Pregelix: Think Like a Vertex, Scale Like Spandex

Yingyi Bu (UC Irvine)
Work with: Vinayak Borkar (UC Irvine), Michael J. Carey (UC Irvine), Tyson Condie (Microsoft & UCLA)
Outline
Introduction
Programming Model
Example Applications
System Internals
Experimental Results
Related Work
Conclusions
Introduction

Big Graphs are becoming common
  - web graph
  - social network
  - ......
Introduction

• How Big are Big Graphs?
  ○ Web: 8.53 Billion pages in 2012
  ○ Facebook active users: 1.01 Billion
  ○ de Bruijn graph: 3 Billion nodes
  ○ ......

• Weapons for mining Big Graphs
  ○ Hadoop/Hive (Facebook)
  ○ Pregel (Google)
  ○ Distributed GraphLab (CMU)
Programming Model (Pregel)

- Think like a vertex
  - receive messages
  - update states
  - send messages
Programming Model (BSP)

Bulk synchronized
A synchronization barrier between iterations
Programming Model - API

- **Vertex (a super class for all applications)**

  ```java
  public abstract class Vertex<I extends WritableComparable, V extends Writable, E extends Writable, M extends Writable>
  implements Writable{

  /**
   * @param msgIterator an iterator of incoming messages
   */
  public abstract void compute(Iterator<M> msgIterator);

  ......
  }

- **Helper methods**
  - `sendMsg(I vertexId, M msg)`
  - `voteToHalt()`
Programming Model - Optional APIs

- **Combiner**
  - Combine messages
  - Reduce network traffic

- **Global Aggregator**
  - Aggregate statistics over all vertices
  - Done for each iteration

- **Early Termination (not in standard Pregel)**
  - Force the job to terminate
Example Applications

PageRank
ConnectedComponents
Shortest Paths
Reachability query

Start the Demo!
Our philosophy

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

- Task scheduling
- Memory management
- Message delivery
- Network management

Vertex/map/msg data structures

Pregelix
- UDAF (combine)
- UDF (compute)
- Barrier
- Msg
- Vertice

Pregel Semantics

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

A general purpose parallel dataflow engine
System Internals - Runtime

- Runtime Choice?

- The UCI Hyracks data-parallel execution engine
  - connection management
  - a set of operators: sorting, grouping, joining
  - task scheduling for jobs (a DAG of operators)
  - index support: B-tree, LSM-Btree, R-tree....
System Internals - Outer Join Execution Plan

**Diagram Description**

- **Barrier** nodes represent the synchronization points where data is received or sent.
- **UDF (compute)** nodes are used to compute the data.
- **UDAF (combine)** nodes combine the data from multiple sources.
- **Msg** nodes represent the messages passed between different components.
- **Vertice B-tree** nodes are used to store and retrieve data efficiently.
System Internals - Inner Join
Execution Plan
System Internals - Implementations

- **Right-outer join**
  - Index merging join
- **Sender-side group-by**
  - Sort + pre-clustered group-by
- **Data redistribution**
  - Hash merging repartitioning connector
  - Sender-side materialization pipelining
- **Receiver-side group-by**
  - Pre-clustered group-by
- **Inner join**
  - Index probing join
- **Set Union**
  - Index set union
System Internals

- Iteration-aware (sticky) scheduling?
  - 1 Loc: location constraints
- Caching of invariant data?
  - B-tree buffer pool -- 1 Loc: never flush dirty pages
  - File system cache -- free

Spark, GraphLab, HaLoop all have caches for this kind of iterative jobs. What do you do for caching?
Experimental Results

- Setup
  - Machines: Yahoo! Research cluster ~ 180 machines. Each has 8 cores, 12GB memory, 4 disk drives.
  - Dataset: Yahoo! webmap (1,413,511,393 vertex)
Experimental Results

- 10 iteration PageRank
- 1x webmap dataset
Experimental Results

- 10 iteration PageRank
- 1x webmap on 88 machines, 2x webmap on 175 machines
Related Work

- **Spark** [NSDI 2012]
  - OutOfMemoryError
- **HaLoop** [VLDB 2010]
  - Only 1.8X from Hadoop
- **Giraph**
  - OutOfMemoryError
- **Mahout**
  - OutOfMemoryError
- **Distributed GraphLab** [VLDB 2012]
  - Haven't tried yet (just published in September...
Conclusions

- Vertex-oriented programming model is simple
- Dataflow implementation is neat and efficient
- We target Pregelix to be an open-sourced production system, rather than just a research prototype:
  - http://hyracks.org/projects/pregelix/
Q & A