The World of Databases in 2020

Stephen H. Gregory

The University of Alabama

CS 301: Database Management Systems

Dr. Susan Vrbsky

October 20, 2020

The World of Databases in 2020

The topic of database management in 2020 is more relevant and spans a greater breadth than ever before. In the 20th century, the world of databases was much simpler; Structured Query Language (SQL) was created in the 70s to handle meticulously well-framed static data many years before Tim Berners-Lee's invention of the World Wide Web, and reigned as the de facto standard for managing the world's data until the time at which Sir Lee's said invention exploded into popularity. However, the exponential growth of data in the world today poses new, exciting challenges for database management, and the aforementioned methods of old do not suffice. Advancements in computational power and artificial intelligence, as well as the capability and ubiquity of the internet, have created a paradigm transformation; data is no longer static, and the amount of power that remains to be leveraged through enormous swaths of data lurking in the ever expanding reaches of the internet is seemingly endless. This massive influx of data in the world, referred to as "big data", has led to the creation of two of the most impressive and innovative technologies of our time: graph databases and cloud computing.

The core concepts of graph databases are remarkably intuitive, being rooted in the principles of graph theory. Whereas an RDBMS (relational database management system) models objects as a set of rows in predefined tables, where matching columns are used to represent relationships between objects (referred to as "foreign keys" in relational databases), graph databases represent information as a directed graph. In such a graph, objects which would have been represented as rows in a relational database are instead represented as vertices in a directed graph and are referred to as nodes. Similarly, relationships between objects are represented as directed edges [2] and are referred to as relationships. In a graph database, there is no concept of a table as is present in relational databases, wherein similar data are grouped together as separate instances of a single relation; rather, the nodes of many graph databases often contain a label which signifies the type of an object [3]. This type of database design is incredibly powerful for a few crucial reasons. Firstly, the very nature of graph theory exists such that relationships are of the utmost importance; "In the graph world, connected data is equally (or more) important than individual data points" [2]. In fact, much of the surge of data in the world

today can be attributed to the concurrent prosperity in the field of machine learning, whose fundamental existence relies upon the relationships between separate, sometimes seemingly disparate, pieces of data [4]. In an RDBMS, gathering information about two points of data that are not directly related to one another via a foreign key constraint is a difficult and unwieldy task, one which often involves multiple complex joins on the tables to which the points belong. This process is not only computationally expensive, often requiring each table involved in the multi-table join to be indexed and searched, it is also incredibly rigid. In fact, RDBMSs are designed from the bottom up for rigidity, and this works well when the relationships between data are fully, comprehensively known at the time of the database's inception. However, expressing relationships among data which are either difficult, impractical or impossible to predict at the time of formulation of the database quickly becomes an unreasonable task to ask of an RDBMS [5]. Conversely, graph databases are made precisely for modeling complex, indirect and previously unexpected relationships between data. Furthermore, queries to a graph database involving complex indirection can be incredibly fast, as the computational problem is transformed from one involving searching for matching fields within tables to one composed of searching in a directed graph, a problem which is both computationally efficient and naturally intuitive [9]. However, for all of the benefits afforded to users of graph databases, there are downsides. Graph databases are still a new concept, and as a consequence, many features of other modern databases are missing in graph databases. For example, distributing a database instance across multiple servers is often impossible with modern graph databases such as neo4j and ArangoDB [6]. Additionally, current graph databases are often unoptimized for the types of high volume data warehousing queries that are important for large databases [6]. However, these limitations of current graph databases are not fundamental to the concept of their existence, and most are a simple product of their youth. Therefore, it seems clear that graph databases will continue to grow in availability, popularity and necessity as the growing interdependence of data and increasing maturity of machine learning techniques drive the emergence of new, highly connected data.

