BI Experts' Perspective

Dipping a Toe into Data Lakes

Oliver Halter, Mark Kromer, Norman Kutemperor, Donald Soares, and Srinivas Varanasi

Oliver Halter is a partner at PwC Big Data Analytics & Information Strategy.
Oliver.Halter@pwc.com

Mark Kromer is currently a senior solution architect for Microsoft's Azure Data Services.
markkromer@hotmail.com

Norman Kutemperor is CTO at Scientel Information Technology, Inc.
scientel@scientel.com

Donald Soares is chief technology officer of the retail and consumer industry at MarkLogic.
Donald.Soares@marklogic.com

Q **Dennis Forbes is the BI director at Children's Toys, which manufactures and sells toys for two- to five-year-olds. The toys are sold online and through mom-and-pop, chain, and "big box" stores. His BI team maintains a data warehouse that supports reporting, dashboards, and forecasts. The team, along with business analysts throughout the organization, is increasingly developing specific-use analytical applications.**

Dennis's data architecture is the traditional hub-and-spoke with dependent data marts for specific business units (e.g., sales) and processes (e.g., supply chain). He sees the time quickly approaching when the business will need to enhance its data management capabilities to accommodate big data. He has been doing his homework on Hadoop/MapReduce, NoSQL databases, and the like, but is having a hard time getting his head around the data lake concept. Some articles say that a data lake will ensure that any data analysts and data scientists need will be available, but others refer to a data lake as a "data swamp." Can you help him understand?

- **What exactly is a data lake and when might one be needed? How is a data lake used?**
- **Does a data lake require implementing Hadoop/MapReduce?**
- **What are the relationships between a data lake and a data warehouse?**
- **Should a company literally put all its data into a data lake?**
- **What are some best practices in implementing a data lake?**
- **What are the potential "gotchas"?**

OLIVER HALTER

What exactly is a data lake and when might one be needed?

A data lake is a high-performance computing repository with low-cost storage capabilities designed to hold large quantities and varieties of both structured and unstructured raw data. Data structures emerge with usage over time rather than being imposed up front with a predefined schema. That provides flexibility and agility to deal with business requirements in a dynamic environment. For example, Children's Toys has to compete and win across every channel (store, mobile and web) to increase their market share, distribution channels, and revenue in an omnichannel retail environment.

Given all the benefits outlined above, a data lake can be a source of competitive advantage for the company to capture, store, and analyze all of the data from multiple channels, and to perform analytics and drive business outcomes.

How is a data lake used?

For Children's Toys, you can think about four categories of usage from the perspective of performance, flexibility, and speed of data access for data management, reporting, and analytics.

First, real-time data ingestion components in the data lake such as message queues and stream processing can be used to capture, store, and analyze real-time information to drive key internal metrics such as sales and inventory. The same process can also capture external channels such as social media, where there is great potential to engage their online audience and capture "at-the-moment" feedback for defining sales and conversion strategies.

Second, the data lake can serve as a staging area for the data warehouse that is normally used for historical reporting and analysis. Existing challenges to the traditional data warehouse in extracting business value from high-volume and complex data sources such as web clickstream, mobile, or real-time social media data can be addressed by capturing, storing, and mining interactions for various types of analytics in the lake in a cost-effective and efficient way.

Third, analysts can use the data lake for discovery and ideation to solve advanced analytics problems in areas such as customer loyalty, product propensity, and inventory management, and to predict data-driven business trends in a rapid "test-and-learn" environment.

Fourth, product development teams can use the NoSQL data store component in the data lake for web and mobile analytics application development. Children's Toys can add significant value to both its customers and suppliers by deploying end-user-friendly applications and exchanging information and analytics insights. There are innovative monetization benefits for such applications as well.

Does a data lake require implementing Hadoop/MapReduce?

A majority of data lake implementations use the Hadoop platform. There are a few other emerging alternatives like Spark, but it's reasonable to say Hadoop is the major player, the most integrated and most commonly used data lake platform, with an ecosystem of tools and technologies within which it can support a wide variety of use cases. MapReduce, a processing engine that runs on Hadoop, is used primarily for batch processing, but it is increasingly being replaced by more user-friendly execution frameworks that may use it behind the scenes.

What are the relationships between a data lake and a data warehouse?

The data lake is not a silver bullet for solving every data and analytics problem. It is an evolving ecosystem and is complementary to a data warehouse. Both of these ecosystems are used for a different set of use cases and are directed at different consumers. For example, the notion of a data warehouse with a rigid, pre-defined schema is not attractive to analytics modelers who are working in a rapid prototyping "test-and-learn" environment, but it might be for predefined management reports running on a predictable basis.

