Big Data Management with Myria

Brandon Haynes and Magdalena Balazinska

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

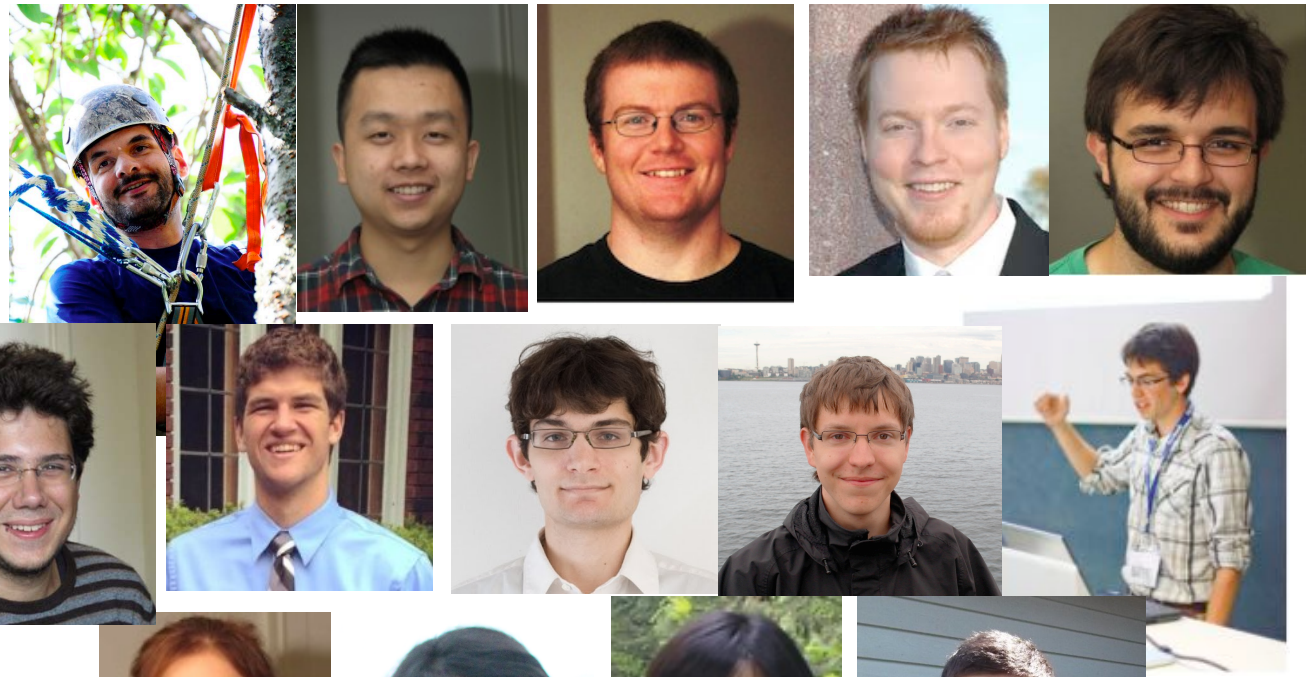
UNIVERSITY OF WASHINGTON

http://myria.cs.washington.edu

Myria Team

Magda Balazinska, Bill Howe, and Dan Suciu (faculty)
Software Engineer: Tobin Baker
Grad Students

- Shumo Chu
- Eric Gribkoff
- Brandon Haynes
- Jeremy Hyrkas
- Paris Koutris
- Brendan Lee
- Brandon Myers
- Ryan Maas
- Dominik Moritz
- Laurel Orr
- Jennifer Ortiz
- Jingjing Wang



Ugrad Students

- Yuqing Guo
- Dan Radion

Alumnae/Alumni: Victor Almeida, Lee Lee Choo, Dan Halperin, Vaspol Ruamviboonsuk, Emad Soroush, Mayukha Vadari, Andrew Whitaker, Shengliang Xu

Acknowledgments

The Myria Team

Our science collaborators

Our sponsors

- National Science Foundation, Moore & Sloan Foundations, Washington Research Foundation, eScience Institute, ISTC Big Data, Petrobras, and EMC

Overview of the Myria Stack

Stack for big data management and analytics

- A new big data mgmt & analytics **system**
  - Available open source
  - Runs in shared-nothing clusters (Amazon EC2)
  - Also runs in an HPC cluster at MIT
  - Think of it as Hive/Hadoop but faster
  - Think of it as Spark but faster
  - This of it as SQL Server or PostgreSQL but more scalable
- An **operational service** deployed at UW

- Developed by the UW database group and eScience

Myria Big Data Management Service

Myria is a cloud service: Just open browser and go!

