Analytic Databases in the World of the Data Warehouse

April 2009

A White Paper by

Dr. Barry Devlin, 9sight Consulting
barry@9sight.com

The majority of companies have implemented their business intelligence (BI) environments according to a physically layered data warehouse architecture and based on traditional general-purpose relational databases.

Specialized analytic databases using technologies such as columnar orientation, massively parallel processing and other techniques have now emerged. Such new DBMSs now offer significantly improved performance for typical BI applications, enable previously impossible analyses and often lower cost implementation.

They also have the potential to challenge the current physically layered Data Warehouse architecture. This paper reexamines the trade-offs that have been made in the layered architecture and argues that analytical databases may enable a move to a simpler non-layered architecture with significant benefits in terms of lower costs of implementation, maintenance, and use.

Contents

BI ThoughtLeader™
by
PARACCEL

The World of the Data Warehouse

In the rapidly changing world that is IT, the data warehouse (DW) might well deserve the title "venerable". The current, widely-accepted architecture first emerged in the late 1980's and has since proven to be remarkably resilient. This points to a set of solid founding principles, but also to the fact that the underlying relational database technology has been relatively stable over the intervening period.

The *founding principles* of the data warehouse are:

1. ***Separation of operational and informational environments:*** For historical reasons, operational data cannot be guaranteed to be consistent, clean or complete. Furthermore, *ad hoc* querying of operational data raises performance, contention and security issues.
2. ***A conceptually single Data Warehouse:*** A complete, generalized, historical view of information is needed for consistent and reliable business intelligence.
3. ***Managed data quality:*** Uniquely sourced data is well-defined and documented in its meanings and manipulation and strictly controlled in its distribution.
4. ***Subsetting of the Data Warehouse:*** Many users require access only to subsets (also known as *data marts*) of the full data warehouse based on ease of use, the need for specialized tools, sand-boxing, security, and other reasons.

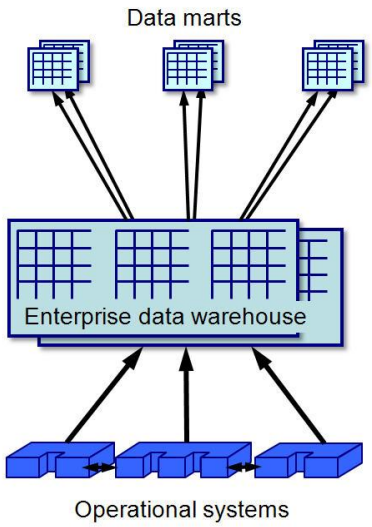

Figure 1:

Data Warehouse Logical Architecture

The resulting *logical* architecture is shown in figure 1. Data is extracted from the operational systems, cleansed, reconciled and stored in a subject-oriented and historical database—the *enterprise data warehouse* (EDW)—aligned to an enterprise-wide data model. To minimize the number and complexity of feeds from the data sources, the number of EDWs is kept to a minimum, ideally one. Subsets of this data, optimized for particular uses, either predefined or *ad hoc*, are fed into data marts for use by decision makers in various tools. Data flows unidirectionally through the environment in order to assure maximum control of data consistency and quality.

From a technology viewpoint, this logical architecture is implemented largely on a relational database platform with physical instantiation of separate EDW and data marts and physical movement of data via ETL (Extract, Transform and Load) tools between the different layers. In such implementations, the EDW is a large relational DBMS, structured close to 3rd normal form and optimized for loading, cleansing and reconciliation of large quantities of operational data. Data marts are implemented in many cases in relational DBMSs, often on a different platform than the EDW, and optimized for rapid response times to end-user queries. Data marts also exist in a variety of other technologies such as spreadsheets for user flexibility or specialized tools for specific types of analysis.

To support more current data needs, a further structure—the *operational data store* (ODS)—has been introduced, either as an additional layer beneath the EDW or as another component of the middle layer. In addition, although excluded from the logical architecture, *independent data marts*—fed directly from the operational systems bypassing the EDW—are often built for performance reasons, especially for very large datasets where near real-time access is required.

