DATA INNOVATIONS

JUNE 2015

SHOWCASE

Hadoop: A Slam-Dunk DM Platform

A CONVERSATION WITH REDPOINT GLOBAL

tdwi
Advancing all things data.

HADOOP: A SLAM-DUNK DM PLATFORM

A CONVERSATION WITH REDPOINT GLOBAL

There are many straightforwardly valuable business applications for Hadoop. Of these, one type in particular—data management—has significant impact for downstream decision-support applications.

For all of its hype, Hadoop is a genuinely disruptive platform for enterprise data management (DM). This is chiefly a function of its combination of cheap, scalable, distributed storage and cheap, scalable, general-purpose parallel processing. This combination is to some extent unprecedented.

So, too, is Hadoop's schema-when-you-need-it data management paradigm, which enables it to store and manage data much more flexibly than a relational database. (Hadoop can ingest and persist data without first applying schema to it; schema can be derived later—at the time of read, for example.) From an information technologist's perspective, Hadoop has all of the makings of a slam dunk: as an elastic storage and compute platform, it's orders of magnitude cheaper than an (R)DBMS; as a DM platform, it's objectively faster and more flexible and robust for schema-when-you-need-it data.

"Many companies have already exhausted their efforts in trying to do schema-on-write in their enterprise data warehouses. There's some support for schema-on-write from some of the vendors, but it's never going to be as flexible as it is in Hadoop," says Jamie Keeffe, product marketing manager for data management with RedPoint Global. The firm helps data-driven organizations unlock the full value of their data to drive customer engagement and profitable, sustained growth.

> Hadoop can ingest and persist data without first applying schema to it.

"When you take Hadoop's computational scalability and cheap storage into account, it becomes clear Hadoop is an extremely useful and extensible platform for data management."

On this basis, it is tempting to conclude that one has only to design, budget for, build, and deploy a Hadoop cluster and the business applications—and business backing—will come.

Right?

Wrong, says Keeffe. Technology "solutions" don't sell themselves, he points out. A vendor can lead or even lure business buyers to a market; it's extremely difficult, however, to get people to deploy big data in a production state in the absence of demonstrably successful production apps that improve upon (or which enable altogether new) business use cases.

Today, it's comparatively easy to spin up a Hadoop cluster as a proof-of-concept in a virtual machine (VM) sandbox or on spare hardware kit. The challenge, now as ever, is to procure backing from an influential business stakeholder to promote that cluster from a proof-of-concept to a production system, complete with its own dedicated resources and (most important) its own budget. To do so, it's first necessary to persuade the business (or at least one influential business stakeholder) about the potential usefulness of one or more production applications running in Hadoop.

"Having Hadoop underneath your desk as an IT person doing a proof-of-concept and proving that it's cheaper storage is one thing, but if you want to get Hadoop into production, you need to find business applications to leverage it. It's as simple as that," Keeffe argues.

Today, there's no shortage of ostensible "business" use cases to which Hadoop could be applied. Some Hadoop vendors, for example, tout a vision of a Hadoop-based "enterprise data hub" that they position as the focal point or organizing center for data management. An increasingly popular prescription is that of the Hadoop "data lake," which functions as an elastic storage "reservoir" for ingesting, persisting, and managing business data.

As business applications go, however, both prescriptions are lean on the specifics and indefensibly vague about time-to-value. (In other words, what pressing business problems does an enterprise data hub address *right now*? What domain-specific business applications does a Hadoop data lake enable *right now*?) They'd rightly invite skepticism—or worse—from a business sponsor. Is it any wonder so many Hadoop vendors market primarily to IT and not to the line of business?

There are a number of straightforwardly valuable business applications for Hadoop, however. Of these, one in particular—data management, or DM—has undeniable business use-case significance.

From a business perspective, DM is "for us, by us." A DM technology that standardizes data load, cleansing, integration/matching, persistent keying, and aggregating across an enterprise and is at the same time tolerant of domain- or process-specific variations has indisputable value. A DM solution that's extensible enough to be cost-effectively implemented across multiple business domains—e.g., to manage HR, supplier, and vendor data, along with product, customer, and virtually any conceivable kind of data—is the Holy Grail of enterprise data management.

To the extent that such a DM solution also delivers rapid time-to-value, it's especially likely to resonate with business backers. For example, how many business teams would agree that they already have access to all the data they want? On top of this, how many businesses would turn down the opportunity to have safe, self-service access to Hadoop and pull in Internet-based data sources to enrich their primary data sets?

