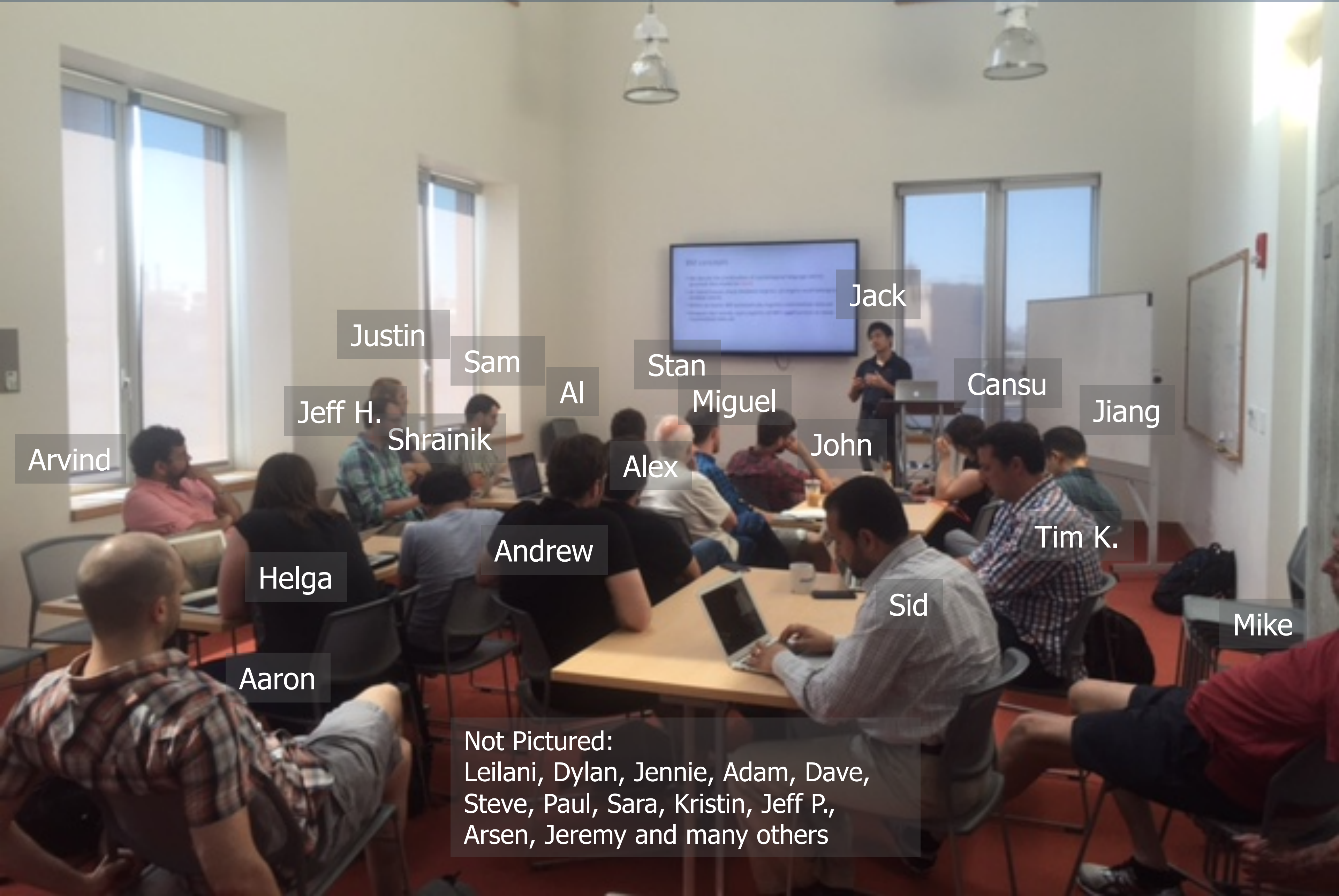
ISTC BIG DATA

Demonstrating the BigDAWG Polystore System for Ocean Metagenomic Analysis

Tim Mattson[1], Vijay Gadepally[2], Zuohao She[3], Adam Dziedzic[4], Jeff Parkhurst[1]

intel 1 MIT 2 BROWN Carnegie Mellon University W UT Portland State UNIVERSITY N 3 4 ISTC BIG DATA

