Scalable Streaming Graph Data Processing

A Position Paper on Design Goals and a Possible Approach

Kurt Rohloff, Jeffrey Cleveland, Kyle Usbeck
Raytheon BBN Technologies
Cambridge, MA USA
{krohloff,jcleveland,kusbeck}@bbn.com

Abstract— **In this paper we provide a vision for scalable, streaming graph data processing. Such a capability is needed to leverage graph data's ability to store complex relationships and the ubiquitous presence of graph data resulting from real life events ranging from social network interactions to financial transactions, among many. We motivate and discuss approaches towards an approach for scalable, streaming graph data processing. We discuss a hypothetical architecture called GEARS to support scalable streaming graph data processing.**

Keywords— big data; streaming; graph data

I. INTRODUCTION

The scalable processing of graph data is and will continue to become increasingly important because of graph data's ability to store complex relationships and to facilitate data mining and rapid retrieval from storage. Graph data is of increasing value because it intuitively represents social interaction networks and spatial-temporal interaction activities between actors. Graph data results from social network technologies such as wikis, email, and chat, as well as from system audit records that keep track of communication network interactions at various layers. Graph data is used by pharmaceutical and other bio firms to identify connection pathways between pharmaceuticals, diseases and gene expressions. Retail and credit card organizations use graph data to analyze customer behavior and financial patterns, both to improve sales and detect fraud. Key forces driving the trend of processing linked data quickly at large scale include linked data's ability to store complex relationships and the ubiquitous presence of linked data resulting from real life events ranging from social network interactions to financial transactions, among many others of interest from analytics and data mining perspectives.

As implied above, the common thread of operational needs driving the development of graph data processing technologies is scalability. Unfortunately, highly scalable cloud-based approaches to traditional information storage, management and processing technologies, such as databases can only be partially leveraged for scalable graph data processing in the clouds. Our contention is that out of recent innovations in cloud-based graph data processing technologies, the most notable and influential have been driven by primarily by high-level design improvements. We review key features of graph data processing driving the need for cloud computing and inform suggestions for continued improvements for scalable cloud-based graph processing. The current state of the art in graph data processing and data mining at scale is not sufficient to handle large amounts of streaming, real-time data. Existing graph processing technologies use batch techniques and are not designed to process streaming data, i.e., to run standing queries against new data as it is received. Further, few open-source technologies today can successfully process graph queries at scale (even using batch techniques). These scalability constraints are the greatest barriers to achieve a fundamental web-scale vision and have hindered the broader standards-based graph data processing technologies.

Now that we as a community are beginning to understand the high-level design concerns for scalable cloud-based computing over graph-data, it is time for us to revisit our initial designs, consolidate the lessons learned and adjust our assumptions to design a next generation of cloud-based graph data technologies. We suggest paying more attention to lower-level details that provide headway for improved performance at large scales.

The current state of the art in processing and data mining at scale over linked data is insufficient to quickly process queries in constantly generated large amounts of streaming, real-time graph data. In this position paper we motivate and discuss approaches towards general scalable, streaming graph data processing.

The remainder of this paper is organized as follows. In Section II we review current approaches to scalable graph data processing and we discuss design tradeoffs that would need to be considered to implement a scalable streaming graph data capability. In Section III we present the GEARS vision for a standards-based architecture for scalable streaming graph data processing.

II. CURRENT APPROACHES AND DESIGN GOALS

Our contention is that the most notable and influential recent innovations for scalable cloud-based graph data processing technologies have been driven primarily by high-level design improvements with a focus on batch processing. See for example the various implementations of MapReduce both via Google [1] and Hadoop [2] which revolutionized graph data processing and analysis. Hadoop abstracts away low-level details of concurrency that normally bedevil system designers. Numerous system architects, employed by companies from from start-ups to major firms like Yahoo have utilized the high-level MapReduce design abstraction provided

by Hadoop for easier development of graph analysis tools while relying on Hadoop's automated replication of distributed data for robustness.

MapReduce approaches often preclude the need to design cloud-based graph data systems and decide on relatively lower-level indexing and data partitioning approaches. However, these indexing and partitioning design decisions are still fairly high level in the context of feasibly potent design improvements at even lower levels of abstraction. This distinction is primarily driven by what has traditionally been one of the greatest technical challenges in building scalable distributed information systems - tracking the location of data over distributed compute nodes.

