International Journal of Research in Engineering, Science and Management
Volume-1, Issue-12, December-2018
www.ijresm.com | ISSN (Online): 2581-5792

Challenges in Handling Databases

Rahul Paldiya
Assistant Professor, Department of Computer Science, Ujjain Engineering College, Ujjain, India

***Abstract*: The world of IT has grown to the point that the rate at which data is being generated calls for appropriate techniques and strategies to handle such large data set called Big Data through an appropriate database management system beyond the traditional DBMS. As at 30 years ago, data of size 100GB could have been regarded as a very large data, but in today's environment, a new database deployment in a large IT industry might start at 1TB meaning that 100GB may be regarded today as a small database size. The transition from desktop computing to mobile computing also has increase the rate of the usage of mobile devices since these devices are mobile, cheap, easily purchased and easily programmed, though having relatively low processing capabilities. As 100GB of database size could be regarded as a small database size for desktop applications, 50GB as well could be regarded as a very large database size for mobile applications. Hence, "Large" is a relative term that changes with time and the nature of the device in question. This paper aims at reviewing the challenges associated with very large databases and recommends the best techniques/strategies to those challenges.**

***Keywords*: Very Large Database, Data Warehouse, Data Mining, OLAP, OLTP, Big Data, Database Partitioning.**

1. Introduction

Very Large Database (VLDB) is typically a database that contains an extremely high number of tuples (database rows or records) or occupies an extremely large physical file system storage space due to wide tables with large number of columns or due to multimedia objects (such as videos, audios, images). However, the most common definition of VLDB is a database that occupies more than 1TB or contains several billion rows. Data Warehouses, Decision Support Systems (DSS) and On-Line Transaction Processing (OLTP) systems serving large numbers of users would fall into this category. Over the years, the definition of a Very Large Database (VLDB) has radically changed. "Large" is a relative term that changes with time, what was large ten or twenty years ago could be small by today's standards, and what is large today will not be so in a few years from now. It is not uncommon to find databases in the tens to hundreds of TB's and even PB's today serving traditional data warehouse and increasingly On-Line Transaction Processing (OLTP) activities.

A. Challenges of large database

- The Very Large Database challenges include the following: There is a steady growth in the size of the database.
- There is no minimum absolute size for a very large database.
- It is not cost effective to perform operations against a system of such size.
- What are the best ways to capture, manage, backup and recovery data in a very large database systems

B. Categories of very large databases

The categories of Very Large Databases include the following:

- Data Warehouses and OLAP (On-Line Analytical Processing) Systems
- Operational Databases such as OLTP (On-Line Transaction Processing) systems.
- Big Data in a Very Large Database

2. Data warehouse

Data warehouse is a database that stores current and historical data of potential interest to managers through the organization. This data originates from operational and historical systems (internal data sources) as well as external systems (external data sources). Data from internal data sources and external data sources are extracted and transformed into the data warehouse. External data sources here could be transactions from websites, data from structured database model (such as relational database model, object-oriented database model, distributed database model, Hierarchical database model, Network database model etc.), or data from unstructured data model (such as HTML, XML, text file, flat file, spreadsheets etc.). Data from these divers' sources are extracted, transformed, and are standardized into a common data model and consolidated so that they can be used across the enterprise for management analysis and decision-making purposes. The data in a data warehouse are made read only data and are available for anyone to access. It does not require frequent update and it helps executives to organize, understand, and use their data during their decision-making process. A data warehouse is designed to support analysis and decision-making. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources. Data warehouse environment can include an Extraction, Transformation, and Loading (ETL) process, On-Line Analytical Processing (OLAP), Data Mining capabilities, Queries and Reports, and other applications that manage the process of gathering data and delivering it to business users.

A. Data warehouse extract, transform, and load (ETL) operations

- *Data Extraction:* This involves gathering data from heterogeneous sources.
- *Data Transformation:* This means converting data from its source format to data warehouse format.
- *Data Load:* This involves sorting, summarizing, consolidating, checking integrity, building indexes and creating partitions.

B. On-line analytical processing (OLAP) system

OLAP system is used to analyze an existing data in a data warehouse, generate summaries, calculate sum/aggregates, and exposes the data pattern because the data in an OLAP system rarely changes. Therefore, the data in an OLAP system can be optimized to perform complex operations such as calculating summaries and aggregates. OLAP requires special data organization, access methods, and implementation methods, not generally provided by commercial DBMSs targeted for OLTP. It is for all these reasons that data warehouses are implemented separately from operational databases. OLAP system also supports data analysis technique called Data Mining.

