



Systems Session Database Day 2015



Dep. of Computer Science & Engineering
University of Washington

The Exciting Times of “Big Data”



Everyone today has a big data problem

- Whether it is a data lake, data swamp, or data stream
- Whether they call it big data, data science, data wrangling, ..



Photo by Gary Bridgman / CC BY

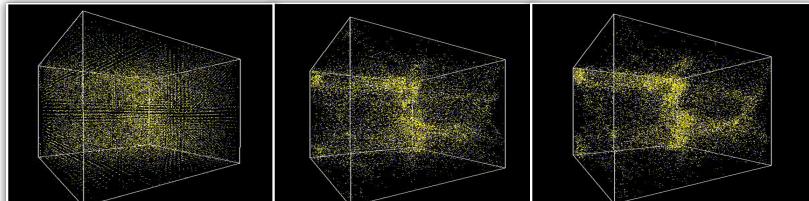
Challenging Application Requirements

Exciting and challenging requirements of campus applications

Use to motivate & test -- Often generalize beyond campus



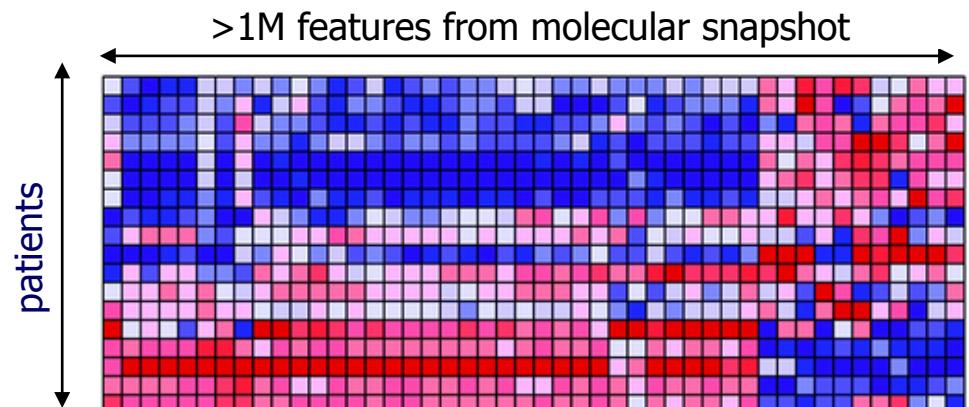
Picture from Deep Lens Survey (DLS: Tyson)



Picture from D. H. Stalder et. al. [arXiv:1208.3444](https://arxiv.org/abs/1208.3444) [astro-ph.CO]

N-body simulation data:

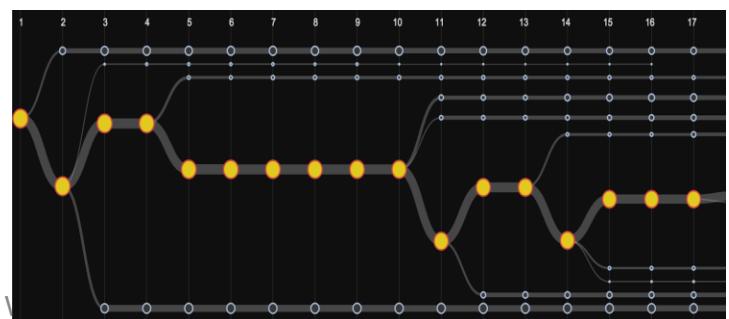
1. Manage hundreds of TB of data
2. Data clustering to extract galaxies
3. Graph analytics to study galaxy evolution



Genome data processing:

Picture from Su-In Lee

1. Linear algebra on large matrices
2. Novel machine learning algorithms



The Challenge

- *Everyone* needs to work with “big data”
 - Big data = data with large volume, velocity, or variety
- Need tools that
 - Can manage big data efficiently
 - Can analyze big data efficiently (complex analytics)
 - Are geared toward being used by data scientists
- Core focus
 - **How to make data scientists maximally productive?**

We Build on Open-Source Tools

Developed **ParaTimer** [SIGMOD10]

- Shows progress of DAGs of Hadoop jobs

Developed **PerfXPlain** [VLDB12]

- Explains the performance of Hadoop jobs

Developed **SkewReduce** [SOCC10] and **SkewTune** [SIGMOD12]