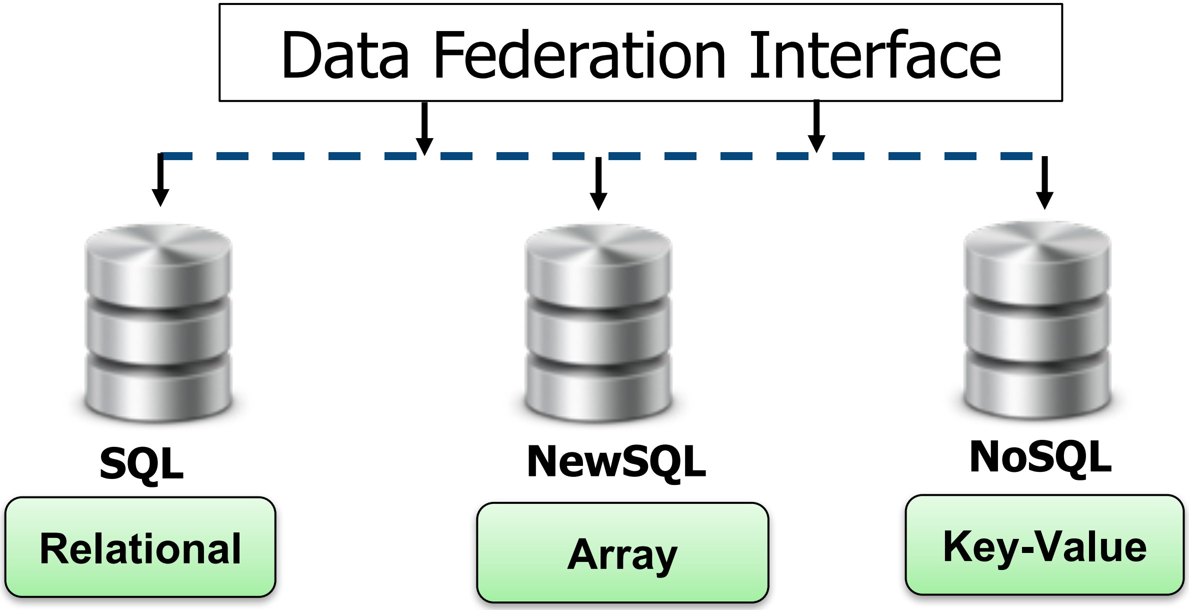
Acknowledgements



Not Pictured:
Leilani, Dylan, Jennie, Adam, Dave,
Steve, Paul, Sara, Kristin, Jeff P.,
Arsen, Jeremy and many others

How do we deal with multiple data bases?

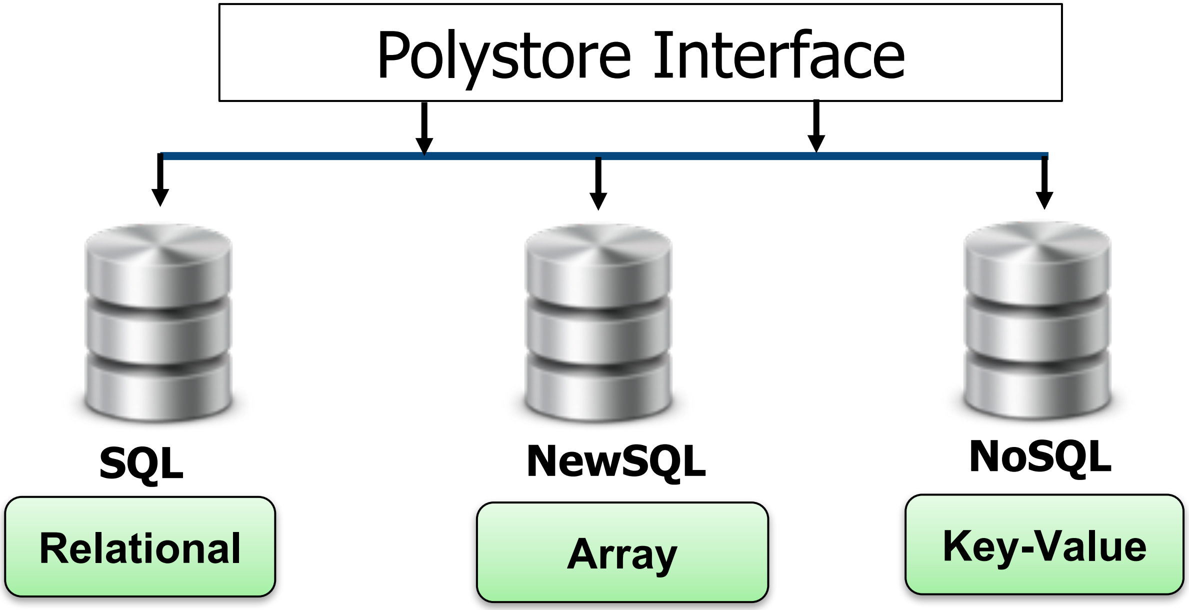
- **Data Federation:** Data stored in a **heterogeneous** set of **autonomous** data stores exposed as one integrated system with on-demand data integration.



- Data Federation … in practice
  - The single interface imposes a single data model
  - The DBMS are autonomous … not integrated.
  - Forces a “One Size Fits All” perspective.

How do we deal with multiple data bases?

