Chapter 3
Database Systems for Big Data Storage and Retrieval

Venkat Gudivada
East Carolina University, USA

Amy Apon
Clemson University, USA

Dhana L. Rao
East Carolina University, USA

ABSTRACT

Special needs of Big Data applications have ushered in several new classes of systems for data storage and retrieval. Each class targets the needs of a category of Big Data application. These systems differ greatly in their data models and system architecture, approaches used for high availability and scalability, query languages and client interfaces provided. This chapter begins with a description of the emergence of Big Data and data management requirements of Big Data applications. Several new classes of database management systems have emerged recently to address the needs of Big Data applications. NoSQL is an umbrella term used to refer to these systems. Next, a taxonomy for NoSQL systems is developed and several NoSQL systems are classified under this taxonomy. Characteristics of representative systems in each class are also discussed. The chapter concludes by indicating the emerging trends of NoSQL systems and research issues.

INTRODUCTION

In the recent years, data is being generated at unprecedented levels. This data is popularly referred to as Big Data. Internet of Things (IoT) applications typically collect streaming data in real time from millions of devices and sensors. According to IDC (Gens, 2015), in 2015, IoT applications will be driven by 15 billion devices and the number of these devices will reach 30 billion by 2020.

DOI: 10.4018/978-1-5225-3142-5.ch003

Currently the data managed by Relational Database Management Systems (RDBMS) is primarily structured. However, unstructured data is growing too rapidly. In 2015, estimated RDBMS data was 32 PB whereas email data was 44 PB and unstructured data was over 226 PB (Dhar, 2013). Natural language in written and spoken forms, graphics, animation and video comprise unstructured data. Email data is semi-structured and falls between structured and unstructured data. Many organizations are exploring ways and means to leverage Big Data to improve their products and services. Typically Big Data is acquired from multiple vendors and integrated through various data transformations. Data vendors are not necessarily the producers of data. Often, these vendors themselves procure data from other vendors, transform and integrate the data to bring additional value and resell the data. Under this scenario, it becomes critical to know the genealogy of data to assess its suitability for a given task.

The overarching goal of this chapter is to examine issues and systems related to storing and retrieving Big Data. More specifically, the chapter begins with describing functional inadequacy of RDBMS. Special data management needs of Big Data applications are discussed next. Some Big Data applications require only a subset of highly optimized RDBMS functionality and configurable data consistency levels. They also employ distributed architectures to achieve horizontal scalability. Furthermore, techniques such as data partitioning, replication, versioning and compression are required to address data volumes and query latency requirements. Massive parallel processing techniques such as MapReduce are used to process ad hoc and compute-intensive queries.

The second part of the chapter enumerates data management needs of Big Data applications. Also, it discusses a wide range of choices provided by NoSQL systems for Big Data Management. The third part of the chapter provides taxonomy for data management systems for Big Data, describes fundamental characteristics of each class and lists representative systems. The fourth part describes data management trends for Big Data and concludes the chapter by indicating future research directions.

FUNCTIONAL INADEQUACY OF RELATIONAL DATABASE MANAGEMENT SYSTEMS (RDBMS) FOR BIG DATA APPLICATIONS

Underlying all the RDBMS (Relational Database Management Systems) is the relational data model for structuring data and the ISO/ANSI standard SQL for data manipulation and querying. The relational data model is based on first-order predicate logic and lends itself naturally for providing a declarative method for specifying queries on the database (Codd, 1970). The SQL language is originally based on relational algebra and tuple relational calculus (Chamberlin, 1974).

RDBMS is inherently inadequate to address the data management needs of Big Data applications. Because semantically related data is fragmented across various tables, it requires several joins to process typical queries. Query latency times are simply too high for these applications. Moreover, impedance mismatch problems arise in RDBMS due to the difference between the relational data model structures on the disk and in-memory data structures of applications. Often Object Relational Mapping (ORM) frameworks such as Hibernate are used to automatically generate the code needed to map relational structures to in-memory application data structures. Though these frameworks help in ORM code generation, the same code exacerbates the query latency times.

RDBMS is principally designed to work on single node computer. RDBMS accommodates increased workload and data volume by increasing computing power through faster CPUs, more memory and bigger and faster disks. This is referred to as vertical scaling. However, RDBMS vendor solutions for vertical