

# **Perfopticon: Visual Query Analysis for Distributed Databases**

Dominik Moritz, Daniel Halperin, Bill Howe, and Jeffrey Heer  
Computer Science & Engineering, University of Washington

CPSC 547  
Thursday, November 12  
By: Dmitry Tebaykin

# Overview

1. Introduction into SQL and databases
2. Why is this paper important?
3. The 4 views of Perfopticon (with analysis and pictures)
4. Could you use Perfopticon?
5. Conclusions

# 1. Introduction into SQL and databases

In our case:

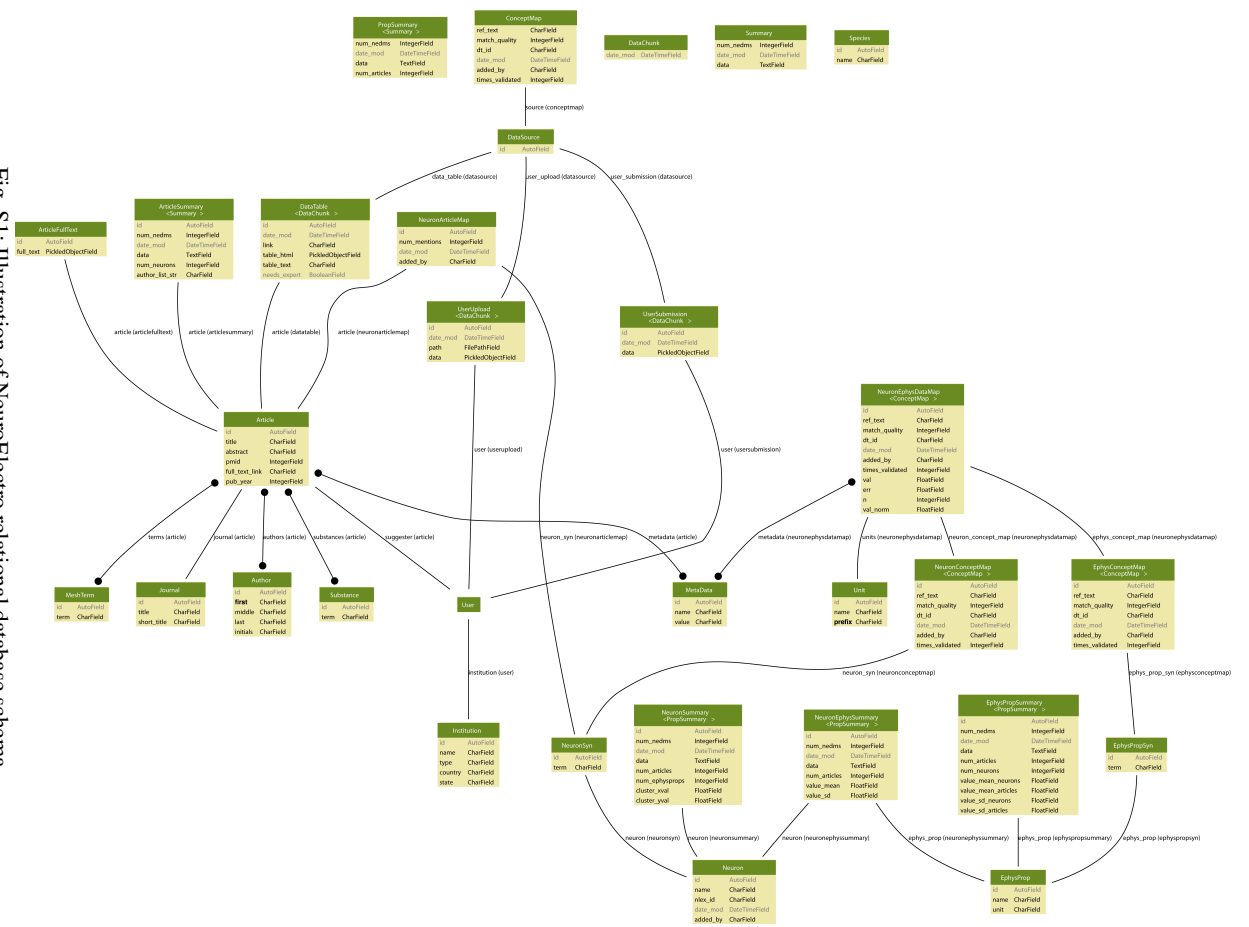
Database - tables of data joined

SQL - language for talking to databases

Examples of questions:

