Section 1: Week 1: Database Bibliography

Nate Bachmeier

TIM-8120: Distributed Systems

December 8th, 2019

North Central University

Bibliography

The growth of data creation from sources such as IoT, Cloud, Big Data, and Mobile (ICBM) is increasing at an exponential pace. This explosive volume of information is forming in different shapes, with varying degrees of structure. Traditional database patterns and practices are unable to manage these data sets efficiently, which is driving enterprise environments to invest in new technologies. Merely adding new widgets to the network topology will not solve the challenges, and existing business processes will also need a revision. Through a combination of these ideas, enterprises can evolve their data pipelines and unlock the insights into more agile data-driven decisions.

# Business Intelligence Tomorrow (2019)

To understand the landscape of future Business Intelligence systems, one needs to look at the challenges of today (Harper, 2019). Harper proposes investment areas into (1) metadata management, semantic understanding, data catalog, data modeling, (2) Natural Language Processing, and (3) Edge computing.

The first aspect deals with the Data Lifecycle Management of ICBM data and its operationalization. Storage prices have decreased significantly, which has led to numerous businesses collecting vast pools of unstructured dark data. One of the principal inhabitants for these businesses is a lack of Information Governance, which includes classification, controls, identification, and monitoring (Ajis & Baharin, 2019). As these organizational systems improve the usability and discoverability of specific data subsets, then data scientists can begin exploring the data and coming up with operational insights.

The second aspect deals with the interaction of users into the data management system. Many of these data repositories rely on query languages, such as Structured Query Language (SQL), to store and retrieve records. These languages introduce a barrier to entry challenges for users of the system, as they need to learn tedious syntax. Instead, Natural Language Processing (NLP) can convert business questions directly into data-driven solutions. When the NLP algorithm is made aware of proprietary object models, it can derive entities, verbs, and other relationships. These capabilities led to a democratization of self-service scenarios across all levels of the organization.

The third aspect is the inclusion of edge computing in data processing architectures. Micro-clouds of IoT and mobile devices are generating vast collections of sensor and machine-to-machine data. Centrally processing these feeds could involve significant network I/O, or is economically prohibitive to move. Instead, a transformation of these ‘high volume/low quality’ feeds into ‘low volume/high quality’ aggregations needs to take place. For instance, one hundred temperature sensors could report one hundred individual measurements or the median of their aggregate value.

# Big Data Quality: A Survey (2018)

Big data is high-volume, high-velocity, and high-variety information that can produce high-value assuming it's high-quality (Taleb et al., 2018). According to ISO 25012, the critical measurement of quality is its fitness for us. Taleb et al. measure this fitness against intrinsic, contextual, accessibility, and representational dimensions. They propose a Process-Driven Analytic Pipeline that contains multiple quality gates. Each quality gate further refines the information into a more standardized and curated representation of itself. For instance, unstructured data becomes annotated, reformatted, and entities extracted. This curation process improves the consumption of data into analytical solutions, enabling more precise estimations.

Their approach is similar to the National Institute of Standards and Technology (NIST) reference model for data lifecycle management, which includes phases Collection, Preparation, Analysis, and Action (Mazumdar, Seybold, Kritikos, & Verginadis, 2019). NIST also proposes spending effort upfront to clean data sets. For instance, extreme outliers might skew the prediction results, and by pre-emptively removing them, a more accurate model defined.

# Business Intelligence for Enterprise Systems: A Survey (2017)

After data curation, experts use machine learning to perform statistical analysis (Duan & Xy, 2017). These algorithms come in two forms, supervised and unsupervised, based on the availability of the predicted value. Supervised algorithms, such as linear regression and decision trees, use examples to build an inference model. Unsupervised algorithms, such as itemset mining and k-clustering, seek out correlations within groups of similar records. In either scenario, the data become split for parameter training (70%) and accuracy validation (30%). Then the model is exposed through RESTful endpoints and consumed by other enterprise applications. Through the use of web service abstraction layers, a clear separation of duties occurs, such that development teams can leverage the predictive capabilities without specialized data science training.

Duan and Xy also describe future challenges for business intelligence in the enterprise environment. They call out the need for outlier detection, management of graph data, and a strategy for edge computation. Graph databases enable the discovery of complex relationships that would not be possible in traditional relational stores (Kronmueller, Chang, Hu, & and Desoky, 2018). In a conventional relational store graph data is tedious to store, because the table structures are globally indexed. Meanwhile, graph databases can maintain local indexes along the edges to quickly traverse to associated nodes.

# A Survey on Graph Database Management Techniques for Huge Unstructured Data (2018)

Many solutions have attempted to provide graph abstractions on top of relational databases or other traditional technologies. These solutions tend to be inefficient due to the overhead of layering technology (Patil, Kiran, Kavya, & Naresh, 2018). Patil et al. also point to adoption challenges caused in-memory storage models limiting the scalability and inconsistency of programming interfaces. They acknowledge that Gremlin and SPARQL are gaining consensus as to the leading query languages, but have a long road to industry standard tooling.

Next, the authors reviewed seven different native graph databases and compared their strengths and weaknesses. Patil et al. claim that Neo4j is the most popular graph database in enterprises. It uses a proprietary language, called *Cypher,* to calculate statistics, traverse relationships, and find bipartite and connected components (Kronmuller, Chang, Hu, & Desoky, 2018). Kronmuller et al. would agree with Patil et al. that *Cypher* is easy to learn and apply to real-world problems.

Another technology reviewed is the Titan Graph, which provides a consistent abstraction layer across different NoSQL stores. Administrators can host a Rexster service to expose RESTful interfaces that accept Gremlin queries. Similar to Cypher, Gremlin can perform many relationships and path operations, such as the shortest path. Titan typically accompanies Apache TinkerPop, a Java-based implementation that can scale to billions of nodes across clusters of horizontal-scaled servers.

Patil et al. conclude their discussion with a review of active research topics within Graph Database research, such as online compression, feature mining, subgraph comparisons, and biology applications. Biology research uses these systems for representing protein-protein interfaces, which can contain enormous relationships between different treatments. Until recently, it has been challenging for smaller organizations to manage big data graphs due to cost restrictions. However, now that public cloud providers are natively offering on-demand pay-for-use economics, they are becoming available to businesses of all shapes and sizes.

# Data Storage Management in Cloud Environments: Taxonomy and Direction (2017)

Storage as a Service (StaaS) is enabling businesses to make agile use of highly reliable, scalable, and flexible resource pools (Mansouri, Nadjaran, & Buyya, 2017). According to the CAP theorem, a distributed store can't provide more than two guarantees between consistency, availability, and partition tolerance (Gilbert & Lynch, 2002). This constraint causes network administrators to choose between different optimization points offered by Relational (SQL), Not Only SQL (NoSQL), and NewSQL solutions. Relational stores focus on consistency over availability, in contrast to NoSQL, which targets availability through eventual consistency models. Between these two extremes is NewSQL, which provides ACID transactions across a storage partition, not the entire data set. This