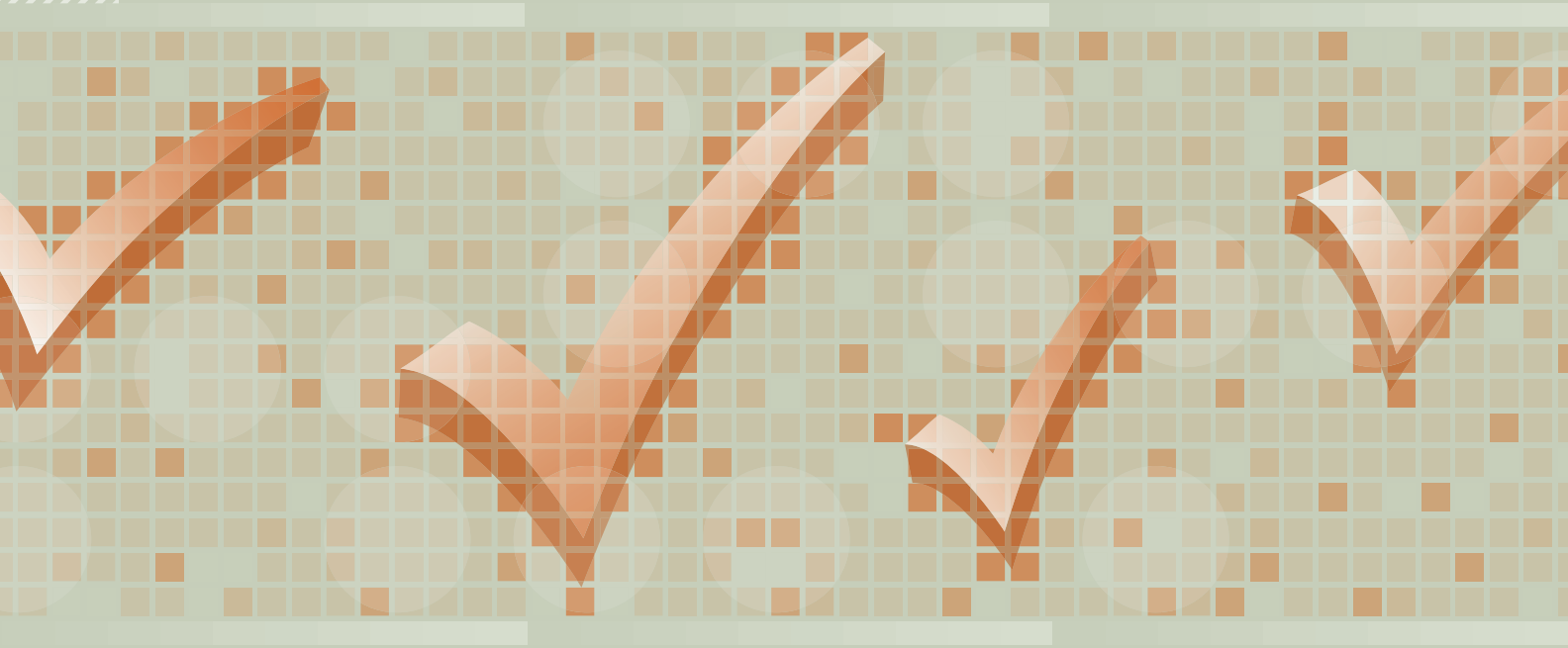
TDWI RESEARCH

TDWI CHECKLIST REPORT

ANALYTIC DATABASES FOR BIG DATA

By Philip Russom



Sponsored by

cloudera INFOBRIGHT PARACCEL™ SAP The Best-Run Businesses Run SAP™

tdwi.org

tdwi

OCTOBER 2012

TDWI CHECKLIST REPORT

ANALYTIC DATABASES FOR BIG DATA

By Philip Russom

TABLE OF CONTENTS

2 FOREWORD

3 NUMBER ONE
Query-Based Analytics

4 NUMBER TWO
Licensing and Deployment Options

5 NUMBER THREE
Storage Strategies

6 NUMBER FOUR
Computing Architectures

6 NUMBER FIVE
In-Database Analytics

7 NUMBER SIX
High-Performance Analytics

7 NUMBER SEVEN
Real-Time Analytics

8 ABOUT OUR SPONSORS

10 ABOUT THE AUTHOR

10 ABOUT THE TDWI CHECKLIST REPORT SERIES

10 ABOUT TDWI RESEARCH

1201 Monster Road SW, Suite 250
Renton, WA 98057

T 425.277.9126
F 425.687.2842
E info@tdwi.org

tdwi.org

© 2012 by TDWI (The Data Warehousing Institute™), a division of 1105 Media, Inc. All rights reserved. Reproductions in whole or in part are prohibited except by written permission. E-mail requests or feedback to info@tdwi.org. Product and company names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

FOREWORD

According to TDWI Research's 2011 Big Data Analytics Survey, 33% of surveyed organizations are contemplating a replacement of their analytic databases, data warehouses, and similar platforms to keep pace with new and intensifying requirements for advanced analytics in a "big data" world. As user organizations make such platform replacements—or add additional platforms to their expanding data warehouse architectures—they are turning more and more to specialized analytic database management systems (DBMSs). For example, a 2012 TDWI Technology Survey reveals that roughly half of organizations surveyed have already deployed one or more standalone analytic databases, and an additional third plan to deploy their first within three years.

A number of trends are driving the adoption of purpose-built analytic databases:

Organizations want more business value from big data. A recent shift in conventional wisdom now says that big data should be leveraged via analytics rather than treated as a cost center.

User organizations want more analytics, in general. For one thing, change is rampant in business, and analytics helps us discover both what has changed and how we should react. In addition, there are business opportunities to seize in the economic recovery, and analytics helps identify customer segments, profitable customers, products of affinity, sales seasonality, and so on.

