

# BIG DATA ANALYTICS

## *- THIS TIME IT'S PERSONAL*

*Robin Bloor, Ph D*



## Executive Summary

1010data is surprising in several ways. As a technology, it does not fall into any obvious category, but instead spans several categories. We have witnessed demonstrations of it, studied detailed technical papers written about it and had long detailed discussions about the technology's capabilities with 1010data staff, including one of the product's creators. The following summarizes our conclusions:

- 1010data provides an Interactive Analytical Service. In respect of its software it is in a category of its own. The software could be thought of as if it were an interactive analytical capability directly wired to a high-performance "big data" database. However, it is more tightly integrated than such a description suggests. The analytical capability is a fundamental part of the database.
- When we say that 1010data's software is interactive, we mean it. Users access and analyze very large amounts of data interactively, as if they were manipulating data in a spreadsheet or using a PC-based statistical analysis tool. It is a "personal" capability.
- 1010data provides a comprehensive set of analytical functions. These include all the usual analytic and financial functions, and also two important but less frequently offered groups of functions that broaden its area of application. These are:
  - Time-Series Analysis
  - Text Analysis
- There are broad areas of commercial application for this technology, not just in the financial sector, where the product cut its teeth, but in telecommunications, utilities, retail, web-based retail, government, social network sites, for analyzing machine generated data and for use in delivering data as a service.
- 1010data is very fast to implement. Customers can and do implement 1010data in a few days, running the software in 1010data's cloud data center or alternatively, on an appliance delivered by and maintained by 1010data. Customers have no need to involve their IT department in installing the capability. Fundamentally, 1010data is providing *Analytical Capability and Database-as-a-Service*.
- The 1010data database delivers extremely high performance, comparable to and often superior to other "big data" database products that were built for speed. The database is highly scalable and capable of being used as a standalone database for SQL queries, if desired. Nevertheless, its performance was designed specifically for interactive analytical use. The fact that it performs well as a standalone column-store database is a bonus.

In summary, 1010data provides a truly interactive analytical capability. The service can be implemented very quickly and can interactively analyze huge amounts of data - it is in use on tables that have hundreds of billions of rows. It has a broad set of analytical capabilities and consequently it has broad areas of application, well beyond the normal range of most analytical BI products.

We recommend that companies which are looking for an analytical platform, especially those that gather, accumulate and analyze large amounts of data, consider it as an option.

## The Genesis Of 1010data

What is 1010data?

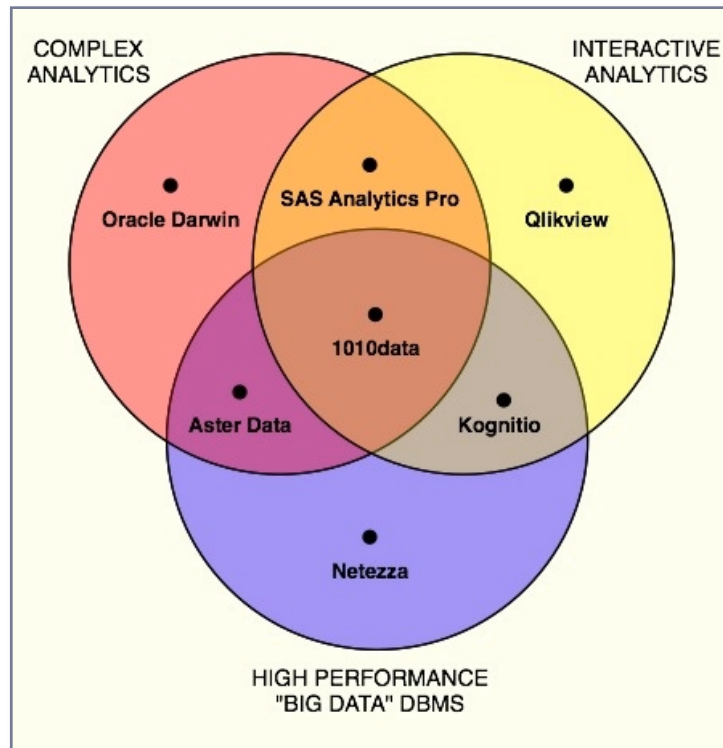
This might seem like a strange question to ask, but the fact is that 1010data is not an easy technology to classify. At the moment there are no products or services that are similar in overall capability to 1010data, although there are quite a few products which overlap some areas of 1010data's application. The broad spread of 1010data's capability has, in our view, led some industry analysts to misunderstand the technology.

Take a look at *Figure 1*. It is a Venn Diagram that positions 1010data relative to several other BI software products. As *Figure 1* indicates, 1010data is not solely a high performance "big data" DBMS, although it can be deployed in that role. Neither is it just a platform for performing complex analytics, although it could also be deployed in that role too. Nor is it just an interactive analytics capability, although in practice, most 1010data customers use the technology in precisely that role.

1010data fulfills all three roles, whereas the products with which it might be compared (some are mentioned in the diagram) can at best claim to fulfill two of these roles. So you can use 1010data as a database that simply satisfies the query demands of a whole clutch of BI tools and you can also employ it as an analytical database that carries out analytical calculations, and many 1010data customers do.

However, in almost every case, 1010data's customers buy the 1010data service for its interactive analytics capabilities. 1010data delivers a high performance *Interactive Analytical Database* and in that respect, it is unique. There are analytical products like Qlikview or Tableau or even MS Excel, that provide an interactive analytical capability, but they do not sit over very large volumes of data. They tend to sit over subsets of data drawn from a data warehouse or data mart.

1010data is interactive in exactly the same way that such PC-based products are. The user does something, such as pose a question, and the software responds quickly with the result. The user then does something with that result, such as join it to some other data, and again the software responds quickly. This way of working is highly productive, allowing a user to explore the data in ways that would be impossible without an interactive interface and fast database responses.



