BIFOLD

Data Visualization – Tutorial

Stefan Halfpap

www.bifold.berlin

https://github.com/klauck/data_visualization_tutorial

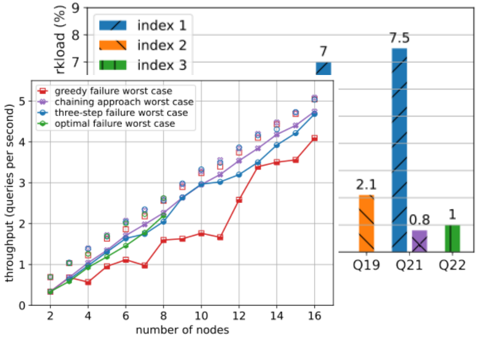
About me

PostDoc in the Database Systems and Information Management (DIMA) group

- Distributed stream processing
- Data management systems

PhD student in the Enterprise Platform and Integration Concepts (EPIC) group

- Physical database optimization
- Main memory database systems



Partial Replication Benchmark: TPC-H #Replicas: 6 Allocation Strategy: Decomposition

Benchmark Queries and their Workload Shares:

Allocation Result

Replication Factor: 2.51 (5.28 GB) Compared to Optimal: +2%

Replica 1 Chunk 1

Replica 2 Chunk 1

Replica 3 Chunk 2

Index Analysis

...eries: (16.7% of Workload Share)

...gments: 1,041 MB (49% of Data)

...eries: (16.7% of Workload Share)

...gments: 615 MB (29% of Data)

```
Exploration of Dynamic Query-Based Load Balancing for Partially Replicated Database Systems

QUERY STREAMS
Benchmark: TPC-H     #Streams: 25     #Nodes: 5     Failed Node: 1     Fragment Allocation Strategy: Rabl and Jacobsen

Benchmark Queries and their Workload Shares:

Data Replication Factor: 2.75     Progress: 262.50s     Queries per Second: 2.28     Average Utilization (Executed/Planned Load): 60%

REPLICA NODE 1 (FAILED)
Executable with 86% Data:                          Planned Load:
Queued:                        In Execution:       Executed Load:                    0%

REPLICA NODE 2
Executable with 27% Data:                          Planned Load:
Queued:                        In Execution:       Executed Load:                    70%

REPLICA NODE 3
Executable with 34% Data:                          Planned Load:
Queued:                        In Execution:       Executed Load:                    67%

REPLICA NODE 4
Executable with 86% Data:                          Planned Load:
Queued:                        In Execution:       Executed Load:                    104%

REPLICA NODE 5
Executable with 43% Data:                          Planned Load:
Queued:                        In Execution:       Executed Load:                    59%
```

Outline

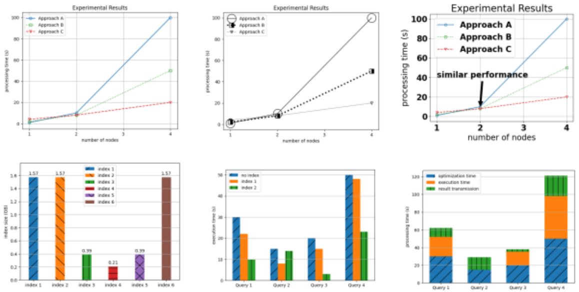
1. Motivation
2. Types and Tools
3. Hands on Examples
4. Best Practices
5. Summary and Discussion

Outline

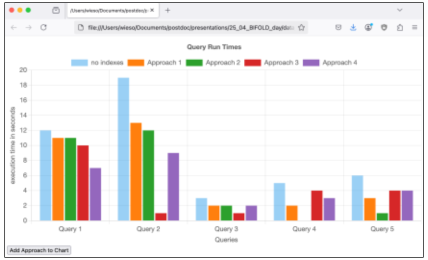
1. Motivation und Use Cases
2. **Types and Tools**
3. **Hands on Examples**
4. **Best Practices**
5. Summary and Discussion