- Based on Hadoop and available as open source

Developed **HaLoop** [VLDB10]

- Faster iterative processing in Hadoop also open source

Developed **Array Proc. Methods** [SSDBM15, ICDE13, SIGMOD11]

- Array storage and query processing in SciDB



Goals of the Myria stack

- Advance state-of-the-art in big data systems
- Focus on efficiency and productivity
- Test on real applications and support real users

Deliverables:

- Built a new **big data mgmt & analytics system**
- Deployed and operate Myria as a **service**

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,
12       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

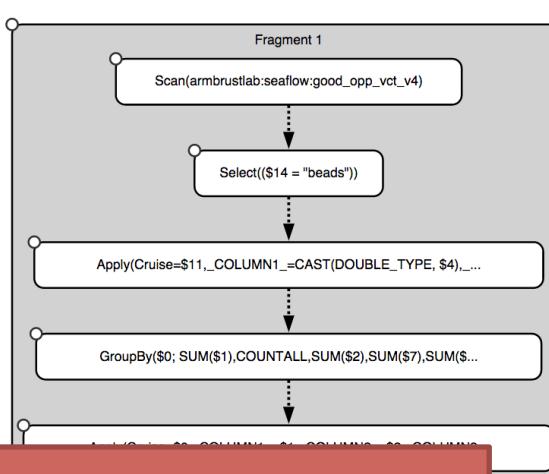
<http://myria.cs.washington.edu>

Examples Datasets **Query Plan** Results

Visualization of the logical and optimized physical query plan.

Code parsed as Relational Algebra

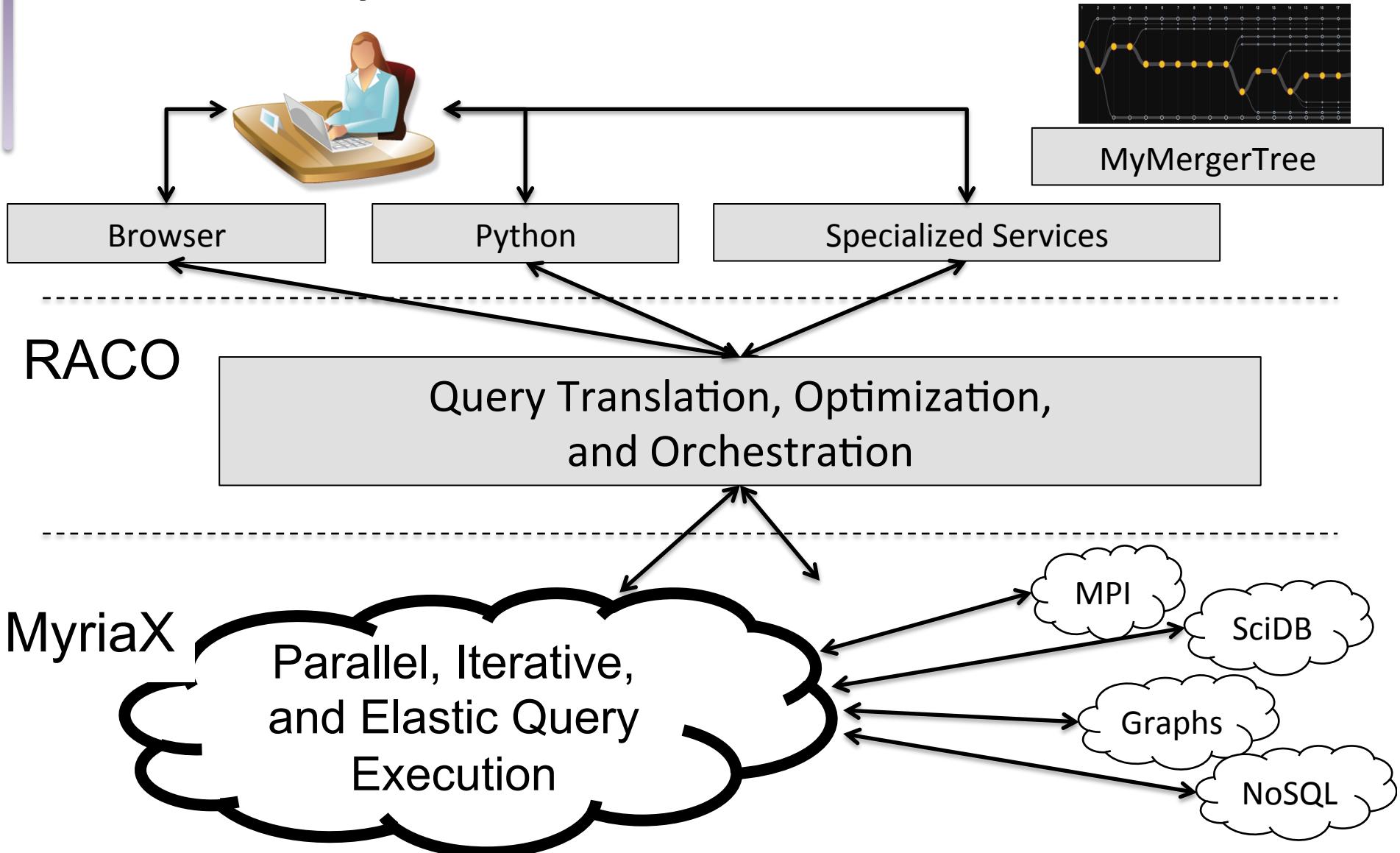
Relational algebra converted and optimized into a Myria Physical Plan