Advanced analytic methods and big data volumes demand a purpose-built analytic database. Today, there's a slow trend away from practicing analytics and DW with relational DBMSs that were originally designed for online transaction processing (OLTP). As a viable alternative, analytic databases provide computing architectures designed for complex queries, analytic algorithms, high performance, terabyte-size scalability, and integration with other platforms in a distributed data environment.

This TDWI Checklist Report presents requirements for analytic DBMSs with a focus on their use with big data. Along the way, the report also defines the many techniques and tool types involved. The requirements checklist and definitions can assist users who are currently evaluating analytic databases and/or developing strategies for big data analytics.

DEFINING ANALYTIC DATABASES

A *database management system* (DBMS) is a vendor-built, enterprise-class software package designed to manage databases, whereas a *database* is a collection of data managed by a DBMS. Hence, an *analytic DBMS* (ADBMS) is a vendor-built DBMS designed specifically for managing data for analytics and data warehousing.

ADBMSs are most often optimized for "extreme SQL," which executes complex queries (that may scan terabytes of data) or routines (that may include hundreds or thousands of lines of SQL code). Some ADBMSs support other in-database or similar analytic processing, such as MapReduce, NoSQL parsing methods, and a variety of user-defined functions for data mining, statistical analysis, natural language processing (NLP), and so on.

Some vendors package or market their ADBMSs as data warehouse appliances, columnar DBMSs, analytic accelerators, in-memory DBMSs, multi-tool analytic platforms, and cloud-based or SaaS-based platforms. Of course, interest is high in the Hadoop family of open source products; although not a DBMS per se, certain combinations of Hadoop products are known for analytics with big data, and so they are viable alternatives to analytic databases, especially when diverse multi-structured data or complex analytic programming are involved.

DEFINING ADVANCED ANALYTICS

Advanced analytics is a term used to describe a collection of techniques, usually for predictive analytics, data mining, statistical analysis, and extreme SQL. Other forms include data visualization, artificial intelligence, NLP, and database capabilities that support analytics (e.g., in-database analytics, in-memory databases, or columnar data stores).

