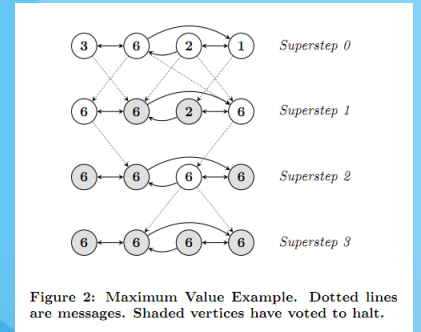Mary Miller
November 25th 2013

Pregel: A System for Large-Scale Graph Processing

Matthew H. Austern, Aart J. C. Bik, Grzegorz Czajkowski, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Malewicz, *Pregel: A System for Large-Scale Graph Processing*, Google, Inc.

Idea

- Graphs are difficult to utilize correctly

- A user takes a directed Graph and feeds it into Pregel.

- Allow for the implementation to take place
    - Looping through the supersteps and running the compute method on every machine until there are no active vertices (all have come to a halt).

- Returned to the user is a directed graph

Implementation Overview

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

- All code copied onto a group of machines.

- One machine assigned master to coordinate the workers.
  - Determines graph's partitions, assigning workers with specified number of partitions (could be user input)

- All other machines are workers to execute the code on specified vertices.

- User's inputs are assigned by the master to the workers.
  - All inputs contain a random number of vertices and edges.

- All vertices become active once all inputs have finished loading.

- Workers perform first superstep (looping through active vertices, done in parallel) followed by compute()

- # of partitions = # of threads

- Workers respond to master by mentioning how many active vertices there will be in the next superstep (at any point the master can obtain the records of these responses)

- Another superstep will be put into motion if there are active vertices or "any messages in transit."

Analysis of the Idea and implementation

- Address's many issues that cannot be handled previously
  - Ex. MapReduce, or writing your own infrastructure.

- Break down into master and workers allows for great organization and a clear end goal being monitored by one machine.

- This simple overall idea, allows for users to be able to utilize the more complex underlying without having to do too much (if they don't want to).

Advantages/ Disadvantages

Advantages:

- Allows for easy user input with API
- Having a backup system, If a round fails then every machine begins again.
- Easily scalable
- Open Source

Disadvantages:

- As information is kept in memory, making sure there is enough RAM.
- User can choose partitions which allows for inefficient use of resources if done incorrectly.
- There are disabled check points.

World use cases

For any large database could use this.

For example:

Amazon, needs to find all values that correspond to a specific product that a customer is looking for on their website.

Facebook, if they need to find a specific person or college,etc.