GIGAOM

Sector Roadmap: Cloud Analytic Databases 2017

Credit: buchachon/ThinkStock

By William McKnight

GIGAOM

Sector Roadmap: Cloud Analytic Databases 2017

12/15/2016

Table of Contents

1 Summary

The cloud is proving immensely useful to providing elastic, measurable, on-demand, self-service resources to organizations. The uptake in 2016 has been phenomenal, continuing the biggest transformation that technology professionals will experience in their careers.

Just about any software, including databases, can be placed in a public cloud these days by simply utilizing cloud resources as an extended data center. This may solve an immediate pressing problem, but the opportunities missed without true cloud integration are huge.

Some relational databases have undergone *significant* cloud-related development in their latest releases. Those will be the focus of this Sector Roadmap, along with the databases built native for the cloud.

We will also only examine true databases as in relational databases, not NoSQL or Hadoop data stores, which are much more integrated with the cloud, but serve mostly different purposes from databases. The platform category distinction should be made first and this report is written to the audience that has workloads meant for databases.

This Sector Roadmap will not develop the value proposition for the cloud itself. The cloud is important enough, and there is enough diversity of offerings, to be a given in a data platform selection process in 2017-18 and beyond.

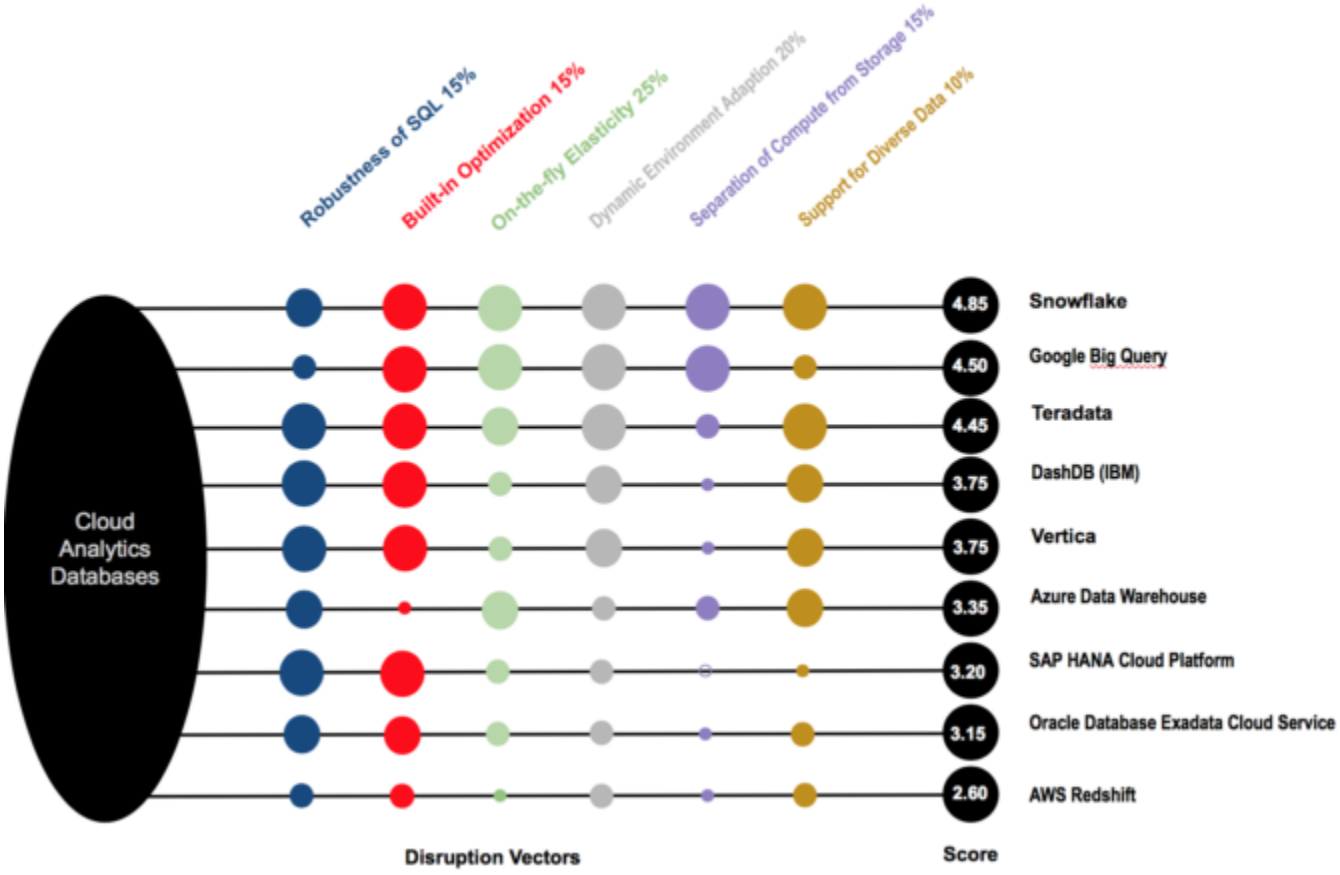
This Sector Roadmap is focused on *analytical* databases and not *operational and transactional* databases. The market is well acquainted with this distinction in database offerings, which not only remains strong but has grown in 2016. Companies who *mismatch* workloads and databases pay a performance and usability price for that decision.

Vendor solutions are evaluated over six Disruption Vectors: Robustness of SQL, Built-in optimization, On-the-fly elasticity, Dynamic Environment Adaption, Separation of compute from storage, and Support for diverse data.