Instead of *advanced analytics*, however, better terms would be *discovery analytics* or *exploratory analytics*. That's because the user is typically a business analyst or data scientist who is trying to discover new business facts that no one in the enterprise knew before. To do that, the user needs large volumes of data that has many details, often sourced from data sets that are as yet untapped for analytics—in other words, big data.

The move into advanced analytics (beyond reporting and OLAP) is one of the strongest trends in BI today thanks to the drivers described early in this report. To guarantee successful analytics, users are adopting big data (as a broad source for discovery analytics) and analytic databases (as platforms purpose-built for high performance with big data and advanced analytics).

NUMBER ONE
QUERY-BASED ANALYTICS

Trends in analytics and DW are sometimes at odds—almost cancelling each other out—and that's currently the case with structured query language (SQL). Most organizations are deepening the amount and sophistication of their SQL usage, while a few others are seeking alternatives to it, as seen in so-called "NoSQL" databases (more on this below).

SQL. Many organizations rely heavily on SQL as the primary approach to advanced analytics. This makes sense because most BI professionals know SQL, and some know it well enough to hand code extreme SQL applications (defined in the next section). Furthermore, almost all tools for analytics, reporting, data integration, and data modeling support or generate SQL code that can be co-opted for analytics. In fact, anecdotal evidence suggests that analytics is driving up the amount of SQL usage across the BI/DW community, whether the SQL is hand coded, tool generated, or a mix of both. Hence, SQL is more important than ever, as are advanced SQL skills.

To supply users' demands for bigger and better SQL, most vendors' analytic databases are built from the bottom up for high performance with today's complex SQL. Furthermore, SQL-based analytic databases have good reputations for optimizing SQL that is of poor quality (due to user inexperience) or inconsistent (because it was pasted together from diverse sources).

NoSQL. SQL is designed to access tables, keys, and other data structures found in relational DBMSs, and access usually goes through a metadata layer. But what if the data is not structured (much less relational), and there is no metadata layer?

To manage and analyze non-relational data, a few organizations are embracing "NoSQL" databases, as well as similar non-DBMS data platforms such as the Hadoop Distributed File System (HDFS). This makes sense when the majority of data types analyzed are not relational and converting them to relational structures is not practical. In other situations, a NoSQL approach is useful when modeling and indexing data in relational schema would inhibit the discovery mission that many modern analytic projects are all about. For these situations, NoSQL databases or Hadoop are viable alternatives to more common analytic DBMSs that are relational with SQL support.

DEFINING EXTREME SQL

To make SQL *analytic*, you have to push it to extremes, as in the practice called *extreme SQL*. Here's how extreme SQL works:

A user collects data from multiple sources, adding or deleting data until the analytic data set represents the problem under study. This large volume of big data is mostly raw operational data, still in the schema of the source systems. It may be mixed with historic data from a data warehouse.

Because it is SQL-based, this form of analytics depends on intense querying. The analyst executes many ad hoc queries until the data reveals the answers needed. Each query becomes more complex with every iteration as the analyst adds more filtering, sources, and logic.

When extreme SQL is a reaction to a critical business event, the urgent need for an answer does not allow time and resources up front for data transformation or data modeling. Instead, these are expressed in SQL, which increases the code's complexity and length, sometimes amounting to hundreds or thousands of lines of SQL code.

Extreme SQL is not only reactive. In most cases, users calmly develop SQL routines following the best practices of programming and data management. The SQL code becomes extreme after years of enhancement, reuse, and automatic generation.

DEFINING NOSQL

A NoSQL database is a DBMS that manages data that is not relational. Thus the DBMS need not—or cannot—support SQL. Although it is not relational, NoSQL data may be structured according to other schema, such as records or value pairs, or the data may have no fixed schema. NoSQL databases are categorized according to schema and storage strategies such as document stores, key-value stores, BigTable implementations, and graph databases.