Myria | Editor | Queries | Datasets | Report an issue | rest.myria.cs.washington.edu:1776 [72/72]

Write your code here, perhaps starting from one of the examples at the right.

```
1  good_opp_vct = scan(armbrustlab:seaflow:good_opp_vct_v4);
2
3  def avg_sd(x):[avg(float(x)),stdev(float(x))];
4
5  beads = select * from good_opp_vct where pop = "beads";
6  bead_stats = select avg_sd(fsc_small) as [fsc_avg,fsc_sd],
7                      avg_sd(chl_small) as [chl_avg,chl_sd],
8                      avg_sd(pe) as [pe_avg, pe_sd],
9                      Cruise from beads;
10
11 store(bead_stats,
   armbrustlab:seaflow:bead_stats_v4_bycruise_untrans);
```

▶ Execute the Query | Parse | Myria JSON

Query Language MyriaL

Developer Options

- ☐ Profile Query

  Profiling will make the query run a little bit slower but allows you to examine exactly how the query was executed.

- ☐ Compile to Multiway Join

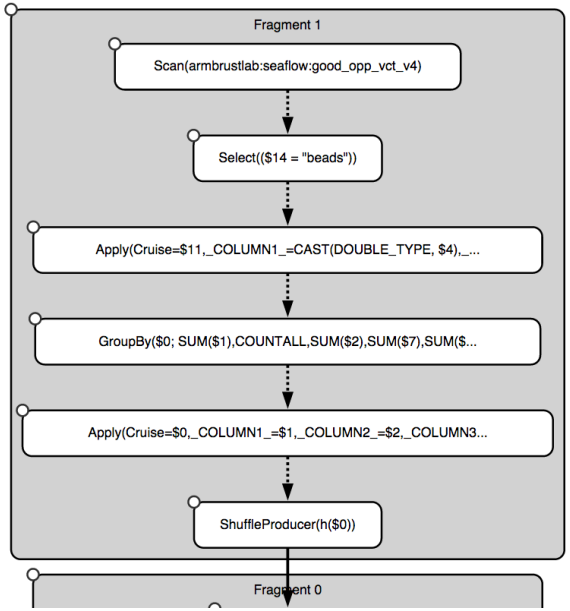
  Compile to multiway join rather than binary joins.

Examples | Datasets | Query Plan | Results

Visualization of the logical and optimized physical query plan.