*Figure 1. A Positioning of 1010data*



## The Influence of Wall Street

Given the uniqueness of 1010data, it is interesting that the genesis and evolution of the technology has been primarily customer-driven. The software engineers, Joel Kaplan and Sandy Steier, who built 1010data, entered the software industry from the IT user's side. Originally they were employed by the Wall Street bank, Morgan Stanley, as part of the bank's Analytical Systems Group, and there they pioneered the use of software as an enabling capability for financial traders.

That was back in the 1980s, when neither the PC nor the spreadsheet were widely used, Windows didn't exist, and Unix servers were as rare as ATMs in Antarctica. Nevertheless, in those days it was possible, using the IBM mainframe, to apply analytics to data and present the results on a green screen. The analytical capability that Wall Street traders wanted at the time was spreadsheet-like in a way. They wanted to manipulate large tables of financial data, sorting it, filtering it and applying analytical functions to it.

Moving from one employer to another, Kaplan and Steier built software that provided such capabilities for many years before they founded 1010data to build their own technology based on their experience and expertise in those financial environments. And naturally, from the get go, they sold 1010data to companies in the financial sector.

So, consider again the location of 1010data in the Venn Diagram of *Figure 1*. The technology didn't start life in some other location on that diagram and gradually migrate to the center. It was designed from the start to be an *interactive analytical database*.

## Another Rare Capability: Database as a Service

We could have made the Venn Diagram more detailed and a little more precise by adding a further circle labeled *Database As A Service* into which, as far as we are aware, only 1010data would fit.

1010data could be characterized as "cloud technology" in the sense that the software is not often deployed in the customer's data center. The company has its own cloud data center and in most customer engagements, it loads the customers' data and runs the software from its data center, while users within the customer organization log in and use the system via a browser.

In practice, 1010data has a minimal to almost zero requirement for Database Administrators. The company forges an agreement on where data will be sourced from and what the update cycle will be, then 1010data's consultants configure and load the data into the database and make the software available to users within about a week, or sooner. So 1010data is primarily a cloud-based service and has been since the company was founded in 2000, just a year after the more widely recognized cloud company, Salesforce.com, was founded.

Nevertheless, 1010data is also deployed and used in other ways. It can be deployed as an appliance, in which case 1010data will deliver a server or cluster of servers to the customer so that the hardware is co-located in the customer's data center, but 1010data will still manage the running of the database.

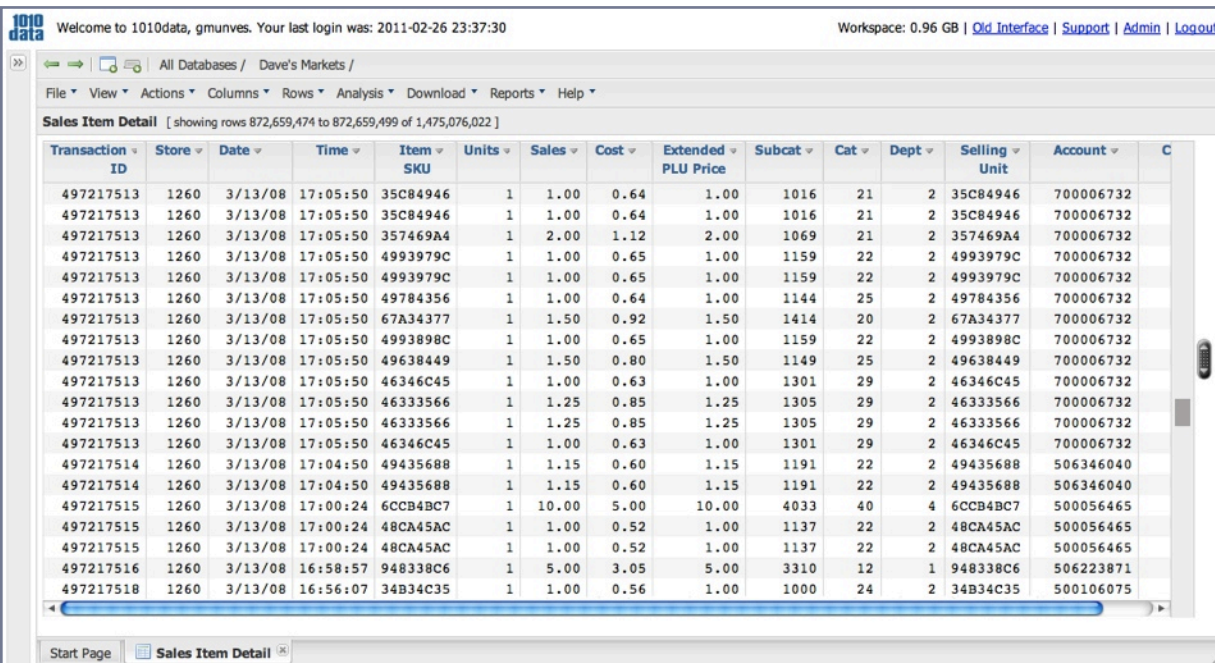
Within 1010data's cloud data center, a customer can either trust 1010data to locate the database or, at extra cost, have its own hardware so that none of its resources are shared.

Another possibility is that the customer simply purchase data and analytical capability as a service. There are a number of generally useful databases that 1010data hosts, particularly databases of financial data, which many customers use. In those circumstances, 1010data is making that data available as a service on behalf of its owner and the customers are licensees that have bought the right to use the data and 1010data's analytical capabilities. Many such licensees also run 1010data on their own data, because they wish to analyze their own data in conjunction with the data that they have licensed access to.

Finally there are some customers on behalf of whom 1010data runs as a private licensing arrangement, where the customer, a retailer for example, makes their sales data available as a service to some of their suppliers. This is similar to the licensing arrangement described above, but the customer controls who is able to use the data and can thus vary the licensing terms to suit its own business priorities.

## 1010data: The Interactive Analytics Capability

1010data's interactive analytics capability is the heart and soul of the technology. Its mode of operation was conceived from the start to involve many users, each interactively analyzing data displayed as a two dimensional table. Users would explore and analyze their data, applying one operation after another. For example, they might sort the table into a different order, or add one column to another, or create a new column in the table calculated from two or three other columns. They might join such data to another table using the values in a common column or they might eliminate some rows of the table using a filter. In short, they would manipulate the data interactively.



