Pregel: A System for Large-Scale Graph Processing and A Comparison of Approaches to Large Scale Data Analysis

By Graham Harrison
5/8/2014

Malewicz, G., Austern, M., Bik, A., Dehnert, J., Horn, I. Leiser, N., Czajkowski, G. (2010, June 11). *Pregel: A System for Large-Scale Graph Processing.* Print

Pavlo, A., Paulson, E., Rasin, A., Abadi, D., DeWitt, D. Madden, S., Stonebraker, M. (2009, July 2). *A Comparison of Approaches to Large-Scale Data Analysis.* Print.

Why Pregel?

- Large Scale Graph Processing (lsgp) is very relevant
  - Social networks, disease outbreak paths, etc.
- Problems with current lsgp solutions
  - Creating your own distributed computing platform (dcp): difficult
  - Existing dcp: not flexible for wide range of lsgp problems
  - Single computer graph algorithms: limit scale
  - Existing parallel graph systems: insufficient fault tolerance
- Solution: Pregel!
  - Scalable, Fault Tolerant, Flexible

Pregel's Implementation

- Scalable to large sizes
  - Computation distributed over multiple computers (nodes)
    - Master: Coordinates workers, does have any portion of graph
    - Workers: Have partition of graph which they perform computations on and send data back to the master
      - Multiple workers allows parallel computation
  - Supersteps: computation focuses on an individual vertex. receives messages from S-1, sends messages to S+1 and modifies the vertex V
- Flexible for many types of graphs and problems
  - Supports many graph data types: decouples input file interpretation from graph computation
  - Users can make new methods by making subclasses of the existing Aggregate class and can modify the Compute method by overriding it
- Fault Tolerant
  - Check pointing: at beginning of Superstep, workers save state of their graph partition
  - If failure occurs, partitions of failed workers reassigned to working ones using data saved from the Checkpoint
  - Recovering a partition of the graph instead of the entire graph saves compute recourses

Analysis

- Simplicity is the key
  - Pregel’s partitioning of a graph to multiple workers makes processing faster and recovery more effective
  - The simplicity of its methods enables it to support a large range of graphs and problems by giving the user more control
  - Focusing computation on single vertices instead of all of them at once is efficient and allows for larger graphs to be analyzed

Parallel DBMS’s

- Language
  - Unlike Pregel which is coded in C++, Parallel DBMSs use SQL to query data
- Schema
  - Pregel: Schema is created manually after load time if ever
  - DDMSs: requires a schema that performs parsing at load time.
- Indexes
  - Pregel: no built in indexes, must be entered manually
  - DBMSs: built in indexes
- Data Distribution
  - Pregel: Data distribution done manually by programmer
  - DBMS: Use query optimizer to distribute data automatically
- Startup time
  - Pregel: There is a delay between startup time and the first computations
  - DBMS: Parallel DBMS are started at OS boot time and are ready instantaneously
- Loading
  - Pregel: does not transform data when it is initially loaded
  - DBMS: Are able to reorganize input data when it is loaded

Advantages and Disadvantages of Pregel vs Parallel DBMSs

- Advantages
  - Because Pregel partitions the graph to different workers, it handles mid computation failures better then Parallel DBMSs
  - Faster Data Loading
    - Because Pregel does not transform data when it is loaded, it is able to initially load data faster than a Parallel DBMS
- Disadvantages
  - More nodes are required with the Pregel system than a Parallel DBMS and must be supported by superior hardware that is expensive
  - Pregel does not perform efficiently as a Parallel DBMS because its messages generate significant overhead and it requires more local files to be created during the Map and Reduce jobs
  - Pregel is stereotypically harder to use the a Parallel DBMS because it uses a low level language (C++) rather than SQL which is used in DBMSs
  - Because Pregel has no built in indexes, its searches are slower than a Parallel DBMS's
  - Pregel requires a greater amount of manual coding to be done than a Parallel DBMS for data distribution, indexing, and schema creating