Key findings in our analysis include:

- Due to the economics and functionality, use of the cloud can now be a given in most database selection in 2017 and beyond.

- Several offerings have been able to leapfrog databases with much more history by being "born in the cloud" and tightly integrating with it through On-the-fly elasticity, Dynamic Environment Adaption, and Separation of compute from storage.
- While traditional database functionality is still required, cloud dynamics are causing the need for more Robustness of SQL, Support for diverse data and other capabilities that may not be present in traditional databases.



Key:

- Number indicates company's relative strength across all vectors
- Size of ball indicates company's relative strength along individual vector

2 Introduction and Methodology

Conventional analytical databases and data warehouse systems are becoming unwieldy, due to their limitations of scaling their architecture and the correspondingly large capital outlay and administrative burden. Thanks to the cloud, analytical engines can be deployed quickly, scaled to fit, and run without a single space in an on-premises server room rack.

This de-coupling of analytical capability from the confines of a data warehouse appliance or local cluster is making more sense for many companies. Organizations can transcend the boundaries of their on-premises solutions (or the acquisition of a new one) in an easy to manage and easy to budget manner. This phenomenon is changing the database industry, forcing vendors to undergo rapid changes to their platforms and has spurred fundamental transition, new innovation, and market share disruption.

Just as critical as this architectural disruption is the need to make business users less constrained in their analysis of data. With traditional data warehouses, the responsibility of IT and its database designers was extended to answering substantially all the questions the business had. Asking new questions later in the process meant changing the data pipeline and the design of the analytic database—a critical bottleneck in the process.

With data volumes and the business' needs for information growing at exponential rates, an on-premises monolithic data warehouse appliance will soon be overwhelmed (if it has not been already). The cloud has created the potential for near limitless capacity. Fully realized, a cloud analytical database solution will propel the capabilities of a company way ahead, but be forewarned, not all platforms leverage this breakout capacity to the fullest. Not all platforms were built for the cloud, and vendors who have made their previously on-premises-only conventional platforms available in the cloud were able to make the full transition. On the other side, newer players with cloud-only offerings may lack the maturity in some of their platform capabilities that a mature organization might expect for their analytical needs.

Methodology

For our analysis, we have identified and assessed the relative importance of six Disruption Vectors. These are the key technologies in which players will strive to gain advantage in the sector. Tech buyers can also use the Disruption Vector analysis to aid them in picking products that best suit their own situation.

The "Disruption Vectors" section of this report features a visualization of the relative importance of each of the key Disruption Vectors that Gigaom Research has identified for the cloud

analytics platform marketplace. We have weighted the Disruption Vectors in terms of their relative importance to one another.

Gigaom Research's analysis process also assigns a 1 to 5 score to each platform for each vector. The combination of those scores and the relative weighting and importance of the vectors drives the company index across all vectors. That produces the Sector Roadmap chart in the company analysis section.

3 Usage Scenarios

Analytic databases in the Cloud offer a number of popular usage scenarios. However today, options for these cloud platforms meet, and sometimes exceed, the same usage capabilities as their on-premises equivalents.

One of the most popular usage scenarios is the cloud analytics platform serving as an enterprise's data warehouse. On-premises data warehouses and appliances are hamstrung by their ability to scale beyond their current footprint—in terms of storage, memory, or even precious rack space in the server room. Companies can literally outsource the physical and administrative burden of an on-premises appliance or warehouse cluster. The EDW offerings in the cloud vary from a turnkey cluster up in running within a public cloud in minutes to a managed cloud offering of a popular platform with all bells and whistles turned on and bundled in one manageable pricing model—scalable to your needs today and tomorrow. You will see an in-depth comparison of these platforms in Section 5.

A second, and possibly even more popular, usage scenario is the analytical cloud database serving as a disaster recovery failsafe mirror of the on-premises warehouse or analytical system. Companies with existing investment in on-premises infrastructure find this method a convenient and cost-effective way to back up their environment and recover from an outage, loss of data, or complete failover. They use the cloud to serve the very important DR function by protecting and not abandoning a significant on-premises investment. Since most of the top-tier warehouse and analytical database vendors have cloud offerings, these companies can create a DR environment in weeks to days without the overhead of maintaining server hardware, finding off-site locations for them, or large CAPEX investments.

Third, companies with pre-existing, significant on-premises analytical systems already in production are using the cloud to created development and test environments. Also vice versa, companies are moving their production environment to the cloud (for a multitude of reasons) and repurposing their on-premises systems for development and testing. The cloud gives the flexibility to stand up anything from a small sandbox to play in or full production-ready environment with the ability to quickly scale two times, three times, and beyond the current on-premises solution.

Finally, another popular use case is using the cloud to create on-demand and workload off-loading stand-up, run, and shut down analytic instances. Many companies do well with their on-premises system, except during times of heavy use—say an end of month financial process—or during a planned outage for maintenance. In this case, a company may temporarily stand up a nearly identical instance of their on-premises system (or just a portion of it) and move user or batch analytical, integration, or processing workloads to the cloud instance. Then when the

operation is complete, any changes to the database are migrated back down from the cloud to the on-premises analytic system (a step that is unnecessary in the case of true separation of compute from storage) and the cloud instance shuts down or is paused. This presents both performance and cost-reducing opportunities for companies.

4 Disruption Vectors

The cloud analytic database platform decision is crucial to enabling the emerging company strategy of analytics.