Welcome to 1010data, gmunves. Your last login was: 2011-02-26 23:37:30  
Workspace: 0.96 GB | [Old Interface](#) | [Support](#) | [Admin](#) | [Logout](#)

All Databases / Dave's Markets /

File View Actions Columns Rows Analysis Download Reports Help

Sales Item Detail [ showing rows 872,659,474 to 872,659,499 of 1,475,076,022 ]

Transaction ID	Store	Date	Time	Item SKU	Units	Sales	Cost	Extended PLU Price	Subcat	Cat	Dept	Selling Unit	Account	C
497217513	1260	3/13/08	17:05:50	35C84946	1	1.00	0.64	1.00	1016	21	2	35C84946	700006732	
497217513	1260	3/13/08	17:05:50	35C84946	1	1.00	0.64	1.00	1016	21	2	35C84946	700006732	
497217513	1260	3/13/08	17:05:50	357469A4	1	2.00	1.12	2.00	1069	21	2	357469A4	700006732	
497217513	1260	3/13/08	17:05:50	4993979C	1	1.00	0.65	1.00	1159	22	2	4993979C	700006732	
497217513	1260	3/13/08	17:05:50	4993979C	1	1.00	0.65	1.00	1159	22	2	4993979C	700006732	
497217513	1260	3/13/08	17:05:50	49784356	1	1.00	0.64	1.00	1144	25	2	49784356	700006732	
497217513	1260	3/13/08	17:05:50	67A34377	1	1.50	0.92	1.50	1414	20	2	67A34377	700006732	
497217513	1260	3/13/08	17:05:50	4993898C	1	1.00	0.65	1.00	1159	22	2	4993898C	700006732	
497217513	1260	3/13/08	17:05:50	49638449	1	1.50	0.80	1.50	1149	25	2	49638449	700006732	
497217513	1260	3/13/08	17:05:50	46346C45	1	1.00	0.63	1.00	1301	29	2	46346C45	700006732	
497217513	1260	3/13/08	17:05:50	46333566	1	1.25	0.85	1.25	1305	29	2	46333566	700006732	
497217513	1260	3/13/08	17:05:50	46333566	1	1.25	0.85	1.25	1305	29	2	46333566	700006732	
497217513	1260	3/13/08	17:05:50	46346C45	1	1.00	0.63	1.00	1301	29	2	46346C45	700006732	
497217514	1260	3/13/08	17:04:50	49435688	1	1.15	0.60	1.15	1191	22	2	49435688	506346040	
497217514	1260	3/13/08	17:04:50	49435688	1	1.15	0.60	1.15	1191	22	2	49435688	506346040	
497217515	1260	3/13/08	17:00:24	6CCB4BC7	1	10.00	5.00	10.00	4033	40	4	6CCB4BC7	500056465	
497217515	1260	3/13/08	17:00:24	48CA45AC	1	1.00	0.52	1.00	1137	22	2	48CA45AC	500056465	
497217515	1260	3/13/08	17:00:24	48CA45AC	1	1.00	0.52	1.00	1137	22	2	48CA45AC	500056465	
497217516	1260	3/13/08	16:58:57	948338C6	1	5.00	3.05	5.00	3310	12	1	948338C6	506223871	
497217518	1260	3/13/08	16:56:07	34B34C35	1	1.00	0.56	1.00	1000	24	2	34B34C35	500106075	

Start Page Sales Item Detail

Figure 2. 1010data's User Interface

A screen shot of the current 1010data interface is shown in *Figure 2* above. The user is always looking at the latest set of results. He or she works incrementally, doing one thing then another, possibly undoing things that didn't work out. Every time they do or undo something

the results swiftly appear before their eyes. Because of the power of the underlying database, the user rarely waits for results. In practice, it is similar to manipulating a spreadsheet, except for one thing; the table may have many billions of rows of data.

## The Query Language

1010data was designed to deliver a practical interactive analytical capability to one of the most demanding IT application areas there is: financial trading. Consequently, it was built with all the analytical functions that a financial trader might want to apply to bond data or stock data or commodity data. The service was quickly adopted by the traders themselves; in banks, investment houses and hedge funds. The typical user of 1010data has analytical skills and can understand and make use of 1010data's analytical query language productively.

An example of the kind of queries that a 1010data user might apply is shown in *Figure 3*. This is, in fact, the 1010data query that gave rise to the result set shown in the screenshot in *Figure 2*. If you are familiar with XML, you'll notice that the query language is XML-based. You may even guess what various lines mean. For example, `<sort col="month" dir="up"/>` is clearly a "sort ascending" directive.

This example is of a compound query that has been incrementally built up by the user using commands from menus or pop-up windows, and queries are often built in that manner. However it could have been written directly as a query, with the user employing the language directly. When users become familiar with the language some prefer to use it in that way, as they find it faster. Nevertheless, they still tend to work in an interactive manner, gradually building up a query and viewing the results as they do so.

## The Report Wizard

In analyzing data, the typical 1010data user will build queries that they wish to save and share with other users. For that reason 1010data has included a report wizard feature that allows users to save a query and then parameterize it for the benefit of other users.

Consider, for example, a table that contains lists of items bought from a chain of retail stores across the USA. Every item is listed with reference to; *the shopping basket, the store, the city, zip*

```
<note type="base">Applied to table: daves.item</note>
<note type="link">The following link is to table: All
Databases/Dave's Markets/Product Master</note>
<link table2="daves.products" col="sku" col2="sku"/>
<willbe name="month" value="yrmo(date)"
format="type:monthshortdash4y"/>
<tabu label="Total Sales by Subcategory, Month"
breaks="deptdesc,classdesc,month">
  <col source="sales" fun="sum" name="sales"
label="Sum of Sales"/>
</tabu>
<willbe name="prev_sales" value="g_tshift
(deptdesc,classdesc;;month;'M';sales;-1)"/>
<willbe name="perc_change" value="100*(sales-
prev_sales)/sales"/>
<sel value="(deptdesc='accessories')"/>
<sel value="(contains(classdesc;'winterwear'))"/>
<tabu label="Monthly Changes in Sales of Winter
Accessories by Subcategory" breaks="month"
cbreaks="classdesc" clabels="short">
  <col source="perc_change" fun="first"/>
</tabu>
<sort col="month" dir="up"/>
```

*Figure 3. 1010data's Analytical Query Language*



code and the state. Additionally, the date and time, mode of payment, price of each item and indicators of special offers taken are shown.

