The State of Graph Processing

APIs, Libraries, Benchmarks, and Programming Languages

Samuel Pollard

January 12, 2017

1 APIs and Libraries

- **Pregel.** API; developed by Google; distributed; vertex centric, Bulk-Synchronous Parallel model; inspired many other platforms such as Giraph and GPS; original paper; 2010.
- **GPS (Graph Processing System).** API; developed by Stanford; distributed; vertex centric, Bulk-Synchronous Parallel model; open source; similar to Pregel but with dynamic graph repartitioning and other enhancements. original paper; website; First appeared 2013; apprears inactive.
- **GraphX** Library.

2 Programming Languages

- **GP (Graph Programs).** Nondeterministic; serial; original paper: [9]; website: `https://www.cs.york.ac.uk/plasma/wiki/index.php?title=GP_(Graph_Programs)`; appears to be more theoretical and used for program verification; still active.
- **Gremlin.** functional, data flow; distributed; a way to interact with graph databases; [10]; website: `http://tinkerpop.apache.org/gremlin.html`.

3 Benchmarks

- **Graphalytics.** CPU and GPU; supports GraphMat, PowerGraph, GraphBIG, Giraph, GraphX, Neo4j, and MapReduce; [2]; still active; so far I can only get PowerGraph and GraphBIG running; website: `http://graphalytics.ewi.tudelft.nl`.
- **GAP (Graph Algorithm Platform).** CPU; shared Memory (OpenMP); `http://gap.cs.berkeley.edu/benchmark.html`; last active October 2016 on Github. [1]
- **GraphBIG.** CPU and GPU; shared Memory and CUDA; last active February 2016.
- **Lonestar.** CPU and GPU; shared memory and CUDA; part of Galois. First appeared 2011; last update appears to be in 2015 though I am in recent (December 2016) contact with someone working on the project.

4 Dynamic Graphs

The primary focuses for dynamic graphs encountered thus far are: dynamic updating of the graph objects themselves and the dynamic partitioning and re-partitioning of the graphs across a distributed architecture. The distinction is made clear below:[1]

Dynamic Partitioning

- **Zoltan.** `http://www.cs.sandia.gov/zoltan/` [5].
- **PTScotch.** `https://www.labri.fr/perso/pelegrin/scotch/` [4].
- **ParMETIS.** `http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview`; [8].

Streaming graphs/support for dynamic updating of graphs. Many of these are advertised to social network websites or consumers of large amounts of streaming data (for example, see `http://storm-project.net`). The key word here is *analytics*.

- **STINGER.** Data structure and library; Georgia Institute of Technology and various national laboratories; streaming model; parallel or serial, distributed or shared; `http://cass-mt.pnnl.gov/docs/pubs/pnnlgeorgiatechsandiastinger-u.pdf`; `http://www.stingergraph.com/`; First appeared 2009; still active on Github.
- **GraphJet.** Java library; parallel but single-machine; original purpose was for real time recommendations from Twitter; `https://github.com/twitter/GraphJet`; [11].
- **GraphStream.** Java library; appears to be serial and focuses mainly on visualization; `http://graphstream-project.org/`; [6] first active 2007 but appears active as of December 2016.
- **Kineograph.** API; parallel and distributed; [3]; 2012; appears to be inactive but influenced GraphX, GraphChi, and PowerGraph.
- **PHISH.** Streaming graph processing; `http://www.sandia.gov/~sjplimp/phish.html`
- **PGX.** Oracle Project; integration with Groovy, Green Marl, Spark, Hadoop; Supports incremental updates though the dynamic aspect is listed as future work. [7].

5 Visualization

There are many graph visualization tools out there, most notably Graphviz (and its associated DOT file format). Likewise, databases such as Neo4j have their own visualization tools. This is concerned only with visualizations which are scalable to a large number of vertices and edges.

- **Gephi.** website

References

[1] Scott Beamer, Krste Asanovic, and David A. Patterson. The GAP benchmark suite. *CoRR*, abs/1508.03619, 2015.

[1] These resources are in part retrieved from `http://scicomp.stackexchange.com/questions/4722/i-am-looking-for-a-parallel-dynamic-graph-library-in-c`.

[2] Mihai Capotă, Tim Hegeman, Alexandru Iosup, Arnau Prat-Pérez, Orri Erling, and Peter Boncz. Graphalytics: A big data benchmark for graph-processing platforms. In *Proceedings of the GRADES'15*, GRADES'15, pages 7:1–7:6, New York, NY, USA, 2015. ACM.

[3] Raymond Cheng, Ji Hong, Aapo Kyrola, Youshan Miao, Xuetian Weng, Ming Wu, Fan Yang, Lidong Zhou, Feng Zhao, and Enhong Chen. Kineograph: Taking the pulse of a fast-changing and connected world. In *Proceedings of the 7th ACM European Conference on Computer Systems*, EuroSys '12, pages 85–98, New York, NY, USA, 2012. ACM.

[4] C. Chevalier and F. Pellegrini. Pt-scotch: A tool for efficient parallel graph ordering. *Parallel Comput.*, 34(6-8):318–331, July 2008.

[5] Karen Devine, Bruce Hendrickson, Erik Boman, Matthew St. John, and Courtenay Vaughan. Design of dynamic load-balancing tools for parallel applications. In *Proc. Intl. Conf. on Supercomputing*, pages 110–118, Santa Fe, New Mexico, 2000.

[6] Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné. GraphStream: A Tool for bridging the gap between Complex Systems and Dynamic Graphs. In *Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems*, EECS '07, Dresden, Germany, October 2007.

[7] Sungpack Hong, Siegfried Depner, Thomas Manhardt, Jan Van Der Lugt, Merijn Verstraaten, and Hassan Chafi. Pgx.d: A fast distributed graph processing engine. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '15, pages 58:1–58:12, New York, NY, USA, 2015. ACM.

[8] George Karypis and Vipin Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *J. Parallel Distrib. Comput.*, 48(1):71–95, January 1998.

[9] Detlef Plump. The graph programming language GP. In *Proceedings of the 3rd International Conference on Algebraic Informatics*, CAI '09, pages 99–122, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] Marko A. Rodriguez. The gremlin graph traversal machine and language. *CoRR*, abs/1508.03843, 2015.

[11] Aneesh Sharma, Jerry Jiang, Praveen Bommannavar, Brian Larson, and Jimmy Lin. Graphjet: Real-time content recommendations at twitter. In *Proceedings of the VLDB Endowment.*