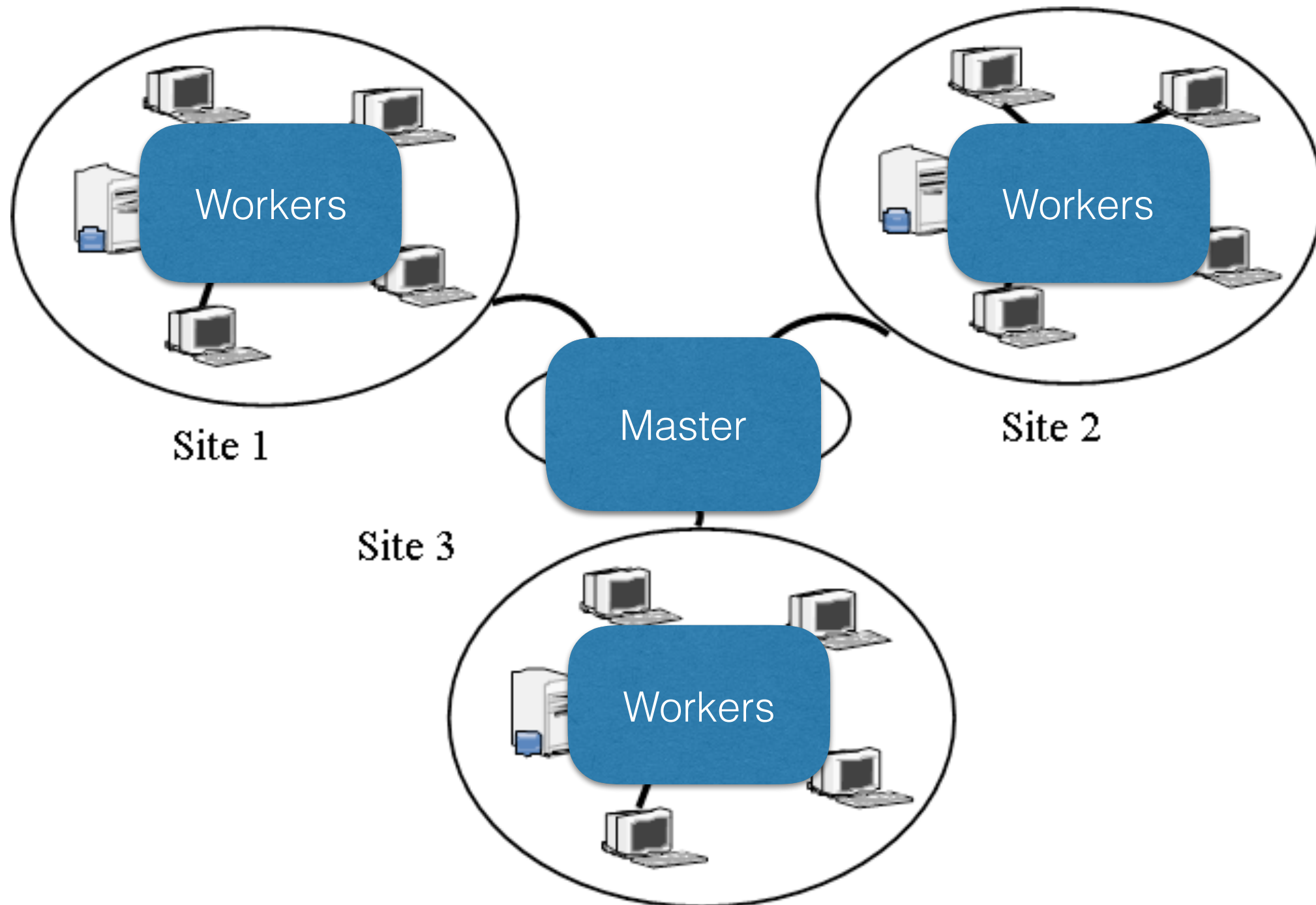
- “What is the age of every student in UBC?”
- “How many people are taking CS547 this term?”