An analyst discovers from analyzing this sales data that certain groups of products always seem to be purchased together. So he or she defines the product group and creates a report that highlights this effect, showing the product grouping and the percentage of the total shopping basket cost the group of products represents.

If the report proves useful, perhaps to the point where the retailer chooses to run specific promotions for the product group, then this query can be saved for reuse as the promotion goes into effect. With its report wizard, 1010data provides the ability to set up some of the data columns as parameters, so that users of a saved report could run the report just for a specific state or for a range of zip codes, or for specific days of the week. Setting up such reports is simply a matter of using the wizard to select which items to use as parameters.

## Functions and Capabilities

The analytical usefulness of 1010data can be seen by examining the various functions it provides to operate on the columns of a table. It helps to think of this collection of functions as being a simple analytical programming language that is used in a step-by-step fashion; one instruction at a time. It enables the user to create new columns in their table, using values in existing columns of the table. A simple example might be: *calculate the value in a new column as being the product of two other columns, to be calculated only if the value in a third column were greater than 100.*

1010data has over 200 built in functions and operators in the following categories:

- Comparisons and Mathematical Operators
- Time and Date Functions
- Financial Functions
- Grouping Functions
- Mathematical and Statistical Functions
- Time-Series Functions
- Text Functions

The first three of these categories are virtually self-describing, containing the kind of functions and operators you would naturally expect, but the final four are worthy of some comment.

### **Grouping Functions**

1010data provides a variety of ways to segment the rows within the table you are working on and then apply statistical and mathematical functions to each segment. In practice, the grouping functions correspond to SQL sub-queries, except that they are far less constrained - in fact unconstrained. The parameters for each such function include a "group by" parameter (for example, group by zip code) and an "order by" parameter (for example, order by age) then a function, such as a running total can be applied. The grouping functions include many sophisticated statistical capabilities and time-related capabilities.



### **Statistical Functions**

The statistical functions include much of what any analyst would expect (averages, modes, standard deviations, etc.). They also include functions that enable regression analysis, which can be used in a variety of ways to explore causal relationships between multiple variables, make sophisticated forecasts and build predictive models. 1010data has been used extensively in the financial sector to model trading situations, for example, when a large number of limit orders may be triggered. Such modeling activity would defeat many analytics products.

### **Time-Series Functions**

The 1010data time-series functions enable analysis on the basis of time or sequence. For example, there is a function that allows you to look at values of a column offset by a specific number of rows or a specific time period. This allows analysts to look for possibly correlated events in a set of data. They might, for example, be looking at commodity prices and stock prices, trying to determine how much the rise in price of a particular commodity (*such as coffee*) impacted the movements of a particular stock price (*Nestlé, perhaps.*) Alternatively a telecom company might be examining information feeds from a sporting event, trying to determine what actions on the field led to sports fans sending SMS messages on their cell phones.

### **Text Functions**

The text functions that 1010data provides are character string functions that allow you, for example, to search a text string for the presence of a specific word or combination of words. They are not semantic functions that try to divine the meaning of text, although it does include some Soundex functions to identify names that sound similar and hence may be misspellings.

Nevertheless, the text functions could be used, for example, to sift text messages, news feeds or tweets to identify those that comment about a particular company or product or stock and you could use these capabilities to identify negative or positive sentiment. Or it might be used to simply correlate tweets that mention a given stock with price movements in that stock.

### **Areas of Application for 1010data**

1010data's natural areas of application are anywhere where the speedy analysis of large collections of data will deliver value. Industry sectors where such activity can make a significant contribution naturally include: telecommunications, banking, financial trading, insurance, utilities, retail and government. It also includes all areas of Internet activity from ecommerce to gaming and social networks, where large amounts of data are gathered on a daily basis and can be usefully analyzed.

Taking a different tack, we can think of 1010data's area of application in terms of general business activities, where analytics is already part of the process or where in the future it is likely to bubble up, such as:

1. **Risk Management:** Risk management has, in recent years, grown from being an activity that was associated with banking, where risk was always the nature of the business, to almost any area of commerce from pharmaceuticals to energy that involves large investments and/or government regulation. Investment by its very nature, is a risky activity; so where there is large investment there is commercial risk.

Risk management virtually demands the analysis of large amounts of data in order to quantify and minimize risk.

2. **Sales and Marketing:** This has always been an application area for analytics. In recent times, computers have become powerful enough to analyze sales at a customer level on an item-by-item basis. Now, this is even the case in the telecommunications industry where the basic customer purchase is a single telephone call or text message. Because of its big data capability, 1010data is obviously suited to this area of activity.
3. **Security:** This is another area where the volume of data has, until recently, defeated attempts to do analysis in a timely manner. In computer networks security breaches nearly always involve anomalous behavior. And that's also true of credit card fraud. Indeed, it's true of most instances of commercial fraud, money laundering and other nefarious activity.
4. **Machine Generated Data:** This is an area of extraordinary data growth in recent times, whether you look at computer gaming, network activity in a computer network, RFID data or just usage logs on web sites. 1010data is relevant here by virtue of speed of analysis and the fact that it can handle very large amounts of data.
5. **Data as a Service.** There are already quite a few companies in many sectors of the economy whose business is in gathering data and making it available as a service. 1010data has the advantage that it assists both in the managing and distributing such data and in providing a wealth of analytics capability that can be extended to any customer.

## Interactive Analytics on Big Data

It helps to contrast 1010data with other products that also provide analytical capabilities. Earlier we summarized the set of analytical functions that 1010data provides. It is very comprehensive, delivering greater capability than is found in many other analytics products. However, even were this not so, 1010data would be notable in being the only product or service that is currently able to provide interactive analytics on large volumes of data.