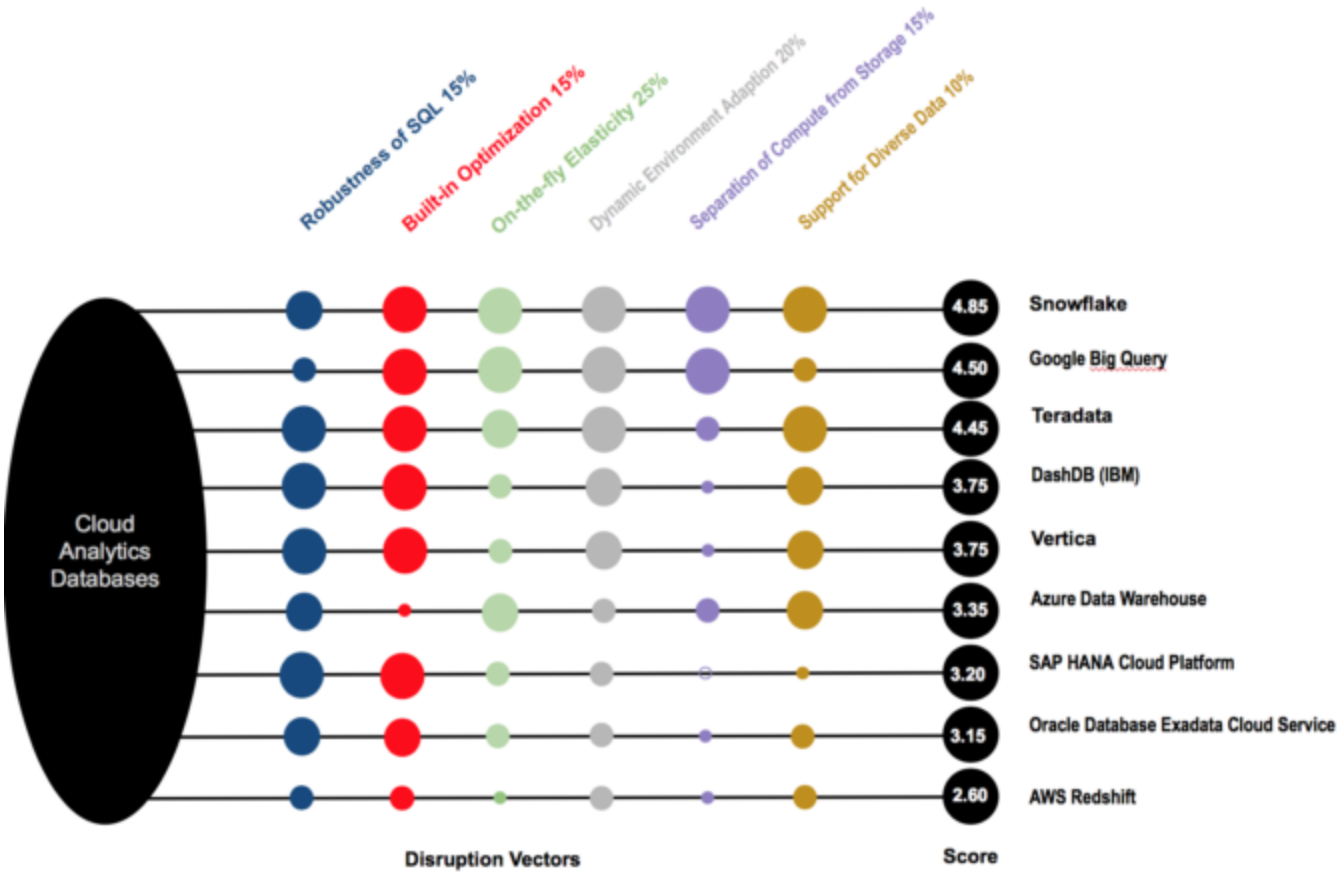
Analytics is a company strategy many are using to identify and control their future. Preciseness becomes required as the data science of the company grows. This comes with increasing amounts of data which has made the cloud truly compelling in 2016.

Below, we discuss the six most important vectors of this disruption. These vectors are *focused on the integration of the database with the cloud* – or what could be compromised by moving a database to the cloud.

These criteria are prominent in a cloud selection because a database's integration with the cloud features strongly in platform success.

The six vectors we have identified are:

- Robustness of SQL
- Built-in optimization
- On-the-fly elasticity
- Dynamic Environment Adaption
- Separation of compute from storage
- Support for diverse data



Robustness of SQL

SQL is the long-standing language of the relational database, supported by thousands of tools and known by millions of users. Backward compatibility to core SQL is essential.

To leverage tools and skills in SQL on the cloud without significant rework, a solution should support standard SQL, not partial or an incompatible variant. The SQL 2011 standard is the latest revision and added improved support for temporal databases, time period definitions, temporal primary keys with referential integrity, and system versioned tables, among other enhancements.

Extensions are welcome and can be found in the company offerings. Examples where SQL is extended include:

- Decorrelation, which allows you to write complex subqueries in any clause of your SQL statement (SELECT, FROM, WHERE, etc.)
- Array and structure datatypes

- Theta JOIN, which offers the ability to use inequalities in join key comparisons, as well as arbitrary expressions as JOIN conditions
- Persisting result sets for a time period
- UNDROP for instant table restore
- CLONE a table, a schema, or an entire database
- Sophisticated analytic and windowing functions
- Bitwise aggregation functions
- Linear regressions functions
- Cardinality estimation functions
- Flex tables

Built-in optimization

Databases have a rich set of techniques that have been developed and honed over the years to optimize performance. A rich cloud analytic database should build on these, including the use of a query optimizer – a key component to consider.

