Large Scale Graph Data Processing

Hien To and Afsin Akdogan

05/01/2013

Outline

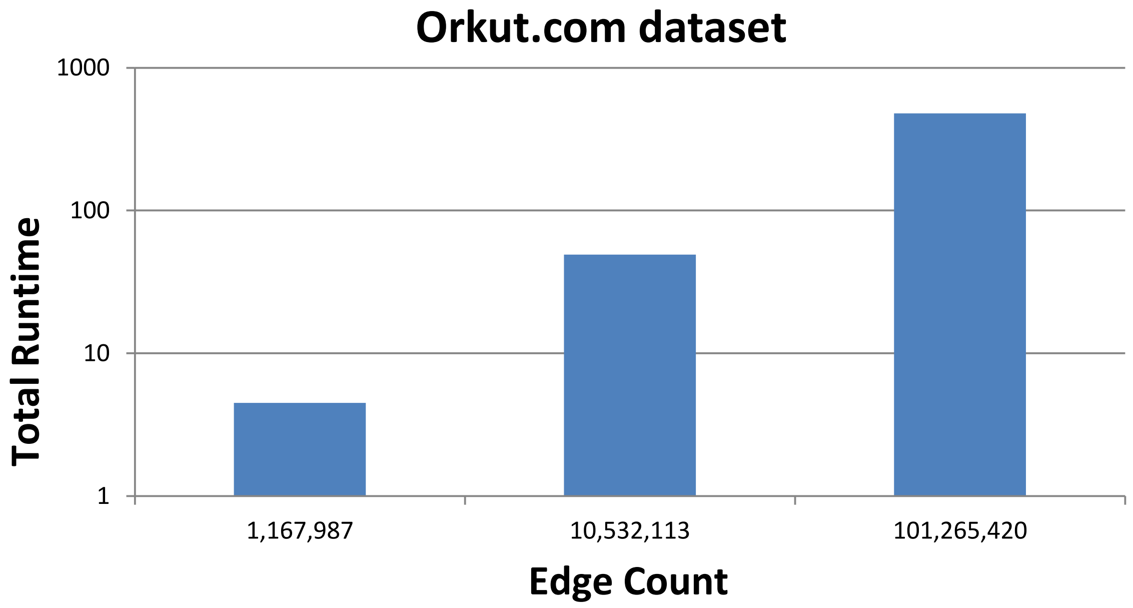
- Experiments
  - Page Rank
  - Weight Bipartite Matching
- Target Architecture
- Programming Models
- Partitioning Techniques

Experiments

- Hardware Specs
  - 4GB RAM
  - Intel Core 2 Duo CPU 3.16 GHz

Experiments

- Page Rank
  - We don’t include disk loading time
  - Whole graph **fits in memory**
  - Running time: O(|E|*I)
    - |E| is the number of edges
    - I is the number iterations. I = 100



Web has 14 billion pages and many more links/edges (source: worlwidewebsize.com)
Given 100M edges, Page Rank takes ~8 hours to compute rankings.

