State of the Art Graph Processing

● Use BGL, LEDA, NetworkX, JDSL, Stanford GraphBase, etc.... on a single node

● Shoe horn into MapReduce (inefficient but fault tolerant)

● Use parallel graph processing (efficient but not fault tolerance)
Examples of Graph Problems

- The web graph: PageRank
- Social graphs: friend group clustering
- Machine learning and other AI
Pregel Graph Model
Pregel Graph Model
Pregel Graph Model
Super-Step Computation

- Global synchronization barrier
- Updates to edge and vertex values
- Possible changes to topology
**Vertex-Based API**

- Vertex and outgoing edges
- Compute(msgs_it)
  - Receive messages
  - Update edge/vertex values
  - Send messages
  - Update aggregators
Global Aggregators

Vertex Sum: 29.4
Edge Sum: 18.3
Implementation

- Partition vertices amongst worker nodes
Implementation

- Master coordinates workers, enforces barrier
Worker Functionality

- Execute Compute() functions
- Combine, buffer and send messages
- Checkpoint state in GFS
- Leaves of aggregation tree
Aggregation

Master

+2

+3
-1

+4
-1
+3
-4

Local Reduce
Local Reduce
Local Reduce
Local Reduce
Master Functionality

- Execute workers in lock step
- Detect failed workers and initiate recovery
- Root of aggregation tree
Examples in this paper

- Shortest path to a node $s$
  - Each iteration, pass on the shortest distance I've seen so far to $s$ via my neighbors.
Examples in this paper

- Shortest path to a node $s$
  - Each iteration, pass on the shortest distance I've seen so far to $s$ via my neighbors.
Examples in this paper

- Shortest path to a node $s$
  - Each iteration, pass on the shortest distance I've seen so far to $s$ via my neighbors.
Examples in this paper

- Shortest path to a node \( s \)
  - Each iteration, pass on the shortest distance I've seen so far to \( s \) via my neighbors.

```
   4
  /   \
  1 —— 2
     |   |
     v   v
   1 —— 7
      ^   |
      |   v
  3 —— 4
     |
     v
  3

0
```

Diagram showing the shortest path from node 0 to node 1, then to node 2, and finally to node 7.
Examples in this paper

- Shortest path to a node $s$
  - Each iteration, pass on the shortest distance I've seen so far to $s$ via my neighbors.
Pregel Graph Model
Examples in this paper

● Shortest path to a node $s$
  ○ Each iteration, pass on the shortest distance I've seen so far to $s$ via my neighbors.

● Greedy clustering
  ○ Each iteration, receive clusters my neighbors know about, add myself, and pass on best ones.

● Page rank
  ○ Each iteration, update page rank based on messages received from incoming links.
Thoughts and Discussion
BSP Model, Pros and Cons
BSP Model, Pros and Cons

Pros
- Simpler API
- Simpler to build
- Easy checkpointing
- Avoid deadlock/races

Cons
- Limits parallelism
- Straggler problem
Pregel vs. MapReduce
Pregel vs. MapReduce

- In-place updates more efficient for graphs
  - Exploits long lived, static state (graph structure) and breaks down without
- Messaging phase looks like all-to-all shuffle
- Implement on MapReduce
  - Two storage types, messages and vertices
  - Map = (message) \rightarrow (dst \text{ vert id, message})
    (vertex) \rightarrow (\text{vert id, state})
  - Reduce = vert id, list[message/vertex] \rightarrow list[message/vertex]
This paper only discusses global checkpoint recovery.

- Local seems difficult, especially if graph permutes or has non-determinism.
Pregel Implementations

- Apache Hama: General BSP
- GoldenOrb: Mostly exact clone
- Giraph: Map-only Hadoop job

http://blog.acaro.org/entry/google-pregel-the-rise-of-the-clones