VER 1
2017

Dimensional INSIGHT

Spectre™: First Look

DIVER PLATFORM 7.0 DATA PROCESSING ENGINE

JAMES CLARK & JEAN COLLINS

Spectre: First Look

Table of Contents

Introduction

Version 7.0 of Dimensional Insight's Diver Platform supports the powerful Spectre engine. The Dimensional Insight Spectre engine combined with cBase, the latest columnar database technology, delivers robust business intelligence for information analysts, managers, and consumers.

Spectre was designed from the ground up to optimize performance and perfect ease-of-use features for all users. Spectre makes the latest version of Dimensional

Insight's flagship product fast, scalable, and manageable to meet the needs of today's enterprise customers.

Overview of Spectre

Spectre and cBase, Dimensional Insight's columnar database technology, give the greatest increase in speed and efficiency to features that are most used by power analysts. The most important feature is the new column-oriented, shareable database storage format optimized for query-time calculations. The design takes advantage of hardware innovations and analysis practices to better handle user behaviors and queries.

Enterprise-level businesses are seeing rapid changes in their business environments, such as the increase in the quantity and diversity of data in varied formats from different platforms and the increased need to combine views. Spectre takes advantage of the latest hardware advances, such as faster core speeds and multiple cores for built-in parallel processing, large amounts of memory, solid state disk (SSD), and advanced compiler technology, to radically boost performance in these environments.

Columnar database design

Spectre uses an in-memory, binary-format columnar database to optimize the most commonly used functions. So, just what is columnar database technology and why is it faster? Spectre's speed comes from perfectly matching the design with function. Mining actionable information from diverse sources requires consolidation of large amounts of data and the ability to quickly access the data in the aggregate as well as the underlying details.

Here's where the design is critical. Typically, a relational database stores fields in a record together, like rows in a table. The records have multiple fields that are potentially of different data types, such as string, integer, fixed, or double. Each record might use a block of memory, or there might be multiple records in a block. Regardless, all of the fields for a record are stored consecutively. This is a great design when you want to retrieve all the fields of a record every time the record is accessed. However, business intelligence queries typically need to access only one or a few fields of each record. For these queries, the row-oriented design is not very efficient.

Contrast the columnar database design with the row-oriented design. In the columnar database design, instead of storing all of the fields for each record together, the records are broken up. The "like" fields for all records, or each column of a table, are stored together in blocks of memory. Now, when you want to perform calculations over the data, such as a SUM, MAX, MIN, COUNT, or AVG, only the relevant columns need to be accessed, thereby making calculations very fast.

Let's use a simplified sales example to show the differences. Suppose you have a table as shown below with quarterly sales figures for four products.

Table 1: Sample quarterly product sales figures

Product	Q1	Q2	Q3	Q4
Product A	5000.00	5200.00	4700.00	7000.00
Product B	1400.00	1100.00	900.00	1200.00
Product C	10000.00	10000.00	10000.00	11500.00
Product D	300.00	1500.00	600.00	2200.00

In a row-oriented database, the records are stored like this:

Table 2: Row-oriented database

Product A,5000.00,5200.00,4700.00,7000.00;
Product B,1400.00,1100.00,900.00,1200.00;
Product C,10000.00,10000.00,10000.00,11500.00;
Product D,300.00,1500.00,600.00,2200.00;

To answer typical business intelligence queries, such as "What were the total Q3 sales?" or "What product had the highest sales figure in Q4?" you need to access all rows in order to get the relevant quarterly sales figures. Alternatively, row-oriented databases can build and maintain multiple indexes to answer this type of query, but the indexes add the overhead of additional disk space and memory.

