Distributed Knowledge Graph Querying on Edge, Fog and Cloud

Shriram R.

Department of Computational and Data Sciences
Indian Institute of Science, Bangalore 560012 INDIA
Email: shriramr@iisc.ac.in

***Abstract*—Knowledge graphs with millions of vertices and edges are currently being used to enable smart devices and knowledge discovery. IoT (Internet of Things) devices are becoming the consumer and producer of such large graphs. Existing graph querying and processing systems are designed to run on commodity hardware. However, IoT applications need lightweight graph processing systems that can scale across large number of devices and handle concurrent queries with low latency. In this project, a lightweight distributed graph querying engine is proposed which can natively scale across edge, fog and cloud devices. It supports declarative query model for vertex, edge, reachability, shortest path and subgraph pattern mining queries. The engine has a novel caching, indexing and query partitioning mechanism which enables to scale efficiently with graph size and query workload. The engine is evaluated on a large scale virtual IoT testbed with massive real-world knowledge graphs. The results indicate that the engine can provide x% lower latency on a diverse query workload and can handle y% more queries concurrently compared to the baseline system.**

I. Problem Description

The problem that is being solved in this project is the scaling of graph querying on edge, fog and cloud devices which can provide low-latency query respone for many concurrent queries on massive knowledge graphs. *Figure 1* shows a representation of typical edge-fog-cloud layout. There is a hierarchical structure where fogs can communicate with edges and cloud and the edges can directly communicate with only fog. Each edge is associated with a single fog depending on its spatial proximity or any other parameter. The bandwidth and latency of edge-fog, fog-fog and fog-cloud network links are significantly different from each other.

The edge, fog and cloud devices are heterogenous and have different compute and storage capabilities. In addition, the device failure rate varies across the layers with edge having the highest failure rate. In this project however, all the devices are considered to be 100% reliable.

Existing graph processing/querying systems are designed for commodity hardware and cannot be efficiently deployed on low powered edge devices like smartphones, smart home appliances etc. These systems do not take into account the network hierarchy and device heterogeneity

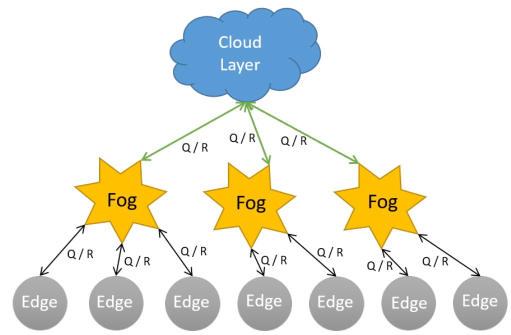

Fig. 1: Edge-Fog-Cloud Layout

that exists on the current edge-fog-cloud layout. The proposed graph querying engine uses GoDB which is a distributed graph database as the backend and implements the novel features on top of the database.

In this project, following are the key contributions:

- **Graph Partitioning:** A novel graph partitioning approach is used which partitions the graph based on the locality of query. This means that the portion of knowledge graph which is required by a query is identified and moved closer to the device performing the query automatically
- **Query Planning:** A query planner which can efficiently partition the query predicates according to graph locality and placement thereby optimizing query cost is implemented
- **Query Caching:** A novel query caching system is devised which caches recent queries by edges and their corresponding results in the fog layer. It exploits the temporal and spatial locality of queries to provide low latency respone. It also has a novel cache replacement policy and can match an incoming query with cache.
- **Graph Indexing** Index data structures from existing literature that can accelerate the processing of queries are built in a distributed fashion and are used for pruning the knowledge graph for processing. Different type of indexes are built for each query type and are maintained dynamically as per updates to the knowledge graph

II. Novelty/Scalability

III. Related Work

IV. Problem Definition and Approach

V. Proposed Experiments

VI. Related Work and Citation

[1] [1].

References

[1] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Computing in Science Engineering (CiSE)*, vol. 15, no. 4, pp. 38–47, 2013.

[2] Y. Simmhan, C. van Ingen, A. Szalay, R. Barga, and J. Heasley, "Building reliable data pipelines for managing community data using scientific workflows," in *IEEE International Conference on eScience (eScience)*, 2009, pp. 321–328.

[3] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance techniques," Computer Science Department, Indiana University, Tech. Rep. TR-612, 2005, extended version of SIGMOD Record 2005. [Online]. Available: http://www.cs.indiana.edu/pub/techreports/TR618.pdf

[4] Distributed Research on Emerging Applications & Machines, "DREAM:Lab Publications Repository," http://dream-lab.cds.iisc.in/publications/, Aug 2014.

[5] Y. Simmhan, Q. Zhou, and V. K. Prasanna, *Green IT: Technologies and Applications*. Springer Berlin Heidelberg, 2011, ch. Semantic Information Integration for Smart Grid Applications.

[6] N. Jamadagni and Y. Simmhan, "Godb: From batch processing to distributed querying over property graphs," in *IEEE CCGRID*, 2016.

[7] S. Aman, Y. Simmhan, and V. K. Prasanna, "Holistic measures for evaluating prediction models in smart grids," *IEEE TKDE*, 2015.

[1] Please refer to the IEEE Taxonomy for more details. http://www.computer.org/portal/web/publications/acmtaxonomy