Databases utilize optimizers to determine the best path, usually among many options, in the data for satisfying a query. Optimizers must get a precise understanding of what data the query is after regardless of how the query is written, forge the best path, and make this determination quickly.

While rules-based optimization is fine, some measure of cost-based optimization, utilizing the data distribution, is required in a robust database.

Moving to the cloud should not impact optimization, which needs to be able to consider the characteristics of the cloud(s) in the path determination. Also since many queries will span clouds and on-premises as more data platforms are explored, the cloud analytic database optimizer needs to work for this emerging set of 'data virtualization' queries.

On-the-fly Elasticity

The promise of the cloud is immediate access to the resources you need. This may mean a growing or shrinking amount. The exact level of resources necessary should not only be what is

provisioned; that is, what should be charged for. A cloud analytic database needs to be able to scale up and down without disruption or delay.

If it takes hours to scale up or down or creates disruption for migration or repartitioning, one of the key benefits of the cloud is lost.

The more proactive and involved a customer has to be in the process of resource determination, the less elastic the solution is. The more granular the growth of the clusters, and the less of a "step ladder" approach to resources, the more elastic the solution is.

Databases in the cloud should not be able to run out of space.

Elasticity extends to upgrades. The cloud analytic database software should be able to be upgraded without any downtime, performance impact or interruption of service.

Dynamic Environment Adaptation

Another aspect of database utilization that must be retained in the move to the cloud is the concurrency. Most databases put in the cloud have expectations around usage that are growing, so arguably a cloud analytic database should provide even higher levels of concurrent access.

Part of usage growth is in the area of complexity of query, variety of user, and the removal of daypart profiling as access goes 24/7/365. The solution must support a more variable usage pattern in the cloud – a scale up in usage as well as storage.

Cloud solutions should be readily available for use by disparate users and applications at any time with consistent performance. Elasticity and the ability to support high degrees of concurrent access and usage are required to meet the demands.

Databases designed for on-premises environments assume that resources are static and plan execution accordingly, or have inconsistent performance.

In the cloud, resources should be able to flex to the requirements at any time, even between when the optimizer creates a query plan and when it is executed. Systems built for the cloud need to dynamically adapt as requirements change.

Separation of Compute from Storage

Taking a cue from the Hadoop clusters, many of these leading cloud analytic databases have the ability to independently scale compute and storage through specification of nodes with emphasis on one aspect over the other. This alleviates compromise in the resource allocation and unused costly resources.

You only pay for what you use, when you use it. Compute and Storage are independently priced and made available. Legacy on-premises software assumes those are all tightly coupled. In that approach, you are forced to pay for compute even when you aren't using is just to get space to store data. And vice-versa.

A solution built for the cloud should allow you to use and pay independently for the storage and compute you need. This arrangement is important to have in the foundation of the cloud analytic database.

Support for Diverse Data

Going into the cloud means closer proximity to the data center of gravity forming in the cloud. The cloud is home to a broad diversity of data forms, from traditional data such as customer profiles to newer machine data such as IoT data sources. A growing majority of these data sources store and transfer data in flexible, schema-free formats.

The cloud analytic database should be able to not only directly absorb the many formats such as JSON, XML, Avro and Parquet, it should be able to interpret the data as columns, seamless to other columns in the database. Forcing a transformation step into the mix adds complexity, fragility, and delay.

It should also be able to treat semi-structured data as efficiently as columnar data and not force any schema definition on the customer.

5 Company Analysis

The companies examined here represent relational databases either built for the cloud or with significant cloud-related development in their latest releases. The vast majority of these databases are being sold as cloud offerings.

These represent the cloud choices for analytic workloads.

Snowflake was built from the ground up in the last few years for the cloud, as was Redshift. Google Big Query is commercial version of Dremel, used for many years inside Google. Azure debuts with origins in Microsoft Parallel Data Warehouse. Teradata has made major strides into the cloud for its enterprise database. Vertica, destined to be part of Micro Focus, has also slid over into the cloud world nicely. dashDB is a combination of DB2 BLU and Netezza, for AWS or SoftLayer. Oracle's Exadata Cloud Service is their 8-year old appliance now available in Oracle Cloud. SAP HANA Cloud is likewise the in-memory appliance in the cloud.

Detailed descriptions of our nine scored vendors follow with discussion of how they each fare across the key Disruption Vectors.

Amazon Web Services Redshift

Amazon introduced its hosted data warehouse platform, Redshift, in early 2013 as a complement to its cloud computing universe, Amazon Web Services. Redshift is available in the cloud and on AWS only. Redshift is offered separately from Amazon Relational Database Service (RDS). Redshift is built using a massive parallel processing architecture.

Redshift could be a consideration for organizations needing a data warehouse with little to no administrative overhead and a clear, consistent pricing model. A Redshift cluster can be created and ready for use in minutes. Redshift pricing is simple as well. Customers pay a certain dollar amount per node per hour with no additional costs, so it is easy to predict and budget. Amazon seems to trust the principle "less is more" concerning its Redshift product.

However, on the other side of the coin, there are tradeoffs to this simplicity. "Getting under the hood" is not easy. There is no command line interface to a Redshift cluster's operating system. The only way to interact with the cluster is through SQL queries. Also, a Redshift cluster cannot be shut down but instead must be deleted. A snapshot of the cluster can be taken, and it can be restored—which can take a few minutes to hours, depending on the data size. Thus once the billing on a cluster begins, it continues until the cluster is deleted, regardless of use.