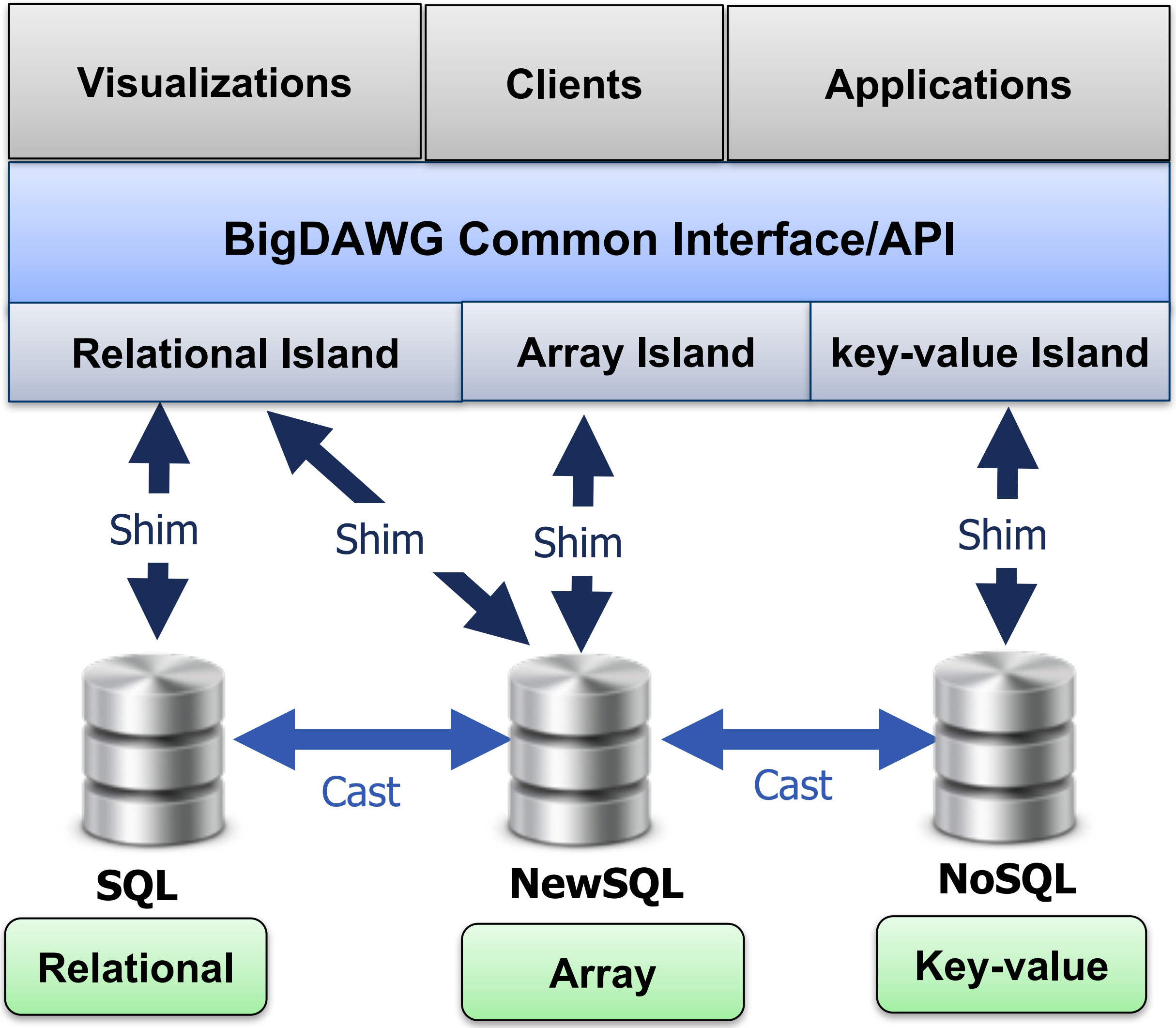
- **Polystore**: data stored in a heterogeneous set of **integrated** data stores is exposed through a common interface but the features of the individual data-stores are visible.