In contrast, in a columnar database, the records are stored by grouping all values for a given field together. For example, all of the product names are together, then all of Q1 sales, then all of Q2 sales, and so forth:

Table 3: Column-oriented database

Product A, Product B, Product C, Product D;
5000.00,1400.00,10000.00,300.00;
5200.00,1100.00,10000.00,1500.00;
4700.00,900.00,10000.00,600.00;
7000.00,1200.00,11500.00,2200.00;

The beauty of this design is how quickly you can answer the business intelligence queries posed earlier. With the self-indexing columnar design, you don't need to access all records for all products to get the quarterly figures. You don't even need to access the sales figures for quarters that you're not interested in. With the columnar database, you only access data elements that are relevant to the query. With databases containing millions of records, you can see how quickly this time savings adds up.

For example, Spectre was tested with actual customer data to determine the time to open a nine-dimension MultiTab, which is a tabular display option that presents multiple dimensions vertically. The input data consisted of 46 million rows, 215 columns, and 102 GB of text. With this large volume of input, the Spectre build time was under 30 minutes, the resultant database size just 20 GB, and the time to open a nine-dimension MultiTab only three seconds.

Terminology

The following terms are commonly used when discussing Diver Platform features. You can refer to this list to familiarize yourself with the definitions.

Build
: The process of building a cBase using the Spectre engine.

Build file
: The configuration file type (*.build) that controls the process of building a cBase using the Spectre engine.

Builder
: The Diver Platform component that summarizes and preprocesses data to create classic models.

cBase
: The file type (*.cbase) that stores Diver Platform data generated by the Spectre engine. A cBase is a columnar data structure optimized to aggregate (summarize) large volumes of data.

Dimension

A collection of related data within a model or cBase used for sorting and filtering data (diving).

DiveLine

The server component of the Diver Platform. DiveLine authenticates users and controls access to data through Diver Platform clients such as ProDiver, DivePort, and DiveTab.

DivePort

The web portal environment of the Diver Platform.

DiveTab

An information delivery platform designed to provide mobile users access to unstructured data, such as presentations and documents, and structured data, such as reports and dashboards.

Diver Platform

The Dimensional Insight software suite that contains all Diver 7.0 software, including Workbench and Spectre. DiveTab is optional. User categories are: Developer, ProDiver, DivePort, and DiveTab.

Dive or diving

The process of accessing smaller segments of data in greater detail using a Diver Platform client.

ETL

The acronym for extract, transform, and load. The process in which data is extracted from one or more sources, transformed into a format suitable for analysis, and loaded to a cBase or model.

Model

The file type (*.mdl) that stores data generated by the classic Builder engine. A model is a presummarized, highly indexed dataset optimized for multidirectional drill-down without directional limits or predetermined drill paths.

MultiTab

A tabular display that vertically combines two or more dimensions.

ProDiver

The desktop analytics client of the Diver Platform. ProDiver is the client in a client/server architecture. Therefore, it requires a connection to a DiveLine server to access data.

Spectre

The data analysis software that is used to build and query cBases. It powers the DiveLine server software for efficient queries from Dimensional Insight clients against those cBases.

Spectre dive file

A Diver Platform file type (*.dive) that specifies a window into Spectre cBase data by enumerating columns and values that appear in a tabular display. Spectre dive files are used in DivePort.

Workbench

An integrated visual development environment to develop, test, and manage projects associated with a Diver Platform application.

Speed

Spectre is built for speed, both for builds and for calculations, significantly boosting productivity of IT staff and response time for clients.

Build times

When your data input amounts to 500 million – 1 billion rows or more, you need builds to turn around quickly. Spectre builds don't pre-summarize data during the build time and instead use query-time calculations. Therefore, Spectre build windows are quite small relative to the data input, which all adds up to getting current data available to users quicker and more frequently.