At PwC, we are seeing more companies following a hybrid model, integrating their data lake and traditional data management infrastructures to solve very interesting business problems. We covered this topic in detail in our article, "Ushering in the Next Generation of Information Architecture," published in TDWI's *Business Intelligence Journal* in December 2015.

Should a company literally put all its data into a data lake?

There are two sides to this debate. One group would argue that funneling all of the enterprise data through a data lake and then distributing it to the downstream applications and users is the best way. The other group would say that all the traditional and proven data sources can flow through systems such as data warehouses and the data lake can be used to process high-volume data sources (such as clickstream and mobile) where information has unproven business value.

There are merits to both these arguments and companies must pick the "right-fit" information flow based on their business objectives, size of their enterprise, maturity of existing capabilities, existing technology investments, and future strategy.

Overall, the debate is constructive because it will introduce new patterns of integrating the data lake with other ecosystems in years to come. Data lakes are also a cost-effective model for storing enterprise data for long periods and with great detail for later exploration.

What are some best practices in implementing a data lake?

- Create a business-driven strategy for implementing a data lake. The road map and case for change must outline the business outcomes and value drivers that necessitate the categories of data assets, analytics techniques, and technologies to be integrated in the data lake.

- Define an operating model and plan to hire, train, and scale talent for execution; it's *critical*.

- Work gradually. Data lakes are a great way of gradually implementing and delivering value by adopting a rapid "test-and-learn" strategy with little engineering needed before value can be created.

- A use-case-driven approach of implementing pilots is an effective means to articulate to end users the value of a data lake platform. Take an incremental approach in selecting high-value use cases for development.

What are the potential "gotchas"?

Strategic positioning: IT organizations often find it difficult to strategically position the use, capabilities, and benefits of the data lake platform with the business stakeholders. Having early adopters of data lakes in the business units as champions during the pilot phase is critical in this regard.

Data swamps: Having a narrow technology-centric view during implementation with no outside-in view from the business, maintaining siloed IT architectures, and a rigid operating model will result in a data swamp—a dumping ground of data with no business context and even less business adoption.

Traditional implementation: The data lake platform solution needs to be implemented incrementally with a joint business and IT team that follows a different process than traditional IT project implementations. The definition of high-value pilots and deployments showing tangible business value, and the ability to "win or fail fast," are paramount. The majority of programs that follow a traditional data warehouse implementation process fail.

Hadoop platform selection: The technical capabilities of the Hadoop platform are evolving at breakneck speeds and there are many available choices for deployment models and software distributions. Hence, companies need to define holistic platform evaluation criteria so they can make choices based on their own business context.

Talent: The talent pool is small. Enterprises need to find a competent team with multiple, cross-functional skills (from developing a strategy through implementing a data lake) so that the right architecture and foundational methods are followed.

Third-party tool integration: You need to pick the right technologies and tools to interoperate seamlessly with the data lake to leverage its scale and computing power.

MARK KROMER

As a BI director with an existing infrastructure, Dennis is wise to evaluate the benefits of the data lake approach to data analytics. When examining big data approaches, he will soon discover three key differences that will likely appear as differentiators from Children's Toys' current traditional data mart approach.

The first important concept to understand is that using a big data way of thinking about the problem is to work from the set of *all* available data. The data mart approach is a way to segment enterprise data that has been extracted from raw data sources, cleansed, and summarized for specific business purposes (and typically stored in a relational database or analytical database).

The most common way to capture and store data for your data lake is in the form of various text and compressed file formats in a distributed file system. A common example would be CSV or JSON text files placed in HDFS, or compressed files in common Hadoop formats such as ORC or Parquet.

The second concept is that of data engineering. A data lake is a collection of valuable data assets that do not bring business value in storage alone. Because we are now storing more raw, granular data, the data volumes are going to be much greater than the filtered data sets made available in a data mart.

Although that has the potential to generate greater business value, it also means that classic ETL (extract, transform, and load) routines and the transformation pipelines that move data will be ineffective (and may not work at all). This is where the data lake concept introduces Hadoop processing engines such as MapReduce, Pig, and Hive, as well as Spark, to perform analytical operations on the data in place in the data lake (HDFS in this case) without the need to transform and move it into a rigid-schema relational database.

> Using a big data way of thinking about the problem is to work from the set of *all* available data.