Fig. S1: Illustration of NeuroElectro relational database schema



# 1. Introduction into SQL and databases

Distributed database system:

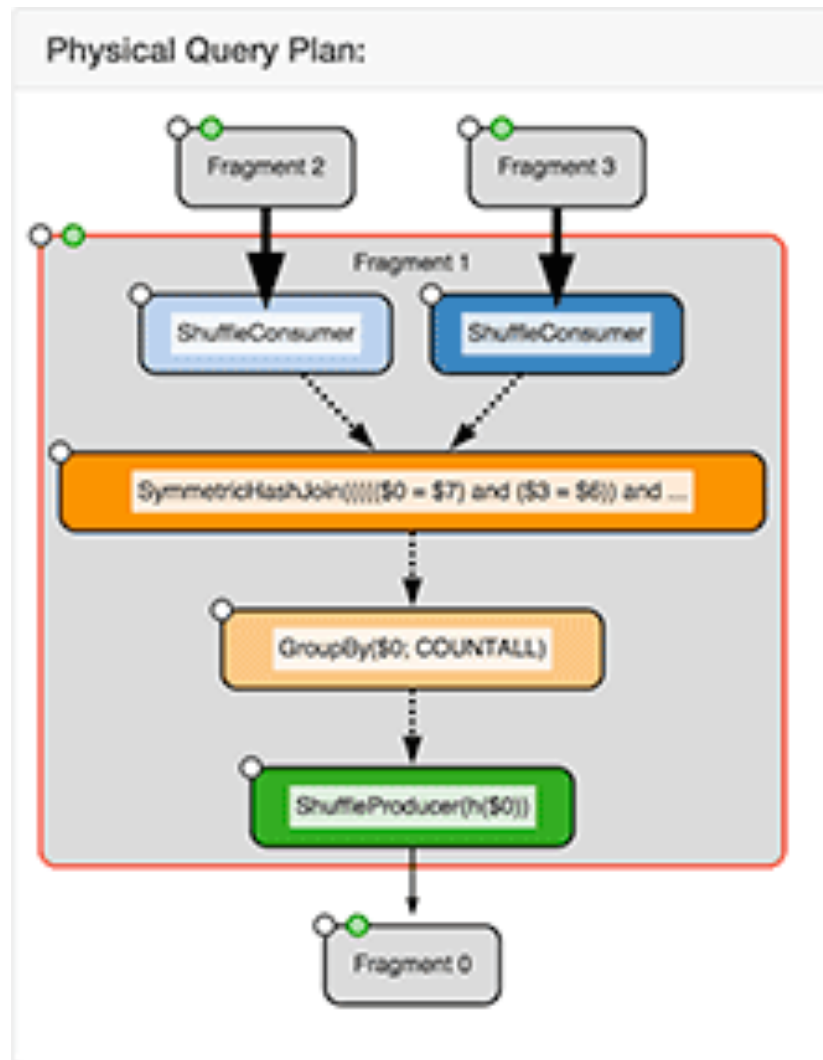


<https://cnx.org/resources/0d203a416b87d2bed544825664c14614602f9385/graphics8.png>





### 3. The 4 views of Perfoption (with analysis and pictures)



View 1	Query plan view
<b>What: data</b>	Directed graph that represents: query plan for data access generated by DBMS
<b>Why: tasks</b>	Locate, identify, compare
<b>How: encode</b>	Shape marks for nodes (execution steps), connection marks for links
<b>How: facet</b>	Coordinate: linked highlighting and navigation with other views

