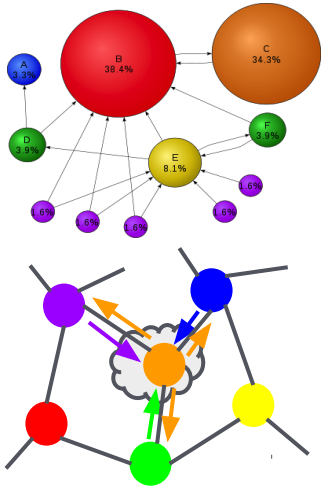
Pregelix: Dataflow-Based Big Graph Analytics

Yingyi Bu
Department of Computer Science, UC Irvine

Big Graph Analytics!

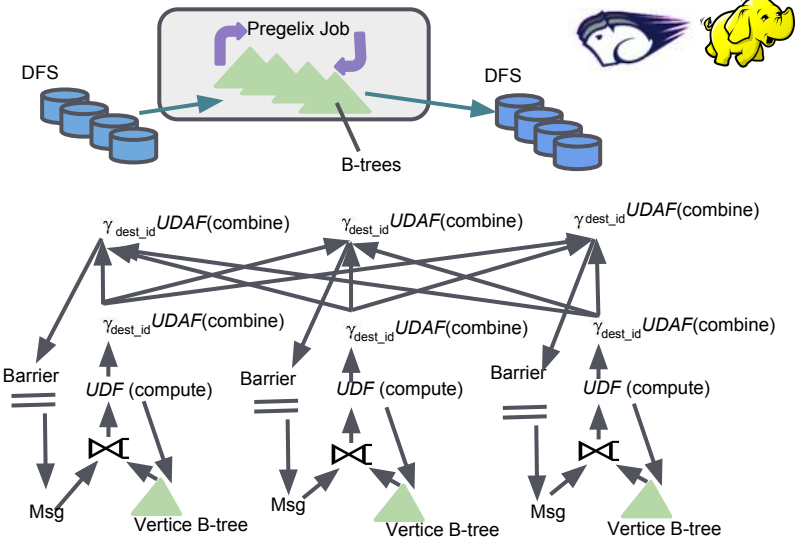


Think Like a Vertex

- Receive messages
- Update states
- Send messages

```
public class PageRankVertex extends Vertex<VLongWritable,
DoubleWritable, FloatWritable, DoubleWritable> {
  ........
  @Override
  public void compute(Iterator<DoubleWritable> msgIterator) {
      .......
      sum = 0;
      while (msgIterator.hasNext()) {
        sum += msgIterator.next().get();
      }
      setVertexValue((0.15 / getNumVertices()) + 0.85 * sum);
      sendMsgToAllNeighbors(vertexValue / getEdges().size());
      ....
  }
}
```

Shared-nothing Parallel Execution



Dataflow Approach

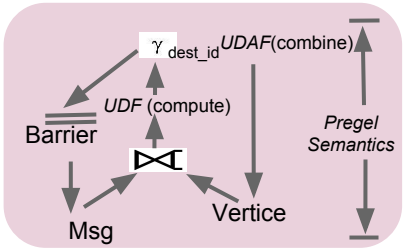
Our philosophy

Stop building one-off systems like Pregel, GraphLab, and Giraph, instead, build them on a **data-flow engine**!

Pregel
GraphLab
Giraph
.....

- Vertex/map/msg data structures
- Task scheduling
- Memory management
- Message delivery
- Network management

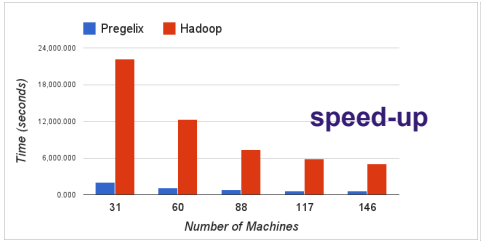
Pregelix

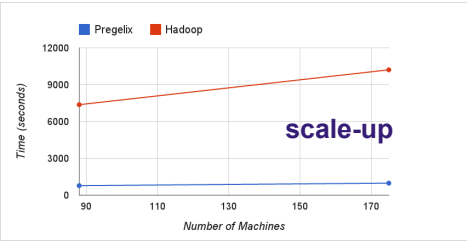


- Task scheduling
- Record/Index management
- Data exchanging
- Buffer management
- Connection management

A general parallel dataflow engine

Experiments





- Machines: Yahoo! Research cluster ~ 180 8 core,/12GB memory/4 disk machines.
- Dataset: Yahoo! AltaVista web graph (1,413,511,393 vertice, adjacency list, ~70GB)

Conclusions

- Vertex-oriented programming model is simple
- Dataflow implementation is neat and efficient
- We target Pregelix to be an open-source production system.

http://hyracks.org/projects/pregelix/

Sponsors: ebay, Google, htc quietly brilliant, Microsoft, NSF, ORACLE LABS, YAHOO!