Most NoSQL DBMSs are designed to manage large, expanding data volumes, as is common with Web data and big data. A NoSQL database typically employs a distributed architecture with data stored redundantly across multiple servers. This way, the system has fault tolerance and can easily scale out horizontally with the addition of more server nodes.

Tools for analytics with NoSQL databases and similar file systems are rare today, so users must hand code their analytic solutions. Hence, NoSQL databases are most often used by programmers—only rarely by BI professionals.

NUMBER TWO
LICENSING AND DEPLOYMENT OPTIONS

There are several licensing and deployment options for analytic databases that users should consider before making a decision.

ENTERPRISE SOFTWARE LICENSES

As with most other types of enterprise software, most analytic databases are available via a traditional software license, to be deployed on server hardware that the user provides and sets up.

DATA WAREHOUSE APPLIANCES

By definition, an appliance involves server and storage hardware, not just DBMS software. However, the supplier of the hardware can vary. A true data warehouse appliance includes hardware along with an analytic database and all other components in a single, turn-key cabinet from a single source. In other cases, a software vendor provides an analytic database along with documentation of one or more reference configurations that can guide the user's purchase and configuration of third-party hardware. Sometimes, a system integrator assembles and sells an appliance. In yet other cases, a software or hardware vendor provides a product bundle that resembles a data warehouse appliance. In these and other scenarios, users get a unified platform optimized for analytics. The time to use is relatively short because little or no system integration work is required of the user.

CLOUD PLATFORMS FOR ANALYTICS

Although cloud computing and similar virtualization techniques are firmly established with operational applications today, clouds are just now starting to be used as platforms for BI, DW, and analytics. Cloud computing can serve as a performance and scalability strategy for analytics, due to clouds' fluid allocation and reapportionment of virtualized system resources. For example, according to users TDWI has interviewed, an analytic database may be set up in a cloud as an "analytic sandbox" to accommodate the large and fluctuating volumes of data (and isolate the unpredictable ad hoc query workloads) that are part and parcel of the work of business analysts and other power users. Furthermore, dynamic allocation and reallocation give the cloud provider fuller utilization of server resources, with less administrative work, as compared to traditional data center approaches. Hence, various types of clouds seem to be a good match for analytic databases and related practices.

SOFTWARE-AS-A-SERVICE (SAAS) FOR ANALYTICS

SaaS may refer to Internet-based software applications that happen to run in a public cloud, or it may refer to a cloud as a platform service upon which you build your own applications. SaaS doesn't necessarily require a cloud, but a few analytic databases are available via SaaS licenses, hosted on a vendor's secure public cloud. For BI users, finding SaaS-based analytic applications that meet your business requirements can be challenging. But SaaS also has appealing benefits such as straightforward software updates for SaaS applications and easy outsourcing of IT and BI infrastructure. Even so, when deploying data outside the enterprise, users must ensure airtight security for those platforms and also ensure they meet government regulations with jurisdiction over any protected information.

OPEN SOURCE FOR ANALYTICS

Open source products were rather rare in users' portfolios of BI/DW products as recently as 2007. For a variety of reasons, open source products that are useful for BI/DW have come out of nowhere to be almost common today. First, Linux led the way and proved that an open source product could be up to the most stringent enterprise requirements for performance, reliability, security, and—of course—low cost. The budget lockdown of the recession led many professionals to try out and eventually use open source products for data integration, reporting, analytics, and database management systems.

Furthermore, let's not forget that many of the columnar databases and data warehouse appliances we trust today began with code from open source databases. More recently, the need to leverage big data and multi-structured data with analytics is driving up the usage of open source Hadoop, helping it propagate from its Internet origins to mainstream firms in several industries.

Although open source software is not a primary requirement for analytics, BI/DW professionals and others should consider the new level of mature functionality, innovation, performance, and low cost of acquisition that open source analytic databases have to offer.