Technically, there are three approaches to analyzing large volumes of data:

1. Analysis using a high performance Analytical Database
2. Analysis of data subsets using interactive Analytical Software
3. An Interactive Analytical Database as provided by 1010data

We illustrate each of these possibilities in *Figure 4 (next page)* in order to compare them. There are two aspects to consider with each of these possibilities. The first is how the user experiences the capability. The second is the work that is involved to make such a capability available. Both are important.

The top diagram (*approach 1*) shows, in overview, how analytical databases are used. The analytical database will be accessible using SQL, which has been extended in some way to accommodate analytical functions along with the data selection and data manipulation directives of SQL. The user might be using some analytical software that has been tailored to work with the analytical database, or they might simply be writing the query themselves. In either case, the query has to be constructed to deliver a specific answer. The database will be built to respond quickly, but will not have been specifically built to respond at interactive

speeds. Even if it does respond very quickly, the analysis tool (or the query interface) will not be interactive. The user may now proceed by narrowing down the question and resubmitting the query. If so, the database will repeat much of the work it just did in getting to the previous answer.

Because the interface is not interactive, the user is more likely to create a batch of queries that they'd like answered and submit them all at once. When they receive the answers, they might then refine some of those queries and proceed. The way of working is different.

In order to support this kind of capability, the IT Department will need to have prepared the analytical database and its associated data feeds. This work can be extensive including data modeling and data tuning activity.

The second option (*approach 2 in Figure 4*) gives the user a truly interactive interface. All that needs to happen to enable the capability

is to extract a subset of appropriate data from a database or data warehouse that holds a much larger collection of data that the user may be interested in. In this situation, the IT Department will need to organize for the extraction of the data and its subsequent refreshing. The downside for users is that there is a limit to the amount of data they can analyze in any given session. So where they need to analyze larger volumes, they are forced either to work in a batch manner (*approach 1 in Figure 4*) if the capability is there, or they are forced to analyze the data in a piecemeal fashion.

1010data (*approach 3 in Figure 4*) enables the user to be interactive no matter how large the volume of data and demands almost no effort from the IT Department. For the user it is dramatically faster, partly because the underlying database has been built for speed and partly because the database works on intermediate results, doing one thing at a time. Because the user isn't forced to use a limiting query language like SQL, they can naturally mix data manipulation activity with analytical functions. They can save their result set at any point and experiment with it in different ways. They can even work collaboratively passing intermediate results to others. The approach is highly flexible.

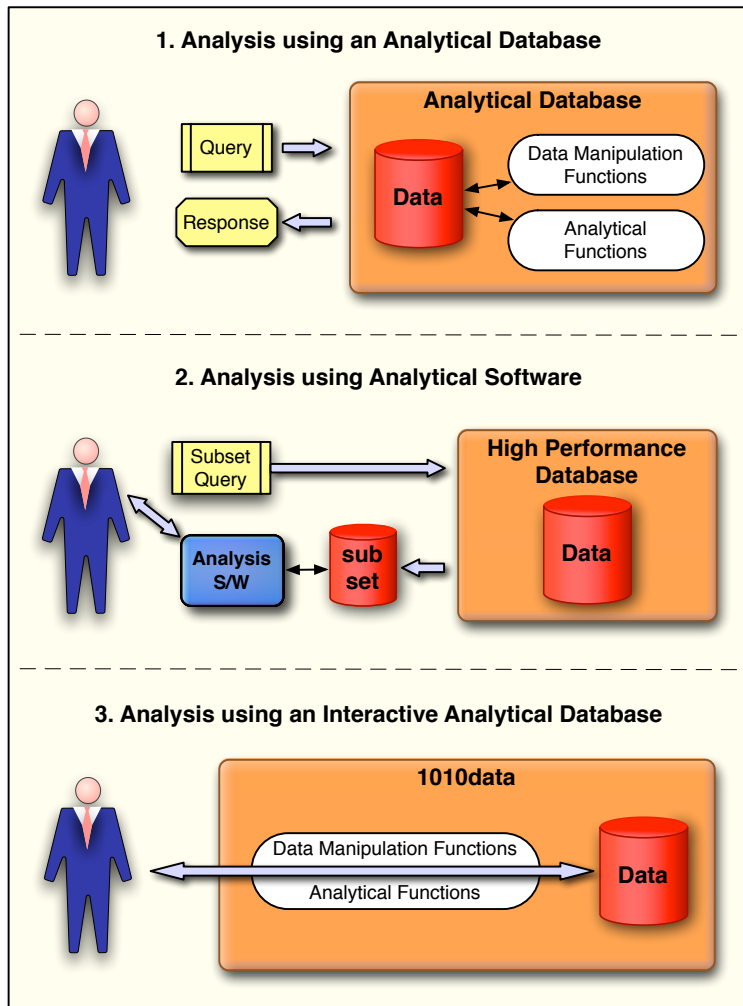


Figure 4. Analytical Approaches Compared

## 1010data Database Performance

1010data has beaten database products such as Exadata and Netezza in customer database benchmarks. This may seem surprising, given the high performance reputation of such products, but in a way it is not. When Steier and Kaplan first built interactive analytical systems, they built data stores based on APLDI, a specialist mainframe database from IBM, which stored data in a columnar manner, just like the many recent BI database products. When they founded 1010data, they built their own column store database to guarantee the performance that their application required.