While the layered DW architecture is the basis for most successful BI implementations today, recent developments in business and technology are beginning to stretch the architecture. On the business side, increasingly rapid decision-making needs and changes in those needs are restricted by the physical layering and distribution of data. And, as discussed further below, developments in database technology, and to a lesser extent in ETL tools, have the potential to disrupt the physically layered implementation approach currently favored.

What is an Analytic DBMS?

Simply put, an analytic DBMS is a relational database that is optimized for analytic use. However, this simple statement deserves deeper investigation. What is meant by analytic usage? How is optimization achieved? And, what are the consequences, both positive and negative of how this optimization is done?

Analytic tasks are generally read-only to "read-mostly" but exhibit very different characteristics depending on the needs of the user performing them and the industry involved. The size of the dataset involved—both width (number of fields) and depth (number of records)—ranges from small to enormous. Query flexibility also varies widely. In many cases, the query is well-defined and its data needs fixed; at the other extreme is the need for *ad hoc* access to any of the company's data. Another axis of variability is the timeliness of the data needed, ranging from monthly and beyond (strategic BI) to near real-time (operational BI).

Even a brief review of the analytic database market reveals that the focus is very clearly on driving query performance at the high end of each of these scales: large datasets, high flexibility and, somewhat less often, near real-time. This focus has led to a number of optimization approaches, used separately or together, at the physical hardware or database organization levels. These approaches include:

- ***Massively Parallel Processing (MPP):*** splits the data into subsets and runs queries in parallel on multiple loosely coupled processors / disks to increase processing power
- ***In memory processing:*** stores the data in RAM rather than on a hard disk to take advantage of much lower access times of memory
- ***Columnar database storage:*** physically stores data in column sequence (rather than the traditional row sequence) to minimize the amount of data read from disk (or memory) in typical analytic queries
- ***Compression:*** again reduces disk or memory I/O and is particularly effective when used in conjunction with a columnar data arrangement

Among the vendors in this market, many of the more established vendors (in this case, "established" meaning more than 4-5 years old) focus almost exclusively on the MPP approach, often offering it as an appliance—a pre-built hardware and software package that can be rolled into the data center and plugged in. More recent entrants use a combination of some or all of the above methods. ParAccel, for example, describes[1] the use of all four approaches (and a few more) to provide highly optimized performance.

Performance and price/performance measures for analytic databases in TPC-H benchmarks and in quoted customer examples over the past few years, show gains of at least 2X to 10X, and in many cases considerably more, over more general-purpose databases. While any decision to invest in an analytic DBMS must, of course, be based on wider criteria and include comparative proofs of concept, these figures are impressive.

Of course, performance optimization is not new in relational databases. Indexes, materialized views, caching and specialized table designs have been the stock in trade for Data Warehouse designers for 20 years now. However, it is worth noting that these traditional approaches are highly specific to the data model in use and the anticipated query patterns against that data. Tuning and optimization is thus a labor-intensive process that delays initial use of the data and often requires rework as usage patterns change. And, complete *ad hoc* usage cannot be optimized by these means.

In contrast, the optimization approaches described above and used in analytic DBMSs are model-independent and generally applicable to all data and most analytical query patterns. They require minimal manual setup or maintenance, thus lowering DBA involvement and costs over the entire lifetime of the system.

The factors just described, along with the dynamics of entry into a long-established market (which Data Warehousing clearly is) have dictated that analytic databases have gained most traction in a segment best called "analytic applications" as opposed to general-purpose Data Warehouses. The difference is important and relates mainly to the scope of the source data.

An analytic application uses a set of data, however large, that is known to be largely internally consistent and clean. In many cases, this data comes entirely from a single source, or at least from a small set of sources that have been designed to work together. For example, the call detail records (CDRs) in a telecommunications company or the checkout records in a retailer, while voluminous, require only minimal external, mostly fixed, reference data to enable the extensive analysis required by business users. As a result, a "stand-alone" analytic application is possible, and an analytic database, perhaps packaged as an appliance, is an attractive and cost-effective proposition.

