GRAPH/Z: A Key-Value Store Based Scalable Graph Processing System

Tonglin Li[1], Chaoqi Ma[1], Jiabao Li[1], Xiaobing Zhou[2], Ke Wang[3], Dongfang Zhao[4], Iman Sadooghi[1], Ioan Raicu[1, 5]

[1]Illinois Institute of Technology, [2]Hortonworks, [3]Intel, [4]Pacific Northwest National Laboratory, [5]Argonne National Laboratory

ILLINOIS INSTITUTE OF TECHNOLOGY

Abstract

The emerging applications in big data and social networks issue rapidly increasing demands on graph processing. Graph query operations that involve a large number of vertices and edges can be tremendously slow on traditional databases. The state of the art graph processing systems and databases usually adopt master/slave architecture that potentially impairs their scalability. This work describes the design and implementation of a new graph processing system based on Bulk Synchronous Parallel model. Our system is built on top of ZHT, a scalable distributed key-value store, which benefits the graph processing in terms of scalability, performance and persistency. The experiment results imply excellent scalability.

Motivation

- ❑ Emerging uses of large graph data sets
- ❑ SQL databases don't handle it well
- ❑ Large data set can not fit in memory
- ❑ Current systems don't allow data change

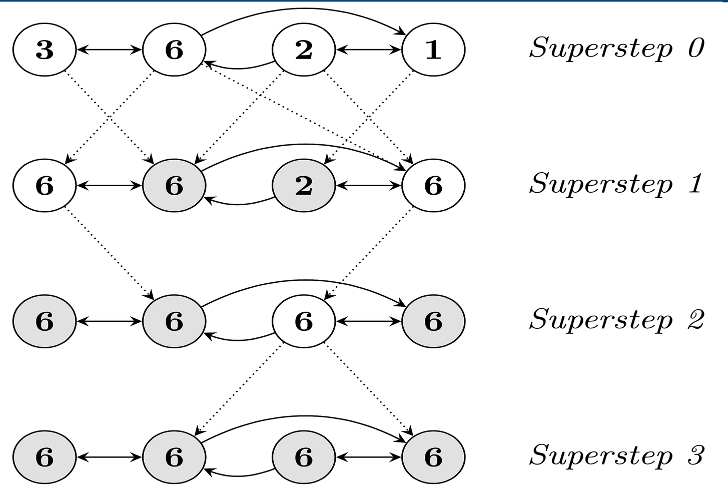
Contributions

- ❑ A BSP model graph processing system on top of ZHT.
- ❑ Utilizing data-locality and minimize data movement between nodes.
- ❑ Benchmarks up to 16-nodes scales.

BSP model

BSP[1] model

- ❑ Think like a vertex
- ❑ Vertices compute
- ❑ Edges communicate



Maximum Value Example. Dotted lines are messages. Shaded vertices have voted to halt. Figure from the Pregel paper, SIGMOD10 [2]

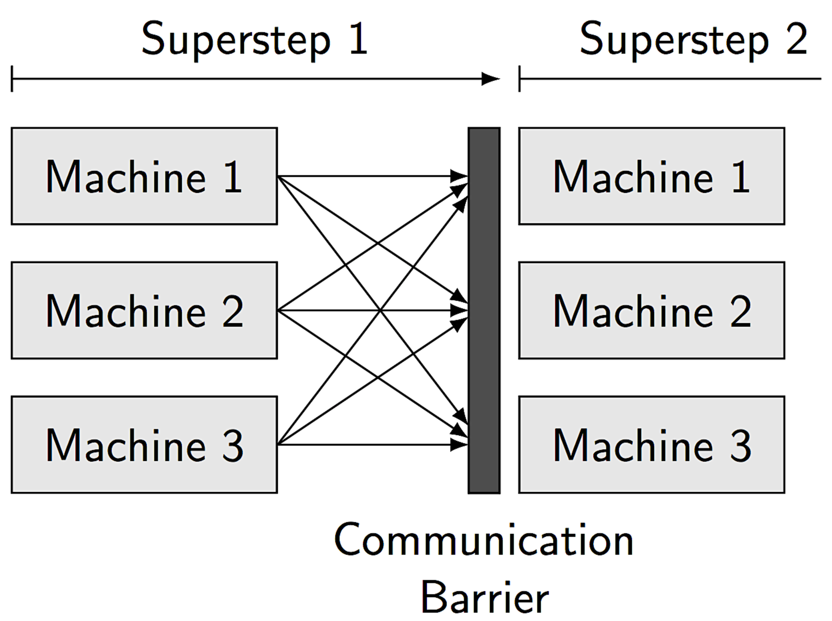
Design and Architecture

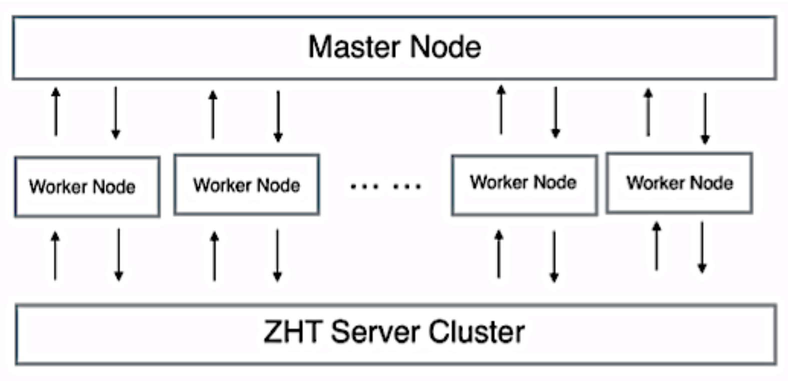
Design

- ❑ A Pregel[2] like graph distributed processing system
- ❑ Master: coordinates synchronization
- ❑ Use ZHT[3-5] as back end
- ❑ Store both intermediate and final result in ZHT