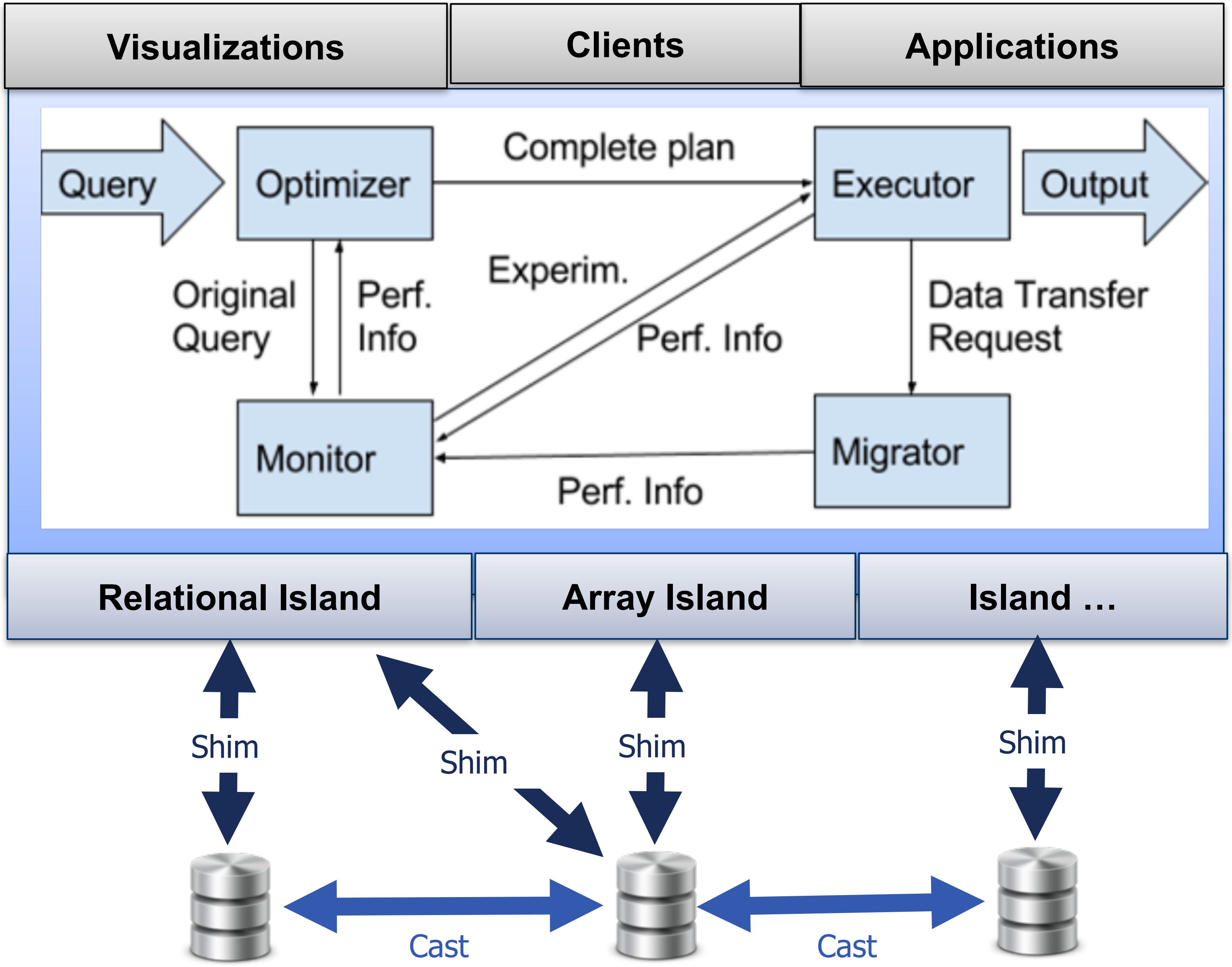
- Polystore Design challenge: Balancing competing forces …
  - **Location independence**: A query does not care which data-store in the polystore system it will target. A huge convenience for programmers.
  - **Semantic Completeness**: Any query natively supported by a data-store in the Polystore system can be expressed.

The BigDAWG Polystore System

- BigDAWG
  - Polystore: match data to the storage engine
- BigDAWG Islands
  - A data model + query operations
  - One or more storage engines
  - “Shim” connects a BigDAWG island to a data engine
  - “Cast” migrates data from one storage engine to another