Compare this, at the other extreme, to the analytic data needed in a large financial institution that has just merged with or acquired a couple of its former rivals. In this case, analysis must be preceded by a significant level of data cleansing, reconciliation and consolidation. A stand-alone analytic application provides much less benefit initially due to the extent of the analysis and modeling required, and may even fly in the face of necessary efforts to consolidate the database platforms of the merged company. In the medium and even longer term, the bank will continue to need to cleanse and reconcile data from still separate operational sources.

Nonetheless, the performance gains mentioned earlier, together with the lower initial and ongoing query performance optimization costs cannot be ignored by the broader data warehouse community, especially in the light of current IT budget constraints. Furthermore, as business users demand increased flexibility in the queries they run and the ability to analyze data ever closer to real-time, analytic DBMS technology appears to offer intriguing possibilities for a change in the physical implementation of the DW architecture.

Positioning Analytic DBMSs in the Data Warehouse Architecture

As of now, the majority of analytic DBMS implementations have skirted the edge of the traditional Data Warehouse world. As described above, they have been used largely to address specific analytic applications, usually with large volumes of (reasonably) consistent source data. They have typically delivered a high return on investment (ROI) by significantly speeding up query response times or by enabling previously-impossible analyses. Because they have been outside of the data warehouse architecture, these types of implementation, however large or powerful, are clearly independent data marts.

For vendors and end users alike, this positioning is very attractive. Because no attempt is made to align with either the model or content of any existing enterprise data warehouse, implementation and ROI can be very rapid. For the IT department, especially those responsible for the enterprise data warehouse and/or data quality, this positioning is problematical in two key areas. First, the creation of a (typically) large database containing data that overlaps with the content of the EDW but is totally independent of it raises the strong possibility of extensive and long-term inconsistencies in the informational data environment. Second, the independent sourcing of the new data mart from the same operational sources as the existing EDW can give rise to additional loading on the operational systems either during the day or in increasingly tight overnight batch windows.

Previous best practices suggest that these issues should be addressed by migrating the independent data mart to a *dependent* position in the architecture—that is, fed from the EDW. However, given the data volumes involved and the additional latency in the data thus introduced, this approach may not find favor with either the end users or IT. A better solution is needed, and the characteristics of the analytic databases themselves suggest some radical positioning possibilities.

Who Needs an EDW Anyway?

Evidence from existing implementations of analytical databases strongly emphasizes their sustained performance over a wide range of query types on very large data volumes without the need

to create a summarized or subsetted layer of data. The relative weakness of general-purpose relational databases in this area was one of the key drivers of the physical instantiation of data marts as a separate layer in the warehouse as opposed to the logistically simpler use of SQL views on the base data. The speed and power of the analytical DBMS can also reduce the need for specialized data stores for particular types of analysis.

The likely outcome of this approach is shown in figure 2, which shows a number of independent data marts fed in parallel from multiple data sources, with each data mart servicing a distinct audience, perhaps functionally or geographically based. In comparison with prior implementations, the use of powerful analytic databases here would reduce the number of data marts needed, while modern ETL tools and operational systems could ease the problem of the multiplicity of data feeds that often plagued this approach.

However, this option simply does not address data quality and consistency issues that are central concerns for any enterprise-wide data warehouse program.

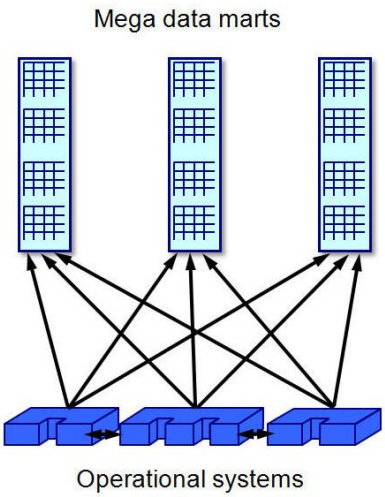

Figure 2: Data Warehouse with Mega Data Marts

A Full Data Warehouse Built on an Analytical DBMS?