The third concept is data exploration, where a more advanced set of data analytics may need to be applied to make sense of and find business value in the massive amounts of new data. This process of data sifting and exploring utilizes advanced analytics as an important part of the new approach to analytics that encompasses what is now sometimes referred to as data science. In this data lake case, the difference between the traditional data warehouse and data mart approach is that we can make broad assumptions about the data due to an anticipated lack of data structure and the granular, raw nature of the data in the lake. A data lake scenario would require two basic steps to analyze the data in place in the lake:

- Bring portions of the files in the lake from HDFS into Spark or R as data frames or RDDs that can be used as data samples to generate string tokens, groupings, text classifications, regressions, and other techniques to find the most relevant features in the data.
- Expose that relevant portion of the lake to business and data analysts by applying metadata HCatalog or Hive table definitions so that Children's Toys' existing BI tools can connect to the relevant data in the data lake. Additionally, the data mining and statistical methods used in step one for data discovery can include new data points and predictive results that can be surfaced to the business through BI tools by generating new files in the lake and appending those results to Hive tables.

Finally, keep an eye on the idea that Dennis noted regarding data swamps. In practice, big data practitioners have often referred to extraneous data stored in a data lake as "exhaust data." This can be thought of as data that was deemed unworthy for the enterprise data warehouse. There is great potential for Children's Toys to gain a competitive advantage through process optimization and gaining new insights into customers through use of big data and data lakes.

However, be diligent and complete the evaluation of the data you are collecting so you realize a positive return on your investment in additional storage and operations. Utilize advanced analytics techniques available in R and MLlib to find patterns in log files and other new data sources that are not obvious simply due to the data volume, complexity, and lack of structure.

NORMAN KUTEMPEROR

The basic requirement of a data warehouse is that all data being accepted into it must conform to its specific format standards before the data can be accepted. A data lake is a true data repository in the sense that data stored in a data lake can all be in completely different formats. Therefore, data lakes are typically utilized by large organizations that acquire various types of data, deploy more than one type of computer system, or deploy several versions of the same computer system due to version incompatibilities.

As a data lake accepts data in any format—which is then kept in its original format and stored until it is processed—generally speaking, one may not be able to perform any processing or analytical functions on the unformatted data in the lake and therefore must have the data formatted and moved to a data warehouse for use.

> NoSQL provides the flexibility with respect to variety, volume, and velocity that is required to create a proper data lake.

Although many consider Hadoop a data lake because it can accept data in many formats, a data lake does not require Hadoop or MapReduce. Hadoop is a repository and it requires other systems to do the analytical processing. MapReduce is a function of analytics. It is the two working together that provides results when it comes to business intelligence. Any good alternative to Hadoop will utilize a robust NoSQL database that can accept data in any format, as well as be made available to processing systems that incorporate both analytical and data reduction algorithms.

A data lake stores all types of data from all types of systems for an indefinite period of time and is typically much larger than a data warehouse. Because this greater size is a basic feature of data lake implementations, Children's Toys should consider putting all its historical data into a data lake as well. The call for certain analytics may only occur at a later date, but when the need does occur, Dennis can fetch the entire data set from the data lake and move it into a new data warehouse for fresh analytics.

In implementing a data lake, Dennis should choose the architecture carefully. Based on new discoveries and capabilities, I'd recommend a NoSQL database to allow the necessary raw data capabilities. NoSQL provides the flexibility with respect to variety, volume, and velocity that is required to create a proper data lake.

NoSQL also addresses the scaling limitations of many database systems that would otherwise place severe restrictions on data lakes. SQL databases tend to be good at scaling up. However, with regard to big data, systems need the ability to cross the physical boundaries of a SQL system and provide for much more expansion. This can only be achieved by systems that can scale out, rather than up, by addressing large compute nodes in real time. NoSQL databases are created with this scalability in mind.

Another important criterion we're seeing is multimodeling, where the system has the ability to process data stored in different formats within a single database—including both structured and unstructured data types. Only select NoSQL databases provide such capabilities. Some notable products provide the ability to store and process with NoSQL capabilities and, at the same time, provide real-time SQL queries on the same database without conversions. One must evaluate carefully to find such rare capabilities.

Multimodeling capabilities are worth searching for, though. They save resources and provide greater performance as the efficiency is increased substantially by storing different types of data in different data stores. Processing is done without creating interim join tables, and that helps tremendously when it comes to big data. Beware not to mistake multimodeling with the varieties of data that exist in a data lake. Multimodeled data in a NoSQL DB is always formatted.

DONALD SOARES

Essentially a data lake (typically Hadoop) is a repository for the massive amounts of disparate data that organizations need to economically store and access. Hadoop cost-effectively stores this data in its native format in a Hadoop Distributed File System (HDFS) and runs large-scale MapReduce jobs for batch analysis. Its key advantages are that data storage in Hadoop is significantly cheaper than traditional databases and does not require upfront schema creation or changes to the data format.