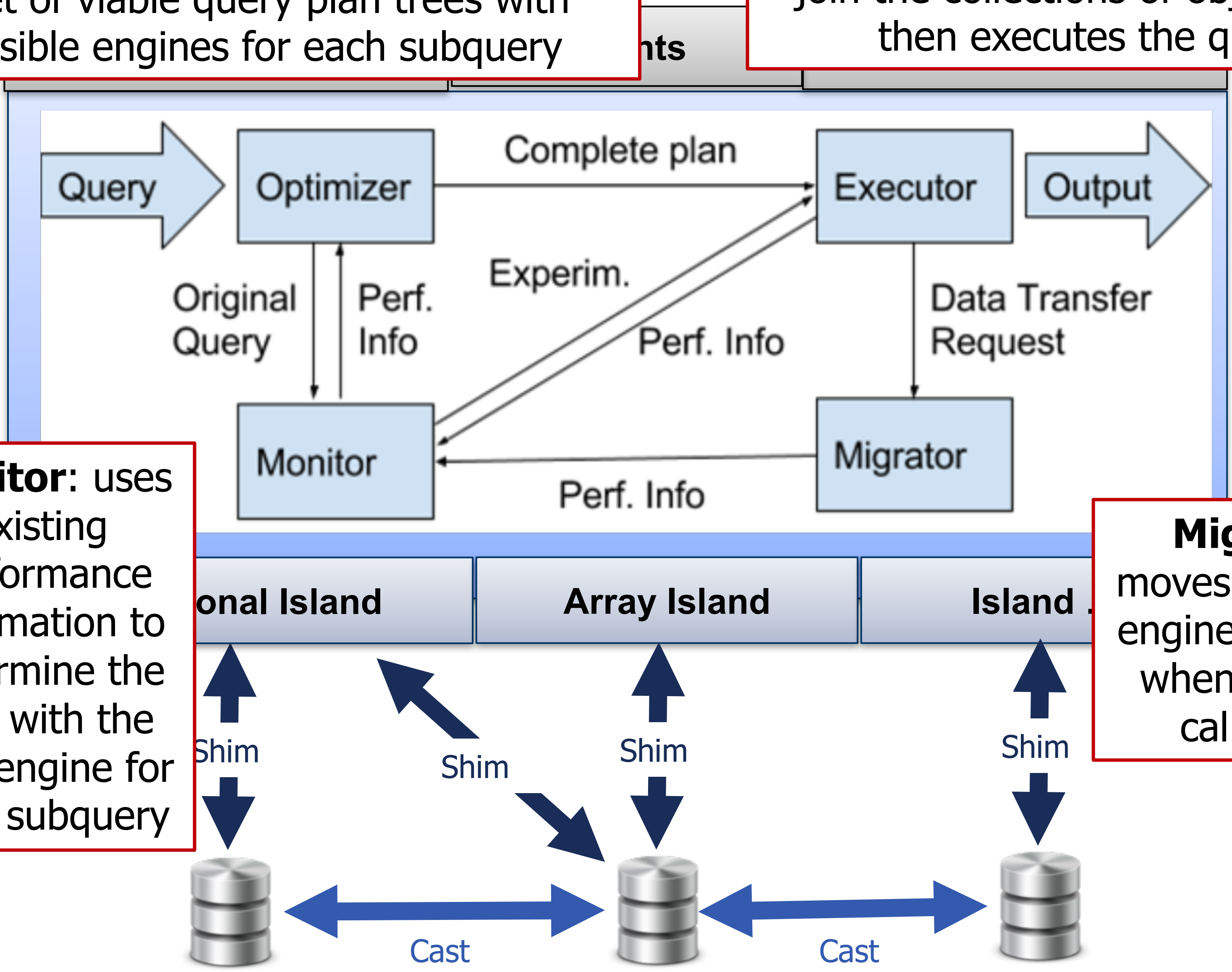
BigDAWG Middleware



BigDAWG Middleware

Optimizer: Parses the query and creates a set of viable query plan trees with possible engines for each subquery

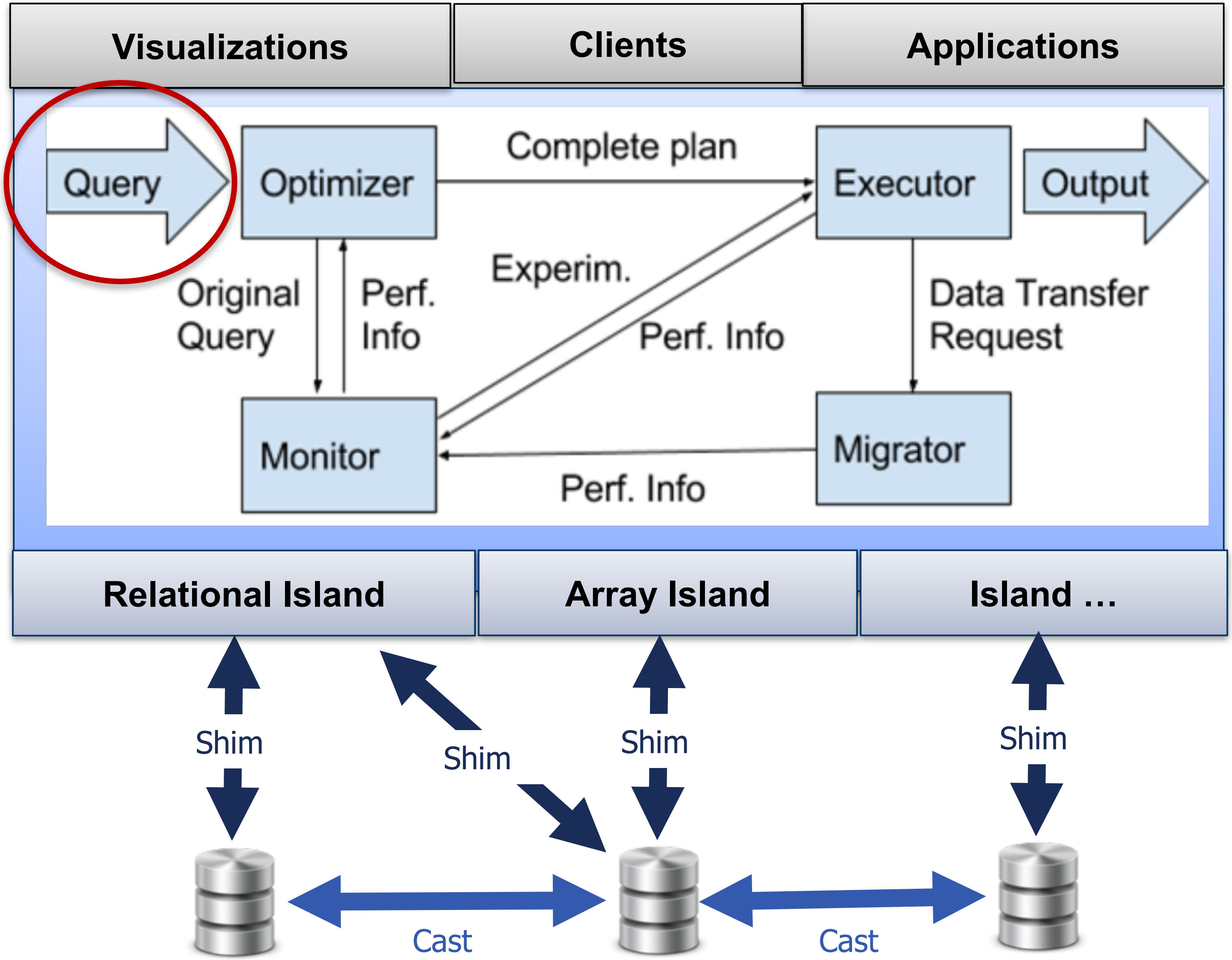
Executor: figures out how to best join the collections of objects and then executes the query



