Query Processing for Complex, Large-Scale Data Analysis

Leonidas Fegaras

University of Texas at Arlington

http://lambda.uta.edu/

09/18/2015

MRQL – a Querying System for Big Data

Project: MRQL – a Querying System for Big Data

Part of the Database Lab (ERB 514)

Current students:

PhD student: Upa Gupta
MS Student: Ahmed Ulde

Looking to recruit new MS and PhD students

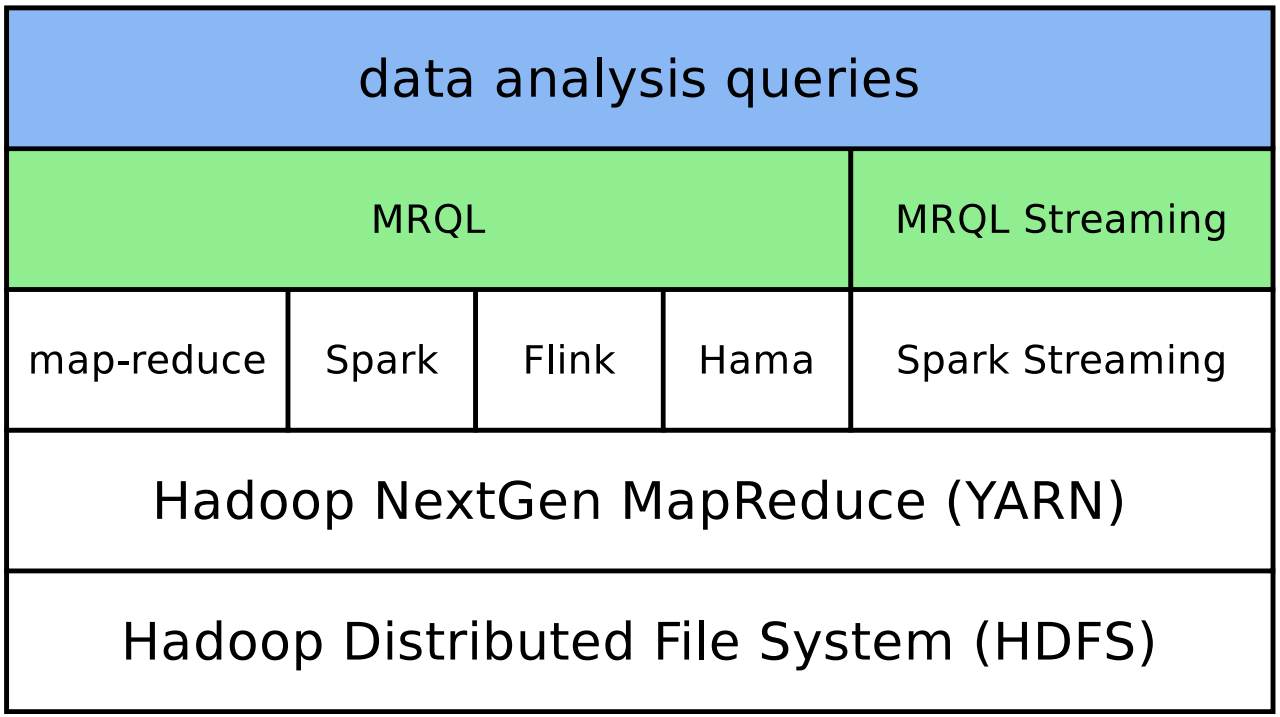
must be interested in:

- Big Data software (map-reduce, Spark, etc)
- open-source programming (Apache software) using Java

may become an Apache contributor and/or developer

MRQL

- Apache MRQL: Started at UTA in 2010 as a research project
- Now, an Apache incubating project
- A powerful and efficient query processing system for complex data analysis applications on Big Data
- Web site: `http://mrql.incubator.apache.org/`