### 3. The 4 views of Perfopticon (with analysis and pictures)

Fragment 0 expand/collapse



Fragment 1 expand/collapse



Fragment 2 expand/collapse



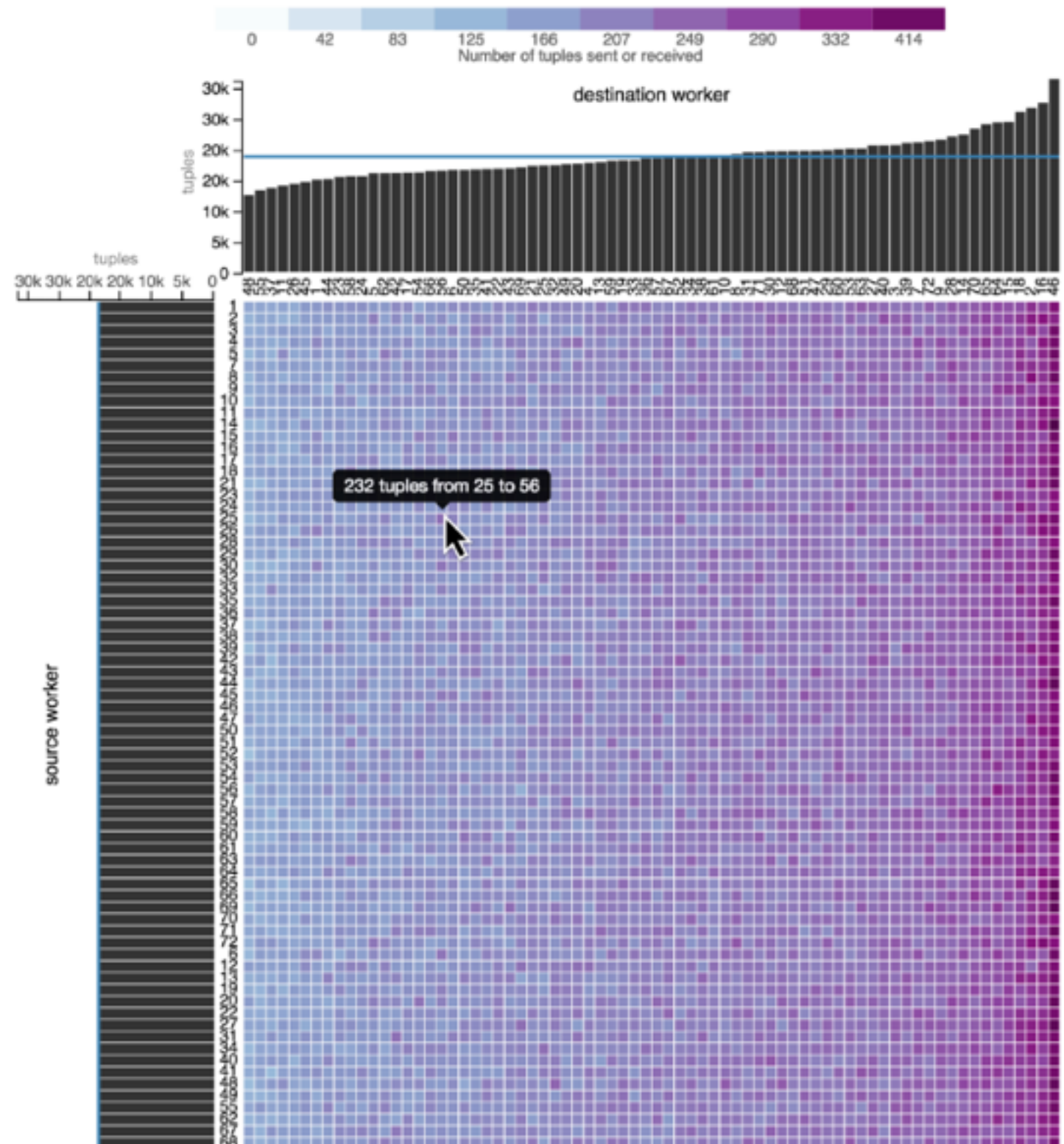
Fragment 3 expand/collapse



<b>View 2</b>	<b>Work distribution view</b>
<b>What: data</b>	Tables from query log files
<b>Why: tasks</b>	Compare, identify outliers
<b>How: encode</b>	Histograms showing execution time of workers
<b>How: facet</b>	Partition: multiple views for each query fragment. Coordinate: linked highlighting and navigation with other views
<b>How: reduce</b>	Navigate