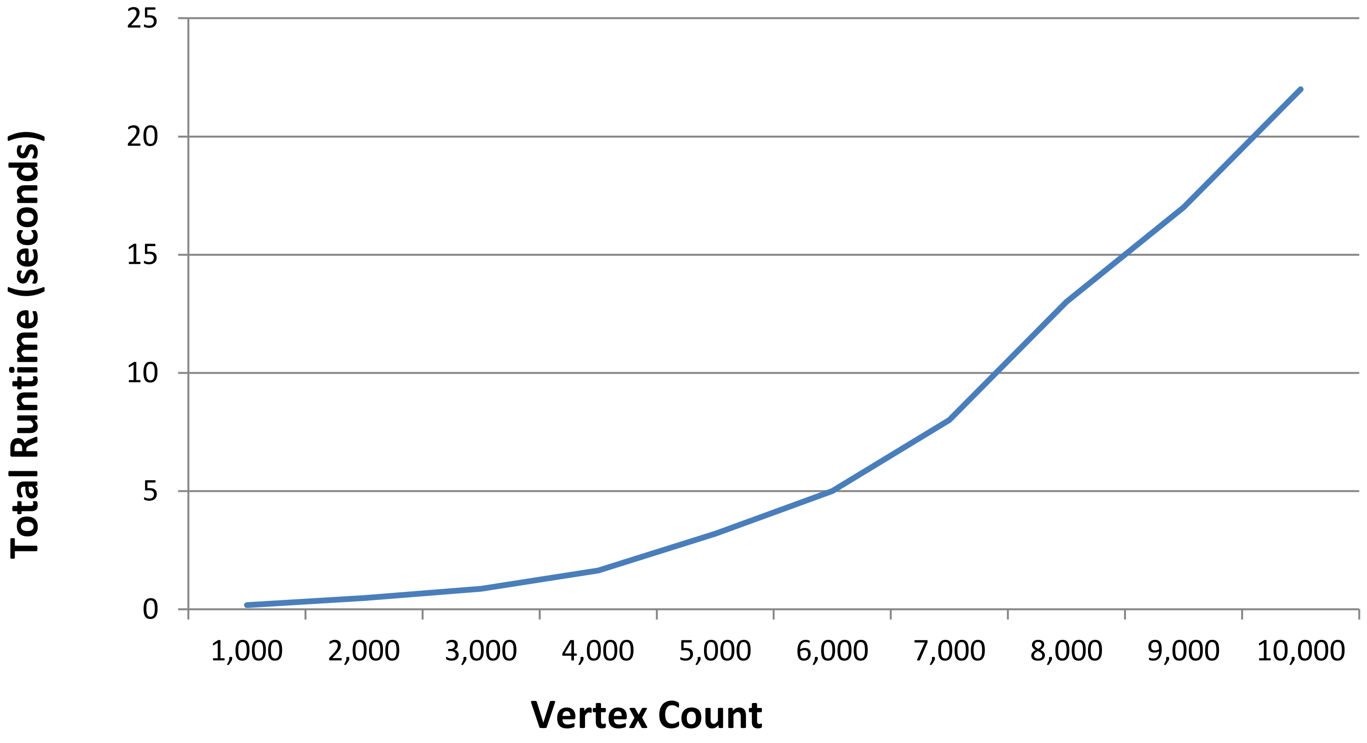
Experiments

- Page Rank

# Edges	Disk Loading (min)	Calculate Page Rank(min)
1M	0.12	4.5
10M	1.25	49
100M	14	471

Experiments

- Weight Bipartite Matching
  - Running time: O(|V|^2)
    - |V| is the number of vertices



Target Architecture

- Architecture
  - In memory
  - Shared nothing
  - Multiple servers, not multi-cores or GPUs

Programming Models

Programming models for large-scale data processing:

A) General-purpose

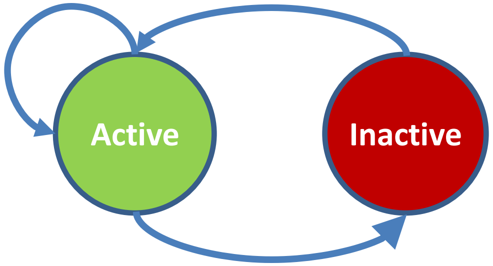
- MapReduce (MR), Pig Latin, Hive, etc.
- A graph algorithm can be expressed as a chain of MR steps. This requires passing the entire state of the graph from one step to the next resulting in enormous amount of Network I/O.

B) Vertex-based model for Graph Processing

- Pregel, GraphLab, PowerGraph

Programming Models

- Vertex-based programming model
  - Only active nodes send message. In MR all nodes need to processed in every MR step. Suboptimal!
  - Keep the vertices and edges in the servers and use network only to pass messages among vertices. Much less network and disk I/O!



- Challenges
  - Modeling TD-VR problem with vertex-based programming model
  - Partitioning technique selection
    - Changing degree of parallelism over the course of execution. Ex: All nodes in a server are inactive and the server stays idle!
    - We will study various graph partitioning techniques. Ex: vertex-cut, edge-cut, etc.

PowerGraph is Scalable

Yahoo Altavista Web Graph (2002):

One of the largest publicly available web graphs

1.4 Billion Webpages, 6.6 Billion Links

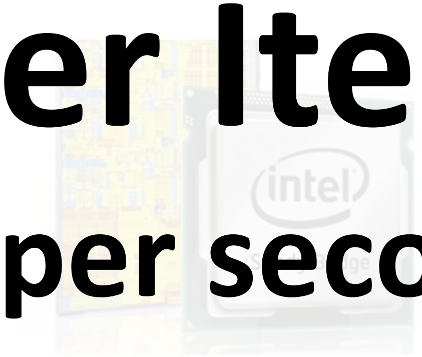
7 Seconds per Iter.

1B links processed per second

30 lines of user code

amazon web services™

64 HPC Nodes



1024 Cores (2048 HT)