

# Simplifying Parallel Graph Processing

## A Survey of Existing Platforms

Samuel Pollard  
University of Oregon  
Eugene, OR, USA  
spollard@cs.uoregon.edu

Boyana Norris  
University of Oregon  
Eugene, OR, USA  
norris@cs.uoregon.edu

### ABSTRACT

This is a survey of existing graph analytics frameworks.

### 1. MACHINE SPECIFICATIONS

Table 1 shows the specifications of the research computer (named Arya).

CPU Model	Intel Xeon(R) E5-2699 v3 @ 2.30GHz
CPU Sockets	2
CPU Cores	72
CPU Clock	3600MHz
RAM Size	256GB
RAM Freq	1866MHz
Max RAM Freq	2133MHz
GPU Model	GM204 [GeForce GTX 980]

**Table 1: Machine specifications.** The disparity between the CPU’s advertised clock speed and the “CPU Clock” row is a result of the Turbo Boost technology which can increase the clock speed to a limit. The manufacturer’s published maximum clock speeds can be found at <http://ark.intel.com>.

### 2. PERFORMANCE

Graphalytics without the use of the Granula plugin produces performance measurement in two forms: runtime in seconds and traversed edges per second.

Table 2 lists performance in milliseconds of runtime according to the graphalytics output. Graphalytics also outputs MTEPS or millions of traversed edges per second. However, the graphalytics version does not make sense in all cases: for example, computing the local clustering coefficient involves traversing each edge multiple times (proportional to the sparsity of the graph), while breadth first search (BFS) traverses each edge exactly once, and on naive implementations single-source shortest paths (SSSP) may have  $O(|E| + |V|^2)$  traversed edges.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Load Testing and Benchmarking of Software Systems (LTB) 2017, L'Aquila, Italy*

© 2016 ACM. ISBN ?.

DOI: ?

In Table 2, BFS is breadth-first search, SSSP is single-source shortest paths, LCC is local clustering coefficient, PR is PageRank, CDLP is community detection using label propagation, and WCC is weakly connected components. For the algorithms used, see [3].

	openg	powergraph
CDLP	181.667	1226
PR	302.333	974
LCC	321.333	1036.67
WCC	87.6667	697.667
SSSP	4061.33	29022.3

**Table 2: Performance Results for the dota-league dataset with 61,670 vertices and 50,870,313 edges.**

### 3. GRAPH PROCESSING TAXONOMY

This is in the spirit of [1]. Here, “|” means “or” and “+” means “and.” FOSS means Free and Open Source Software. The quotes around “yes” for HPC mean that the product claims to be amenable to high performance computing. Whether these actually achieve their goal is one of the purposes of this project.

### 4. CONCLUSION

We have presented an updated survey of parallel graph processing frameworks supplementary to [1]. From this, we have selected a representative subset of frameworks on which performance is analyzed and have stored these results in a database. To facilitate parallel graph processing, hardware information and performance results are automatically populated (as were all the tables in this paper). These performance results are then used to provide simple recommendations of the optimally-performing framework given a particular algorithm and problem size.

### 5. REFERENCES

- [1] N. Doekemeijer and A. L. Varbanescu. A survey of parallel graph processing frameworks. Technical report, Delft University of Technology, 2014.
- [2] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 17–30, Hollywood, CA, 2012. USENIX.

Name	Type	HPC	Parallelism	Target	FOSS	Source	Notes
PowerGraph	Framework	“yes”	both	CPU	yes	[2]	<sup>a</sup>
GraphBIG	Benchmark	“yes”	shared	CPU GPU	yes	[4]	<sup>b</sup>

**Table 3: Tools used for graph processing**

<sup>a</sup>The current version is a closed-source product by Turi though PowerGraph v2.2 is on Github.

<sup>b</sup>Only works on Linux.

- [3] A. Iosup, T. Hegeman, W. L. Ngai, S. Heldens, A. P. PÁlrez, T. Manhardt, H. Chafi, M. Capota, N. Sundaram, M. Anderson, I. G. Tanase, Y. Xia, L. Nai, and P. Boncz. Ldbc graphalytics: A benchmark for large-scale graph analysis on parallel and distributed platforms, a technical report. Technical report, Delft University of Technology, 2016.
- [4] L. Nai, Y. Xia, I. G. Tanase, H. Kim, and C.-Y. Lin. GraphBIG: Understanding graph computing in the context of industrial solutions. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC ’15, pages 69:1–69:12, New York, NY, USA, 2015. ACM.