Evolving Data Warehouse and BI Architectures: The Big Data Challenge

Dai Clegg

Dai Clegg has been an evangelist for big data and an agile development practitioner at Netezza, IBM, Oracle, and Acunu.
dai.clegg@btinternet.com

Abstract

It would be a mistake to discard decades of data warehouse architectural best practices under the assumption that warehouses and analytics stores for big data are not relational nor driven by data modeling (top-down or bottom-up). First, a significant portion of analytic and reporting data is (and will remain) relational. Second, the heterogeneity of data marts combined with enterprise warehouses and operational data stores will remain and be amplified by marts based on new types of data stores.

That said, existing best practices are not enough. Following a period of some stability, the last decade has seen the emergence of MPP data warehouse appliances, then Hadoop and MapReduce, NoSQL databases, and real-time analytics, together with a host of variants, hybrids, and add-ons. The solution space—the collection of available options—has grown rapidly. Consequently, building an effective data warehouse and BI strategy has become even more complex. Meanwhile, the problem space has become more demanding. New sources of social media data and machine-generated data (including the Internet of things) have created opportunities to answer new questions.

Without underrating the value of its history, it is clear that data warehouse architecture is entering a new phase, and history is not enough.

The Physical Architecture Landscape

The big-data-driven silo-ization of platforms has finally fractured the enterprise data warehouse once and for all. Pundits and vendors have widely acknowledged this in the last two years, but predictably, vendors' visions seem to map remarkably closely to their existing product

portfolios, and analysts are often equivocal about solutions, preferring to just evaluate product categories. A single enterprise-scale organization today might well be running Hadoop for data mining and other mass-batch operations, relational data marts for domain-specific, performance-sensitive reporting and analytics, and NoSQL for real-time and time-series data. There might be any combination of on-premises and cloud-based storage, platforms, and solutions.

There are compelling opportunities to use social and Internet-of-things data sources, and these justify application architectures designed to support specific use cases. The breadth of platforms, database back-ends, and visualization requirements means that BI tooling is unlikely to be standardized or, if it is, is unlikely to remain so for long.

We are presented with a wider problem space at the same time as a much wider solution space.

New Technologies and New Philosophies

The fundamental reason to have any computer system is that it produces something of value, not because it stores the data from which value is derived. Storage is merely the means to the end.

Recognizing this leads to the accumulation of data in separate silos, each supporting the use case for which it was created. If silos duplicate data, then so be it; their existence is justified by the value of what they produce. That happened frequently in old data warehouse architectures, a phenomenon sometimes known as data mart proliferation, and it will continue, although more technology choices are available now. It used to be a matter of different relational database vendors arguing about obscure features or arcane licensing arrangements. Now you can hear hundreds of vendors pitching their magical wares.

The change in technology has to drive a change in philosophy. We have moved on from the ideal (often *only* an ideal) of an integrated enterprise data warehouse to a federation of different technologies addressing different use cases and categories of use cases. Gartner has called this the Logical Data Warehouse (Beyer, 2011). IBM has kept the name (enterprise data warehouse) but redefined, or at least evolved, the meaning (Kobielus, 2013). We have moved away from a data-driven view of the warehouse (or whatever we call it now) to a use-case-driven approach. Now we confront a proliferation of use-case-specific architectures—and a danger of uncoordinated data silos all across the enterprise.

Vendors' visions usually address this danger from the technology perspective by focusing on what data is stored in which virtual or tin boxes. More important is how we strike the balance between freeing up teams to deliver the right solution with maximum value for their specific challenges and optimizing across all these solutions so the enterprise gets the best return on its corporate data investment.

This is a new and even more demanding role for the corporate data architect. Line-of-business-sponsored teams are demonstrating ROI to their stakeholders. They will not wait for corporate guidelines about which Hadoop distribution to use, or whether they may spin up a new instance of their favorite open source product in a public cloud.

There is no time to pre-develop appropriate corporate standards, even if we knew how to do so.

The Impact of Big Data across the Data Warehouse Ecosystem

The traditional view of data governance has data stewards and business rules operating to admit newly acquired data to "golden status," where it can be used for reliable reporting and analysis.[1] Once again, the world of big data disrupts that view. We do not have rules about how to accept or reject data, and besides, the value of any single data item almost certainly fails to justify the cost of validating it.

[1] A good summary of this approach is available at http://www.dashboardinsight.com/news/news-articles/the-increasing-convergence-of-mdm-and-data-governance.aspx.

Big data use cases often involve analyzing a mass of individual records to derive insight. If anything can be validated, it is the overall reliability of the data—and that is only possible after the fact. This means that much big data analysis must take place outside the master data, which is the crux of the problem. We only invest in analytics in order to inform decisions, based on trusted data. We do not know in advance whether our big data sources are trustworthy guides to decision making.

The resolution is inevitably in iteration. For example, a simple application developed at the University of Southern California successfully predicted the success of new movie releases.[2] The data was a subset of the Twitter stream, but the success of the application's predictions quickly built trust. Even though any individual data item (a tweet) was of dubious value, the final insight was of value. Trust in the conclusions also has value.