Run-time performance

Spectre and the columnar database design make run-time performance extremely fast. This is because the Spectre engine algorithms optimize run-time performance for some of the most commonly used computations. In particular, Spectre is optimized for these specific functions:

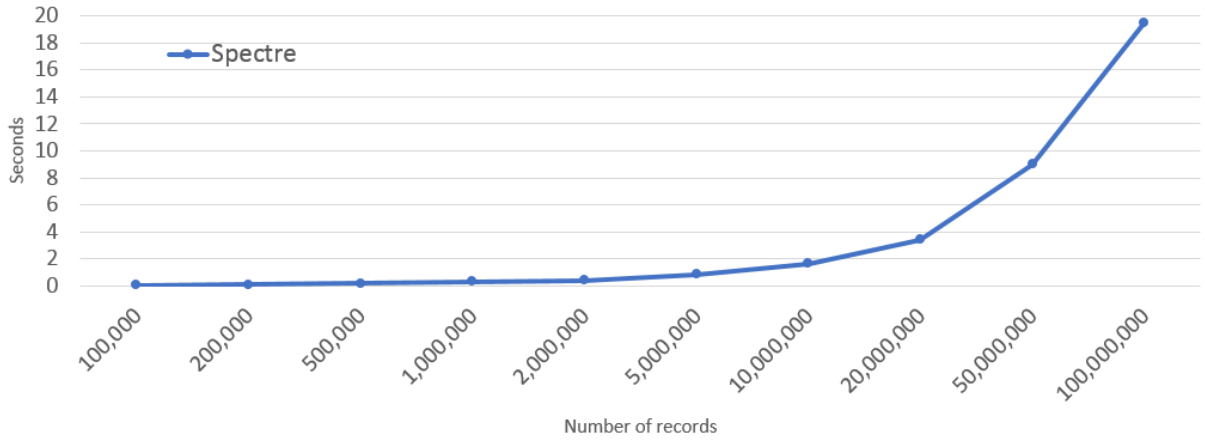
- Dimension Counts (DimCounts): Calculation of the number of unique values for a selected dimension.
- Groups: Functionality to combine multiple diveable entities into a single unit for diving.
- MultiTab: A tabular display option that presents multiple dimensions vertically.
- Multimodel: A combination of several models that contain the same or similar sets of dimensions, summaries, and info fields.

- Time Series: Functionality to present columns that are limited by time-based periods.

In Spectre, there are no limits for file size, column count, and number of dimensions. The overall file size is small relative to the data input. The Spectre calculation engine is based on the LLVM compiler technology, which compiles formulas into machine code that is optimized for the processor on which it is running. The machine code runs raw as opposed to being run through an interpreter. All these design optimizations combined with the columnar database shave off computation processing time and deliver fast run-time performance.

Figure 1 shows the response time of Spectre when diving into a 100 million row data set with a three-dimension MultiTab. Spectre takes a mere 20 seconds to deliver the results.

Figure 1: 3-Dimensional MultiTab response time over number of records



Scalability

The Spectre design is robust enough for challenging enterprise-level business intelligence analysis and delivers fast performance without taxing resources.

Database size

Spectre does not maintain separate database indexes, so the on-disk size of the columnar cBase format is small relative to the data input. Spectre can handle large data volumes in a single cBase, which minimizes maintenance tasks.

Cache

Cached dives deliver fast response while avoiding stale results. As an in-memory data engine, Spectre answers most queries without needing to access the hard drive. The results are cached for reuse the next time you make the same dive. Spectre can share the cache among multiple users with the same access when they do the same dive. People with different access levels use a different cache, so as not to compromise security.

When you rebuild a Spectre cBase, you can refresh the cache entries of previous dives. With this option, Spectre refreshes the cache of actual dives so that when you dive into previous dives, Spectre returns refreshed, not stale, results.

Memory

When Spectre loads parts of a cBase into memory, it does so in a way that is shared across multiple Spectre processes. If two different dives are running at the same time, the dives do not use twice as much memory for the database. Frequently, the database required for a dive is already in memory, so Spectre doesn't need to load it from disk. Users with different access share this memory, and Spectre processes ensure that each user gets the right results based on his or her access.

