Enabling Graph Analytics with Graphulo

Lauren Milechin[*], Vijay Gadepally[+^], Dylan Hutchison[$], Jeremy Kepner[+^]
[*]MIT EAPS, Cambridge, MA, U.S.A.
[+]MIT Lincoln Laboratory, Lexington, MA, U.S.A.
[^]MIT CSAIL, Cambridge, MA, U.S.A.
[$]University of Washington, Seattle, WA, U.S.A.

ABSTRACT

Abstract – Graphulo is a tool built for Apache Accumulo to enable in-database graph analytics. This allows analysts to perform graph analytics on large graphs that do not easily fit into local memory. In order to reduce the barrier of entry for analytics developers, we have developed interfaces to two common analytical programming environments: MATLAB®/Octave through the D4M library and Apache Pig.

I. INTRODUCTION

Graphulo is an extension for Apache Accumulo, a high performance NoSQL key-value store. It implements GraphBLAS kernels as server-side iterators, which can be combined to execute graph algorithms [1]. Further, Graphulo provides implementations for several of the more common algorithms, such as Breadth First Search, Jaccard Index, and k-Truss Subgraph and supports common graph schemas, including Adjacency, Incidence, and Single-Table schemas [2].

Graphulo has been shown to scale and perform well within the Accumulo computing environment when compared with local computation, and continues to perform well when local computation fails on larger graphs due to memory constraints [3] [4]. Graphulo is implemented in Java, a common language for database applications, and this can be a barrier to entry to some analysts. Therefore, we have provided two interfaces in analytical programming environments, allowing more analysts to take advantage of the features that Graphulo as to offer.

II. MATLAB®/OCTAVE AND D4M

The MATLAB®/Octave interface, which is accessible through the D4M library. D4M provides a flexible and extensible data representation, manipulation, and analysis [5]. One advantage of D4M is its ability to connect to a variety of types of databases. These connectors have been benchmarked and shown to have high and record-breaking performance [6] [7].

The D4M 3.0 release integrates an interface to Graphulo with these database capabilities [8]. This benefits those analysts that are most comfortable with the matrix and linear algebra-based syntax of MATLAB and D4M. Graphulo algorithms and kernels can be initiated directly from MATLAB® or Octave using the D4M library, without the overhead of writing and compiling a Java program.

III. APACHE PIG

Apache Pig is a platform in the Hadoop ecosystem for analyzing large datasets [9]. It provides a declarative, SQL-like language called Pig Latin to initiate Map-Reduce jobs on Hadoop. Pig supports custom user defined functions (UDFs) in Java that can be called from Pig Latin, allowing features to be added as needed.

We have implemented Graphulo calls as Pig UDFs, thereby creating SQL-like environment to interact with Accumulo and perform graph analytics. We have also provided UDFs that use our custom D4M Accumulo connector to insert into and query from Accumulo. On ingest, data formatted as graph edge lists can be auto-organized into a Graphulo-supported schema for ease of use, and triples can be inserted with no manipulation if needed. The Graphulo-Pig interface can also query Accumulo data, and will take advantage of available transpose tables on query (a query issued on a column will search the rows of the transpose tables to improve performance).

IV. CONCLUSIONS AND FUTURE WORK

Graphulo allows graph analytics to scale beyond the constraints of local computation. The MATLAB®/Octave D4M and Apache Pig interfaces make this technology available to a wider range of researchers and analysts. D4M is also available as a package in the Julia programming language [10]. Future work includes adding database capabilities to the D4M.jl package, including an interface to Graphulo.

V. ACKNOLWEDGEMENTS

The authors wish to acknowledge the following individuals for their contributions: Michael Stonebraker, Sam Madden, Bill Howe, David Maier, Chris Hill, Alan Edelman, Charles Leiserson, Dave Martinez, Sterling Foster, Paul Burkhardt, Victor Roytburd, Bill Arcand, Bill Bergeron, David Bestor, Chansup Byun, Mike Houle, Matt Hubbell, Mike Jones, Anna Klein, Pete Michaleas, Julie Mullen, Andy Prout, Tony Rosa, and Chuck Yee.

REFERENCES

[1] V. Gadepally, J. Bolewski, D. Hook, D. Hutchison, B. Miller and J. Kepner, "Graphulo: Linear Algebra Graph Kernels for NoSQL Databases," in *IEEE High Performance Extreme Computing (HPEC)*, 2015.

[2] D. Hutchison, J. Kepner, V. Gadepally and B. Howe, "From NoSQL Accumulo to NewSQL Graphulo: Design and Utility of Graph Algorithms inside a BigTable Database," in *IEEE High Performance Extreme Computing (HPEC)*, 2016.

[3] D. Hutchison, J. Kepner, V. Gadepally and A. Fuchs, "Graphulo Implementation of Server-Side Sparse Matrix Multiply in the Accumulo Database," in *IEEE High Performance Extreme Computing (HPEC)*, 2015.

[4] T. Weale, V. Gadepally, D. Hutchison and J. Kepner, "Benchmarking the Graphulo Processing Framework," in *IEEE High Performance Extreme Computing (HPEC)*, 2016.

[5] J. Kepner, W. Arcand, W. Bergeron, N. Bliss, R. Bond, C. Byun, G. Condon, K. Gregson, M. Hubbell, J. Kurz, A. McCabe, P. Michaleas, A. Prout, A. Reuther, A. Rosa and C. Yee, "Dynamic Distributed Dimensional Data Model (D4M) Database and Computation System," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[6] J. Kepner, W. Arcand, D. Bestor, B. Bergeron, C. Byun, V. Gadepally, M. Hubbell, P. Michaleas, J. Mullen, A. Prout, A. Reuther, A. Rosa and C. Yee, "Achieving 100,000,000 database inserts per second using Accumulo and D4M," in *IEEE High Performance Extreme Computing (HPEC)*, 2014.

[7] S. Samsi, L. Brattain, W. Arcand, D. Bestor, B. Bergeron, C. Byun, V. Gadepally, M. Houle, M. Hubbell, M. Jones, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, J. Kepner and A. Reuther, "Benchmarking SciDB Data Import on HPC Systems," in *IEEE High Performance Extreem Computing (HPEC)*, 2016.

[8] L. Milechin, V. Gadepally, S. Samsi, J. Kepner, A. Chen and D. Hutchison, "D4M 3.0: Extended Database and Language Capabilities," in *IEEE High Performance Extreme Computing (HPEC)*, 2017.

[9] The Apache Software Foundation, "Apache Pig," 2017. [Online]. Available: https://pig.apache.org/. [Accessed 2017].

[10] A. Chen, A. Edelman, J. Kepner, V. Gadepally and D. Hutchison, "Julia Implementation of the Dynamic Distributed Dimensional Data Model," in *IEEE High Performance Extreme Computing (HPEC)*, 2016.