If we can establish trust in the raw data, we can be more confident in any conclusions drawn from the data. However, just because we cannot trust the raw data (at least not in isolation) does not mean we cannot trust conclusions coming from that data. We simply need another way to establish that trust. In this case, it was repeated accuracy of predictions.

There are similar impacts to other elements of the traditional data warehouse ecosystem. As a parallel to governance, the traditional view of data integration is that the data is cleansed, correlated, aggregated, and conformed to a data model before it is used for analysis. However, if we move some of the analysis downstream, we can do more intelligent filtering and operate on raw data in whatever format it arrives in.

Early Hadoop adherents suggested all analysis could be done more cheaply and effectively on raw data. Although that has not been widely accepted, the desire to move some analytics downstream to deliver earlier insight and more compact and richer data to upstream processes is reflected in a number of vendors' reference architectures[3] and clearly has value.

It is not only the core data stores that are disrupted. The entire data warehouse ecosystem is being re-made.

Responding to Uncertainty

Suddenly there are potentially many ways to respond to a requirement, and there are new requirements that have never before been addressed.

There is a case to be made for piecemeal replacement of some components of an existing architecture with cheaper, better, or faster components. This is what drove the success of data warehouse appliances (which are much faster, although not necessarily cheaper) and it is also driving Hadoop adoption for a number of critical use cases (much cheaper for complex processing of very large data volumes). However, the number of components or functions that lend themselves to a lift-and-shift replacement is limited.

Direct replacement will not address new use cases, and even in the cases it does address, the right replacement technology will differ according to the requirements. For example, a relational data mart might be replaced by Hadoop if the use case is data mining. It might be replaced by a NoSQL database if it is struggling to ingest high-velocity data. It might be replaced by a NewSQL instance if the priority is low-latency, ad hoc queries. In other words, the use case is the driver.

To get the most out of the wide and still-growing span of big data solutions, we need to understand exactly what we are asking of them. We also need to accept that in the future, heterogeneous data stores will be at the heart of our BI and data warehouse architectures. This means the challenge of the big data warehouse architect is not just about understanding the data; it encompasses

[2] See http://www-01.ibm.com/software/ebusiness/jstart/portfolio/annenbergMovie.html. The embedded clip shows the application running on an iPad at about 3:30.

[3] For an example, see Oracle's big data reference architecture at http://www.oracle.com/technetwork/topics/entarch/articles/info-mgmt-big-data-ref-arch-1902853.pdf.

understanding why we want that data and what value we seek from it.

Tell Me a Story

Use-case analysis has been around for a long time.[4] The term *use case* has passed into the language; it simply means any need to be met by software. It has been refined and used widely and effectively as a software development concept. Another idea that has taken firm hold as part of several agile approaches is the *user story*.

The meaning of these two terms has been hotly debated,[5] but they both describe what some system or software is supposed to do at some level of detail. They definitely have that much in common. Although the two terms are used interchangeably in many situations, one formulation of user story states that the stakeholder must be able to express what he or she wants in a single sentence that has the form:

> "As a <role>, I want to <goal> so that <reason>."

This is sometimes described as the start of a conversation from which the implementation details will eventually emerge. From the architect's perspective, this ability to refine iteratively means we only have to drill down into the requirements far enough to understand how the story fits in our architecture or to understand what changes to our architecture we will need to make to support it. "Use case" is used more widely, but the role/goal/reason formulation of a user story adds another dimension. It ascribes ownership and value, which provide critical context when identifying the technology needed to deliver the goal. At an architectural level of abstraction, other differences between the two become blurred.

User stories have long been used in agile development and specifically by agile BI teams, but they usually drive implementation of specific features in a specific release of a specific product. Using them to drive prioritization at the architecture level is a more recent innovation, but it will become more widespread as more practitioners exploit big data technology—not for technology's sake, but in order to provide genuine benefit to their organizations.

A Simple User Story Example

Let's say an architect has identified two user stories:

> "As a marketer, I need to understand how website activity is affected by various factors about visitors, such as customer/non-customer, in order to optimize real-time marketing programs."

and

> "As CMO, I need to understand the ROI of the website using metrics such as lead funnel, eventual conversion, revenue per customer, etc., in order to prioritize marketing budget."

User stories have long been used in agile development and specifically by agile BI teams, but they usually drive implementation of specific features in a specific release of a specific product.

To achieve the first goal, we must analyze web log data, which is not naturally in relational form. To achieve the second goal, we will need some of that same web log data integrated with customer data in the existing relational warehouse.

[4] Ivar Jacobson first described use case modeling as part of his object-oriented software engineering methodology in 1992. See http://www.ivarjacobson.com/resources/resources/books/#use case modeling.

[5] See http://c2.com/cgi/wiki?UserStoryAndUseCaseComparison.

This simple example raises a number of issues. A least-change response might be to find an extension to our current data integration technology that reads web logs, models the data structure of those logs, loads them into the existing warehouse, integrates with the customer data, and builds the two analyses requested. However, this approach will not give our marketer real-time responsiveness, which was the reason given in the story. We will also have to load a mass of web log data into the warehouse that we will only need for short-term analysis of the marketer's story. The CMO's story only needs detailed web log data relating to identified customers, and maybe not even that. Aggregates for everything else will probably suffice.