Monitor: uses existing performance information to determine the tree with the best engine for each subquery

Migrator: moves data from engine to engine when the plan calls for it

BigDAWG Middleware



A Big DAWG Query

```
bdarray(
  filter(
    bdcast(
       bdrel( select bodc_sta, time_stp, interp_sal
             from sampledata.main)
      , intrp_salinity
      , '<bodc_sta:int64, time_stp:datetime, interp_sal:double> [i=0:*,1000,0]‘
      , array)
   , interp_sal < 35))
```

A Big DAWG Query

Using the array island, issue the island’s filter operation

filter([source_array], [logical_expression])

```
bdarray(
  filter(
     bdcast(
        bdrel( select bodc_sta, time_stp, interp_sal
               from sampledata.main)
       , intrp_salinity
       , '<bodc_sta:int64, time_stp:datetime, interp_sal:double> [i=0:*,1000,0]‘
       , array)
   , interp_sal < 35))
```

Result is an array with rows for which interp_sal is less than 35

A Big DAWG Query

Create the array for the filter op by casting the table formed by this subquery from the relational island to the array island

```
bdarray(
   filter(
      bdcast(
         bdrel( select bodc_sta, time_stp, interp_sal
                  from sampledata.main)
         , intrp_salinity
         , '<bodc_sta:int64, time_stp:datetime, interp_sal:double> [i=0:*,1000,0]‘
         , array)
      , interp_sal < 35))
```

Bdcast ([source_query], name, [Dest_schema_parameters], [target])

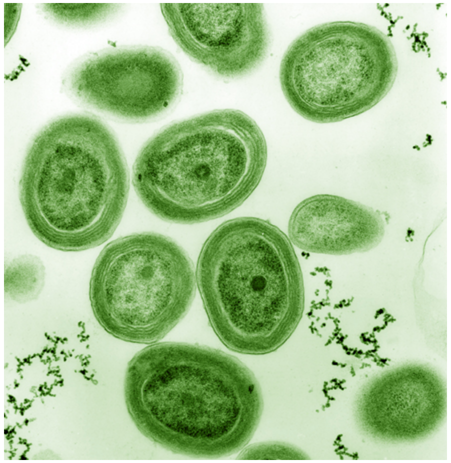
A Big DAWG Query

The array created is named “intrp_salinity”. It has three attributes (bodc_sta, time_stp, and interp_sal) with unbounded number of rows (i=0:*) broken down into chunks of size 1000 with 0 overlap

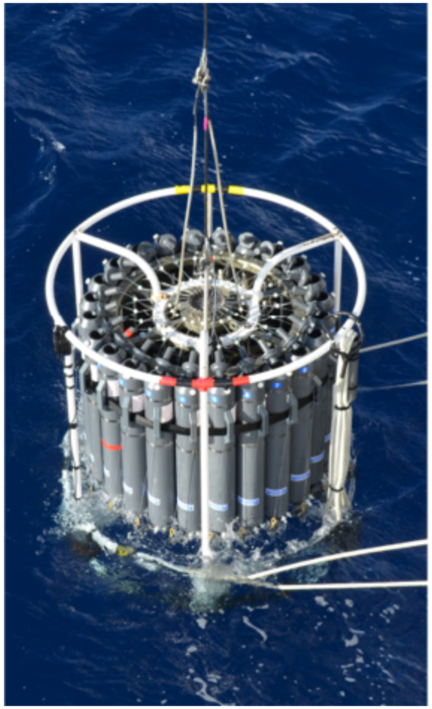
```
bdarray(
   filter(
      bdcast(
          bdrel( select bodc_sta, time_stp, interp_sal
                   from sampledata.main)
          , intrp_salinity
          , '<bodc_sta:int64, time_stp:datetime, interp_sal:double> [i=0:*,1000,0]‘
          , array)
      , interp_sal < 35))
```

The most populous species on Earth

- Prochlorococcus: A tiny marine cyanobacteria … yearly abundance is around $3*10^{27}$ critters.
  - Discovered in 1986 by Chisholm (MIT), Olson (Woods Hole) and collaborators.
- We need these guys … they are the primary producer in the ocean and provide 15-20 % of our O_2.