C. Data Mining

Data mining is the use of statistical tools to extract information from a very large database. Data mining software finds hidden patterns and relations in a large pool of large data and generates rules that would be used to predict future behavior and guide the managers for their decision-making. Database mining activities differ from traditional database interrogation in that data mining seeks to identify previously unknown patterns as opposed to traditional database inquiries that merely ask for the retrieval of stored facts. Moreover, data mining is practiced on static data collections, called data warehouses, rather than “online” operational databases. Data-mining tools use advanced techniques from knowledge discovery, artificial intelligence, neural networks etc. to obtain “knowledge” and apply it to business needs. The discovered knowledge is then used to predict/forecast the outcome of an event. Data mining describes a new breed of specialized decision support tools that automate data analysis. In short, data mining tools initiate analysis to create knowledge. Such knowledge can be used to address any number of business problems. For example, banks and credit card companies use knowledge-based analysis to detect fraud, thereby decreasing fraudulent transactions. The explosive growth of databases makes the scalability of data-mining techniques increasingly important.

D. Benefits of data warehouse

The following are the basic benefits of a data warehouse:

- It does not only provide important information, but it provides improved information that can be used by decision makers.
- It can model and remodel data.
- It enables decision makers to access data as often as possible without affecting the overall performance of the operational system in question.
- Data extracted from a data warehouse through data mining and OLAP systems help organizations refocus on their business.

E. Scalability requirements of data warehouses

Data Warehouses often contain large tables and require techniques both for managing these large tables and for providing good query performance across these large tables. Database Partitioning helps scaling a data warehouse by dividing database objects into smaller pieces, enabling access to smaller and more manageable objects. Having direct access to smaller objects addresses the scalability requirements of data warehouses:

- *Bigger Databases:* The ability to split a large database object into smaller pieces transparently provides benefits to manage a larger total database size. You can identify and manipulate individual partitions and sub-partitions in order to cope with large database objects.
- *Bigger Individual tables:* It takes longer to scan a big table than it takes to scan a small table. Queries against partitioned tables may access one or more.

3. Online transaction processing systems

Online Transaction Processing (OLTP) systems are one of the most common data processing systems in today’s enterprises. Classical examples of OLTP systems are order entry, retail sales, and financial transaction systems. OLTP applications typically automate clerical data processing tasks (such as banking transactions, telecom transactions etc.).

A. Characteristics of an OLTP system

The basic characteristics of an OLTP system include:

- *Short response time:* The nature of OLTP environments is predominantly any kind of interactive ad hoc usage, such as telemarketers entering telephone survey results. OLTP systems require short response times for users to remain productive.
- *Small transactions:* OLTP systems normally read and manipulate highly selective, small amounts of data; the data processing is mostly simple and complex joins are relatively rare. There is always a mix of queries and DML workload.
- *Data maintenance operations:* It is not uncommon to have reporting programs and data updating programs that need to run either periodically or on an ad hoc basis. These programs, which run in the background while users continue to work on other tasks, may require many data-intensive.
- *Large user populations:* OLTP systems can have

immeasurably large user populations where many users are trying to access the same data at once.

- *High concurrency:* Due to the large user population, the short response times, and small transactions, the concurrency in OLTP environments is very high.
- *Large data volumes:* Depending on the application type, the user population, and the data retention time, OLTP systems can become very large.
- *High availability:* The availability requirements for OLTP systems are often extremely high. An unavailable OLTP system can impact a very large user population, and organizations can suffer major losses if OLTP systems are unavailable.
- *Lifecycle related data usage:* Similar to data warehousing environments, OLTP systems often experience different data access patterns over time.

4. Very large database and big data

Big Data simply means complex and very large data sets from new data sources, voluminous enough that traditional data processing software tools are inadequate to process and manage them. Big Data makes it possible for one to gain more and complete answers to unharnessed problems in a business environment. More and complete answers simply mean having more confidence in the data which involves having a completely different approach to problems solving.

A. *Characteristics of big data*

- *Volume:* The quantity of generated and stored data is very large. *Velocity:* The speed at which the data is being generated is very fast.
- *Variety:* The nature of data varies (video, audio, geospatial data, 3D data, text files etc.).
- *Veracity:* The quality of data can vary (data inconsistency), affecting the accuracy of analysis

5. Very large database partitioning

The key component for solving some of the challenges associated with very large databases is partitioning. Partitioning addresses key issues in supporting very large tables and indexes by letting you decompose them into smaller and more manageable pieces called partitions, which are entirely transparent to an application. SQL queries and Data Manipulation Language (DML) statements do not need to be modified in order to access partitioned tables. However, after partitions are defined, Data Definition Language (DDL) statements can access and manipulate individual partitions rather than entire tables or indexes. This is how partitioning can simplify the manageability of large database objects.