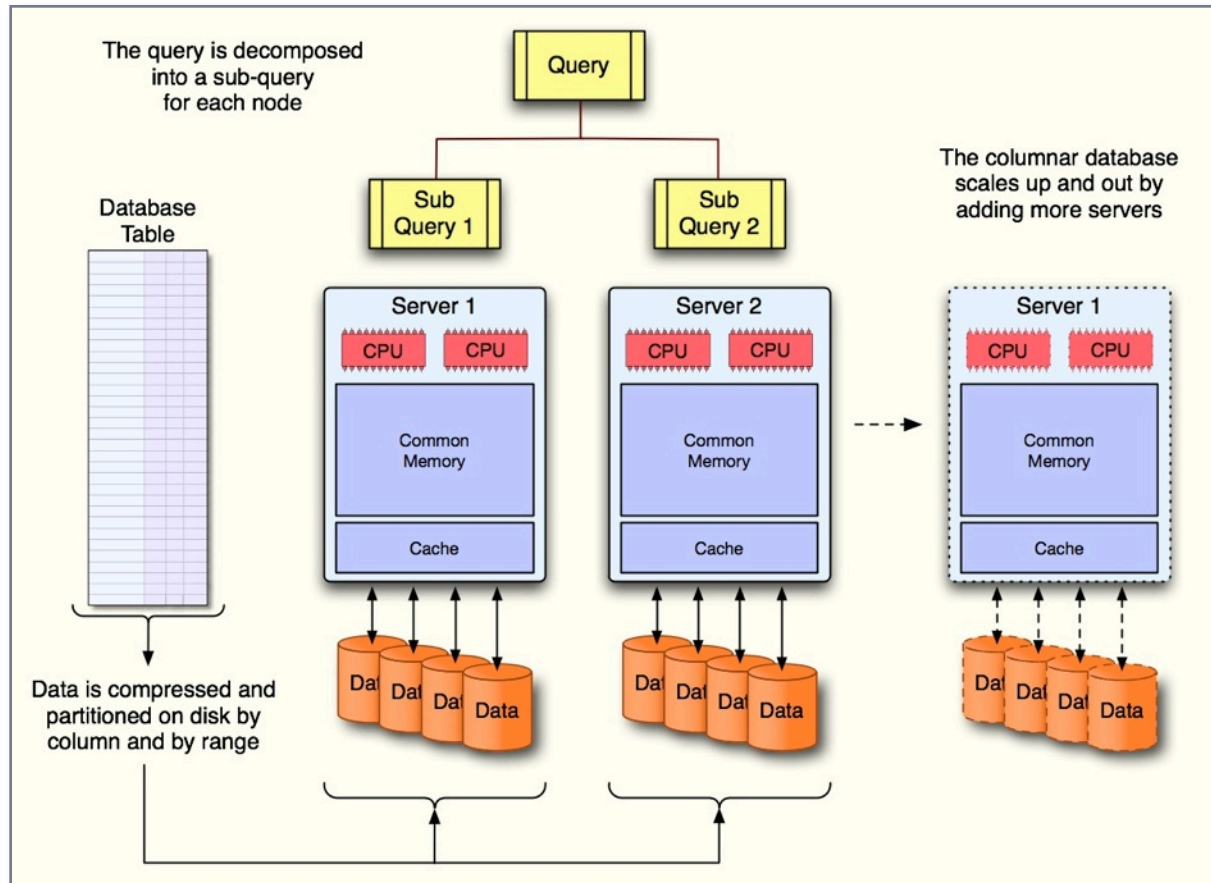


Figure 5. Query Decomposition In A Typical Column Store Database

The way 1010data works, and indeed the way that most column-store databases work, is illustrated in *Figure 5*. Two specific performance techniques are employed. Database tables are partitioned into columns and each column is distributed across multiple disks and multiple machines. When a query is processed, the query is resolved into sub-queries, one for each server based on the location of data and hence can run in a highly parallel manner.

Additionally, data in each column is compressed dramatically using various compression techniques, such as tokenization. With 1010data, a very high rate of data compression can be achieved. Column-store databases generally don't require indexes, although some databases, including 1010data, provide it. For the sake of performance, data is extensively partitioned and read serially from disk. It is then cached in memory, for quick access.

All column-store databases use similar parallelization, data partitioning and data compression techniques to achieve good performance, although they vary in the efficiency of its implementation. However, the way such products manage the distribution of processing varies. 1010data's approach to distributing its processes, illustrated in *Figure 6*, may be unique to the product because it focuses on enabling interactive use.