NUMBER THREE
STORAGE STRATEGIES

The importance of data storage grows as big data grows. Data processing is inherently I/O-bound, which can be a performance bottleneck; so, depending on data and the analytic questions asked of it, advanced analytics may likewise be I/O-bound. Here's what to look for in an analytic database if you need to get the most performance out of storage:

In-memory databases avoid most I/O operations. An influx of in-memory databases for analytics has been enabled by 64-bit computing and slowly dropping prices for server memory. For example, the in-memory database may be a cube for OLAP, a table of metrics for performance management, analytic models that are rescored in real time, or an entire relational database.

Solid-state drives (SSDs) greatly accelerate I/O. Traditional, spinning hard drives have become larger, cheaper, and more reliable—but no faster. Based on flash memory, with no moving mechanical parts, SSDs provide a speedy (albeit expensive) alternative. Although they are still rare in IT, a number of data warehouse appliances already incorporate SSDs.

Fully parallelized data loaders are indispensable. This is especially true with big data. Some form of trickle loader for your analytic database may be required for real-time analytics.

The columnar data store optimizes physical storage. It automatically collates data into proximal locations on disk that are naturally optimized for the columnar queries of analytics. As a bonus, storing group data by column compresses data dramatically and alleviates the need for most custom indexes.

DEFINING COLUMNAR DATABASES

Many people are confused about what a columnar database is and does, so let's clear up the confusion with some quick definitions of relevant terms:

Relational database. A DBMS where data is modeled or structured using tables to group related data and keys to further define relationships between tables.

Row-oriented data store. In most relational DBMSs, data is physically stored as table rows. This makes sense in OLTP environments, where each row is a transaction that is committed to or retrieved from the database, but it makes less sense with analytic queries, which usually parse columns.

Columnar data store. Data is physically stored by table columns, which represent attributes of a record, even though the data is structured in a relational model. The close proximity speeds up the DBMS's retrieval of data for a specific column. It also enables the DBMS to create statistics and lists about the content of columns, plus heavily compress data, which in turn speeds up columnar queries.

Columnar database. This is a relational DBMS with an integral columnar data store. The entire DBMS (not just the columnar storage engine) is optimized for columnar queries.

A columnar DBMS is also an analytic DBMS. Analytics requires high performance for complex queries against large data sets, and that's what the columnar DBMS enables.

NUMBER FOUR
COMPUTING ARCHITECTURES

SMP AND MPP FOR ANALYTICS

Symmetrical multi-processing (SMP) has seen three decades of successful use in data warehousing and analytics. Roughly two-thirds of data warehouses are running on an SMP platform today, showing that SMP is still the mainstay when it comes to computing architectures for warehousing and analytics. For most DWs, SMP provides the performance, scalability, and straightforward maintenance needed. Yet, one-third of organizations are planning to migrate away from SMP.

Massively parallel processing (MPP) is the computing architecture to which many organizations plan to migrate. MPP has been around since the 1980s, but has seen prominent use only in recent years. Most of the new analytic database brands that have arrived recently are based on MPP, and these databases have helped evangelize MPP's advantages for warehousing and analytics.

MPP involves multiple nodes that work in parallel on subsets of the same computational problem, but without sharing memory or other resources. This is good for problems that can be parallelized effectively, including most analytic computations. MPP also supports the trend toward scaling out instead of scaling up. On the downside, large multi-node MPP configurations can be complex to set up and maintain, which may not be practical for the departmental use that is common with analytic databases.

DATA WAREHOUSE ARCHITECTURES FOR ANALYTICS

As organizations adopt more analytic methods and deploy more analytic applications, they introduce more data-oriented workloads into their BI and DW environments. When the architecture of a DW is designed and optimized for the most common deliverables—namely standard reports, dashboards, and OLAP—it makes sense to put other workloads (including those for advanced analytics) on separate platforms on the edge of the DW (instead of inside the DW proper). This, in turn, leads to a distributed architecture for the DW. Thus, the introduction of workloads for advanced analytics is one of the prominent trends driving DWs toward distributed architectures.

