X-Stream: Edge-centric Graph Processing using Streaming Partitions

Amitabha Roy, Ivo Mihailovic, Willy Zwaenepoel (EPFL - SOSP'13)

Presented by: Amir H. Payberah amir@sics.se Feb. 7, 2014



Graphs









- Large graphs are a subset of the big data problem.
- Billions of vertices and edges, hundreds of gigabytes.
- ► Normally tackled on large clusters.
 - Pregel, Giraph, GraphLab, PowerGraph ...
 - Complexity, power consumption ...

Could we compute Big Graphs on a single machine?



Presented by: Amir H. Payberah (SICS)

Challenges

- Disk-based processing
 - Problem: graph traversal = random access
 - Random access is inefficient for storage

Challenges

Disk-based processing

- Problem: graph traversal = random access
- Random access is inefficient for storage

Medium	Read (MB/s)		Write (MB/s)	
	Random	Sequential	Random	Sequential
RAM	567	2605	1057	2248
SSD	22.64	355	49.16	298
Disk	0.61	174	1.27	170
Note: CAlerta and lines AV blacks (dishared and) 1CM should				

Note: 64 byte cachelines, 4K blocks (disk random), 16M chunks (disk sequential)

Eiko Y., and Roy A., "Scale-up Graph Processing: A Storage-centric View", 2013.

Proposed Solution

Solution

X-Stream makes graph accesses sequential.

X-Stream Contribution

- Edge-centric scatter-gather model
- Streaming partitions

Edge-Centric Scatter-Gather Model

Scatter-Gather Programming Model

- State stored in vertices.
- Vertex operations:
 - Scatter updates along outgoing edges
 - Gather updates from incoming edges



Vertex-Centric Scatter

Iterates over vertices

for each vertex v
 if v has update
 for each edge e from v
 scatter update along e

Vertex-Centric Scatter-Gather (1/5)



src	dest	
1	3	
1	5	
2	7	
2	4	
3	2	
3	8	
4	3	
4	7	
4	8	
5	6	
6	1	
8	5	
8	6	

edges

for each vertex v
 if v has update
 for each edge e from v
 scatter update along e

Vertex-Centric Scatter-Gather (2/5)



edges

if v has update for each edge e from v scatter update along e

Vertex-Centric Scatter-Gather (3/5)



edges

if v has update for each edge e from v scatter update along e

Vertex-Centric Scatter-Gather (4/5)





for each vertex v
 if v has update
 for each edge e from v
 scatter update along e

Vertex-Centric Scatter-Gather (5/5)





for each vertex v
 if v has update
 for each edge e from v
 scatter update along e

Vertex-Centric vs. Edge-Centric Access



Edge-Centric Scatter

► Iterates over edges

for each edge e
 if e.src has update
 scatter update along e

Edge-Centric Scatter-Gather (1/5)



cuges		
src	dest	
1	3	
1	5	
2	7	
2	4	
3	2	
3	8	
4	3	
4	7	
4	8	
5	6	
6	1	
8	5	
8	6	

edues

for each edge e
 if e.src has update
 scatter update along e

Presented by: Amir H. Payberah (SICS)

Edge-Centric Scatter-Gather (2/5)



for each edge e
 if e.src has update
 scatter update along e

Edge-Centric Scatter-Gather (3/5)



for each edge e
 if e.src has update
 scatter update along e

Edge-Centric Scatter-Gather (4/5)



for each edge e if e.<mark>src</mark> has update scatter update along e

Presented by: Amir H. Payberah (SICS)

Edge-Centric Scatter-Gather (5/5)



for each edge e if e.src has update scatter update along e

Presented by: Amir H. Payberah (SICS)

Vertex-Centric vs. Edge-Centric Tradeoff

Vertex-centric scatter-gather: Rai

EdgeData RandomAccessBandwidth

- Edge-centric scatter-gather: <u>Scatters×EdgeData</u> <u>SequentialAccessBandwidth</u>
- ► Sequential Access Bandwidth ≫ Random Access Bandwidth.
- Few scatter gather iterations for real world graphs.

Streaming Partitions

Problem

• Problem: still have random access to vertex set.



Problem

• Problem: still have random access to vertex set.



Solution

Partition the graph into streaming partitions.

Partitioning the Graph (1/2)



edges

Partitioning the Graph (2/2)



Random access for free.

- A subset of the vertices that fits in RAM.
- All edges whose source vertex is in that subset.
- ▶ No requirement on quality of the partition, e.g., sorting edges.
- ► Consists of three sets: vertex set, edge list, and update list.

Streaming Partition Scatter-Gather

- The scatter phase iterates over all streaming partitions, rather than over all edges.
- The gather phase iterates over all streaming partitions, rather than over all updates.
- The vertex sets and edge lists remain fixed during the entire computation.

• The update list of a partition varies over time.

X-Stream Edge-Centric Scatter-Gather

```
// Scatter phase
for each streaming_partition p {
  read in vertex set of p
  for each edge e in edge list of p
    edge_scatter(e): append update to Uout
}
// Shuffle phase
for each update u in Uout {
  let p = partition containing target of u
  append u to Uin(p)
destroy Uout
// Gatter phase
for each streaming_partition p {
  read in vertex set of p
 for each update u in Uin(p)
    edge_gather(u): apply update u to u.destination
  destrov Uin(p)
```

Summary

- X-Stream
- Scatter-Gather model
- Edge-centric: sequential access to the graph edges
- Streaming partition: vertex set, edge list, and update list

Questions?

Acknowledgement

Some slides were derived from the slides of Amitabha Roy (EPFL)