### 3. The 4 views of Perfoption (with analysis and pictures)

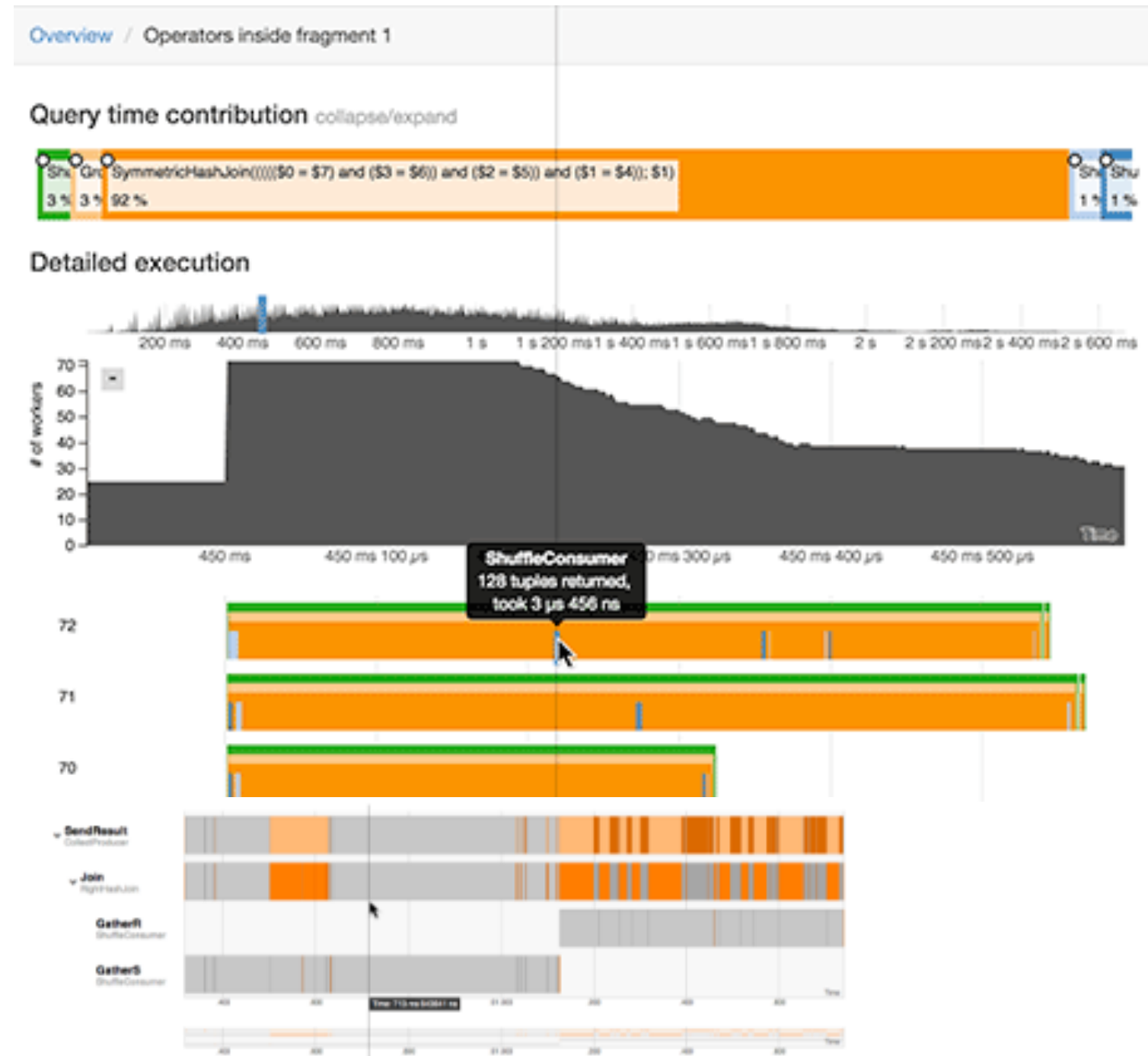
<b>View 3</b>	<b>Communication view</b>
<b>What: data</b>	Table: two continuous variables (amount of data sent and received by workers)
<b>Why: tasks</b>	Compare, identify outliers, summarize
<b>How: encode</b>	2D matrix alignment of area marks, diverging colormap
<b>How: facet</b>	Coordinate: linked navigation with other views





# 3. The 4 views of Perfoption (with analysis and pictures)

<b>View 4</b>	<b>Local execution view</b>
<b>What: data</b>	Tables from query log files
<b>Why: tasks</b>	Compare, identify outliers
<b>How: encode</b>	Histograms, bar charts (colour indicates active/inactive/wait states)
<b>How: facet</b>	Partition: multiform views. Coordinate: linked highlighting
<b>How: reduce</b>	Navigation



## 4. Could you use Perfopticon?

- Built into Myria (Giant online database), requires log files for the query executions with slight modifications.
- Their example: Myria, added 3 lines to log file per query execution step.
- The tool has a front-end component, upload your query log files and view the results.

## 5. Conclusions

- Perfopticon can be used effectively for query and database optimization (Emma, the oceanographer, managed to speed up her query and Chu S. et. al created a better table joining algorithm).
- Provides the ability to spot underperforming or overtasked nodes and drill down into the problem.
- Might work for non-relational databases as well.
- Needs more validation.