Code parsed as Relational Algebra

Relational algebra converted and optimized into a Myria Physical Plan



Myria Demo on Amazon

http://demo.myria.cs.washington.edu

Myria Is a Cloud Service

Browser

MyMergerTree

Python/iPython

RACO

Web UI

Language Parser

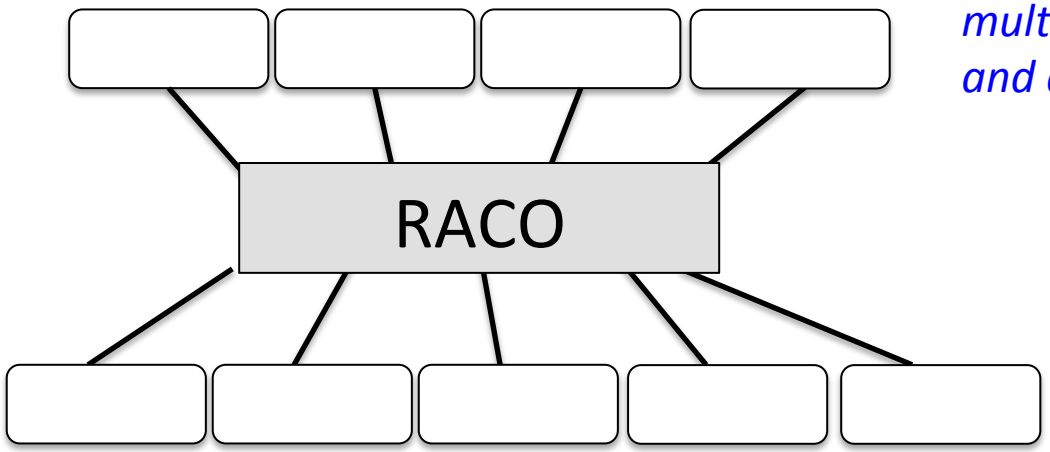
Query Optimizer

Google App Engine

MyriaX