Figure 3 has a distinct flavor of "Back to the Future" with a single block standing for the data warehouse component. Early work[2] on the data warehouse architecture envisaged a single level data warehouse; the introduction of two (or more) physical layers in the model emerged later as issues with the performance of complex join queries in general-purpose relational databases arose. However, layering presents some distinct problems itself: data latency is increased and maintenance of the added transforms can be costly as user needs change regularly.

At this stage of their development, analytical databases have shown significant gains in complex query performance on multi-terabyte size databases, potentially allowing a move away from the widespread, institutionalized introduction of a data mart for every set of analysis needs. Unfortunately, this approach does little to reduce many users' predilection for loading warehouse data into spreadsheets (the unmanageable "spreadmart" phenomenon) at the drop of an analytical hat; but at least IT resources can be redirected from data mart creation and maintenance to providing a more managed approach to supporting these users.

In the case of very large or distributed warehouses, one could envisage the need for some amount of duplication of data or replication of subsets of the data based perhaps on geographical location or major functional boundaries in the organization. This would be very much the exception rather than the norm it is today, and it is likely that there are still improvements in query optimization and workload management in the analytic database technology that could further reduce the need for such mirroring within the data warehouse. Where the number of such mirror datasets is small, it would make sense to populate them in parallel with the primary warehouse, as shown in figure 3, rather than in a subsequent layer populated from the warehouse.

Some current analytical database implementations also demonstrate efficient and ACID-compliant (Atomicity, Consistency, Isolation, Durability) loading of data while production queries are running. This is a key requirement for *operational BI*, which requires close to real-time data. This suggests that operational BI needs can also be served by the same data warehouse environment, provided that intra-day data is consistent with the usually better integrated and cleansed overnight data.

As shown in figure 3, another key activity in the data warehouse is data conditioning, such as cleansing and reconciliation, which is a vital prerequisite to end user access. In the layered architecture, data conditioning occurs in both the EDW and in the ETL feeds to the EDW. In some cases, a physical *data staging layer* is introduced in the implementation where such work is performed. In other cases, conditioning occurs mainly in the ETL layer, with an implied sequence of steps where transform (or conditioning) precedes loading. In the same way that analytic databases cause us to question the need for layering with the user access side of the warehouse, they also allow us to revisit the data conditioning process.

While the data warehouse industry in general continues to speak of ETL, some analysts and vendors suggest alternatives such as ELT (Extract, Load and Transform) or ETLT (Extract, Transform, Load and Transform). The reason for this is that a significant proportion of the transform function required can be efficiently performed in a relational database. In many cases, apart from the final step where cleansed data is written back to the warehouse, much of the process involves the type of look-ups, comparisons and queries for which analytical databases have been optimized. For example, Merkle, a leading database marketing agency, has begun using the ParAccel Analytic Database to integrate and consolidate many terabytes of highly redundant consumer name and address data in a process that resembles some parts of data warehouse conditioning. The implication is clear: the possibility exists to reexamine and redefine the roles of traditional ETL tools and analytical databases in loading the data warehouse.

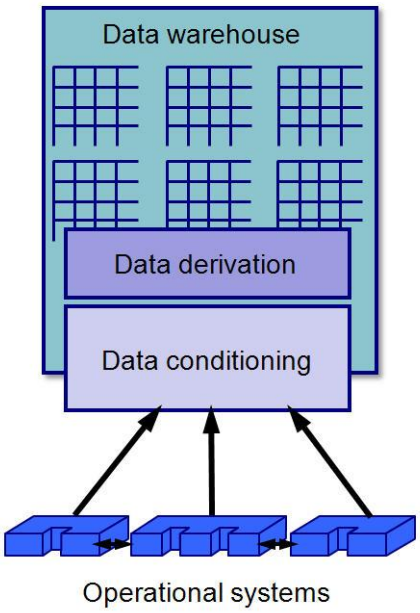

Figure 3:

Data Warehouse Single Layer Architecture

The data derivation function is closely related to data conditioning, but creates new data elements in the warehouse that do not exist in the source data. In a layered architecture, much of this function relates to the creation of summaries and subsets of warehouse data for loading into data marts. As mentioned earlier, analytical databases significantly reduce or perhaps eliminate the need for such structures in order to tune performance. However, the derivation function does not disappear entirely from the architecture. There will remain the need to generate and store derived data elements that do not exist in the source data where such derived data is widely used and/or particularly difficult to calculate. This trade-off of processing and maintenance costs against storage and ease of use for the business users of the warehouse differs from layering in that it does not lead to wholesale duplication of base data.

