HPC vs. The Cloud: A Graph Based Benchmarking

Lance Fletcher (TAMU & LLNL), Trevor Steil (LLNL), Grant Johnson (LLNL), Roger Pearce (LLNL)

Introduction

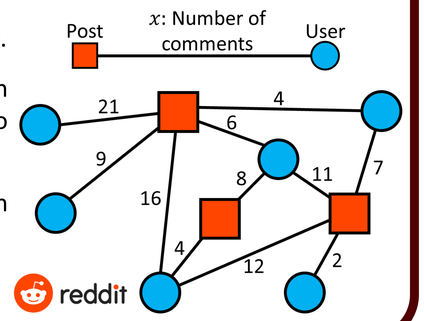
With cloud computing offering many benefits, such as flexibility, scalability, and cost-effectiveness, it has emerged as a capable HPC platform for many applications. This poster aims to address the following questions:

- *How will our MPI-based Graph algorithms perform in the AWS HPC environment?*
- *Will our latency hiding techniques in YGM [3] overcome the increased latency of AWS's EFA network?*

We have designed a set of benchmarks based on realistic analysis situations of opensource Reddit data consisting of over 14-billion comments. The irregular scale-free topology of this real data causes irregular communication patterns and provide a unique set of benchmarks.

Methods

- The comparison was conducted between LLNL's Ruby, a traditional HPC, and Amazon EC2 Hpc6a instances. Each compute node on Ruby has dual Intel Xeon CLX-8276L processors totaling 56 cores along with 192GB of DRAM per node. Each EC2 Hpc6a instance has 96 third-generation AMD EPYC™ cores along with 384GB of DRAM.
- Both systems have 100 Gbps networks (OmniPath vs EFA).
- Benchmarks are written in C++ using YGM [3] – an asynchronous communication library which is designed to handle irregular communication amongst processes.
- The algorithms were run on a bipartite reddit graph consisting of users and posts [4].



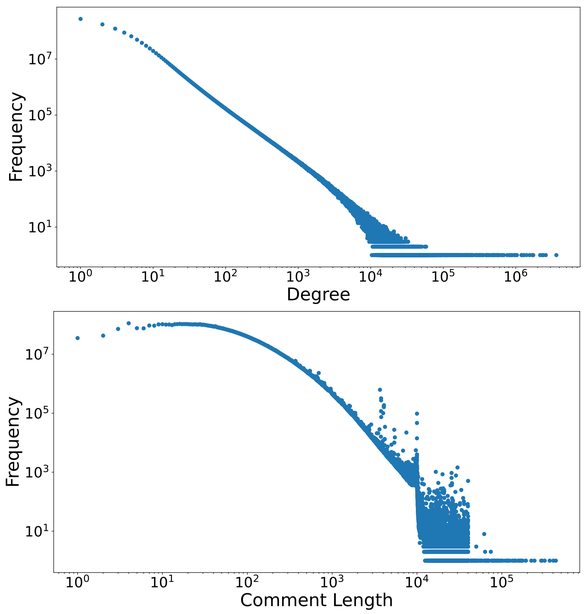
References & Acknowledgements

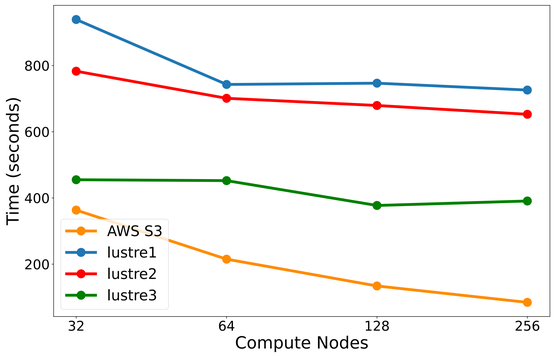
These experiments were performed at Lawrence Livermore National Laboratory HPC facilities. We appreciate the EC2 Hpc6a compute time donated by AWS for these experiments.

[1] M. Eidsaa and E. Almaas, 's-core network decomposition: A generalization of k-core analysis to weighted networks', *Phys. Rev. E*, vol. 88, p. 062819, Dec. 2013.
[2] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi, 'K-Path Centrality: A New Centrality Measure in Social Networks', in *Proceedings of the 4th Workshop on Social Network Systems*, Salzburg, Austria, 2011.
[3] YGM. https://github.com/LLNL/ygm
[4] J. Baumgartner, "pushshift.io website," https://pushshift.io, 2021

