Optimizing Analyses of Large Datasets (with Myria)

Daniel Halperin
Director of Research, Scalable Data Analytics
UW eScience Institute
UW Computer Science & Engineering
Distributed Datasets

- Relation $\text{TwitterK}(x, y)$: edge table from the Twitter social graph for 1000 people
- Distributed randomly among 72 machines
Project in MapReduce

- **SELECT** $x$
- **FROM** TwitterK

- What does the distributed plan look like?
Project in MapReduce

Fragment 0

Scan(public:adhoc:TwitterK)

Apply(x=$0)

Store(public:adhoc:JustX)
Project in MapReduce

Fragment 0

Scan(public:adhoc:TwitterK)

Apply(x=$0)

Store(public:adhoc:JustX)
Project and Filter

- **SELECT** \( x \)
  **FROM** TwitterK
  **WHERE** \( y < 50 \)

- What does the distributed plan look like?
Project and Filter

Fragment 0

Scan(public:adhoc:TwitterK)

Select(($1 < 50))

Apply(x=$0)

Store(public:adhoc:JustX)
Distributed Join

- SELECT A.x AS src, A.y AS link, B.y AS dst
FROM TwitterK AS A, TwitterK AS B
WHERE A.y = B.x

- What does the distributed plan look like?
Distributed Join

Fragment 1
- Scan(public:adhoc:TwitterK)
- ShuffleProducer(h($1))

Fragment 2
- Scan(public:adhoc:TwitterK)
- ShuffleProducer(h($0))

Fragment 0
- ShuffleConsumer
- ShuffleConsumer
- SymmetricHashJoin(($1 = $2); $0,$1,$3)
- Apply(src=$0,link=$1,dst=$2)
- Store(public:adhoc:TwoHopsInTwitter)
Flow cytometry data

SeaFlow

Laser

Microscope Objective

Nozzle

Pine Hole

d1
d2

FSC (Forward scatter)

Lens

Red fluo

Orange fluo

Jarred Swalwell
Francois Ribalet
Ginger Armbrust

Francois Ribalet

Jarred Swalwell

Ginger Armbrust
Flow cytometry data

SeaFlow

Francois Ribalet
Jarred Swalwell
Ginger Armbrust

- Continuous observations of various phytoplankton groups from 1-20 µm in size
  - Based on RED fluo: Prochlorococcus, Pico-, Ultra- and Nanoplankton
  - Based on ORANGE fluo: Synechococcus, Cryptophytes
  - Based on FSC: Coccolithophores
Flow cytometry data

- Cells(Cruise, Day, File, ID, [Size, Red, Orange...])
- Classification(Cruise, Day, File, ID, Type)
Join and Aggregate

- **SELECT** Cell.Cruise, **COUNT(*) AS** BigCells
  **FROM** Cell, Classifications
  **WHERE** Cell.ID = Classifications.ID
  **AND** Cell.Cruise = Classifications.Cruise
  **AND** ... = ...
  **AND** Cell.Size > 10000
  **AND** Classification.Type <> 'Noise'
  **AND** Classification.Type <> 'Beads'

- What does the distributed plan look like?
Triangles on Twitter

- SELECT \( A.x, B.x \) AS \( y \), \( C.x \) AS \( z \)
  FROM TwitterK AS A, TwitterK AS B, TwitterK AS C
  WHERE \( A.y = B.x \)
    AND \( B.y = C.x \)
    AND \( C.y = A.x \)

- Or in Datalog:
  \[ \text{Triangle}(x,y,z) \iff \text{TwitterK}(x,y), \text{TwitterK}(y,z), \text{TwitterK}(z,x). \]
Join1 (two-hops), followed by Join2 (filter to triangles)

• We looked at this plan in the profiler:
  • Answer had ~212K tuples, took ~17.8s
  • 54M tuples shuffled
  • Most work on worker 46, during two-hop path generation
  • We visualized fragments and communication to verify
  • Two-hop paths: 50.8M produced vs answer of 212K
HyperCube shuffle + Multi-way join

• Two key papers:
  Afrati & Ullman 2010 (Shuffle into hypercube);
  Veldhuizen 2012 (LeapFrogJoin)
HyperCube shuffle + Multi-way join

- Took ~2.4s, 212K results
- 14M tuples shuffled (50M)
- Work better balanced
- Multi-way join (LeapFrogJoin) required sorted data, so some blocking operations in pipeline while waiting for data from scan fragments to arrive.
Paper about our Myria Demo (with more stuff!)

• **Demo of the Myria Big Data Management Service**
  Daniel Halperin, Victor Teixeira de Almeida, Lee Lee Choo, Shumo Chu, Paraschos Koutris, Dominik Moritz, Jennifer Ortiz, Vaspol Ruamviboonsuk, Jingjing Wang, Andrew Whitaker, Shengliang Xu, Magdalena Balazinska, Bill Howe, and Dan Suciu.
  Demo at *SIGMOD 2014*.

• [http://r.halper.in/paper/halperin_myria_sigmod14_demo.pdf](http://r.halper.in/paper/halperin_myria_sigmod14_demo.pdf)

• See the website: [http://myria.cs.washington.edu](http://myria.cs.washington.edu)