Unfortunately, recent Hadoop implementations focus on batch processing and are not ideal for the manipulation of streaming graph data due to unnecessary parsing overhead (especially for large files), and incompatibility with inherently stream oriented data types (video, audio) since breaking up causes issues with codecs. Specialized graph data processing frameworks have mostly been closed source and non-generic, tailor-made for specific applications as seen in the Complex Event Processing (CEP) community. However, open source cloud-based stream processing frameworks are emerging, particular built for Semantic Web applications, albeit at relatively small scales [3].

Most of the few cloud-based streaming platforms (such as IBM System S) do not natively support graphs and/or are closed-source (such as StreamBase). Open source platforms, such as JBOSS Drools Fusion, provide complex event processing as part of business logic integration, but are unable to scale to the data types and quantities needed in the DoD enterprise.

With this context of prior work in mind, our vision is that a graph data stream processing capability is needed to support:

- Distributed stream processing to improve response times and throughput.
- Meta-analytics to improve data co-location, query decomposition across distributed compute and storage nodes. This could include sliding window reconfiguration of data based on data and query performance sampling.
- Heterogeneous clouds of mixed capabilities to leverage "special" resources. Ex: GPUs, DSPs, FPGAs, etc…
- Wide-area operations to move processing to the data. For example, leverage existing and ongoing work on tactical information management.
- Easy integration of analytics and reporting to end users.

Tentative solutions should also be able to avoid extensive data pile searches and simplify data aggregation and computation federation to design effective heuristics for highly connected subgraph collocation to avoid inter-node processes caused by data splitting.

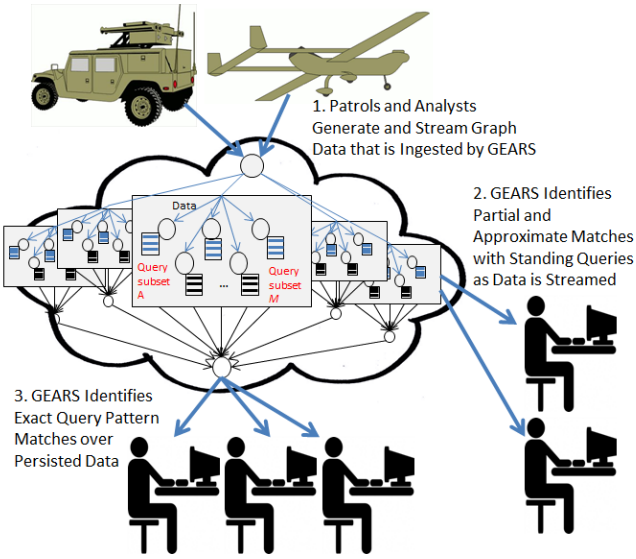

Figure 1: GEARS Concept of Operations

A possible approach would be to utilize meta-analytics to support data and processing colocation by supporting the intelligent management and distribution of streams to available processing nodes and avoid following data "links" across multiple compute nodes. For the majority of graph data processing, "neighborhoods" of data are all that is needed, so data co-location reconfiguration managed over sliding-window meta-analytics would improve performance over time by reducing inter-process bandwidth and latency with intelligent query decomposition.

This solution would need to analyze data at multiple layers:

- As data gets ingested and processed at the syntactic layer (fast).
- As algorithms run and transform that streams or produce new streams.

As such we see the primary technical challenges as needing capabilities for:

- A stream-processing framework that can host standing user queries and know on which cloud compute nodes to execute these queries to minimize communication bottlenecks.
- A heterogeneity-aware distributed file system that tracks the best place to store ingested data and stores the data in these locations with distributed replication.
- An extensible sub-graph matching environment that processes user queries over possibly distributed data and disseminates the appropriate results to the appropriate users.

- Possibly multiple sub-graph matching toolboxes with interfaces for users to write custom applications to execute queries in graph query languages such as SPARQL.

Overcoming these challenges is only possible today because of the recent work in highly scalable data processing, recent advances in open-source streaming software, and recent research in cloud-based real-time publish-subscribe systems.

III. The Hypothetical GEARS Architecture

We propose the GEARS (Graph-focused Extensible Allocation of Resources with Streaming) cloud-based information environment for key DoD Big Data challenges. GEARS (as seen in Figure 1) provides a large scale stream processing capability that can:

- Ingest streaming graph data, including spatial-temporal data;
- Run standing queries in real-time for sub-graph pattern matches with partial and approximate matches;
- Persist the data and update previously persisted data; and
- Alert relevant users when graph pattern matches are found.