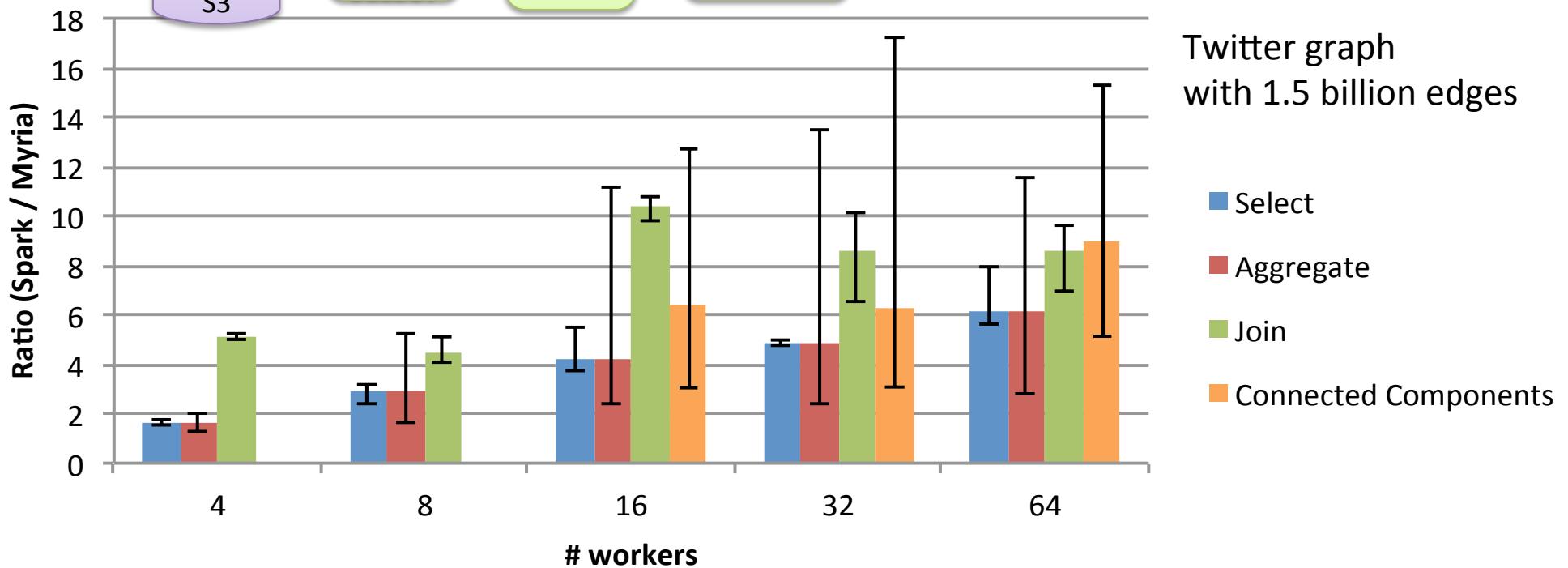
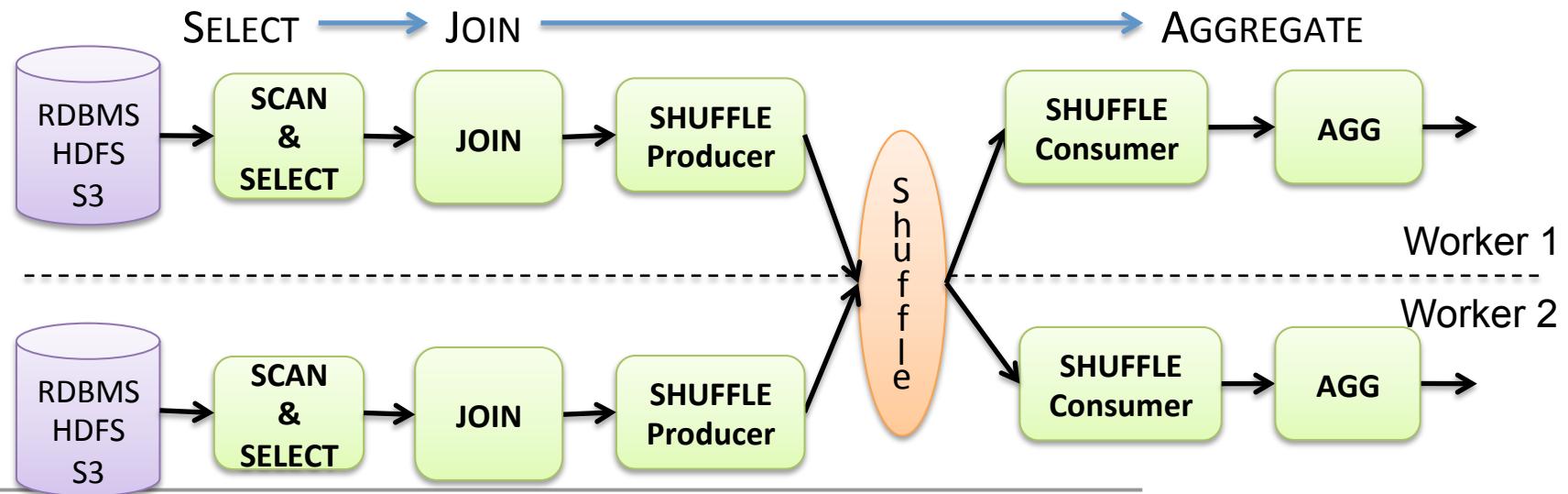
```
graph TD; subgraph Fragment1 [Fragment 1]; S1[Scan(armbrustlab:seaflow:good_opp_vct_v4)]; S2[Select((\$14 = "beads"))]; S3[Apply(Cruise=$11,...)]; S4[GroupBy($0; SUM($1), COUNTALL, SUM($2), SUM($7), SUM($...));]; end; S1 --> S2; S2 --> S3; S3 --> S4;
```

