



Creating insightful data analytics

Advanced analytics for better business decisions with HP Vertica Analytics Platform

The case for user-defined extensions

The HP Vertica Analytics Platform supports many user-defined capabilities: functions, transforms, aggregates, and loading. This support brings the power and flexibility of procedural code to your data—whether it's structured, semi-structured, or unstructured—leveraging a parallel computing environment. You can load and transform data from a variety of sources, perform rich statistical analytics, and visualize the results using industry-standard tools. With HP Vertica Platform, you can expect real-time results for better business decisions.

A closer look at user-defined extensions

SQL is a declarative language, requiring you to state the required outcome. The database then develops an optimal procedure and computes the answer. The Optimizer in the HP Vertica Analytics Platform does both. It understands how data is distributed on a cluster as well as the most cost-effective join order, and it generates efficient query plans (the steps in a procedure). However, you may find it difficult to express some operations in SQL, especially with unstructured or semi-structured inputs. User-defined extensions are a good alternative.

Maintaining and reusing code

With SQL, you need a schema to organize data. Procedural code, however, lets you manipulate your data arbitrarily. Each method has advantages. User-defined extensions have clearly defined inputs and outputs, which the HP Vertica Analytics Platform presents in searchable tables. It automates the physical design and manages arbitrarily large input and output streams. The Optimizer schedules procedures into the most cost-effective phase of a query plan without compromising accurate results. User-defined extensions scale to fit the size of your data. You can maintain and reuse code without compromising scale.

An example of log parsing

The HP Vertica Analytics Platform includes a software developers' kit (SDK) with an Apache web log parser. The log parser enables you to combine procedural code with SQL and execute data in parallel. HP Vertica Analytics Platform distributes procedures to each node in a cluster, managing version control and updating procedures if a node goes down.

Registering a procedure is simple:

```
CREATE LIBRARY TransformFunctions AS <path_to_file>;  
CREATE TRANSFORM FUNCTION ApacheParser  
NAME 'ApacheParserFactory'  
LIBRARY TransformFunctions;
```

HP Vertica Analytics Platform automates the physical design. If you know procedural languages, you do not need database administrator (DBA) skills.

Loading semi-structured data into HP Vertica Analytics Platform is also easy:

```
CREATE TABLE raw_logs (data VARCHAR(64000));  
COPY raw_logs FROM <path_to_file>;
```

Now that you have loaded your parser library and have the raw log data, you can run arbitrary queries as if the log data were structured. Registering your library lets HP Vertica Analytics Platform know the inputs and outputs of your user-defined extensions. You can either use the Apache Parser to create a table as select (CTAS), or directly query the raw logs as if they contained structured data. In either case, your user-defined transform provides column names:

```
SELECT COUNT( user_agent ), user_agent  
FROM (SELECT ApacheParser(data)  
OVER (PARTITION BY 1) FROM raw_logs) AS x  
WHERE response_code = '304'  
GROUP BY 2 ORDER BY 1 DESC LIMIT 5;
```

Count	User agent
2752	Mozilla/5.0 (Macintosh; Intel® Mac OS X 10.7; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
1898	Mozilla/5.0 (Windows® NT 6.1; WOW64; rv:8.0) Gecko/20100101 Firefox/8.0
803	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
584	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
493	Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0

The browsers are ranked according to a specific response code, but you can query on other relationships and use this type of function as input for sessionization. You can assign a session ID to each person visiting your website and use pattern matching and event series joins to identify and rank unique click paths.

Window functions and parallelism

The log parsing example uses a PARTITION BY clause. HP Vertica Analytics Platform distinguishes between user-defined functions—which take one row of input for each row of output they produce—and user-defined transforms—which take an arbitrary number of rows for input and produce an arbitrary number of output rows. Both extension types need to be set up independently from iteration, so that they are not called multiple times to process multiple lines. The PARTITION BY clause is specific to user-defined transforms and tells which rows belong together within procedure instantiation.

For example, a simple user-defined transform may compute the weighted average for a stock exchange. By partitioning by “symbol” in a stock ticker table, each instantiation of the weighted average function operates against the price for each ticker symbol. The Optimizer knows that the table contains multiple symbols, so HP Vertica Analytics Platform can scale the throughput by running multiple threads.

The Apache Parser is a user-defined transform, but it processes each line independently. You can partition the collection of input rows arbitrarily and still get correct output rows. You can safely create an arbitrary number of partitions, and HP Vertica Analytics Platform can run multiple threads of your user-defined transform in parallel:

```
SELECT COUNT( user_agent ), user_agent  
FROM (SELECT ApacheParser(data)  
OVER (PARTITION BY HASH(SUBSTR(data,1,20)))  
FROM raw_logs) AS x  
WHERE response_code = '304'  
GROUP BY 2 ORDER BY 1 DESC LIMIT 5;
```

This partitioning tells HP Vertica Analytics Platform to look at the first 20 characters of each raw log line and use a hash of those characters to group log lines into a partition. Each thread of the Apache Parser uses one or more partitions of the raw logs. The result is the same. In this case, HP Vertica Analytics Platform can use more CPU cores across the cluster.

When reviewing the query, the Database Designer within HP Vertica Analytics Platform recommends, and can automatically implement, the following change to the raw_logs table:

```
CREATE TABLE raw_logs (data VARCHAR(64000))  
SEGMENTED BY HASH(SUBSTR(data,1,20)) ALL NODES;
```

The optimization applies the same distribution to the data that the Database Designer recommends. As a result:

- Lines of raw log data optimally distribute across the cluster nodes as they load into the database.
- The Optimizer schedules Apache Parser threads on each cluster node, feeding them with log lines local to the node. This reduces the network traffic between nodes, thus reducing the cost of the query.

The user-defined extension framework combines intuitive procedural code with easy maintenance and operational efficiency. You can directly address structured, semi-structured, and unstructured data. Just like SQL operations, user-defined extensions work closely with the database-designed, column-aware Optimizer to scale both procedural and declarative executions seamlessly for any data size or analytic workload. You can run extensions in a separate, isolated process so that your user-defined code does not affect the resiliency or security of your deployment.

User-defined functions in R

Millions of users around the world use the open-source R programming language for statistical computing. Thousands of pre-built analytics packages are based on R. However, R is currently a standalone application, lacks parallelism, and requires data to be in memory.

HP Vertica Analytics Platform supports R for data analysis and addresses many of its limitations. In addition to leveraging the thousands of contributed packages from CRAN (Comprehensive R Archive Network), you can create your own custom R-language functions. With HP Vertica Analytics Platform, R functions are automatically distributed and executed in parallel across your selected window of data, so you get real-time processing and analysis for your data. In-database R execution avoids data extraction and down sampling for even higher performance.

The columnar storage of HP Vertica Analytics Platform is a natural fit for quickly performing R computations outside of the HP Vertica Analytics Platform environment as well as within, allowing you to reap all of the benefits of using R on a high-performant, cost-effective platform, and more.

For more information

Test drive the HP Vertica Analytics Platform at vertica.com/evaluate.

Get connected

hp.com/go/getconnected

Get the insider view on tech trends, support alerts, and HP solutions.



Share with colleagues

© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

4AA1-2497ENW, Created September 2012