As BI director at Children's Toys, Dennis faces three major challenges:

- Integrating internal data sources from different business units such as sales, marketing, and operations to run the business and derive better insights
- Cost-effectively enhancing this data with external sources
- Ensuring business users can use the data to support operations and provide analytics applications to grow revenues and profits

We'd recommend that Dennis consider combining Hadoop with an enterprise-grade NoSQL database to run real-time applications. This also ensures that his business data is protected with certified security, alongside high availability and disaster recovery.

Here's how Dennis should proceed. Let's start with the business. How will the data be used?

Hadoop works well with distributed, cost-effective storage. It's also great for batch analytics and off-line, high-latency processing. However, in retail the most effective promotions occur in real-time, whether in-store or online. Dennis's retail clients will want to be able to identify the consumer upfront and then promptly provide them with real-time offers and promotions based on their past toy purchases and known interests.

Marketers also want to alert customers immediately when the latest hot toy becomes available and his supply chain team may want to know ASAP if a certain toy is sold out during Black Friday in order to rush in replacements. Hadoop lacks real-time operational capabilities and is therefore not well suited for these retail use cases.

Dennis should ask his business team if they want to observe and report on past performance or if they want to "run the business" and create real-time applications. If the business team chooses the latter, Dennis will want to use Hadoop for data storage and link it with an enterprise-grade NoSQL database to run real-time and operational applications. A NoSQL database with a flexible data model is a better choice here than a relational database because an RDBMS would require considerable data modeling and significant ETL to flow data from Hadoop.

Stop the Progression from Data Lake to Data Swamp

Remember Hadoop is really a complex ecosystem with multiple parts for data ingestion and storage, computation, search, and query as well as management and coordination. Dennis needs to ensure that his users at Children's Toys have the technical skills required to perform these functions. Be mindful that most Hadoop installations do not meet revenue growth and cost savings goals because users lack the skills and are unable to successfully integrate the various components. Lack of skills often leads to data silos as well as technical silos.

Dennis can't take the risk of having multiple business functions dump data into Hadoop and then be unable to access it later. This is what turns a data lake into a swamp. However, partnering with an enterprise NoSQL database that indexes all data up front, with built-in search and semantic capabilities, would go a long way to ensuring the contents of the data lake stay clean, transparent, and easy to navigate.

Enrich the Product Data

Toys are an amazingly complex category in terms of data. It's not just the product and supplier name, it's also targeted demographics, film and video game associations, safety and usage instructions, as well as pricing that make up the product experience. Using semantics to enrich data linkages and setting up the metadata makes perfect sense. For instance, Dennis will want to ensure that the latest release of Chewbacca and Han Solo Star Wars action figures are paired properly with the spaceships intended for them otherwise Han may have to leave the oversized Chewie behind in the swamp.

Enterprise Applications

Finally it comes down to consumer, product, and supply chain data, all of which are critical from a security standpoint. Most open source applications, including Hadoop, lack built-in security features. Dennis and Children's Toys just can't take the risk of a security breach that impacts consumer credit card or product design data. Choosing an enterprise NoSQL database should provide the role-based security, ACID compliance, high availability, and disaster recovery that ensure the integrity of the data.

To date, Hadoop deployments have been tales of IT trumpeting storage cost savings while frustrated business users feel they have not realized similar benefits.

By combining Hadoop with an enterprise NoSQL database, Dennis will be able to overcome Hadoop's technical limitations and meet the needs of Children's Toys today and into the future. ■

Instructions for Authors

The *Business Intelligence Journal* is a quarterly journal that focuses on all aspects of business intelligence, data warehousing, and analytics. It serves the needs of researchers and practitioners in this important field by publishing surveys of current practices, opinion pieces, conceptual frameworks, case studies that describe innovative practices or provide important insights, tutorials, technology discussions, and annotated bibliographies.

The *Journal* publishes educational articles that do not market, advertise, or promote one particular product or company.

Visit tdwi.org/journalsubmissions for the *Business Intelligence Journal's* complete submissions guidelines, including writing requirements and editorial topics.

Submissions

For complete submission guidelines and suggestions, visit tdwi.org/journalsubmissions

Materials should be submitted to:
Peter Considine
Managing Editor
Email: journal@tdwi.org

Upcoming Deadlines

Volume 21, Number 4
Submission Deadline: August 5, 2016
Distribution: December 2016

Volume 22, Number 1
Submission Deadline: November 18, 2016
Distribution: March 2017