Parallel Query Execution

Myria Supports Different Back-Ends

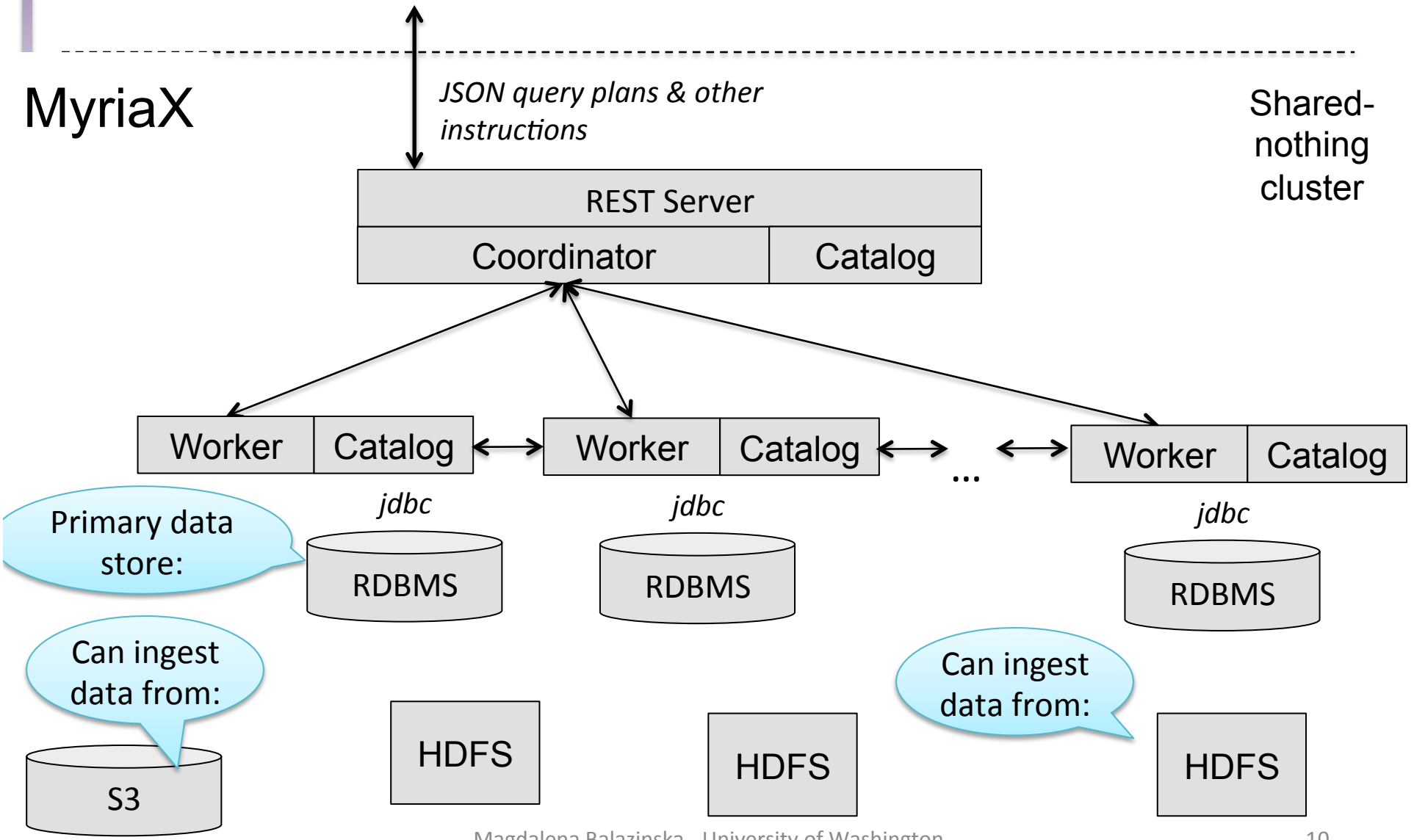


multiple languages and apps

multiple Big Data systems

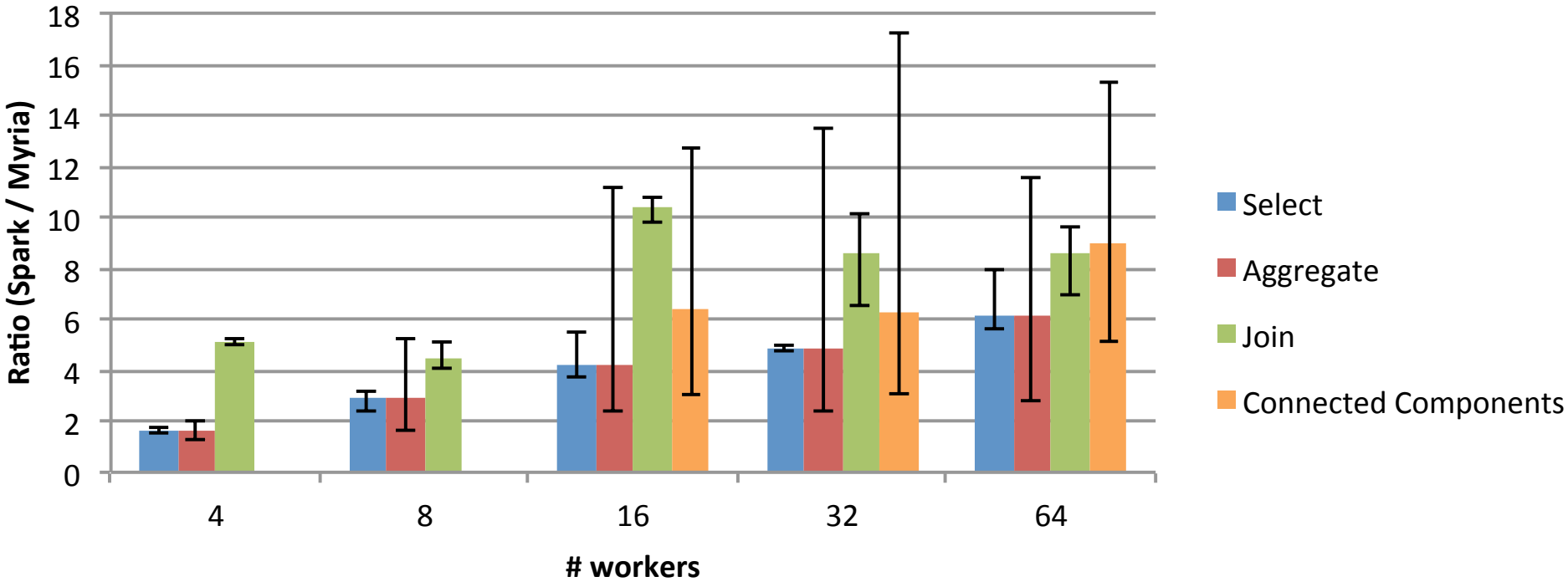
RAGO Goals: Optimizing middleware for big data systems – write once, run efficiently anywhere: Hadoop, Spark, MyriaX, straight C, RDBMS, PGAS, MPI, …