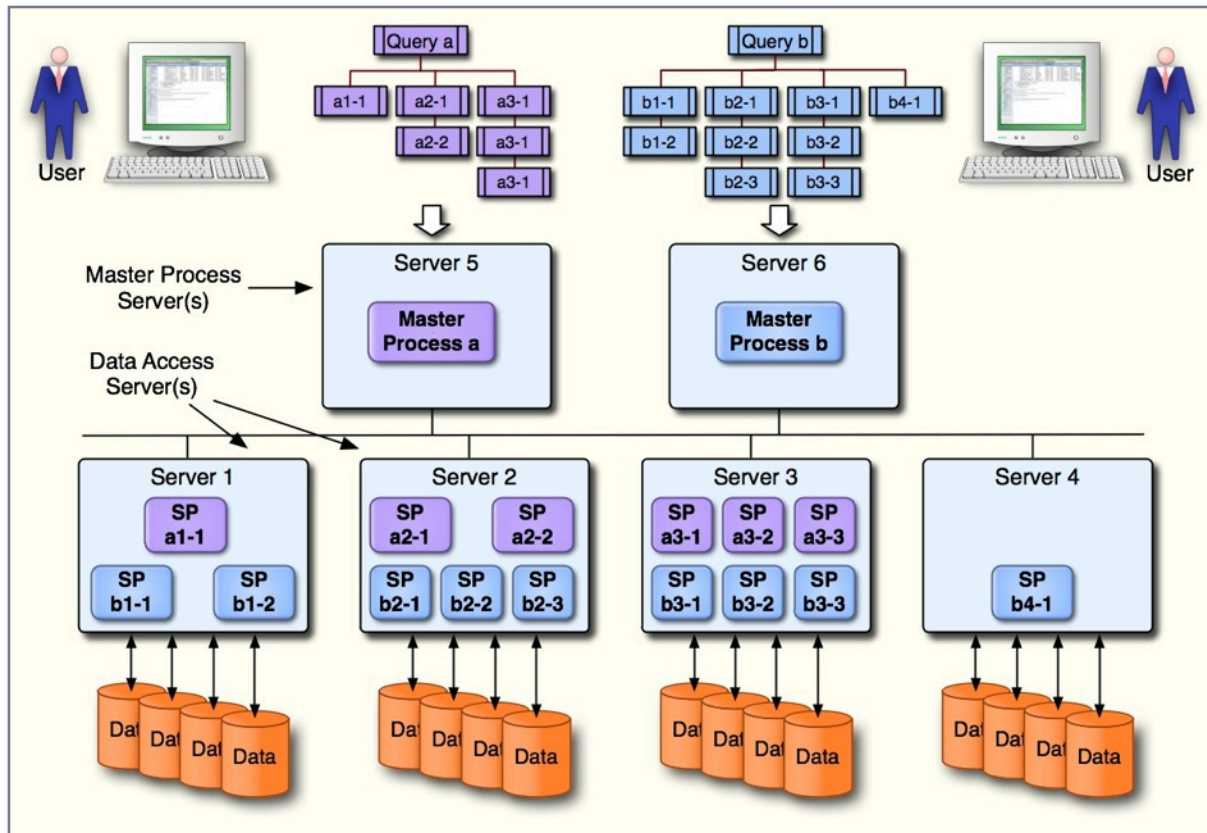


Figure 6. 1010data's Organisation of Processes

It works in the following way. Each database user fires up a *Master Process*. (The user might be a person using the interactive interface or simply a process sending a SQL query to the database.) The master process manages the resolution of all queries sent by its user and can run on a separate server (as illustrated) although in most implementations, it doesn't.

The master process doesn't simply split the query up between servers, but, as *Figure 6* illustrates, it segments the query according to its knowledge of where the relevant data for that query resides and the best plan in respect of processing efficiency for retrieving it. As such it may choose to initiate multiple processes on each data access server.

So in the illustration, *Master Process a* fires up *Slave Process a1-1* on *Server 1*, *a2-1*, *a2-2* on *Server 2* and *a3-1*, *a3-2*, *a3-3* on *Server 3*. Similarly, *Master Process b* fires up 2 *Slave Processes* on *Server 1*, 3 on *Servers 2* and *3* and 1 on *Server 4*. In reality, of course, the situation gets far more complex than shown in the illustration. The point to note is that the 1010data architecture is highly parallelized. Users run a single query at a time, and can run multiple simultaneous queries if need be, by using another ID and master process. Each query runs in parallel across multiple servers and then runs in a parallel manner within each server.



1010data normally deploys on commodity servers with 2 CPUs and as many cores per CPU as is currently being delivered on such hardware. So, in effect, queries are parallelized across servers, CPUs and CPU cores.

### **The Resilience of 1010data**

The first point to note is that 1010data normally runs within its own data center, which is built on a highly reliable cloud infrastructure. Servers are all run with RAID disks for reliability so that actual server failure is rare.

However, when any failure does occur, whether through server failure, infrastructure failure or software failure, 1010data is built to swiftly recover. A high level of resilience is achieved through data duplication, mirroring data in a manner similar to the way that RAID configurations mirror data across multiple disks. So in the event of server failure, the same data is available on other servers.

What happens is that the current state of the user's session is saved, so the only thing that may need to be recomputed is the current query, if it was interrupted. When such a failure happens the user receives an error message and can, with the click of a mouse, resubmit the query. The current master processes that were running slave processes on the failed server simply fire up new slave processes on the other servers that hold the necessary data and continue as before.

When a master process fails, the session is lost and the user will have to redo the whole session. However the software provides a save capability, which allows the user to save a session. So if the user has made use of that, recovery will start from the last saved state.

### **The Scalability of 1010data**

Scaling is achieved by adding servers, and this can be done without taking the application down or even pausing it. Just as *Figures 5 and 6* suggest, there is no "choke point" in 1010data's architecture. The only shared processes are management processes, which can themselves be replicated. The architecture is therefore a "shared-nothing architecture" that is highly scalable over many servers.

Adding data access servers naturally improves query performance as it increases available CPU power, available memory and disk i/o channels all of which work in parallel. If there are data hot spots which are frequently accessed, the bottleneck can be removed by data replication and the workload managed accordingly - but the need rarely arises.

Normally, 1010data is implemented on commodity servers with two CPU chips and some extra memory (currently that's either 72 gbyte or 144 gbyte). Because of the very high level of parallelism, increasing the number of commodity servers almost always proves to be the best way to scale up or to improve performance. Using specialized servers with, for example, extra memory and memory channels almost always proves to be a more expensive solution than simply adding inexpensive server resources.

At the time of writing, the largest 1010data installation runs on several hundred servers presiding over several trillion rows of data.



## Updating The Database

Data update works in the following way. An updating process runs regularly (or continually) feeding additional data into the database. This runs independently to all other processes, so master processes are unaware of the new data being added. The effect of this is that the data a user is working on never changes for the life of the user's session. This is the case even if a user's session lasts longer than 24 hours a rare event, but 1010data allows for it.

So from the user's point of view the data is refreshed at the start of a new session when a new master process is created. However, from the database's perspective, the database is being updated all the time.

The situation sometimes arises that the user wants to know when a major refresh of the data has occurred. This might happen, for example, where 1010data is being updated with a full day's data at a specific time of day. To cater to this, 1010data displays a System Message Board that reports on such updates. Users could also simply create a new session and examine the last record added - as long as they remember to close the session if an update has not happened.

## 1010data's Data Philosophy

Finally, to best understand 1010data, it helps if you realize that the technology implements an approach to data that is different from approaches employed by most other analytical databases. This unique approach is easy to understand and it makes an important contribution to the overall functionality of the product.

Part of this data philosophy can be summarized with the following statements:

1. All data records are snapshots of state and hence can be time-stamped.
2. Almost all data is ordered (by time if by nothing else.)
3. Data updates delete information and hence should be treated as "additional information."