Redshift SQL is based on PostgreSQL 8.0.2. PostgreSQL 8 was released in 2005 and version 9.x has a fairly sizeable list of features not supported by Redshift. Customers looking for a robust set of SQL capabilities may be disappointed to find the inability to perform some of the following operations in Redshift, including but not limited to:

- • Table partitioning
- • Constraints, such as unique, foreign keys, primary keys, and checks
- • Indexes
- • Array and row constructors
- • Stored procedures
- • Triggers
- • Full text search

Redshift does not have built-in optimizers per se, but relies on best practices by users to design tables and improve queries to maximize performance. It does, however, offer a workload management feature to define query queues and assign executions at run-time to prevent bottlenecks during peak usage.

Redshift does not offer “on-the-fly” elasticity. Expanding a Redshift cluster requires a new, larger cluster to be allocated and migration from the smaller environment. This involves either downtime or a series of manual data copying to sync data from an old to a new cluster. Unlike EC2 instances where additional storage volumes can be initialized and attached to a running virtual machine, Redshift storage capacity is fixed to the node type (compute dense or storage dense). Therefore, to expand storage requires more nodes, which requires a new cluster to be spun up.

Also, Redshift does not separate storage from compute resources. However, if a customer needs the data to persist but desires the cluster to go away, unloading data onto S3 (or copying it back) can be done with a single simple command, but it is a manual process, nonetheless.

In support of diverse data, Redshift has a few nice features like the JSON_EXTRACT_PATH_TEXT function. However, it does not support large text columns like BLOBs and so it cannot store raw data in a VARCHAR field larger than 64K.

All-in-all, Redshift shines for its simplicity, but has some gaps along the lines of our Disruption Vectors.

Google BigQuery

Google BigQuery has its origins in Dremel, the internal Google paper that specified the database that Google has used internally since 2010. Google BigQuery is Dremel for the rest of us, externalized to the cloud (Google Cloud). Though built on-premises initially, from a commercial perspective, it exhibits as a seamless cloud database with full elasticity and complete separation of storage and compute (and the memory shuffle tier).

Optimization has occurred around the shuffle step. Shuffle is an integral part of the manner in which most cloud analytic database queries occur, which is analogous to MapReduce. Shuffle has become a bottleneck.

By dedicating nodes to hosting remote memory and persisting the intermediate results, similar to how Spark works with its RDDs, BigQuery ensures the shuffle is performed in memory. Furthermore, the entire shuffle does not have to complete before data is made available to the workers (the next/last step of the query processing).

BigQuery has a massive amount of hardware available to devote to queries for the length of time it takes. The customer can choose the jurisdictions of their storage (i.e., US, Europe) and greater regional specification is coming in 2017.

It is another of the low-touch databases. There are no indexes, no column constraints, and no system tuning. Database administration is reduced to environment specification and engagement and access control. Its optimizer is cost-based and rule-based.

Recently Google announced ANSI 2011 SQL compatibility with nested and repeated data types, etc.. BigQuery has some work to do on their SQL, as in complex JOINs and update DML. And for now, User Defined Functions are utilized for JSON and XML data type storage.

It has ODBC and JDBC compatibility and is building an ecosystem of partners that can bypass xDBC.

BigQuery is designed to work seamlessly with Google's NoSQL database BigTable as well as Google Drive, though a data virtualization approach.

Google also takes down the cost after 90 days, helping to make it one of the lower cost analytic cloud databases.

HPE Vertica in the Cloud

Since its inception, Vertica Analytics Platform was designed to run on clusters of commodity enterprise servers to provide high availability and scalability. Vertica in the Cloud was a natural evolution for the platform. Much like other conventional platforms that have been elevated into the cloud, the Vertica cloud platform has all the same features and touted performance as its on-premises forbearer. For example, Vertica in the Cloud's column-orientation can manage petabyte-scale, fast-growing volumes of data and provide fast query performance when used as a data warehouse or other analytical purposes. To illustrate its strength, the platform boasts a service-level agreement with one of its customers of 60TB per hour data ingest.

Vertica Analytics Database is offered on AWS and Azure.

Like its predecessor, Vertica in the Cloud has its own flavor of SQL called Vertica SQL or VSQL, for short. In fact, most command line query executions are evoked with the VSQL command from a Linux console. VSQL supports a subset of ANSI SQL-99. There are some distinctions worth noting—such as an UPDATE actually being a DELETE followed by an INSERT and the CONNECT, COPY FROM, and EXPORT TO VERTICA commands for connecting to and shipping data to and from external sources and other Vertica clusters in batch. The syntax differs very little from ANSI SQL, making it easy to adopt and use.

Vertica is fully ACID-compliant with sophisticated and resource-costly operations to maintain ACID, thus trading availability for consistency.

Vertica has built-in optimizers for columnar storage, compression, and continuous performance. For example, Vertica has a FlexStore feature that reads columns from database objects called projections, which are optimized collections of table columns. Such groupings can be advantageous for correlated columns and for columns that are always accessed together for projecting but not necessarily filtering or joining; however, there is an administrative burden in maintaining projections. Vertica uses both encoding and compression to save disk space. Encoding increases performance with less disk I/O during query execution and increases storage. Also, in certain cases, compression rates of 50-90% are achievable with Vertica. When properly configured, Vertica's optimizers will help users enjoy high concurrency. However, the challenge of diverse and highly variable usage patterns will challenge administrators to continually monitor and refresh projections.