Distributed DW architectures are both good and bad. They're good if your fidelity to business requirements and DW performance leads you to a standalone analytic database, and that database integrates well with the other platforms in the distributed architecture. But they're bad when edge systems proliferate uncontrolled, like the errant data marts we all fear. So far, the new generation of analytic databases are controlled and governed by users far better than the marts of yore. But you have to be diligent with edge systems of all types to avoid abuses.

NUMBER FIVE
IN-DATABASE ANALYTICS

For many years, most analytic tools based on data mining, statistical analysis, and other non-relational approaches required that analytic data managed in a relational DBMS be moved from that database into a flat file or special tool environment for processing. That's because these tools were optimized for high performance with flat files or their own proprietary environments, and because they tend to make many passes through a data set, which exacerbates workloads for DBMS access.

There were many problems with this process. It is time-consuming and therefore antithetical to real-time operation. As data evolves into big data, it becomes ever less tenable to move, extract, and transform large volumes. The process demands tools and development for ETL, plus a hefty data staging area.

As an antidote to this problem, a new trend takes analytic algorithms to analytic data, instead of vice versa. In-database analytics is the most prominent technique for this new analytic data paradigm.

In-database analytics is typically enabled by executing user-defined functions (UDFs) in an analytic database, where each UDF contains one or more algorithms for mining, statistics, predictive models, natural language processing, and so on. Some of these UDFs are miniaturized versions of vendor tools for advanced analytics; others are hand coded by users. UDFs aside, some analytic platforms enable in-database analytics via a library within the DBMS, which simplifies the creation and invocation of analytic logic.

In-database analytics enhances speed and scale for large-volume analytics. Depending on how you set it up, it also reduces architectural complexity. And it's all about leveraging the power of an analytic DBMS or other relational DBMS. For these reasons, users are adopting in-database analytics aggressively. In fact, it is the fastest-growing option for high-performance data warehousing, according to a 2012 TDWI Research survey.

Note that functionality available via in-database analytics varies and will continue to evolve. When evaluating an analytic database, ascertain what forms of UDFs, which vendors' analytic tools, and what UDF languages (for programming, queries, modeling, analytics, and so on) are supported by the analytic database.

NUMBER SIX
HIGH-PERFORMANCE ANALYTICS

Just about everything we do nowadays in BI, DW, and analytics has some kind of performance requirement, for both business and technology reasons. Luckily, the latest generation of analytic databases is doing its part to enable high performance, as are the hardware components that databases run on.

Parallel processing is the true enabler behind many high-performance capabilities. For example, think of the SMP and MMP computing architectures discussed earlier. Ideally, every component of an analytic database should be parallelized, from query optimizers to data loaders. Hadoop coupled with MapReduce is a great example; this combination provides distributed parallel processing and data collection for a wide range of processing—including analytics. Furthermore, most analytic database components should take advantage of the parallel capabilities of hardware. For example, multi-core CPUs available at reasonable prices have recently taken software to a high level of performance that wouldn't be possible without them.

Workload management tools are built into most vendor brands of DBMS, but tool capabilities vary. A good workload management tool enables you to define resources and execution characteristics for individual workloads, plus rules for how concurrent mixed workloads will share resources for job priority, load balancing, and availability.

Server memory prices are dropping, so more organizations are upgrading their BI, DW, and analytic servers with additional memory; 256 gigabytes seems common, although some systems are treated to a terabyte or more. "Big memory" speeds up extreme SQL, multi-way joins, and analytic model rescores, thanks to fatter pipes and less I/O to land data to disk. It also enables the new generation of in-memory databases described earlier.