Supporting a complete data warehouse environment is clearly a bigger stretch for the vendors of analytical databases than the pure user access side of the warehouse. Query optimization, especially when writing to the database, ACID considerations and workload management would all require deeper analysis and probably upgrading. ETL and database vendors would need to work together to define their new roles vis-à-vis one another and to create a design and maintenance interface that spans the boundary between them. Nonetheless, the benefits for the data warehouse are considerable. Another layer of storage could be (at least partially) removed from the physical architecture, reducing latency for operational BI needs and driving down development and maintenance effort for IT.

Conclusion

Analytical databases, be they software or appliance, proprietary or open, columnar or row-based have gained considerable traction in the data warehouse market today. Their popularity stems from the significant query price/performance they offer, as well as their lower implementation and ongoing maintenance costs. These benefits have been particularly emphasized by the understandable vendor focus on analytic applications that can often stand independently of the enterprise-wide data warehouse environment.

While commercially successful and providing customers with strong return on investment, this positioning has diverted attention away from the significant possibilities this technology provides in the wider data warehouse environment. While still in a relatively early stage of development, the combination of technologies used by analytic databases appears to offer the possibility to move away from the multi-layered physical architecture that has been prevalent in data warehouses for most of the past two decades and adopt a simpler, cleaner and more easily managed single layer model. Such a move offers potentially large savings in data warehouse implementation projects and ongoing reductions in database administration costs.

This proposition is far from proven, and in order to take it forward, analytic database vendors will have to place greater focus on maturing the technology with robust query optimization, workload

management, loading and updating features. They will also need more sophisticated performance monitoring and tuning function, as well as a closer relationship with ETL vendors to optimize data conditioning in both batch and continuous feed. If successful, the data warehouse landscape will be dramatically changed, bringing analytic database vendors into direct competition with the established vendors of traditional relational databases.

After a long period of relative calm since the data warehouse architecture wars of the nineties, it seems that analytic databases have the potential to create a new revolution in the way warehouses are structured. To paraphrase that alleged, old Chinese curse: "May we live in interesting times"!

Dr. Barry Devlin is among the foremost authorities on business insight and data warehousing. He is a widely respected consultant, lecturer and author of the seminal book, "Data Warehouse—from Architecture to Implementation". Barry's current interest extends to a fully integrated business, covering informational, operational and collaborative environments to offer an holistic experience of the business through IT. He is founder and principal of 9sight Consulting, specializing in the human, organizational and IT implications and design of deep business insight solutions.

About ParAccel

ParAccel, Inc. is the proven leader in scalable analytic performance and price-performance. The ParAccel Analytic Database™ is a new generation, MPP-Columnar DBMS that is delivering breakthrough analytic performance and price-performance in customer environments. It is available as software or a virtual or packaged data warehouse appliance on standard hardware from all major vendors. ParAccel's management team includes technical founders and industry veterans from noted data management companies Netezza, Oracle, Teradata, Gupta, SenSage, PointBase, and IBM. ParAccel helps leading companies like Autometrics, Merkle and TRX to extend their analytic performance advantage to further differentiate their products and services. ParAccel is based in California with offices in Cupertino and San Diego. For more information please contact us at info@paraccel.com or 866-903-0335, or visit us at www.paraccel.com

ParAccel, Inc.
9920 Pacific Heights Blvd.
Suite 450
San Diego, CA 92121
www.paraccel.com

[1] "*The ParAccel Analytic Database: A Technical Overview*", www.paraccel.com/PADB_techncal_overview

[2] *"An architecture for a business and information system"*, B. A. Devlin, P. T. Murphy, IBM Systems Journal, Volume 27, Number 1, Page 60 (1988)