Like many other platforms with on-premises predecessors, Vertica storage is tightly coupled to compute resources. This has a number of implications. Obviously, without separation, data must be loaded onto and off-loaded from Vertica to take advantage of another engine for processing or additional workloads. Also, compute resources cannot easily be "spun down." Thus, Vertica users will be unable to save money during times when data is at rest. Another added

consideration of Vertica's tight coupling of storage makes spinning up additional nodes require planning and some downtime.

In terms of support for diverse data, Vertica has a number of capabilities, such as JSON schema on read, and the ability to handle most of the conventional non-relational data types and formats.

IBM dashDB

dashDB is a combination of DB2 BLU and Netezza for SoftLayer and AWS. dashDB is actually becoming the main code base from which all of DB2 follows. There is a dashDB for Analytics and a dashDB for Transactions that come with preconfigured configurations for analytics and transactions respectively. We obviously focused on dashDB for Analytics and will reference this product as dashDB.

The contributions from Netezza to dashDB are largely in the area of in-database analytics. dashDB is not utilizing Netezza hardware and its physical configuration of components that includes overloading the FPGA. It is the software side of Netezza that has been integrated with dashDB.

dashDB is available as a managed service from IBM, regardless of the public cloud chosen. In this arrangement, the customer manages everything "from the table down" and IBM manages everything "from the tablespace up."

Since dashDB is strongly based on DB2, it inherits its SQL and the rich cost-based optimizer and statistic-gathering capability of the DB2 database. This is one of the most, if not the most, powerful optimizers in all databases. dashDB also inherits the concurrency capabilities of DB2, not Netezza. This is preferable.

dashDB does not grow on demand. There are various terabyte levels you can choose from and growth is monitored by IBM personnel and processes and you are moved to a larger plan as the data grows. There are brief outages for software upgrades and queries can be rolled back if caught in cluster loss.

Separation of compute and storage is not part of dashDB today.

The dashDB approach to diverse data is to convert it to table structure in the schema. So the JSON and other forms of tagged data are interpreted and moved into table format. One reason it's done this way is that a huge part of the value proposition of dashDB is the columnar capabilities from DB2 BLU.

dashDB only stores data in their individual columns. However, when columns are determined to be frequently accessed, they are moved into memory. Data compression with dashDB preserves the order of the data and allows SQL to be processed while the data is still compressed. Storing data in JSON format would pack a lot of data into a column and this approach would lose some of its value.

The columnar approach is preferable for an analytic workload and the smart use of memory finds a great balance for the analytic workload. There is still some work to do around elasticity. However, DB2, with the smart inclusion of some Netezza software assets, make dashDB a contender for the cloud analytic database workload.

Microsoft Azure SQL Data Warehouse

The Azure SQL Data Warehouse is one of the newest offerings, making its debut for public use in mid 2016. It runs a massive parallel processing architecture and shared nothing architecture on cluster nodes each running Azure SQL Database—which shares the same codebase as Microsoft SQL Server. The cloud platform has an on-premises predecessor—Microsoft's Parallel Data Warehouse.

Microsoft-native on-premises shops will find the Azure platform comfortably similar to SQL Server. The platform uses Transact-SQL, just like SQL Server, and can even be accessed from familiar tools, like SQL Server Management Studio (SSMS). Transact-SQL, while widely used and known, has some significant departures from ANSI SQL.

SQL Data Warehouse does not yet have full built-in optimizers. Unlike its predecessor, SQL Server, SQL Data Warehouse does not automatically detect, create or update column statistics for query planning and execution optimization. Microsoft reports they plan to implement this in future releases. Currently, the platform requires manual maintenance of statistics—thus the execution plans created by the optimizer are only as good as the last refreshed statistics.

Microsoft does offer the ability to pause and scale the system. Pausing the cluster halts the billing temporarily while the system is not in use—a nice ability – and manual intervention to control costs. Scaling the system can be achieved easily through the Azure portal or PowerShell commands. Another cost-saving alternative to pausing the warehouse would be to scale the Data Warehouse to the smallest size, leaving it available for small query loads. Scaling up is just as easy. However, while robust, it is not truly "on-the-fly." Any queries that are executing when the pause or scale commands are executed are cancelled. If only SELECT statements are running, this is fine, but if a transactional statement is running, it must be completely finished or rolled back before the cluster will pause or scale.

Microsoft does offer some separation of storage and compute. The SQL Data Warehouse data is stored in Azure Blob storage, which allows customers to scale storage without changing compute resources, or vice versa.

Microsoft also does a decent job supporting diverse data types. From built-in JSON and XML functions to PolyBase, the Azure SQL Data Warehouse can query both structured and unstructured data types. While semi and unstructured data are treated as external resources by SQL Data Warehouse, they are still available for accessing, importing, and exporting with T-SQL statements.

While late to the game and with some features still withstanding, Azure SQL Data Warehouse is an attractive option—particularly for organizations who already have an extensive Microsoft footprint.

Oracle Exadata Cloud Service

Oracle introduced its Exadata platform in the cloud in late 2015. Like its on-premises predecessor, Oracle Exadata Cloud Service runs Oracle Database software on high-performance engineered systems with scale-out architecture. Since its storage is tightly coupled to its dedicated compute servers (and not commodity servers), Exadata Cloud Service is only available within the Oracle Cloud.