How many business teams would agree that they already have access to all the data they want? How many businesses would turn down the opportunity to have safe, self-service access to Hadoop?

For several reasons, Keeffe contends, Hadoop is a logical platform for DM deployment. What's more, a Hadoop-based DM solution can be linked with any number of pressing business needs. TDWI Research famously estimated that "bad" data costs U.S. businesses $600 billion a year.

At a minimum, bad data delays decision making through decision-by-committee as a means to mitigate risk and responsibility when data is not trusted. In practice, bad data translates into lost or misdirected shipments, incorrect orders, wasted (or poorly targeted) marketing initiatives, and a litany of other bad business outcomes. To the extent that a Hadoop-based DM solution is price-competitive with traditional DM offerings, improves business outcomes, and enables fundamentally new applications, services, and capabilities, it should have irresistible cachet with business stakeholders, especially inasmuch as traditional DM just isn't getting the job done relative to big data.

"Your typical enterprise DM solution will consist ... of an ETL component, data quality—or, at least, of parsing, standardization, cleansing, and de-duplication routines—data integration, which matches, links, and merges cleansed data, and a workflow in the form of automation and monitoring that instantiates all of this in a managed and repeatable way," he says.

"There's no reason this configuration couldn't be replicated or migrated in Hadoop or hybrid environments," he continues. "When you consider Hadoop's strengths, there's every reason it *should* be part of the DM processing environment."

What Makes Hadoop Different

The Hadoop platform combines a scalable distributed storage layer with a general-purpose parallel processing engine. This combination is similar to that of a massively parallel processing (MPP) (relational) database management system, with the obvious difference that an MPP (R)DBMS is designed for a very specific type of data processing workload—viz., query processing. In addition, MPP database systems are optimized for SQL programmatic operations and designed to ingest and persist data in a schema-on-write structured (tabular) format.

Hadoop is different. Its storage substrate is a file system (namely, the Hadoop Distributed File System, or HDFS), which permits it to land and store data objects of any type. (At the file system level, Hadoop persists data in any of several data serialization formats—including "Avro" data files and Sequence Files. It can also store data in the form of text, CSV, JSON, or XML files, as well as in formats such as Parquet files—a column store format—or Record Columnar Files.) An MPP (R)DBMS, by contrast, imposes strict constraints on the types of data it can efficiently ingest.

An MPP database is disadvantaged, too, when it's tasked with managing and processing heterogeneous data types: as with all (R)DBMSs, it's optimized for the structured data that the SQL language is used to manage and manipulate. True, most MPP databases implement functional workarounds (e.g., UDFs that embed procedural code), but HDFS gives Hadoop a clear and decisive advantage when dealing with multi-structured data types.

Hadoop, then, comprises a general-purpose distributed storage and parallel execution environment.

It boasts another, critical advantage with respect to an MPP database: it is orders of magnitude cheaper. Thomas Davenport, a senior adviser at Deloitte Analytics and a senior research fellow at MIT's Center for Digital Business, puts the cost of Hadoop-based storage at 23 *cents* per gigabyte, and that of online storage in a data warehouse at 19 *dollars* per gigabyte. In comparison with an MPP database platform, then, it's possible to inexpensively store, manage, and process data of any type in the Hadoop environment.

MIT's Center for Digital Business puts the cost of Hadoop-based storage at 23 *cents* per gigabyte, that of online storage in a data warehouse at 19 *dollars* per gigabyte. In comparison with an MPP database platform, then, it's possible to inexpensively store, manage, and process data of any type in the Hadoop environment.

This, says RedPoint Global's Keeffe, is why Hadoop makes for a slam-dunk DM platform. He cites RedPoint Global's own Data Management Platform for Hadoop, which he says leverages all of Hadoop's strengths to achieve a modern, agile, and extensible DM solution. "Because we're able to store all of the data in its raw state without first applying schema to it, we can be agnostic with respect to data types, so even though customers are primarily focused on SQL analytics today they'll soon be mixing in data from social and cloud sources," he explains.

"This isn't a problem for us. From our perspective, it could be a new app that we're preparing the data and cleaning it for or it could be a very old traditional Teradata app or Oracle app. Because of how we use Hadoop to store, manage, and process data, we can accommodate any use case."