- We are working with the Chisholm Lab (MIT).
- Collect water samples around the world
- Sequence sea water to Measure populations (metagenomics) and correlate with features of the system.
- Challenges that are faced by researchers:
  - The volume and variety of data make it difficult to integrate, explore and/or summarize
  - Extracting sequences related to organisms is a computational and data management problem
  - Correlating metadata with sequence data is messy



Oceanographic Data Components -current status-

- Genome Sequence Data
  - For every individual sample, we quality controlled, trimmed and (sometimes) paired sequence data. Each sample contains many different DNA sequence reads from a particular sample corresponding to different DNA samples.
- Discrete sample metadata
  - Recording of nearly 500 different entities for water samples (ocean chemistry)
- Sensor Metadata
  - Information about recordings, where they took place
- Cruise Reports
  - Free form text reports written as cruise logs
- Streaming Data
  - Data collected from SeaFlow* system.

*http://armbrustlab.ocean.washington.edu/resources/seaflow/

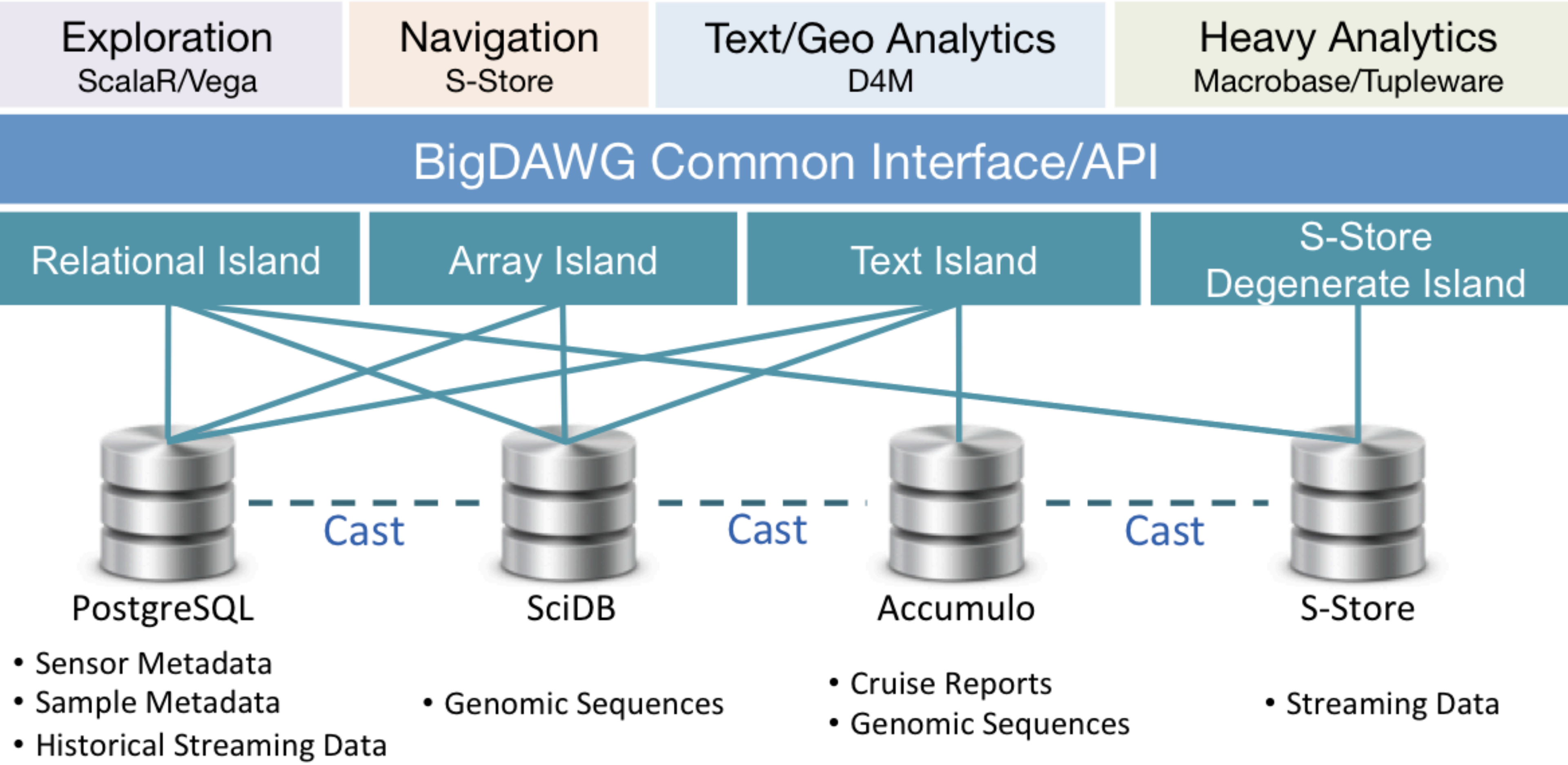
Oceanographic Data Components
-current status-

- Genome Sequence Data
  - For every individual sample, we quality controlled, trimmed and (sometimes) paired sequence data. Each sample contains many different DNA sequence reads from a particular sample corresponding to different DNA samples.
- Discrete sample metadata
  - Recording of nearly 500 different entities for water samples (ocean chemistry)
- Sensor Metadata
  - Information about recordings, where they took place
- Cruise Reports
  - Free form text reports written as cruise logs
- Streaming Data