Low per-user overhead

Spectre delivers fast user performance with a low idle-connection cost per user. This supports more simultaneous users without a linear increase in memory and processor usage. Spectre operations are optimized for run time, so that user connections are closed when the operation completes, and resources are not devoted to idle user sessions. Resources are devoted to active users and concurrent dives.

Manageability

Users need rapid information access and IT needs to make sure they can manage and support user requirements. The Diver Platform with Spectre does both with Workbench for developers and DiveTab to keep your mobile workforce connected on the go.

Workbench

Workbench, an integrated development environment (IDE), helps developers manage the entire back-end process, from data source to portal. With Workbench, developers can create and manage integrated, streamlined ETL processing from within a single application. The Workbench IDE combines a host of tools in a visual environment for efficient, centralized project management. Workbench integrates the following tools in a modular design:

- Spectre: Define, build, and access cBases.
- DiveTab: Build data presentation portals for mobile users.
- DiveLine: Configure and control access at the server level in cBases.
- Production: Configure and automate scheduled tasks.
- Visual Integrator: Define steps to extract, transform, and load raw data into cBases.
- Visual Builder: Define and build cBases.
- DiveMaster: Organize cBase data by categorizing columns.
- DIAL: Prepare and schedule email delivery of reports for targeted users.

Spectre configuration and scripts use a single text-based scripting language, which developers access and edit with the robust Workbench editor. The scripting language is simple and powerful for builds and dives. Workbench speeds development with highlights for important parts of the script, code suggestions, and descriptive help for syntax errors. Developers will appreciate how straightforward certain tasks are, such as setting up a new build script or viewing a copy of the build script and log within every cBase.

DiveTab Client

Powered by Spectre, the DiveTab client is a tablet-based mobile technology for self-service reporting and analysis that drives data-driven decision making and information delivery using dashboards.

DiveTab uses the speed of Spectre's in-memory and columnar data management technologies for rapid and secure access to your data and other resources, such as presentations and documents, from one central location.

With DiveTab, you can connect to and synchronize (download) your data from up to 10 different server hosts for the most current information. During a work

session, you can stay connected to a host or use the application in a disconnected mode with offline usage and local caching.

Not only is DiveTab available for the iPad, but Dimensional Insight also offers DiveTab for the PC with a similar look and feel to the iPad app. The DiveTab functionality is the same on both platforms for a smooth transition when on the road or in the office.

Installation

Installation for a large deployment might seem daunting, but Dimensional Insight's global team of business intelligence consultants assist with the design, implementation, and customization of your application. Dimensional Insight's consulting service plans offer the flexibility to deliver complete turnkey solutions or remote support for your internal IT team or any level of service in between putting you in control of your application.

The consolidated installation package for Diver Platform 7.0 and Spectre optionally includes required third-party software to handle installation of pre-requisites. Workbench gives you centralized project management to manage the entire back-end process from a single tool.

All Dimensional Insight clients are fully Spectre-enabled, and Spectre-enhanced back-end processes are fully supported in Workbench. Spectre takes center stage in the Diver Platform 7.0 architecture. The cBases developed with Spectre can take advantage of the Dimensional Insight clients:

- ProDiver, desktop analytics client
- DivePort, web portal environment
- DiveTab, tablet-based mobile client

Summary

To recap, Dimensional Insight's Spectre engine is designed from the ground up to optimize powerful business intelligence. Whether your business is healthcare, supply chain, goods and services, or another business, the Spectre engine delivers fast, scalable, and manageable business intelligence for enterprise customers.

- Speed
  - Fast build times keeps information current.
  - Fast run-time performance for most frequently used functions boosts productivity.
- Scalability
  - Columnar databases reduce build times and use less disk space.
  - Refresh of cached dives and reuse across users with the same access keep results fresh and minimize per-user overhead.
- Manageability
  - Spectre with the Workbench IDE simplify back-end processes with centralized project management.
  - Capture the power of Spectre in DiveTab for business intelligence on-the-go.