Traditional DM platforms aren't designed for (and can't easily be retrofitted to accommodate) big data's Not Only SQL paradigm, which was designed to ingest and

manage multi-structured data along with structured (relational) data. In this regard, Keeffe argues, traditional DM and MDM solutions tend to "co-opt" Hadoop to work with or in an existing architecture instead of making the Hadoop platform the focal point of DM for data quality and integration.

"Traditional DM vendors have tried to co-opt Hadoop by embracing it as a platform for cheap dark storage. This makes sense, because they're trying to protect their traditional database management market shares," he says, "so what they do is they position Hadoop as a data lake, which means they assume you're storing all or only dark data in Hadoop and HDFS. They move that data out of Hadoop across the wire to their DM hub, process it there, and then move that data back across the wire back into Hadoop, none the wiser for the journey."

This double movement is nothing less than wasteful, according to Keeffe. It ignores the fact that Hadoop isn't merely an inexpensive platform for data storage—it's an *extremely flexible* inexpensive platform for computationally intensive data processing. To wit: it's possible to land and to persist data in Hadoop without predefining a schema for it, which means that schema can be enforced *retroactively*—i.e., when data is accessed.

This flexibility makes Hadoop an extremely useful platform for DM, as use-case requirements evolve or change. On the one hand, data can be prepared or cleansed (if necessary) on access; on the other hand, schema—or, rather, standards and definitions—can be derived or applied, as needed, when data is accessed.

Traditional DM implementations leverage Hadoop as a scalable storage substrate but neglect its most disruptive features: its flexibility with respect to how it stores/manages data and its ability to perform parallel processing *at the site of data itself*. Data doesn't have to be moved to and from Hadoop because it's already there; DM, data quality, data integration, and matching workloads can be performed on data *in situ*—with processing parallelized automatically across the Hadoop cluster—without having to move data at all.

"The need to move large data sources across the wire into a traditional DM solution for processing ... creates a huge amount of network traffic. Depending on how much data has to be moved, it can take an extremely long time," Keeffe points out. "What would work better is to use Hadoop to do schema-on-read for these DM jobs, so you're moving the work to where the data resides, and you can store aggregations however you want in HDFS or in traditional EDWs for legacy-app consumption of enriched EDW data sets."

This is the approach RedPoint Global takes with its Data Management Platform for Hadoop. "Most of these other DM vendors are focused on SQL, but because of our architecture, we're focused on data in any format. We can reach out and pull out any data in any kind of format and ETL or ELT it into Hadoop, but that's only the first step. Instead of working out things in SQL in Hadoop, RedPoint actually keeps the data in its raw format in Hadoop. We're doing cleaning, de-duping, and key management on the data in its native state without changing the data. Just as important, from a business user perspective, achieving a single raw data repository that multiple business teams can access will effectively dismantle the congruity issues experienced with traditional data silos."

"We're doing cleaning, de-duping, and key management on the data in its native state without changing the data. … A single raw data repository that multiple business teams can access will effectively dismantle the congruity issues experienced with traditional data silos."

A Hadoop-centric approach has other advantages, too, he maintains. "Your DBA is already an expert with the internal data assets that you're integrating with the new external data sources, and they also have an understanding of the business cases and rules, which is a huge advantage when you're doing data quality and data integration. The ability to apply these in-house resources, this in-house knowledge, is a significant advantage," Keeffe argues.

"In the traditional model, you're paying to educate these ETL resources"—i.e., developers or architects who are either versed in the use of specific ETL tools or who have deep, domain-specific knowledge about data engineering itself. "In most cases, they'll only touch these ETL processes once. They code the ETL once, and there's no repeatability. You bring in outside people,

or you educate your own, they run the ETL tool, it populates the schemas, and then they're done. In the Hadoop environment, you can go back to the data as often as you want as requirements change and you can even schedule, automate, and evolve those jobs."

"In the Hadoop environment, you can go back to the data as often as you want as requirements change and you can even schedule, automate, and evolve those jobs."

Finally, there's the fact that Hadoop itself is a fine-grained parallel processing environment. This has everything to do with Hadoop's new resource negotiator, YARN (yet another resource negotiator), which permits Hadoop to schedule, monitor, and manage both native frameworks and *third-party* frameworks in addition to the bulk processing traditionally performed by MapReduce.

YARN officially debuted with Hadoop 2.0. It's implemented as a cluster management framework—what Hortonworks (which contributed most of the YARN code) dubs a "data operating system"—and is designed to provide cluster-wide scheduling, monitoring, and management capabilities.