Hard drives have become dirt cheap while their capacity and reliability have improved. However, they are still based on spinning disks, which haven't sped up much. To compensate, some data warehouse appliances include their own local disks with the ability to execute data processing logic at the storage level, which offloads the server and reduces the amount of data moved to the CPU. **Columnar data stores** likewise help compensate for slow seek speeds in storage. **Solid-state drives (SSDs)** have seek speeds that are a fraction of traditional drives, which is great if the data needed is already on a SSD.

NUMBER SEVEN
REAL-TIME ANALYTICS

Real-time operation is the most influential trend in BI, DW, and analytics today. It's driven by the business need to receive and react to time-sensitive data as soon as possible. For example, over the last 10 years, operational BI has incrementally marched from offline operational reports to on-demand management dashboards. Now it's time for operational analytics to make a similar journey into real time.

Real-time practices such as operational BI and operational analytics require fresh data, and the definition of "fresh" varies from one time-sensitive business process to another. To enable the multiple data speeds and frequencies required, analytic databases must support a number of capabilities or integrate with third-party tools that support them.

In-memory databases and similar in-memory data caches have become common in recent years. For example, the practices of performance management and operational BI assume that business users can refresh dashboards and other reports frequently to get fresh data. When the metrics, KPIs, and operational data assumed by these applications are cached in server memory or managed as an in-memory database, refreshes are fast.

Streaming data shoots out of a number of systems in real time, including Web servers, robots and other machinery, sensors, social media, supply chains, RSS feeds, and so on. This is one of the reasons big data is getting so big. All these sources produce valuable information, and some of it merits analytic processing in real time. Hence, an analytic database needs high-speed bulk loaders and trickle-feed interfaces to capture and process streaming data. For many applications, an analytic database should be complemented by an analytic tool for **complex event processing (CEP)**, which can be programmed to spot opportunities and problems in streaming data in real time.

High availability is a requirement that is often overlooked. Operational BI and operational analytics are not real time unless the databases involved are up and running 24/7. Hence, an analytic database that supports real-time applications must have a strategy for high availability.

ABOUT OUR SPONSORS

cloudera

www.cloudera.com

Cloudera, the leader in Apache Hadoop–based software and services, enables data-driven enterprises to easily derive business value from all their structured and unstructured data. Cloudera's Distribution including Apache Hadoop (CDH), available to download for free at www.cloudera.com/downloads, is the most comprehensive, tested, stable, and widely deployed distribution of Hadoop in commercial and non-commercial environments. For the fastest path to reliably using this completely open source technology in production for big data analytics and answering previously un-addressable big questions, organizations can subscribe to Cloudera Enterprise, comprised of Cloudera Manager software and Cloudera Support. Cloudera also offers training and certification on Apache technologies, as well as consulting services. As the top contributor to the Apache open source community and with tens of thousands of nodes under management across customers in financial services, government, telecommunications, media, Web, advertising, retail, energy, bioinformatics, pharma/healthcare, university research, oil and gas, and gaming, Cloudera's depth of experience and commitment to sharing expertise are unrivaled.

INFOBRIGHT

www.infobright.com | www.infobright.org

Infobright's columnar analytic database is designed for applications and data marts that analyze large volumes of "machine-generated data" such as Web data, network logs, telecom records, stock tick data, and sensor data. It is an easy-to-use, low-cost, and high-performance solution, providing:

- Fast query performance, especially for ad hoc analysis, even as data volume grows
- Outstanding load speed—it can load 2 TB per hour into a table
- It is a self-tuning database that eliminates the "DBA tax": no indexes, no projections, no manual tuning
- Industry-leading data compression, reducing storage requirements by up to 90%, which lets you store much more history online, with far less hardware than any other approach
- Scales from gigabytes to over 50 TB of data using an industry-standard server
- Available as a free, open source edition and an enterprise edition product

Infobright has more than 300 customers and 1,000 installations. It is used by 8 of the top 10 global telecommunications service providers for near-real-time analysis of telecom network and call detail data, and by leading online and mobile advertising companies for Web data analytics. Infobright is especially well suited as the embedded analytic database within a SaaS or ISV solution due to its fast query performance, ability to store much more data in less space, and attractive licensing model. To learn more, visit us at www.infobright.com and www.infobright.org.