Features

- ❑ Handle large data sets
- ❑ Don't need to fit all data in main memory
- ❑ Dynamic data modification during running
- ❑ Load balance
- ❑ Fault tolerance
- ❑ Support checkpointing





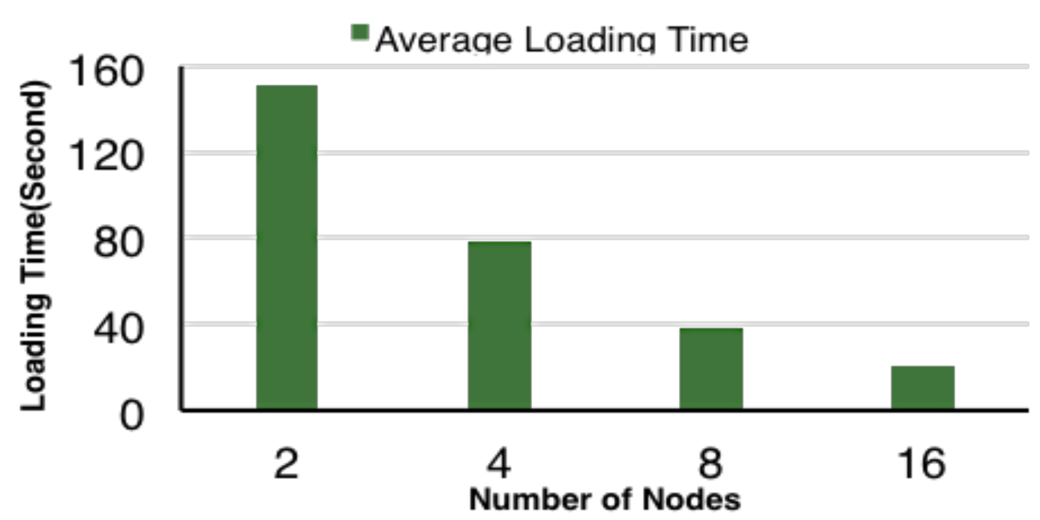
Graph/Z system architecture

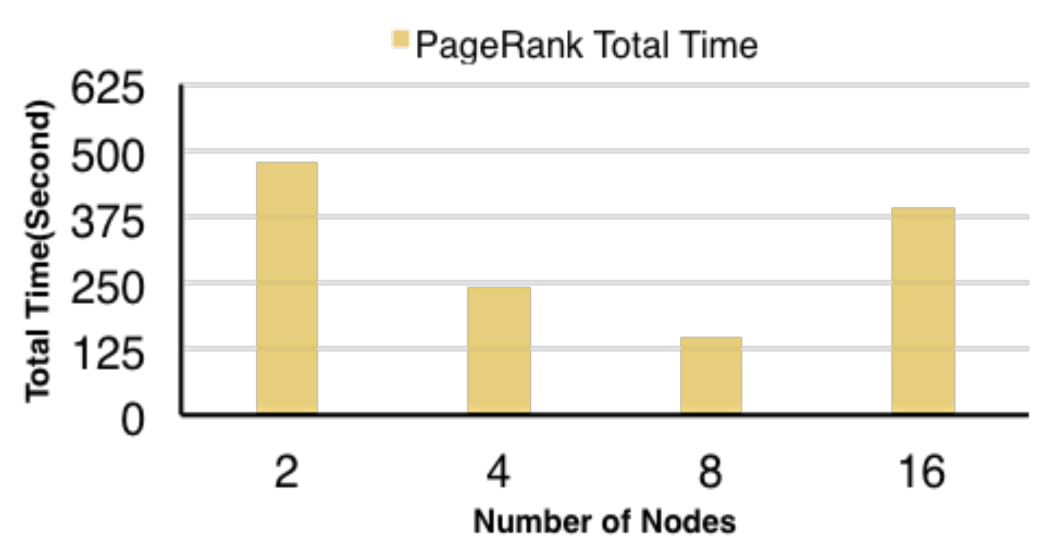
Evaluation

Experiment setup

- ❑ Test bed
  - 2-16 m3.2large Amazon EC2 spot instances
  - 2.5 GHz Xeon, 30GB RAM
- ❑ Data set
  - Web-Google from SNAP (Stanford Network Analysis Project)
  - 1M vertices and 5M edges

Preliminary results





Reference

1. Leslie G. Valiant, A Bridging Model for Parallel Computation. Comm. ACM 33(8), 1990, 103–111.
2. G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing,", SIGMOD 2010.
3. T. Li, X. Zhou, K. Brandstatter, D. Zhao, K. Wang, A. Rajendran, Z. Zhang, and I. Raicu, "ZHT: A light-weight reliable persistent dynamic scalable zero-hop distributed hash table," IPDPS '13.
4. T. Li, R. Verma, X. Duan, H. Jin, and I. Raicu, "Exploring distributed hash tables in high-end computing," *SIGMETRICS Performance Evaluation Review.*, 2011
5. T. Li, X. Zhou, K. Wang, D. Zhao, I. Sadooghi, Z. Zhang, and I. Raicu, "A convergence of key-value storage systems from clouds to supercomputers," , CCPE 2015

Acknowledgement

This work is supported in part by the National Science Foundation grant NSF-1054974.
This work used Amazon cloud resources through Amazon research grant.

U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

amazon webservices™