Myria Is a Parallel Data Mgmt System

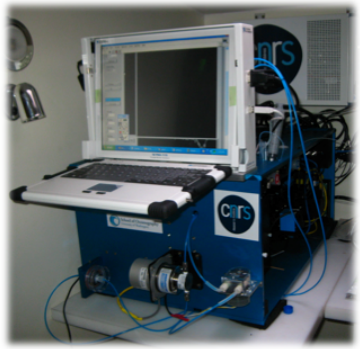


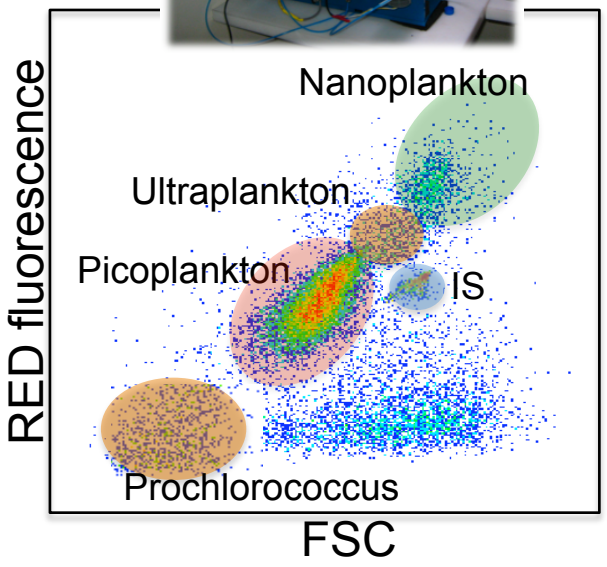
Performance Example

Input data: Twitter Graph 1.5 billion edges and 41 million vertices

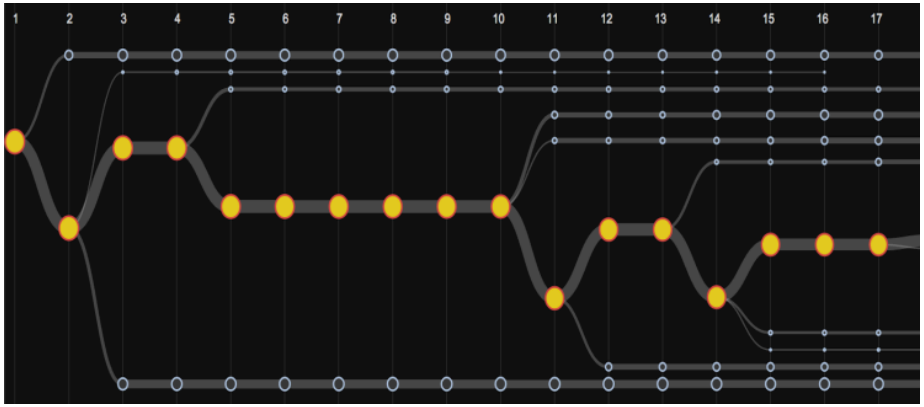


Example Myria Applications

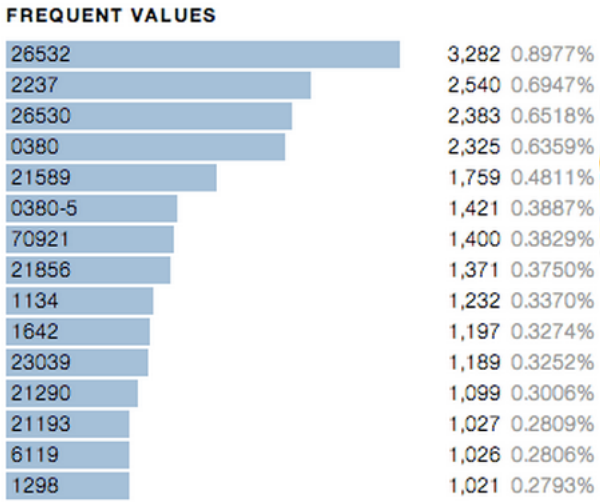




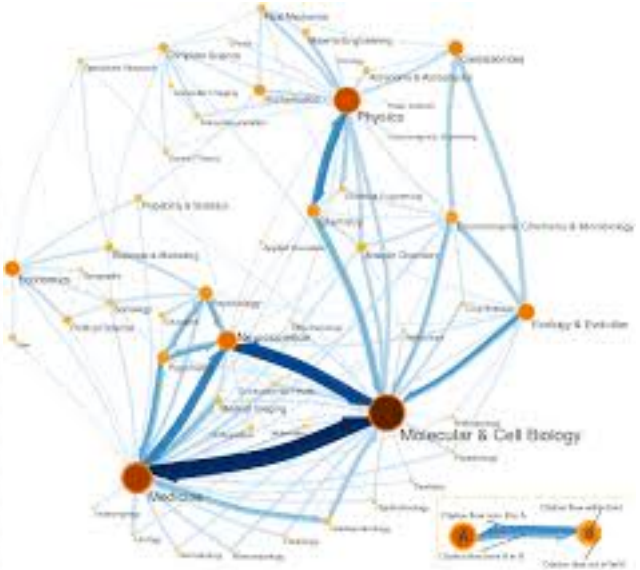
Environmental Flow Cytometry



Galaxy Simulations



Retail Analytics



Bibliometrics

Myria Demo

http://demo.myria.cs.washington.edu

- Demonstrating Web interface to Myria
  - Analyzing pre-loaded data
  - Analyzing data stored in S3
- Demonstrating Python interface to Myria
  - Basic Python and iPython
- Demonstrating spinning up your own cluster
  - Demo with Amazon EC2 but private cluster also OK

Myria Documentation

Docs: http://myria.cs.washington.edu/docs/index.html

Issues: Please post on github
https://github.com/uwescience/myria-stack

Users mailing list: myria-users@cs.washington.edu
To subscribe:
https://mailman.cs.washington.edu/mailman/listinfo/myria-users