MRQL

- is more powerful than existing Big Data query languages
- is able to capture most complex data analysis tasks declaratively
- is platform-independent:
  - the same query can run on multiple platforms on the same cluster: Hadoop map-reduce, Spark, Flink, and Hama
  - allowing developers to experiment with various platforms effortlessly
- a common front-end for the multitude of distributed processing frameworks emerging in the Hadoop ecosystem
- a tool for comparing these systems (functionality & performance)

Possible MS theses

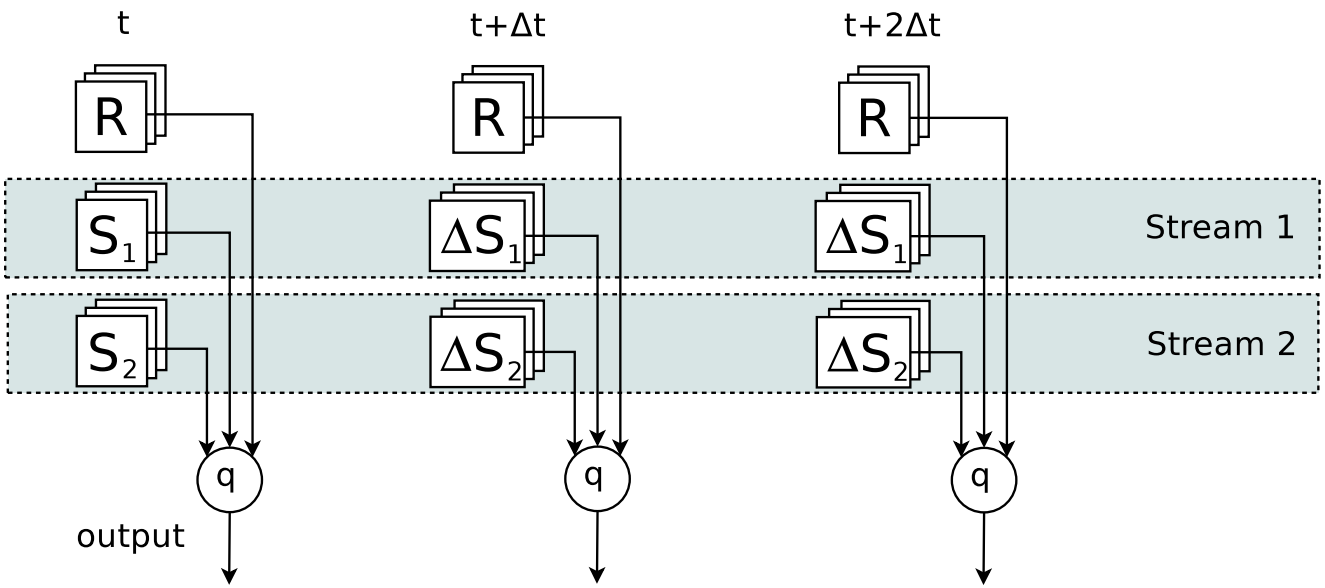
- Data analysis applications: implement certain data analysis algorithms in MRQL

  pagerank, clustering, matrix factorization, ...
- MRQL Streaming: Support for continuous queries on big data streams

  Currently, works on Spark Streaming. Need to make it work on Flink Streaming and Twitter Storm
- Incremental query processing

Distributed stream processing engines (DSPs)

- Support for **continuous queries** over multiple streams of data
- Data come in incremental batches ΔS
- Batch streaming based on **sliding windows**



Query $q(S_1, S_2; R)$ over one invariant and two streaming data sources

- Currently, works on Spark Streaming

Current project: incremental query processing

- *Problem:* translate any batch program (eg, PageRank) to an incremental program automatically
- *Solution:* Break the query $q(S_1, S_2; R) = a(h(S_1, S_2; R))$ such as:

$$h(S_1 \uplus \Delta S_1, S_2 \uplus \Delta S_2; R) = h(S_1, S_2; R) \otimes h(\Delta S_1, \Delta S_2; R)$$

- Requires program analysis & transformation