We might alternatively use a NoSQL data store to enable real-time ingestion of web logs and live feedback to the marketer. To satisfy the CMO's story, we might use the same store to build aggregates that are modeled for easy, subsequent integration into the warehouse, where they can be used with the existing customer data.

The marketer's story is met entirely in the NoSQL data store. The CMO's story piggybacks on the real-time ingestion, locates aggregation processing in the same store, and adds a simple loader to assemble its data from two sources.

This alternative duplicates some data, but only aggregates base data that is already being stored. It meets the marketer's real-time requirement and adds minimal processing and storage requirements to the more expensive warehouse.

Another alternative might involve a dual feed of the web logs into the NoSQL store for the marketer and into existing data integration technology, where it will be aggregated and fed to the warehouse.

Even this simple example gives us a number of alternatives to consider. It also gives us the basis for making decisions about those alternatives. It identifies the necessary features (real-time ingestion and processing) and comparative cost (aggregate in NoSQL or store in NoSQL and aggregate in existing data integration technology). Thinking about the problem from the perspective of what is to be done rather than what is to be stored helps us to navigate the tricky territory of new technologies.

The View of New Technology from the Stakeholder's Perspective

Driving from use cases or user stories is not a substitute for understanding the capabilities of new, competing technologies, but it provides a rational and informed means of evaluating them—at least from the perspective of the individual stakeholder.

This approach is close in philosophy to the classic Kimball-esque data mart view, although without the assumption (valid when all data was going into a relational database) that data modeling is the prime concern. Now the prime concern is finding the right technology to meet functional needs, regardless of data store. It carries with it the same problem of how to reconcile bottom-up and top-down priorities. That is nothing new; only the mechanisms for achieving that resolution are potentially new.

> Because the landscape is moving so fast, now is not the time to design a grand architecture and plug requirements into it.

In the view of the corporate data architect, this attitude may appear to encourage anarchy. Instead, it is intended to be an encouragement to seek value in a new situation. As Rear Admiral Grace Hopper famously said, "Humans are allergic to change. They love to say, 'We've always done it this way.'" In a new situation, old policies and processes may survive, but they must be challenged. One way to retain balance is for the corporate architecture group to become an owner of stories. This group represents genuine stakeholders—corporate governance, cost control, and so on. However, user stories' emphasis

on role, goal, and reason takes the debate back to value, where a given story can be measured against the value of other stories waiting to be implemented.

Conclusion

A recent TDWI Research report identified 15 new technologies that 23 percent or more of respondents planned to adopt (Russom, 2014). The second and fourth top technologies on the list of respondents' priorities concern business sponsorship and business drivers. Where there is so much potential choice, the most appropriate way to drive decision making is not from the competing claims of vendors of often still-maturing technologies. It is from the imperatives of business.

Because the landscape is moving so fast, now is not the time to design a grand architecture and plug requirements into it. Now is the time to grasp the most high-value use cases and realize that value for the organization. This means data silos will appear all around, but these are not the data silos of the past, with relational data models that could be merged (in principle). Much of the data will be transitory. For example, the social media response to a product launch is of no value three months later.

A new, extended data warehouse driven solely by line-of-business use cases would be an overrreaction. Some classes of use cases have to be supported, such as the standard operational metrics of customer, product, finance, employee, and so on. Going "year zero" about a unified logical data model is going too far. However, a decision to expand or change the technology base (for example, NoSQL, HDFS, in-memory stores, changing the balance of in-cloud and on-premises applications) needs to seriously consider the use cases that will benefit from the change.

If an organization is considering new technology, it must ask: "Do we need this? What benefit will accrue and when?" There must be good answers to these questions.

Old certainties have been unmade and the new certainties are not yet made. That means a period of transition for data warehouse architects. Whether we will reach another period of stability soon, or at all, is an open question. Meanwhile, a use-case-driven approach to data analytics architecture appears to be the best strategy. Those organizations that remain cognizant of the bigger, strategic picture while they focus on tactical gains will have the greatest success. ■

References

Beyer, Mark [2011]. "Mark Beyer, Father of the Logical Data Warehouse," Gartner blog, November 3. http://blogs.gartner.com/merv-adrian/2011/11/03/mark-beyer-father-of-the-logical-data-warehouse-guest-post/

Kobielus, James [2013]. "The Enterprise Data Warehouse is Virtualizing into the Big-Data Cloud," The Big Data & Analytics Hub blog, IBM, June 27. http://www.ibmbigdatahub.com/blog/enterprise-data-warehouse-virtualizing-big-data-cloud

Russom, Philip [2014]. *Evolving Data Warehouse Architectures in the Age of Big Data*, TDWI Best Practices Report, April, pp. 29–34. http://tdwi.org/research/2014/04/best-practices-report-evolving-data-warehouse-architectures-in-the-age-of-big-data.aspx