The Exadata Cloud Service gives organizations the same power as on-premises resources hosted in Oracle's public cloud with cloud licensing models and elasticity determined by consumption needs. For example, Exadata Cloud Service lets you choose metered (monthly) or yearly subscription use of Oracle 11g and 12c database services including use of all of Oracle's features and options—such as Real Application Clusters (RAC), In-Memory, Advanced Compression, just to name a few—which would normally be offered a la carte on an on-premises Oracle system. These subscriptions include on-demand hourly bursting for peak periods. To use Exadata, organizations must start with at least 16 Oracle cores, but to use the Oracle Database Cloud Service on commodity hardware, organizations can get started with as few as 2 Oracle cores.

Oracle SQL is widely known and makes use of a robust SQL syntax. Companies already familiar with Oracle platforms will be able to adopt it easily.

Since Exadata Cloud Service has its roots in the on-premises platform, it leverages the built-in Oracle optimizations, including its popular cost-based optimizer (CBO)—although the rule-base optimizer is still available for backwards compatibility. The Oracle CBO supports many well-known features such as partitioning, reverse and function-based indexes, SAMPLE clauses in SELECT statements, query rewrite with materialized views, and so on.

Elasticity on Oracle Exadata Cloud Service is not fully “on the fly.” To expand the cluster, a customer must manually activate more cores. However, organizations can leverage pay by the hour to “burst” some additional resources for peak usage. Another upside is that the Exadata Cloud Service model apportions physically-dedicated 1/4 racks to clients. This gives consistent high performance, because there are no “noisy neighbors” to compete for resources. Overall, Oracle scores high in performance but adapting and scaling is a manual process.

Finally, Oracle Exadata has some support for diverse data types. For example, they offer JSON and XML data types that can be decomposed and scanned in storage. Oracle has supported large objects for some time.

The on-premises Oracle databases and platforms have enjoyed wide use for years and represent a substantial share of those markets. Likewise, many of these existing customers are making their entrée into the cloud by using it for disaster recovery and replication of their on-premises Oracle Exadata system. This is a common trend for companies with established legacy platforms who are not quite ready to jump into the cloud wholesale for their data warehouse and other analytical uses.

SAP HANA Cloud Platform

SAP has introduced its touted in-memory computing appliance SAP HANA (alongside a host of other cloud offerings, such as BusinessObjects Cloud). Known for its in-memory processing with the aim of high performance, customers can leverage the platform through a managed private cloud. Additionally, in late 2015 at SAP TechEd Barcelona, SAP introduced new business services for HANA Cloud including support for virtual machines, prebuilt Ariba cloud integrations, beta versions of workflow and business rules services. These include smart data streaming for high speed IoT scenarios that could enable enterprises to bridge on-premise and cloud landscapes.

SAP HANA Cloud Platform, persistence service (HANA)’s SQL capabilities, like its on-premises version, is compatible with ANSI 92 and extended using their SQLScript host of functions. SAP HANA also uses Continuous Query Language (CQL), similar to SQL, for processing streaming data.

SAP HANA Cloud has built-in optimizers, both rule-based and cost-based. Their rule-based optimizer parses the query into a normalized relational algebra tree as the input for the cost-based optimizer which creates the query execution plan. The optimizer also caches plans to avoid repeated efforts for optimizing the same query and re-used for any identical SQL statements that follow.

SAP HANA has the ability to scale up and out. It can scale down by reorganizing data to empty a node and then deleting it from the cluster. Scaling out to larger instant types will require a reboot, which, depending on the memory size of SAP HANA, might take up to 1 hour to reload all data into memory for the largest instance types. SAP HANA can grow and shrink memory on-the-fly on a single server but if it involves adding nodes, some downtime can be expected.

Since nearly all data is stored in-memory, the platform does not separate compute from storage. Data must be on-loaded and off-loaded to the platform memory.

SAP HANA Cloud does not yet support storing raw JSON and XML data. It is on their product roadmap for 2017.

All told, customers wishing to leverage the analytical power and speed of the SAP HANA in-memory platform or to back up their existing on-premises deployments will find the cloud offering an attractive one—offering all the same capabilities in a managed private cloud.

Snowflake Elastic Data Warehouse Service

Snowflake Computing was founded in 2012 with the purpose of building the first data warehouse built for the cloud. Snowflake is the only stand-alone product in our vendor list that is not part of a larger company.

The independent development has allowed Snowflake to forge a tight integration with the cloud. In addition, their modern approach has seen many features added to the database that are immensely useful in data warehousing, regardless of hosting. You can tell the data warehouse pedigree from the development.

With superior performance and the most hands-off model of ownership, Snowflake is the epitome of data warehouse as a service. The model, cost, features, and scalability have already caused some to postpone Hadoop adoption.

In its multicluster, scale-out approach, Snowflake separates compute from storage. It is fundamental to the architecture where multiple, independent compute clusters can access a shared pool of data without resource contention. Customers pay for what is used without a stair-step approach to resources and pricing. The cost model is simple at terabytes per year or compute hours. For primary storage, Snowflake uses Amazon's Simple Storage Service (S3). Snowflake also uses an SSD layer for caching and temp space.

Snowflake customers deploy a wide array of modern BI and visualization tools, some utilizing the ODBC and JDBC connectivity. Snowflake also offers a web interface.

Snowflake SQL includes support of objects in JSON, XML, Avro and Parquet using a special datatype that can handle flexible-schema, nested, hierarchical data in table form. There are no indexes either, as zone maps are used for an abstract understanding of data in the database. SQL extensions include UNDROP and CLONE. Features include result set persistence and automatic encryption.

No down time is required for anything including upgrades or cluster expansion.

Concurrency, a clear challenge in database scale-out, is a focus at Snowflake. Their automatic concurrency scaling is a single logical virtual warehouse composed of multiple compute clusters split across availability zones.