YARN introduced a new compute paradigm, complete with a MapReduce reboot (MapReduce 2.0) and enabled third-party batch processing frameworks to co-exist on the same plane as MapReduce. Subsequently, two new native "frameworks" (i.e., engines) emerged. The first, Tez, provides a reduced latency framework for queries; the second, Slider, provides a long-running service framework that can be leveraged for services like machine learning. Thanks to YARN, it's now possible for third-party engines (or "frameworks," in the YARN lexicon) to run the same way and with the same granularity of control as MapReduce.

Prior to YARN it was virtually impossible for third-party engine-level applications to tightly couple with Hadoop and natively manage—schedule, monitor, tune, or scale up or down—the performance of cluster resources. Nevertheless, Cloudera and a number of other vendors built third-party engines (such as Impala) that were designed to execute in—and which could, by means of distribution-specific tools, be managed by—Hadoop. These tools are either tied to specific Hadoop distributions, such as Cloudera, or implemented *outside* of a Hadoop cluster.[1]

Why does this matter? As any DBA in an MPP environment can tell you, clusters must be carefully sized and tuned to maximize performance. In the same way, MPP systems provide control over the use and allocation of resources in a cluster—including, crucially, the ability to manage and balance (by dynamically scaling up or scaling down) workloads. In a production environment, a runaway workload can cause (costly) data loss, sabotage other running workloads, or result in (costly) downtime.

Thanks to YARN, Hadoop can exercise comparatively granular control over how, where, and when a workload runs. It can efficiently parallelize a data processing workload across server nodes within a specific rack, across nodes housed in other racks, and, at a more granular level, across the microprocessor cores that power those nodes. Hadoop's data replication scheme distributes a minimum of two copies (for a total of three) of a data or file "block" across a cluster, replicating once to a data node that is local to—i.e., in the same rack as—the original block and again to a data node that is located in a physically separate rack. Hadoop and YARN can also scale workloads to exploit available SMP (symmetric multi-processing) resources—as well as processor-specific optimizations—at the node level.

This makes Hadoop especially well-suited for performing the identifying, matching, de-duping, and

[1]For example, a typical Hadoop configuration consists of a cluster of nodes running Linux. The Hadoop system administrator configures each node to balance the needs of the host OS with Hadoop. Some third-party "Hadoop" engines actually install locally, in Linux, and copy data from a clusterwide context—in HDFS—to local storage on one or more nodes for processing. In the same way, their management tools install locally, in the Linux OS, and are managed locally. They provide little to no insight into or control over how Hadoop provisions and manages cluster-wide resources.

linking operations that are common in data quality and integration processing, Keeffe points out. "If you can provision a job at the core level, so that Hadoop can assign it to individual cores that are possibly underutilized, that's a huge benefit. Those jobs are going to benefit significantly."

Spinning a YARN

It's easier to use YARN to do this. In fact, a YARN-native application—which is written to YARN itself, not to an intermediary framework, such as Tez—is able to exercise extremely granular control over how, where, and when it runs. A YARN-native application can also exploit YARN's ability to schedule, execute, and manage interactive and real-time/streaming workloads, something that was impossible in the "legacy" MapReduce 1.0 paradigm.

The batch-only MapReduce 1.0 paradigm is less-than-optimal for certain kinds of data processing workloads, and, particularly, for business intelligence (BI), data warehousing (DW), and even many kinds of data engineering workloads, including ETL transformations, data quality routines, and data integration processing. (Even though MapReduce saw early use as a brute-force ETL tool, most ETL vendors opted to use third-party libraries, running, in the pre-YARN context, co-located with/in the nodes of a Hadoop cluster, to perform ETL transformations. MapReduce 1.0 was a rigid batch paradigm, which means jobs or operations couldn't be pipelined: instead, they had to run separately and in sequence.)

For data management workloads, it may appear cleaner, faster, and cheaper to exploit YARN-native intermediary frameworks such as Tez than to write code for MapReduce. Although this is true in one respect, this approach has a number of drawbacks, starting first and foremost with performance: a YARN-native application designed specifically for data integration or data quality can run more dynamically in Hadoop and will be much faster than an application that uses Hadoop's Tez or MapReduce to implement the same functions.

An innovative third approach is to write your own YARN-native application designed specifically for parallel processing, thus eliminating the need to code to any intermediary framework. This is the approach championed by RedPoint Global's Keeffe. Not surprisingly, it's the approach RedPoint Global took with its Data Management application for Hadoop.