A. *Benefits of partitioning*

- *Performance:* Partitioning provides several performance benefits by limiting the amount of data to be examined or operated on, and by providing data distribution for parallel execution.
- *Manageability:* Partitioning allows tables and indexes to be partitioned into smaller, more manageable units, providing database administrators with the ability to pursue a "divide and conquer" approach to data management. It enables database designers and administrators to tackle some of the toughest problems posed by cutting-edge applications.
- *Availability:* Storing different partitions in different table-spaces allows the database administrator to do backup and recovery operations on each individual partition, independent of the other partitions in the table. Partitioning is a key tool for building multimegabit systems or systems with extremely high availability requirements.

B. *Partitioning strategies*

Using data distribution methods such as Range, Harsh, and List, a table can be partitioned either as a single list (Single-Level Partitioning) or as a composite partitioned table (Composite Partitioning).

1) *Single-level partitioning*

In a single-level partitioning, a table is defined by specifying one of the data distribution methodologies (Range Partitioning, List Partitioning, and Hash Partitioning), using one or more columns as the partitioning key.

- *Range Partitioning:* Range partitioning divides a table into partitions based on a range of values. You can use one or more columns to define the range. It is the most common type of partitioning and is often used with dates.
- *List Partitioning:* List partitioning enable you to explicitly control how rows map to partitions by specifying a list of discrete values for the partitioning key in the description for each partition. The advantage of list partitioning is that you can group and organize unordered and unrelated sets of data in a natural way.
- *Hash Partitioning:* Hash partitioning is the ideal method for distributing data evenly across devices. Hash partitioning is also an easy-to-use alternative to range partitioning, especially when the data to be partitioned is not historical or has no obvious partitioning key. Hash partitioning maps data to partitions based on a hashing algorithm to the specified partitioning key. The hashing algorithm evenly distributes rows among partitions, giving partitions approximately the same size.

2) *Composite partitioning*

Composite partitioning is a combination of the basic data distribution methods. A table is partitioned by one data distribution method and then each partition is further subdivided into sub-partitions using a second data distribution method. All sub-partitions for a given partition together represent a logical subset of the data.

6. The very large databases problem

One problem with Very Large Databases (VLDB) is 'How do we backup and recover a VLDB of 50TB – 100TB in size?'

A. What is the best way to manage data?

One of the best solutions to managing VLDB data is to:

- Store data in its native format so that the recoveries are instant.
- Keep daily copies for a few days, weekly copies for a month, and monthly copies for a few months. This ensures a minimum of storage consumption with the incremental forever strategy to satisfy retention needs (cost effectively).
- Compress static data to further reduce storage consumption
- Replicate data to remote clouds or sites with only changed blocks and compression to reduce bandwidth consumption.

B. What is the best way to recover data?

The best solution to recover data for VLDB is to:

- Restore a full VLDB from a reduplication appliance (an appliance that eliminates redundant duplicate data).
- Select the point in time usually an incremental backup from wherever it resides, and the backup Server will have to restore that incremental, as well as the prior full backup.
- Then both the full and incremental database backup images will be rehydrated from their reduplicated state, so the backup software can read the data in its proprietary backup format, combine them, and then converting and writing the database back to the target host in the native application format.

7. Conclusion

We have reviewed Very Large Databases, and examined the challenges associated with them. We also, discussed strategies for solving those challenges; and why we believe those strategies such as partitioning will continue to enhance the performance, manageability, and availability of a wide variety of database applications and help in reducing the total cost of ownership for storing large amounts of data for many years to come. Partitioning, indeed, has emerged as a new methodology and a critical feature for solving and managing problems associated with very large databases. We also discussed the trends responsible for the steady growth in database size as well as the categories of very large databases including Data Warehouse and On-Line Analytical Processing (OLAP) Systems, Operational Databases such as On-Line Transaction Processing (OLTP) Systems, and the concept of Big Data with respect to Very Large Databases (VLDBs). We went further to recommend that the best approach to complex and very large data sets from new data sources, voluminous enough that traditional data processing software tools are inadequate to process and manage them is to keep such data in the cloud than using the traditional approach. For the purpose of selecting the right tools for the pool of our big data sets, we analyzed and compared the very large database big players and their degree of support for data warehouses of all sizes. At the same time, we analyzed the nature of the architecture and the level of functionality they have provided in the market and how consistent they have demonstrated customer satisfaction, strong support and strong vision with respect to VLDs emerging architectures and functionalities over the years. Finally, we concluded by answering the big questions with respect to very large databases such as the best way to capture data, the best way to manage data, and the best way to recover data.

References

[1] Actifio (n.d.). The Very Large Database Problem: How to Backup & Recover 30–100 TB Databases. http://cdn2.hubspot.net/hubfs/214442/Actifio_For_Very_Large_Databases_White_Paper.pdf

[2] Belden, E., Avril, P., Baer, H., Dijcks, J.-P., Fogel, S., Ganesh, A, Wie, M. Van de. (2016). Oracle ® Database.