Finally, Snowflake has a native connector for Spark built on the Spark Data Sources API.

Snowflake has jumped constraints found in databases from earlier development and honed a very promising cloud analytic database. Eminently elastic on a foundation of separation of compute and storage, Snowflake offers as close to a hands-off approach as we found. Snowflake is market-leading in what you would want for a multi-purpose cloud data warehouse/ analytical database.

Teradata Cloud

Even though we are focused on the integration of the database with the cloud to determine the efficacy of the cloud analytic database, the database itself is a major part of the proposition.

Teradata's offerings stand out for data warehouse and data mart appliance platforms. Its Active Enterprise Data Warehouse line, based on the Teradata Database, supports an outsized percentage of large-scale data warehouses today. All database functions in Teradata systems are always done in parallel, using multiple server nodes and disks with all units of parallelism participating in each database function.

The Teradata Optimizer is grounded in the knowledge that every query will be executing on a massively parallel processing system. The Teradata Database Adaptive Optimizer also adjusts to the environment it resides in by utilizing a cost coefficient that gives it a cost profile for the specific hardware.

Teradata manages contending requirements for resources through dynamic resource prioritization that is customizable by the customer. The server nodes interconnect was designed specifically for a parallel processing multi-node environment. This interconnect is a linearly scalable, high-performance, fault-tolerant, self-configuring, multi-stage network.

Teradata has optional columnar structure for tables, effectively mixing row and column structures directly in the DBMS. XML and JSON are supported as a native SQL data type. AVRO support is coming in January 2017.

Teradata Everywhere is bringing its high-performance database (same codebase) to a suddenly large variety of platforms, especially in the cloud. This has migration and portability advantages. Teradata is now available on Amazon Web Services, Teradata Managed Cloud, VMware and the future of Teradata, IntelliFlex. It will also be available on Microsoft Azure soon. Teradata is available in its massively parallel (MPP) form on AWS on up to 64 nodes and is available in most AWS regions globally now.

There are customizable hash maps for data co-locating strategies and eliminating downtime during node expansion via table-level data distribution.

Under the Borderless Analytics initiative, Teradata QueryGrid and Teradata Unity were extended to Teradata Everywhere, as well as allowing any cross-platform queries, should the need arise or make sense, to be part of the architecture.

However, Teradata's architecture is a traditional architecture in which a single compute cluster is tightly coupled with data storage. This will be fixed in IntelliFlex, which will scale them separately.

Customers do have to monitor space as well, which is metered out in AMPs of a few terabytes at a time.

Overall, the strength of the Teradata Database makes Teradata in the cloud a very compelling cloud analytic database.

6 Key Takeaways

Our Disruption Vectors indicate that a tight integration with the cloud is imperative to long-lasting cost-effective success for the analytic workload in a cloud database. Snowflake is built as a cloud analytic database from scratch. Its independent development, and focus on analytic features, has enabled Snowflake to hit the mark on all of our Disruption Vectors.

Google's offering is quite compelling as well and Teradata's Everywhere strategy takes its offerings to the cloud quite well. Vertica's performance demands attention and dashDB is a strong contender with some strong accommodation for the analytic workload.

Teradata and IBM have offerings that should not only see their on-premises customers extending with them to the cloud, they should attract new customers. Vertica's cloud offering should do the same for its HPE heritage customers.

For companies willing to invite Google into the enterprise, Google Big Query may shine bright here as well.

Other takeaways

- The cloud now offers attractive options with better economics, such as pay-as-you-go which is easier to justify and budget, better logistics (streamlined administration and management), and better scale (elasticity and the ability to expand a cluster within minutes).
- Many enterprises are fully embracing a cloud-based analytics strategy and making it accessible across their current infrastructure and data ecosystems. Many others are adopting it as a secondary platform for purposes such as disaster recovery and offloading analytic and data transformation workloads.
- Depending on needs, customers may choose to weight the Disruption Vectors differently than we have here. Ultimately all of these databases are viable. You should know what you're acquiring, what you are missing and what it is costing you.
- Being "born in the cloud" does appear to offer advantages. While on-premises-first development brings a robust database to the table, not all functions are always part of the cloud solution and not all of the organizations behind them have made the transition to cloud.
- Down the road, cloud database capabilities will likely include being able to dynamically spin up a cloud instance, migrate workloads, and synchronize small-scale data on the fly.

- Current and future capabilities make a powerful argument for considering cloud databases for analytics in the overall data ecosystem of an organization.

7 About the Author: William McKnight

William is President of (McKnight Consulting Group Global Services). He is an internationally recognized authority in information management. His consulting work has included many of the Global 2000 and numerous mid-market companies. His teams have won several best practice competitions for their implementations and many of his clients have gone public with their success stories. His strategies form the information management plan for leading companies in various industries.

William is author of the books *Integrating Hadoop* and *Management: Strategies for Gaining a Competitive Advantage with Data*. William is a popular speaker worldwide and a prolific writer with hundreds of published articles and white papers. William is a distinguished entrepreneur, and a former Fortune 50 technology executive and software engineer. He provides clients with strategies, architectures, platform and tool selection, and complete programs to manage information.

8 About Gigaom Research

Gigaom Research gives you insider access to expert industry insights on emerging markets. Focused on delivering highly relevant and timely research to the people who need it most, our analysis, reports, and original research come from the most respected voices in the industry. Whether you're beginning to learn about a new market or are an industry insider, Gigaom Research addresses the need for relevant, illuminating insights into the industry's most dynamic markets.

Visit us at: Gigaom.com/reports.