An innovative third approach is to write your own YARN-native application designed specifically for parallel processing, thus eliminating the need to code to any intermediary framework.

"By running the way we do (i.e., as a YARN-native application in Hadoop), RedPoint is able to accelerate the data processing required for identity resolution, linkage, and keying, and that, in turn, eliminates a lot of new application development that would be required. It allows you to meet or surpass mandated performance benchmarks, and it allows you to automate the execution of a broad set of jobs that can scale across your entire data processing environment, both in and out of Hadoop," he says.

Is "YARN-native" the same thing as "YARN-ready?" Hortonworks, for example, sponsors a "YARN Ready" certification program, which means that a YARN-ready application is "able to use the resources of the customer's Hadoop system to process Hadoop data in place, without interfering with other YARN-ready tools and applications." What does this mean in practice? Or, put differently, why is YARN-native a superior model to YARN-ready?

For one thing, Keeffe points out, a YARN-native app is able to exploit YARN's fine-grained control over resource use, parallelization, and other aspects of cluster management. More important, the category of YARN-ready applications is potentially so expansive as to be meaningless. According to Hortonworks, for example, any application that invokes MapReduce is "YARN-ready." (From Hortonworks' perspective, a YARN-*unready* application is one that bypasses an intermediary framework such as MapReduce to "read data directly from HDFS, and thus competes with YARN for cluster resources.")

In the same way, an app that uses Hive to query against data in HDFS is no less YARN-ready. Under the covers, Hive—which compiles SQL queries into MapReduce jobs—exploits Tez or MapReduce to figure out how to run these jobs.

A YARN-native app, on the other hand, generates and submits an ApplicationMaster to YARN itself. YARN, in turn, takes care of scheduling and running the ApplicationMaster and its collection of processing tasks. There are two primary advantages to this approach. First, writing to YARN makes it possible to exploit computational patterns that are not available to other kinds of applications, especially those that generate MapReduce code. This is because a purpose-built ApplicationMaster can more efficiently invoke and coordinate data management processing routines and functions using processing patterns and techniques that are not available to MapReduce programs. A data management application operating under YARN has a more flexible and complete computational environment at its disposal and is able to optimize for specific use cases more fully than an application based on MapReduce or Tez.

Second, YARN has enabled ISVs such as RedPoint to create productive data management applications for end users, thus eliminating any need for programming. Hadoop is a young technology and its application is still dominated by legions of dedicated programmers and technologists, but this model works only for large and specialized organizations. Application development using the native Hadoop tools—MapReduce, Tez, Hive, and Pig to name a few—is notorious for its requirement of rare and leading-edge software development skills. Just as data management functions in "classic" server environments matured into turn-key applications requiring little or no coding, so must these functions evolve in the Hadoop space to be productive for regular use.

"Without an ApplicationMaster, you have to code. Some kind of coding will have to be done, whether that's code you write in Java, Pig Latin, or Python. Alternatively, The ApplicationMaster tells YARN how to run a workload," says Keeffe. "YARN's primary purpose is to serve as a kind of data processing ecosystem for Hadoop. As part of this ecosystem, Hadoop developers have built out a number of standalone services—things like MapReduce 2.0, Tez, Slider, and then they have this fourth service for apps that are able to generate an ApplicationMaster and architect it in such a way that YARN can stand them up and have them be automatically instantiated."

In most cases, a YARN-native application will achieve superior performance because it bypasses Tez, Slider, and other "intermediary" interfaces, Keeffe claims. "The ApplicationMaster is an engine-level path that bypasses those intermediary interfaces into YARN, which means your application code is basically processing natively in Hadoop. This is what 'native' is all about," he argues.

"Another big benefit of doing this is that YARN's governance capabilities are being leveraged by those applications that can run natively, so YARN cleans up after them, scales them, and redistributes or resizes them if the configuration of a Hadoop cluster itself changes. Those applications leave no overhead in Hadoop. You have to manage these other third-party engines manually: if you resize your Hadoop cluster, you have to reconfigure them manually."

Accelerated Data Management Is a Killer App

Advanced analytics is frequently touted as a killer app for big data. Keeffe believes this puts the proverbial cart before the horse, however. If data is to be analyzed, it must first be identified and prepared—i.e., cleaned, transformed, and conformed. At big-data scale, data movement becomes a non-trivial problem. It's one thing to move 1, 10, or 20 *gigabytes* of data across a network—it's quite another to move 1, 10, or 20 *terabytes*. Thus the hard problem of data integration that's poised to get much more difficult at big-data scale. The real killer app for big data, he argues, is one that applies technology to accelerate data preparation for any application.