## Snapshots of State

To expand on the first of these ideas, it's a simple fact that everything, large or small, is in flux to some extent. When computer systems record information about something; a customer, a product, an invoice, the data can be classified as one of two things:

1. **A State Description of an Entity:** The data relates to the description of some entity (customer, product, etc.) at a specific point in time. For example, a customer currently has the name "John Doe" and lives at "1001 Main Street, Somewheresville." As the entity changes, so the description changes. So this information record is simply recording the current state. It therefore can be time-stamped.
2. **An Event Description:** The data relates to an event of some kind, for example, the purchase of some goods. As such, the time of the event is important because it collects together multiple items of information about the state of entities involved in the event. For example, with a purchase; *price*, *discount*, *delivery address* are all data items that may change in time, but had that specific value for that specific event. For that reason this information too should be time-stamped.



You need only consider the impact of not recording the time of some events to understand why time is so important. Let's say, for example, that a specific stock has a price of \$2.05. We have also recorded it as having a price of \$50.24 and \$20.14. It is impossible to imagine anyone writing an application to manage stock prices and not recording the time of the prices. Without information about time, the data tells you almost nothing other than that the stock price has not been constant. Those data snapshots could even be 30 years apart or out of order. Add a timestamp and the data suddenly has useful context.

Applying time-stamps to data records is obviously important, and yet many databases do not enable that as a matter of course. 1010data does. It does not insist on time-stamping records because with some entities change is so slow, that a single record rather than a time series is practical and less confusing for the user. Nevertheless, if records are delivered to the database without a time stamp, 1010data can apply one based on arrival time and it can subsequently assume that the records are ordered by their order of arrival and by time.

### **The Ordering of Data in 1010data**

The tables held by 1010data are usually ordered and stored in order by time. Consequently, what is the previous record and what is the subsequent record is usually known. The reason we draw attention to ordering here is that the Relational Theory of Data, which guided the whole relational database movement for years, insisted that data had no inherent ordering, and should not have. Order could be imposed by indexing the data or by sorting it, but the idea was that the data itself should be independent of order.

This odd and clearly incorrect idea caused problems for many years, particularly with data whose order by time was important. This became clear with the advent of Complex Event Processing, which sought to analyze data in event streams, and find useful correlations (often time-dependent correlations) between events. Relational databases are inadequate for such applications and so entirely new time-based data stores were invented for the purpose. Since then there have been a plethora of new databases that no longer adhere to the artificial restrictions of Relational Theory. 1010data is one of these.

### **Data Updates**

Where 1010data stores data in time order it treats updates as additions of new data and treats deletions as notifications that "a record ceased to be valid at a specific time." There is a sound analytical reason for adopting this approach.

Consider the impact of "data corrections" in a BI database. Sometimes the value of a given item in a record at a given time is simply wrong. Perhaps a birth-date was recorded as 3/12/1971 when it should have been 3/12/1991. If such data applied to, say, an insurance policy, then for a period of time, the system could be over-estimating the risk of mortality for that individual. It is even possible that a bad business decision has been made because that data was incorrect.

1010data is an analytical database, not a transactional database, so while it is clearly important for the birth date to be accurate in a transactional system, in an analytical database it is important to know that the data values changed and to know the time period during which the first value applied.



## No ETL

1010data is not a data warehouse in the usual sense, so cumbersome data warehousing practices are unnecessary. While it is true that a company might choose to load a great deal of its data into 1010data for analysis purposes, it would not require involvement in ETL (Extract, Transform, Load) activities or data modeling to assemble the data. It might extract the data from an existing data warehouse or it might source it from operational systems, or even from several different places, but there would be no need for ETL or specific data quality procedures or a data modeling effort. As a consequence, little effort is required before users can engage with the data.

## NoSQL

1010data could be classified as a “NoSQL” database. The term “NoSQL” can be confusing because it does not mean that a database doesn’t use SQL, it stands for “Not Only SQL” implying that the database supports SQL (at least to the level of ODBC and JDBC) but also supports other ways to access data.

There are a variety of databases that fall into the NoSQL category. Some are object databases, some are graph databases and some are document stores. Yet others are key-value stores, like Google’s BigTable that tend to be associated with the use of Hadoop and MapReduce. 1010data isn’t at all similar to any of these kinds of database, but it is a NoSQL database.

1010data’s query language is fundamental to the technology and to the way it is used. It could even be argued that the 1010data query language is the heart of the technology and that the high-performance database is a necessary but secondary component required to support the application of the language.

The use of SQL is so pervasive among BI tools vendors that it was natural for 1010data to provide such products with an interface that parses SQL and translates it into the 1010data query language to execute it. By virtue of that, a company is not obliged to sacrifice its legacy of SQL code if it wishes to consolidate data within 1010data. It was serendipitous that queries submitted through 1010data’s SQL interface proved to deliver exceptional performance. It’s valuable and in a sense it’s interesting, but it’s a side show. 1010data is first and foremost an interactive analysis technology.

It is also important to understand that SQL itself was never designed for interactive use in the way that 1010data’s analytical query language was. It includes only data manipulation functions and no analytical function. In order to use it for analysis, product vendors have to engineer around its limitations, in a tortuous attempt to try to ensure that analytical functions and data manipulation functions have equal priority. 1010data circumvented this problem entirely by developing an analytical query language that subsumed SQL.

## In Summary

1010data delivers an interactive analytical capability that can be quickly implemented to access and analyze huge amounts of data, up to trillions of rows. It has a broad set of analytical capabilities and consequently it has broad areas of application, beyond the normal range of most analytical BI products. We recommend that companies looking for an analytical platform, especially those that gather, accumulate and analyze large amounts of data, consider it as an option.



### **About The Bloor Group**

The Bloor Group is a consulting, research and technology analysis firm that focuses on open research and the use of modern media to gather knowledge and disseminate it to IT users. Visit both [www.TheBloorGroup.com](http://www.TheBloorGroup.com) and [www.TheVirtualCircle.com](http://www.TheVirtualCircle.com) for more information. The Bloor Group is the sole copyright holder of this publication.

□ 22214 Oban Drive □ Spicewood TX 78669 □ Tel: 512-524-3689 □

[www.TheVirtualCircle.com](http://www.TheVirtualCircle.com)

[www.BloorGroup.com](http://www.BloorGroup.com)