Developed in collaboration with Hortonworks
  – Became a top-level Apache project in June 2015
  – Check out orc.apache.org

• HDFS C++ API
  – Developed in collaboration with Hortonworks
  – Integrated into Apache Hadoop in July 2015

• APIs for other file formats and filesystems can be integrated into Vertica
Key Differentiators
No Limits Data Exploration

• **Richest and Proven SQL Query Engine**
Dramatically reduce complexity of your data architecture with a time-tested, unified query engine

• **Enterprise-Ready, Stable, Trusted**
Improve productivity with a robust, highly optimized, enterprise-ready SQL engine and first class support

• **Simplicity and Fast Insights in Data Exploration**
Leverage our Database Designer to simplify the management of your queries

• **Easy Path to Vertica Enterprise**
Perform data discovery in Hadoop and boost analytics performance with Vertica EE

• **Rich Ecosystem**
3rd Party Integrations

HADOOP

ANSI SQL ENGINE

VERTICA

HADOOP

ANSI SQL ENGINE

VERTICA

HADOOP

ANSI SQL ENGINE

VERTICA

HADOOP

ANSI SQL ENGINE

Thank You