What you will learn

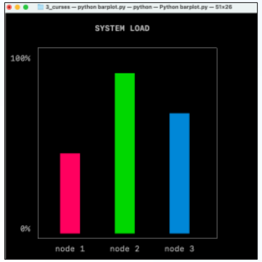
Matplotlib for static graphs



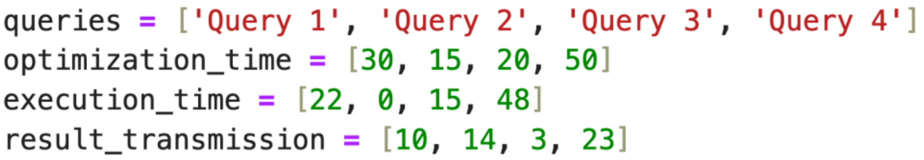
(Intro) Chart.js for interactive web applications



(Intro) Python curses for terminal applications

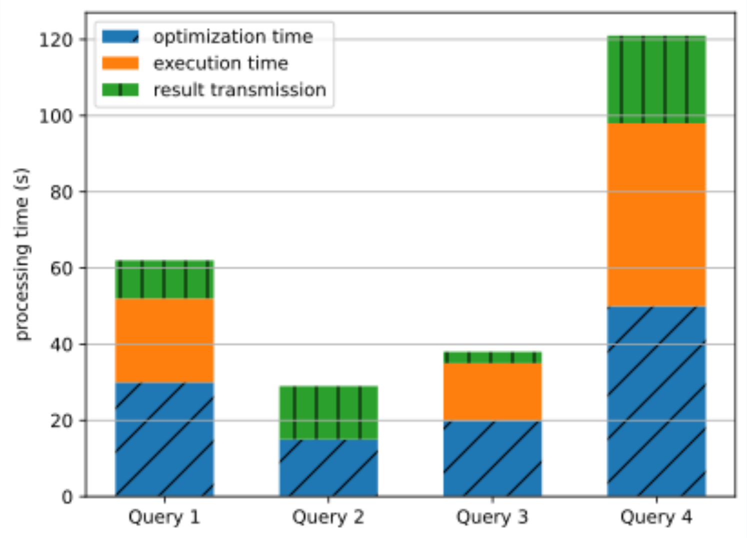

3_curses — python barplot.py — python — Python barplot.py — 51×26
SYSTEM LOAD
100%
0%
node 1
node 2
node 3

Motivation



```
queries = ['Query 1', 'Query 2', 'Query 3', 'Query 4']
optimization_time = [30, 15, 20, 50]
execution_time = [22, 0, 15, 48]
result_transmission = [10, 14, 3, 23]
```

vs.



Motivation and Use Cases

- **Humans can** more **easily perceive visual differences** than sequences of text or numbers

Data visualizations

- Exploring, understanding, and communicating data

Essential research skill

- Early prototypes -> systematic experiments -> final publications
- Papers/theses, presentations, or applications/demonstrations

Types and Tools

What to use?

It depends!

Types and Tools

What to use?

It depends on..

the **purpose**, **data**, and **own experience**.

Types and Tools

Factors that influence data visualization types and tools

- **Purpose**
  - Self usage vs. communicate to others vs. products
  - Papers/theses, presentations, or applications/demonstrations
  - Customization, automation, reproducibility
  - Static vs. dynamic
- **Data**
  - Size, dimensions, and kind (2D vs. 3D; metric, categorical, maps, …)
- **Your own experience**
  - Beginner
  - Intermediate
  - Advanced

Types and Tools

Factors that influence data visualization types and tools

- **Purpose**
  - **Self usage vs. communicate to others** vs. products
  - **Papers/theses, presentations, or applications/demonstrations**
  - **Customization, automation, reproducibility**
  - **Static vs. dynamic**
- **Data**
  - Size, dimensions, and kind (**2D** vs. 3D; **metric, categorical**, maps, …)
- Your own (**programming**) experience and **available software**
  - Beginner (OpenOffice Calc or MS Excel, Google Sheets, web applications)
  - Intermediate (Tableau Software or other BI tools)
  - **Advanced (visualization libraries, e.g., Matplotlib, chart.js)**