Fragment 0

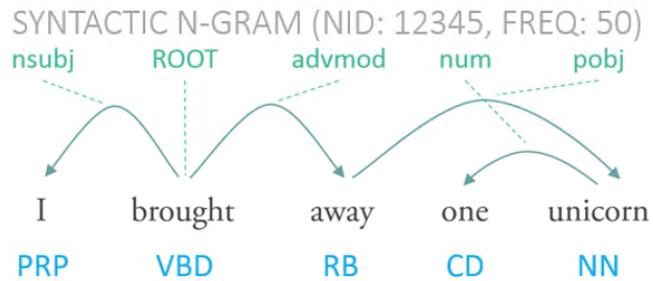
Myria Is a Cloud Service



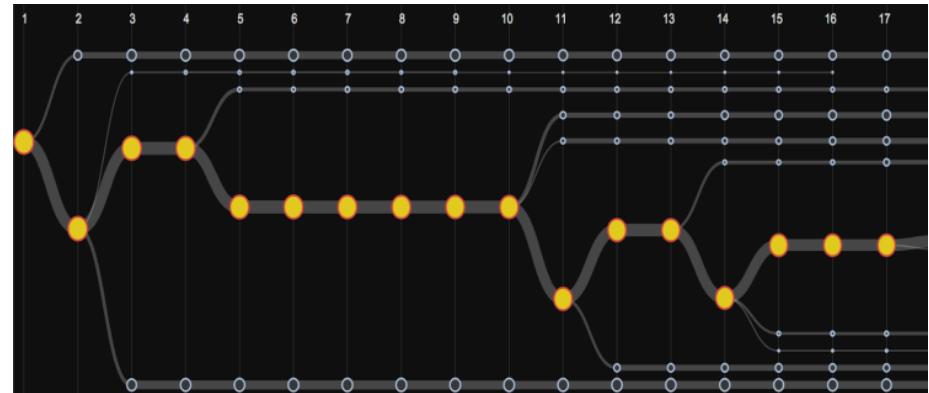
MyriaX Query Execution Engine



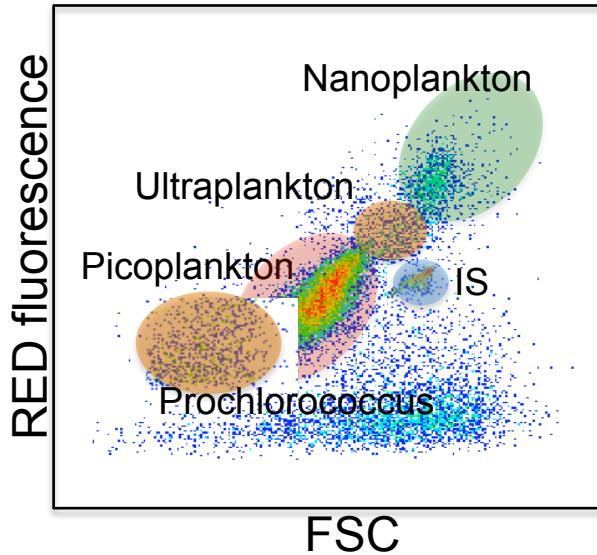
Example Myria Applications



Natural Language Processing



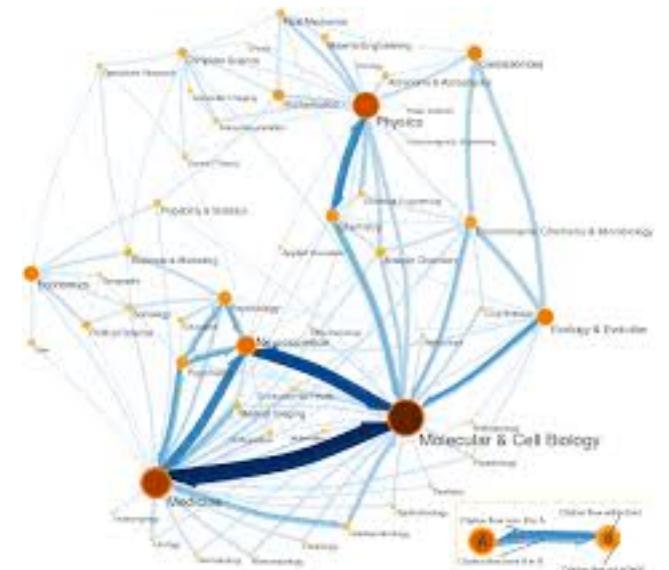
Galaxy Simulations



Environmental
Flow Cytometry



Telescope Images



Bibliometrics ¹⁰

Some of Key Research Themes

Efficient big data management and analytics

- Efficient multi-join query processing (**Shumo**)
- Iterative & in-memory query processing (**Jingjing**)
- Data summarization (**Laurel**)

Effective operation as a cloud service

- Personalized Service-Level Agreements (**Jennifer**)
- Query time guarantees (Brendon & **Jennifer**)
- Predictable and explainable performance (**Parmita & Helga**)

Easy to use even for complex tasks

- Cross-system analytics – Auto connectors (**BrandonH.**)
- Cross-system analytics – Algebra (**Dylan**)
- Linear algebra support (**Ryan**)

Beyond Big Data

Transaction Processing

- Making optimistic concurrency control faster (**Bailu**)
- With Johannes (Microsoft) and Lucja (Cornell)