RedPoint Global's Data Management Platform for Hadoop combines data storage and schema flexibility with a YARN-native parallel processing engine that's optimized specifically for DM that can scale across both

traditional and Hadoop environments. This combination helps simplify and accelerate the work of data quality and integration—thanks chiefly, argues Keeffe, because it's matched on the front end with a visual, point-and-click/drag-and-drop UI. This front-end component uses proven pre-defined tool functions to complete data management activities such as:

- Loading/extracting
- Address/spatial quality
- Transforming
- Parsing
- Neural network (machine learning)
- Complex processing
- Public tool repositories
- Web services
- Matching
- Profiling

RedPoint Global's Data Management Platform for Hadoop combines data storage and schema flexibility with a YARN-native parallel processing engine that's optimized specifically for DM that can scale across both traditional and Hadoop environments.

"Today, two-thirds of a quant's or data scientist's time is spent loading, cleaning, and preparing data for analytics and predictive modeling. You have these extremely smart, extremely knowledgeable, extremely rare people spending most of their time finding the data they need, getting access to it, moving or copying it, and preparing it to be analyzed," he points out.

"You can make quants and data scientists much more successful if you can reduce the time and effort required to get them up and running. If you can do this and also provide a place for IT and business people to collaborate, you can enable a single data scientist to span (i.e., service) multiple business departments, if not the entire enterprise. *That's* killer," he concludes.

"With Hadoop and its benefits (i.e., commodity distributed storage and general-purpose parallel processing), you can achieve a homogeneous data pool in record time with line of business collaboration, where any data can be easily integrated. This makes sense on another level. Data scientists are actually looking to integrate and enrich their EDW data with data from some of these new big data sets, and if you're a business analyst, you want to have raw data from operational systems so that you can track and look at behaviors/context/relationships over time. That's why you want to be able to do your data management on Hadoop, because it pays to take your work load processing to where the data is."

"With Hadoop and its benefits (i.e., commodity distributed storage and general-purpose parallel processing), you can achieve a homogeneous data pool in record time with line of business collaboration, where any data can be easily integrated."

RedPoint

www.redpoint.net

RedPoint Global is a recognized leader for empowering data-driven organizations to unlock the full value of their data and drive customer engagement with profitable, sustained growth. RedPoint Data Management delivers an innovative approach to make big data accessible to business users while buffering Hadoop's complexity.

From a single graphical interface, RedPoint makes it possible to enrich primary data sources with big data, within the same project workflow, and leverage in-house resources who possess data and use cases expertise. RedPoint Data Management provides a full set of data quality and data integration functions that include ETL, cleansing, matching, de-duping, merging/purging, householding, parsing, standardization, and keying with persistent entity resolution, and linkage. In addition, RedPoint scales to process the raw data granularity in Hadoop that exposes the consumer behaviors that fuel predictive modeling and machine learning.

In the 2015 Gartner Critical Capabilities for Data Quality Tools report, RedPoint received the highest product scores in the Operational/Transactional Data Quality and Data Integration Use Cases; second highest in Data Migration, and third highest in Big Data and Information Governance Initiatives, holding upper quartile ratings in 6 out of 6 use cases reviewed. For more information visit http://www.redpoint.net/tdwi or email: contact.us@redpoint.net.

tdwi
Advancing all things data.

tdwi.org

TDWI is your source for in-depth education and research on all things data. For 20 years, TDWI has been helping data professionals get smarter so the companies they work for can innovate and grow faster.

TDWI provides individuals and teams with a comprehensive portfolio of business and technical education and research to acquire the knowledge and skills they need, when and where they need them. The in-depth, best-practices-based information TDWI offers can be quickly applied to develop world-class talent across your organization's business and IT functions to enhance analytical, data-driven decision making and performance.

TDWI advances the art and science of realizing business value from data by providing an objective forum where industry experts, solution providers, and practitioners can explore and enhance data competencies, practices, and technologies.

TDWI offers five major conferences, topical seminars, onsite education, a worldwide membership program, business intelligence certification, live Webinars, resourceful publications, industry news, an in-depth research program, and a comprehensive website: tdwi.org.