Pregel: A System for Large-Scale Graph Processing

Written by:

Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski

Brenden Bishop – November 23rd, 2013

WHAT IS IT ALL ABOUT?

- Pregel is a type of graphing algorithm. Normal graphs are made up of vertices and edges that send and receive messages in the form of programming iterations. This approach allows for a variety of different mapping algorithms to execute the graph.

- Pregel, one of these algorithms, is designed to be an efficient, scalable and fault tolerant implementation on clusters across thousands of computers. The outcome is a framework for handling large graphs that is open and easy to program.

IMPLEMENTATION

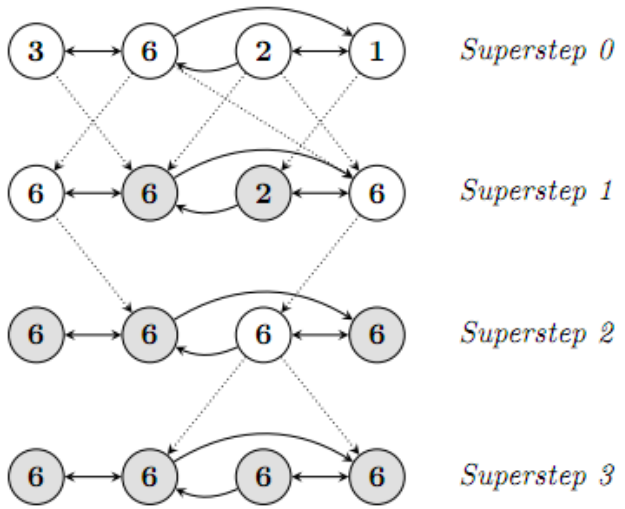
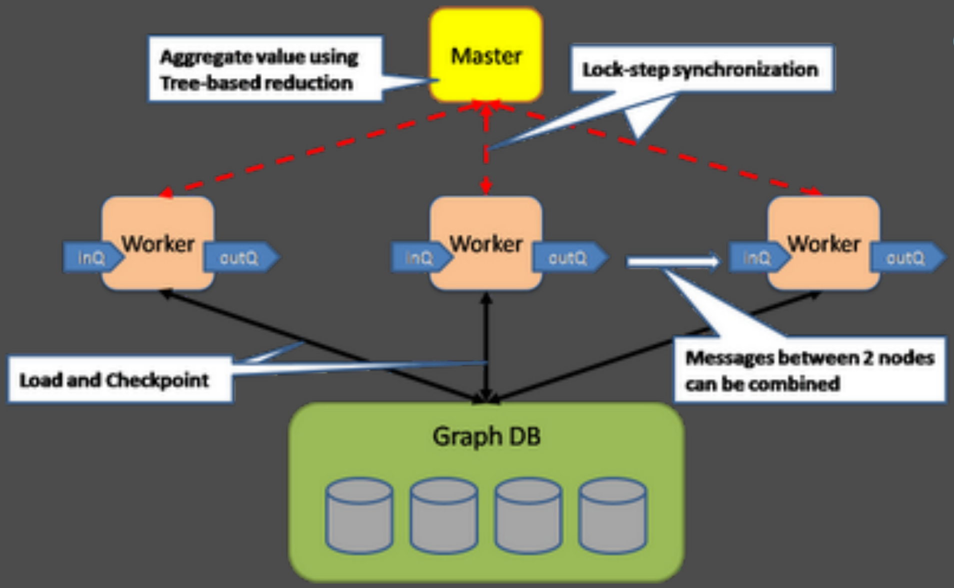


Figure 2: Maximum Value Example. Dotted lines are messages. Shaded vertices have voted to halt.

- Pregel is a C++ API based on Google’s clustered architecture of indices. Each cluster contains thousands of computers(organized into racks) that are interconnected locally, but distributed geographically.

- When implementing a Pregel program, one must subclass the predefined Vertex class. This class define three value types, that are used with Pregel, associated with vertices, edges, and messages. Each vertex has its own value of the declared type.

MY ANALYSIS



- I believe that Pregel is an interesting approach to an algorithm for handling graphs. It is basically a recursive approach to dealing with the complexity of navigating and dealing messages throughout indices. Most graph implementations contain the complexity and restrictive algorithm to those objectives.

Pregel, on the other hand, allows for simple and open implementation allowing the user to modify states of the indices as well as the path of messages to navigate the graphical structure of the nodes/indices.

From a programming perspective this algorithm does allow for more freedom and does make the process simpler, and although it inherits messaging and fault tolerance, I still think that it may cause much more room for user error.

Advantages and Disadvantages

- Pregel’s greatest advantage is scaling well on distributed clusters of computers.
- Expressive and easy to program.
- Deals well with a variety of different implementations, it is not restrictive.
- Has iterative computations which makes it substantially less complex.
- Has a linear run-time.

- Stability and effectiveness can rely heavily on the user or one implementing it.
- The entire computation resides in RAM, which is a limitation.
- Mainly designed for graphs where communication occurs only over edges.
- Performance will suffer when most vertices continuously send messages to most other vertices.

Real-World Use Cases

- Apache Giraph is an open source implementation of Pregel.
- Pregel runs on standard Hadoop infrastructure.
- Facebook is Apache Giraph and therefore based on Pregel.
- Web graph: PageRank (influential vertices) the likelihood of choosing a random link from a webpage.
- Social graph: popularity rank, personalized rank, shortest paths, shared connections, clustering (communities).
- Advertisements: target ads likelihood of ads reaching a certain population.
- Communication network: maximum flow, transportation routes, tracking and accidents can be sent as messages through indices.