Types and Tools

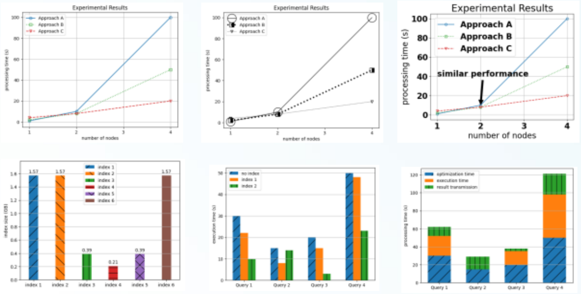
Suitable visualization libraries depend on

the programming language and your specific needs

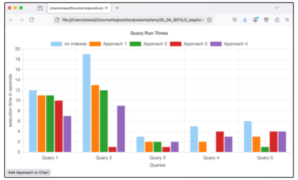
Hands on

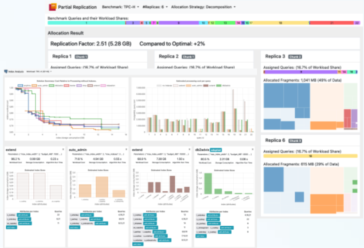
https://github.com/klauck/data_visualization_tutorial

- Matplotlib for static graphs

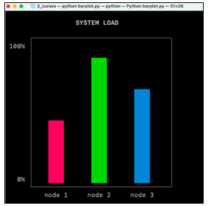


- Chart.js for interactive web applications





- Python curses for terminal applications



Exploration of Dynamic Query-Based Load Balancing for Partially Replicated Database Systems
QUERY STREAMS
Benchmark: TPC-H
#Streams: 25
#Nodes: 5
Failed Node: 1
Fragment Allocation Strategy: Rabl and Jacobsen
Benchmark Queries and their Workload Shares:
Data Replication Factor: 2.75
Progress: 262.50s
Queries per Second: 2.28
Average Utilization (Executed/Planned Load): 60%
REPLICA NODE 1 (FAILED)
REPLICA NODE 2
REPLICA NODE 3
REPLICA NODE 4
REPLICA NODE 5
Executable with 86% Data:
Planned Load:
Executed Load:
Queued:
In Execution:

Best Practices

- Use what fits best to you(r needs)
- Organize the data to visualize (structure, version, queryable)
- Configurable, automated, and reproducible visualizations
- Visualization styles
    - Keep it simple
    - Keep it clear/self-contained
        - Use titles, labels (e.g. axes with units), annotations, legends
        - Avoid distortion and misrepresentation (do not mislead the audience)
    - Use appropriate sizes, markers, colors, patterns
    - Use consistent styles, colors, and names
    - Chart type, data selection, scaling (linear vs. logarithmic)

See also: Friends Don't Let Friends Make Bad Graphs
https://github.com/cxli233/FriendsDontLetFriends

Data Visualization – Tutorial Summary and Discussion

Suitable Types and Tools

depend on **purpose**, **data**, and **own experience**

Best Practices

- Organize the data to visualize
- Configurable, automated, and reproducible
- Appropriate visualization styles

Hands on Examples

https://github.com/klauck/data_visualization_tutorial

Matplotlib for static graphs

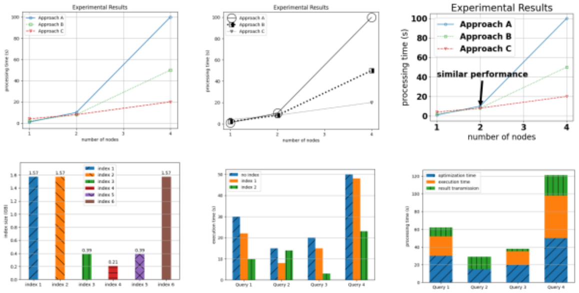
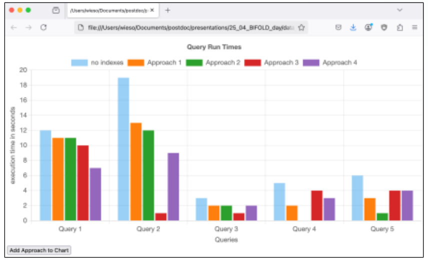
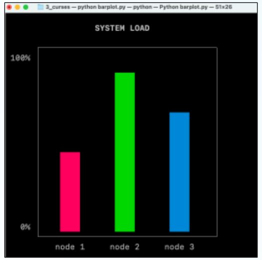


Chart.js for interactive web applications



Python curses for terminal applications



SYSTEM LOAD
100%
0%
node 1
node 2
node 3