Our solution and approach with GEARS is directly informed from recent experience addressing graph processing challenges for proprietary customers. From our work on a scalable graph processing tool study for a proprietary customer, we identified a pressing need for a highly scalable and extensible pattern matching and querying capability over graphs. This need drives the GEARS goal for highly scalable graph data processing. Our GEARS concept is also informed by the missions of identifying spatial-temporal graph patterns using a streaming graph service. As data collection occurs, new graph edges on relevant behavior need to be added to the graph and analysts need to identify as soon as possible behavior that matches a pattern of interest. Of particular interest for integration of legacy systems, GEARS is compatible with the technologies already in wide use for scalable data processing, such as Hadoop and Accumulo.

The proposed GEARS architecture can be seen in Figure 2. The GEARS architecture will operate in a heterogeneous compute infrastructure using resource management software, such as Apache Zookeeper. GEARS will provide a toolbox for developers to build high-performance graph querying applications over data streams. We will design and develop four software layers that correspond to the four technical challenges we identified above:

•G2O graph-optimized stream computing framework.

•GDFS heterogeneity-aware distributed file-system.

•GEM extensible sub-graph matching environment.

•GSGM sub-graph matching toolbox.

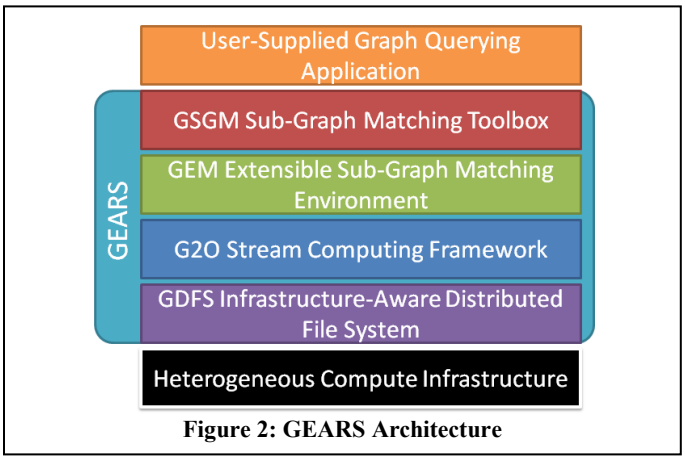

Figure 2: GEARS Architecture

The GEARS Graph-Optimized (G2O) stream computing framework is a customization of existing open-source stream processing frameworks such as the S4 and Storm open-source stream computing environment with native support for distributed streaming graph-data manipulation to leverage the local graph stores embedded in GDFS.

The GEARS Infrastructure-Aware Distributed File System (GDFS) manages the storage of graph data on the cloud-computing nodes. GDFS maintains automated data replication to ensure that data is preserved when compute nodes fail. GDFS supports co-location of data replications based on sub-graph connectedness, temporal similarity and geographical similarity to improve response time to the most critical queries which depend on graph and spatial-temporal colocality.

The GEARS Extensible sub-graph Matching (GEM) environment automates the decomposition of streaming input data and processing. GEM manages the decomposition of query requests into operations performed on localized G2O processing engines and federates responses that are fed to user applications.

The GSGM subgraph matching toolbox provides support for specific query languages such as the SPARQL query language and/or various types of reasoning over graphs such as OWL reasoning. We will develop several initial graph matching toolboxes and design this capability so that further toolboxes can be developed and contributed by 3rd parties.

References

[1] Dean J. and Ghemawat S., MapReduce: Simplified data processing on large clusters. In Proceedings of the USENIX Symposium on Operating Systems Design & Implementation (OSDI), pp. 137-147. 2004.

[2] Hadoop. (2010). Apache Hadoop. Retrieved from http://hadoop.apache.org/

[3] Mendes, Pablo N., Alexandre Passant, and Pavan Kapanipathi. "Twarql: tapping into the wisdom of the crowd." *Proceedings of the 6th International Conference on Semantic Systems*. ACM, 2010.

[4] Rohloff, Kurt, and Richard E. Schantz. "High-performance, massively scalable distributed systems using the mapreduce software framework: The shard triple-store." *Programming Support Innovations for Emerging Distributed Applications*. ACM, 2010.