ABOUT OUR SPONSORS CONTINUED

ParAccel™

www.ParAccel.com

ParAccel Analytic Platform extends all of the power of an analytic database to deliver high-performance, deep analytics on massive amounts of data for the entire interactive datasphere (relational, machine, Web). Companies already using ParAccel consistently deliver an environment for unconstrained analytics, driving business momentum, innovation, and competitive advantage.

ParAccel platform capabilities span the entire TDWI checklist:

- ✓ ADBMS
- ✓ In-memory option
- ✓ In-database analytics
- ✓ Analytic library
- ✓ Extensibility via UDF
- ✓ On-demand platform interaction
- ✓ On-demand data integration
- ✓ On-demand access to streaming data (2.5M records/second)
- ✓ Massive scalability (100s of TB)
- ✓ Fast parallel load (40 TB/hour)
- ✓ Columnar database (not just store)
- ✓ Compression (10+ times)
- ✓ MPP
- ✓ Real-time analytics
- ✓ High availability
- ✓ SQL access
- ✓ SQL optimization
- ✓ Extreme SQL optimization
- ✓ Execution optimization
- ✓ Hadoop analytics (interactive, bi-directional, data, and processes)
- ✓ High-performance analytics for data in NoSQL archives
- ✓ Discovery analytics
- ✓ Ad hoc queries
- ✓ On demand queries
- ✓ No indexing required
- ✓ No projections required
- ✓ Enterprise licensing
- ✓ Offerings: appliance, cloud, SaaS, commodity hardware
- ✓ SSD support
- ✓ Distributed analytic processing
- ✓ Workload management
- ✓ Interactive workload management
- ✓ Big memory
- ✓ SAN integration

SAP® The Best-Run Businesses Run SAP™

www.sap.com

As market leader in enterprise application software, SAP (NYSE: SAP) helps companies of all sizes and industries run better. Founded in 1972, SAP (which stands for "Systems, Applications, and Products in Data Processing") has a rich history of innovation and growth as a true industry leader. Today, SAP has sales and development locations in more than 75 countries worldwide. SAP applications and services enable more than 109,000 customers worldwide to operate profitably, adapt continuously, and grow sustainably. By using SAP analytic databases, enterprises can capture, store, and analyze massive volumes of data in real time. SAP HANA is a completely reimagined real-time database platform that streamlines applications, analytics, planning, and predictive and sentiment analysis to make your business operate in real time. SAP Sybase IQ has more deployments than any other column database in the market and supports petabyte-sized data warehousing environments.

ABOUT THE AUTHOR

Philip Russom is the research director for data management at The Data Warehousing Institute (TDWI), where he oversees many of TDWI's research-oriented publications, services, and events. He's been an industry analyst at Forrester Research and Giga Information Group, where he researched, wrote, spoke, and consulted about BI issues. Before that, Russom worked in technical and marketing positions for various database vendors. Over the years, Russom has produced over 500 publications and speeches. You can reach him at prussom@tdwi.org.

ABOUT THE TDWI CHECKLIST REPORT SERIES

TDWI Checklist Reports provide an overview of success factors for a specific project in business intelligence, data warehousing, or a related data management discipline. Companies may use this overview to get organized before beginning a project or to identify goals and areas of improvement for current projects.

ABOUT TDWI RESEARCH

TDWI Research provides research and advice for business intelligence and data warehousing professionals worldwide. TDWI Research focuses exclusively on BI/DW issues and teams up with industry thought leaders and practitioners to deliver both broad and deep understanding of the business and technical challenges surrounding the deployment and use of business intelligence and data warehousing solutions. TDWI Research offers in-depth research reports, commentary, inquiry services, and topical conferences as well as strategic planning services to user and vendor organizations.