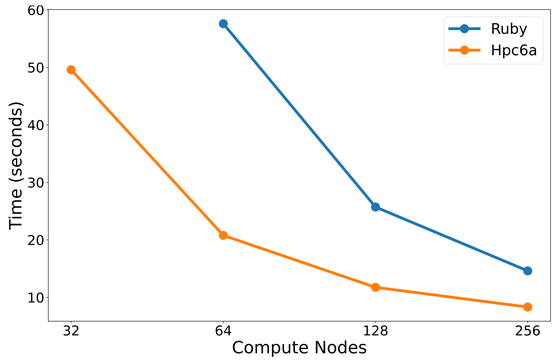
Results

- The Reddit graph is constructed from 14.3 billion comments made on the social network from 2005 to 2022 [4].
- Bipartite weighted graph between author and page is built using a counter for the weight.
- The graph contains **1 billion nodes** and **8.2 billion edges.**
- The two charts below show vertex degree and comment length distributions, respectively.

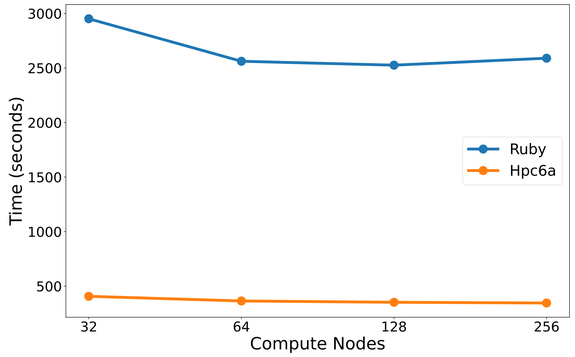




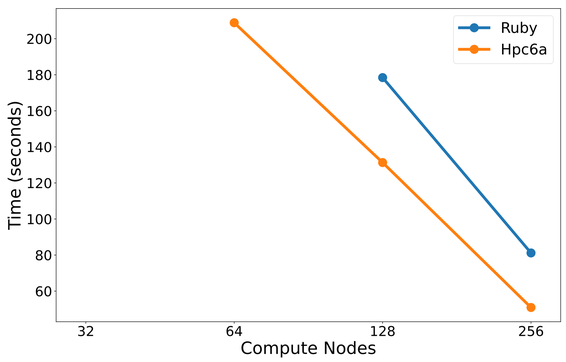
Ingest: An input comparison between Ruby's parallel file systems lustre{1,2,3} and AWS S3 Bucket storage performed by ingesting 14.3 billion JSON records, a total of 15TiB.



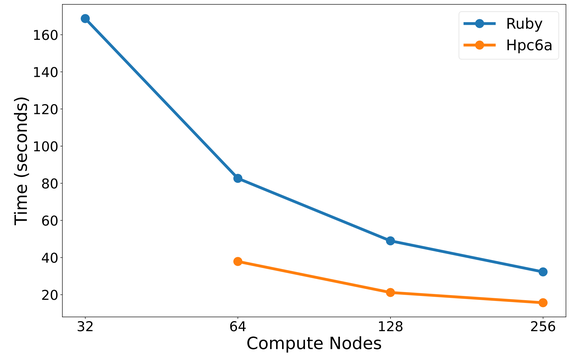
Sort: A pivot-based sorting of all 14.3 billion reddit comment strings including duplicates.



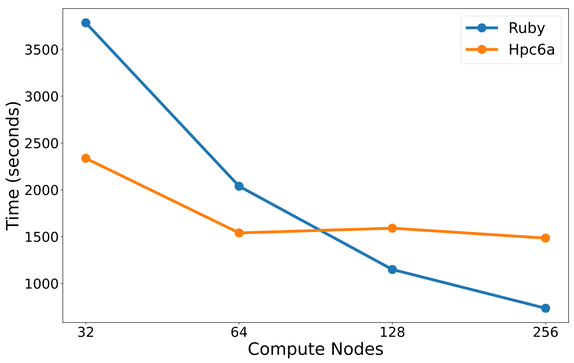
***K*-path Centrality:** A node centrality metric based on determining the number of random walks of length K that pass through a node [2]. In this experiment, K = 10 and 500 million random walks were taken.



Jaccard Similarity (Truncated): Calculates a similarity score between every pair of connected nodes. Did not include nodes with degree > 10,000. Outputs a matrix with 237.3 billion non-zero entries.



Connected Components: A label-propagation implementation which finds the number of connected components comprising the reddit graph. Requires 45 barriers (BSP steps) for the Reddit graph.



***s*-core Decomposition:** A generalization of the k-core decomposition which can be applied to weighted graphs [1]. Requires 5013 barriers (BSP steps).

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52 07NA27344 (LLNL-POST-848130). Funding from LLNL LDRD project 21-ERD-020 was used in this work.