While novel and promising, the immediate importance of the graph database pales in comparison to the pervasive influence of cloud computing on database management. The emergence of cloud computing services such as Amazon Web Services (AWS), Microsoft Azure and Google Cloud allows businesses and individuals to host their databases on remote servers that they do not need to maintain in a way that provides high availability and fast, reliable access [8]. As could be expected, there are a multitude of benefits associated with storing data in a cloud computing environment. The overhead associated with storing data in local servers (space allocation, hardware costs, security implementation, database design) is considerable [7], and the use of cloud computing services largely eludes such costs. Secondly, cloud computing services are designed for security and optimization simply by nature of their size and expansive pools of capital resources [8]. Interestingly, one of the most common hesitations cited by businesses when contemplating the adoption of cloud infrastructure is data security. However, this concern is easily remedied when considering, "On the other hand, one can argue that cloud providers—whose very business existence depends on ensuring a secure environment—should be better skilled at security and compliance issues than any individual company" [7]. There are a few disadvantageous factors to consider when discussing the prospect of leveraging the cloud, however. Perhaps most visibly, reliable network connection is a necessity; accessing data with a cloud infrastructure requires a constant connection to the internet, and, depending on the size of the database, necessitates a high level of bandwidth to process and transmit large amounts of data over the internet quickly and reliably [8]. In addition, the monthly fees paid to a cloud infrastructure provider may be greater than the cost associated with self-maintained, on-premises data storage. As can be seen, the compromises associated with storing data on the cloud are serious. However, these negative factors pale in comparison to the proposed benefits of using the cloud. Moreover, as time continues, advancements in AI and machine learning, as mentioned previously, will allow companies such as Amazon, Microsoft and Google to create applications which can help businesses leverage all of the hidden knowledge that lay dormant within the connections between data in their databases, while eschewing the need for such businesses to develop such pipelines and application from scratch themselves.

Database management has maintained its status as a critical problem in the technology industry since computing became commercially viable in the middle of the 20th century, but only recently has the subject line changed. Data in 2020 is both highly abundant and highly interconnected, and as such, it is important to develop database management technology that is simultaneously highly available, secure, reliable, cheap and easy to work with, while being architected with complex, unexpected relationships in mind. The complexity of data and its often intractable relationships are exactly what graph databases are made for, so it is not unrealistic to expect continuous growth in the popularity and maturity of the technology for years to come. Likewise, availability, security, reliability, scalability and ease of use are the factors that have rendered cloud computing an omnipresent force in the world of databases, and there is no reason to believe that such a trend will reverse its course anytime soon.

References

[1] Hillam, Jared. “Data Detective Board.” *INTRICITY*, www.intricity.com, 2 Oct. 2020, www.intricity.com/whitepapers/data-detective-board-whitepaper.

[2] Sasaki, Bryce Merkl. “Graph Databases for Beginners: Why Graph Technology Is the Future.” *Neo4j Graph Database Platform*, neo4j, 16 Sept. 2020, neo4j.com/blog/why-graph-databases-are-the-future/.

[3] Edlich, Stefan. “What Is a Graph Database?” *Aws*, Amazon, 2011, aws.amazon.com/nosql/graph/.

[4] Agrawal, Ajayand, Gans, Joshua, and Goldfarb, Avi “How to Win with Machine Learning.” *How to Win with Machine Learning* , Harvard Business Review, 18 Aug. 2020, hbr.org/2020/09/how-to-win-with-machine-learning.

[5] Epinomy. “Structured Data.” *Structured Data Tables*, Epinomy, 2017, epinomy.com/structured-data-tables.

[6] Rund, Ben. “The Good, The Bad, and the Hype about Graph Databases for MDM.” *Upside: Where Data Means Business*, Tdwi, 14 Mar. 2017, tdwi.org/articles/2017/03/14/good-bad-and-hype-about-graph-databases-for-mdm.aspx.

[7] Steier, Sandy. “To Cloud or Not to Cloud: Where Does Your Data Warehouse Belong?” *Wired*, Conde Nast, 7 Aug. 2015, www.wired.com/insights/2013/05/to-cloud-or-not-to-cloud-where-does-your-data-warehouse-belong/.

[8] Dignan, Larry. “Top Cloud Providers in 2020: AWS, Microsoft Azure, and Google Cloud, Hybrid, SaaS Players.” *ZDNet*, ZDNet, 1 Oct. 2020, www.zdnet.com/article/the-top-cloud-providers-of-2020-aws-microsoft-azure-google-cloud-hybrid-saas/.

[9] Angles, Renzo; Gutierrez, Claudio (1 Feb 2008). "Survey of graph database models" (PDF). ACM Computing Surveys. 40 (1): 1–39. CiteSeerX 10.1.1.110.1072. doi:10.1145/1322432.1322433. Archived from the original (PDF) on 15 August 2017. Retrieved 28 May 2016.