Overall: Diverse, Fast, and Big
-Great fit for BigDAWG -

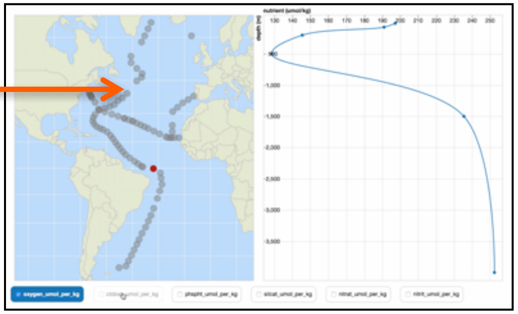
*http://armbrustlab.ocean.washington.edu/resources/seaflow/

BigDAWG and our Ocean Metagenomic Demo

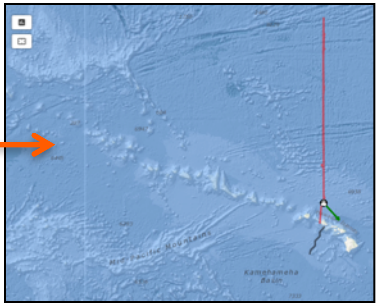


Application Overview

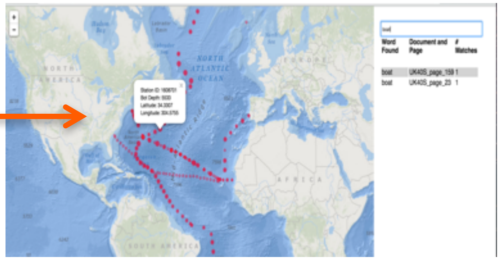
Exploration
(see the entire dataset)



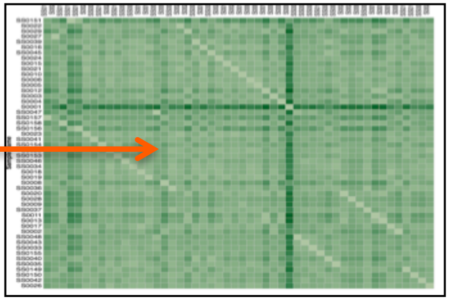
Navigation
(make cruises more efficient)



Geo-Analytics
(leverage the unstructured data)

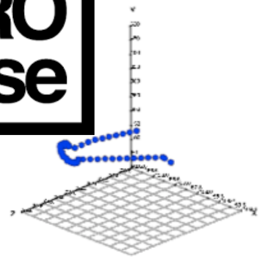


Genomic Processing
(look for interesting trends in genomic data)

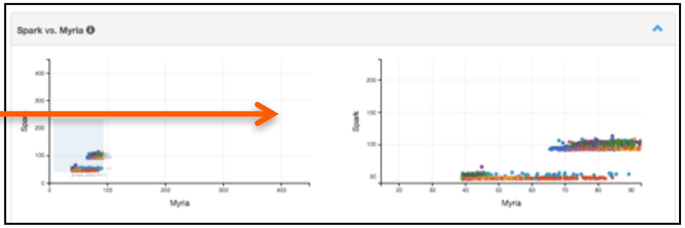


Heavy Analytics
(cut across data set for deep analytics)

MACRO base



Performance Modeling
(see how well the system performs)



Conclusion

- Polystore systems are an important tool for dealing with heterogeneous data.
  - A single high level data management system that is composed of many individual storage management systems.
    - Storage management matches the data for a better performance.
    - Analytics embedded into the storage managers to keep computing near the data.
- BigDAWG is an effective Prototype to prove the concept.
  - There is a great deal of work needed to turn it into a general purpose tool for data scientists.
  - Early results, however, are encouraging
- Prochlorococcus is really cool.   Take a deep breath and think about how much we enjoy the work of this little critter.

BigDAWG Open Source Release in Q1’2017

References (All in the HPEC’2016 Proceedings)

- **The BigDAWG Polystore System and Architecture** *Vijay Gadepally, Peinan Chen (MIT), Jennie Duggan (Northwestern University), Aaron Elmore (University of Chicago), Brandon Haynes (University of Washington), Jeremy Kepner, Samuel Madden (MIT), Tim Mattson (Intel), Michael Stonebraker (MIT)*

- BigDAWG Polystore Query Optimization Through Semantic Equivalences *Zuohao She, Surabhi Ravishankar, Jennie Duggan (Northwestern University)*

- **The BigDawg Monitoring Framework *Peinan Chen, Vijay Gadepally, Michael Stonebraker (MIT)***

- **Cross-Engine Query Execution in Federated Database Systems *Ankush M. Gupta, Vijay Gadepally, Michael Stonebraker (MIT)***

- **Data Transformation and Migration in Polystores** *Adam Dziedzic, Aaron J. Elmore (University of Chicago), Michael Stonebraker (MIT)*

- **Integrating Real-Time and Batch Processing in a Polystore** *John Meehan, Stan Zdonik Shaobo Tian, Yulong Tian (Brown University), Nesime Tatbul (Intel), Adam Dziedzic, Aaron Elmore (University of Chicago)*