

Seminar Large-scale Data Engineering (LDE)

03 Experiments, Reproducibility & Presentations

Dr.-Ing. Patrick Damme

Technische Universität Berlin

Berlin Institute for the Foundations of Learning and Data

Big Data Engineering (DAMS Lab)



Last update: May 5, 2024

[Credit: Based on “Introduction to Scientific Writing”/
”03 Experiments and Reproducibility” by Matthias Boehm
(TU Graz, winter 2021/22)]



Announcements/Org



- **Hybrid Setting with Optional Attendance**

- In-person in TEL 811 MAR 0.003
- Virtual via zoom

<https://tu-berlin.zoom.us/j/67376691490?pwd=NmlvWTM5VUVWRjU0UGI2bXhBVkxzQT09>



- **Assigned Seminar/Project Topics**

- Every participant should have received a topic notification by now
- Start of self-organized seminar/project work

- **Exam Registration in Moses: Deadline Jun 3, 23:59**

- Mind the regulations on exam deregistration



Agenda



- **Experiments and Result Presentation**
- **Reproducibility and RDM**
- **Scientific Presentations**

Experiments and Result Presentation

In Computer Science (Data Management)



[**Credit:** Ioana Manolescu, Stefan Manegold:
Performance Evaluation in Database Research:
Principles and Experiences, **ICDE 2008**]

Motivation



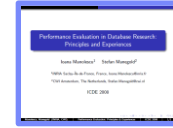
- **Worst Mistake: Schrödinger's Results**
 - Postpone implementation and experiments till last before the deadline
 - No feedback, no reaction time (experiments require many iterations)
 - **Karl Popper:** falsifiability of scientific results
- **Continuous Experiments**
 - Run experiments during survey / prototype building
 - Systematic experiments → observations and ideas for improvements
 - Don't be afraid of throwing away prototypes that don't work
- **Good Research Fires Itself**
 - Initial experiments give directions for further improvements
 - Problem-oriented methodology

Types of Experiments

- **#1 Exploratory Experiments**
 - Tests for functional correctness
 - Unstructured experiments for initial feedback → evaluate feasibility
- **#2 Micro Benchmarks**
 - Measure specific aspects in controlled and understandable scope
 - Bottom-up approach
- **#3 Benchmarks**
 - Evaluate on community/own benchmarks
 - Examples: TPC-C, TPC-H, TPC-DS, JOB, MLPerf, TPCx-AI
- **#4 End-to-end Applications**
 - Evaluate in larger scope of real datasets and query workloads
 - Examples: Customer workload, ML pipelines (data preparation, training, evaluation)

From Idea to Experiments

[I. Manolescu, S. Manegold: Performance Evaluation in Database Research: Principles and Experiences, **ICDE 2008**]



■ Overview

- Proper planning helps to keep you from “getting lost”
- Repeatable experiments simplify your own work
- There is **no single way** how to **do it right**
- There are **many ways** how to **do it wrong**

■ Basic Planning

- Which **data** / datasets should be used?
- Which **workload** / queries should be run?
- Which **baselines** are relevant?
- Which **hardware & software** should be used?
- **Metrics**: What to measure? How to measure?
- **Comparison**: How to compare? How to find out what is going on?

Dataset Selection



■ Synthetic Data

- Generate data with specific data characteristics
- Systematic evaluation with data size, sparsity, distributions etc.

Representative
of real data
distributions?

■ “Real” Data Repositories

- Wide selection of available datasets w/ different characteristics
- UCI ML Repository: <https://archive.ics.uci.edu/>
- SuiteSparse Matrix Collection: <https://sparse.tamu.edu/>
- Google dataset search: <https://datasetsearch.research.google.com/>
- Common Datasets in ML: ImageNet, Mnist, CIFAR, KDD, Criteo
- Common Datasets in DM: Census, Taxi, Airlines, DBLP, benchmarks etc.

Representative
for variety of
workloads /
common case?

Workloads: Benchmarks



■ Overview

- Community- and organization-driven creation of agreed benchmarks
- **Benchmarks can define a field** and foster innovation

■ #1 Data Management

- Query processing: 007, TPC-C, TPC-E, TPC-H, TPC-DS (w/ audit), SSB
- Join ordering: JOB

■ #2 “Big Data”

- MR/Spark: BigBench, HiBench, SparkBench
- Array Databases: GenBase

■ #3 Machine Learning Systems

- SLAB, DAWNbench, MLPerf, MLBench, AutoML Bench, Meta Worlds, TPCx-AI

[Michael J. Carey, David J. DeWitt,
Jeffrey F. Naughton: The oo7
Benchmark. **SIGMOD 1993**]



[<http://www.tpc.org/tpch/>]

(See AMLS course for details)

■ #1 Primary Baseline

- Existing algorithm or system infrastructure
- Main comparison point, usually with same runtime operations
- **Beware:** Avoid speedup-only results
(need absolute numbers for grounding)

■ #2 Additional Baselines

- Alternative systems with different runtime and compiler
- Usually, not directly comparable but important for grounding
- E.g., SystemDS: R, Julia, Spark, TensorFlow, PyTorch, ...
- E.g., DAPHNE: the same, plus numpy, pandas, DuckDB, ...
- **Potential hindering factors:** commercial and closed-source systems, software licenses

■ Problem of **Weak Baselines**

- Authors want to show improvements
- Successive improvements over state-of-the-art don't add up



[Timothy G. Armstrong, Alistair Moffat, William Webber, Justin Zobel: Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. **CIKM 2009**]



[Maurizio Ferrari Dacrema, Paolo Cremonesi, Dietmar Jannach: Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. **RecSys 2019**]

Experimental Setting (Hardware & Software)



Hardware Selection

- Multiple nodes for distributed computation
- Avoid too outdated hardware (irrelevance)

Find Balanced Level of Detail

- Underspecified:
“We ran all experiments on an Intel CPU”
- Over-specified:
cat /proc/cpuinfo
cat /proc/meminfo

```
mboehm@alpha:~$ cat /proc/cpuinfo
processor       : 111
vendor_id     : GenuineIntel
cpu family    : 6
model         : 85
model name    : Intel(R) Xeon(R) Gold 6238R CPU @ 2.20GHz
stepping     : 7
microcode    : 0x5002f01
cpu MHz      : 2201.563
cache size   : 39424 KB
physical id  : 1
siblings     : 56
core id      : 30
cpu cores    : 28
apicid      : 125
initial apicid : 125
fpu         : yes
fpu exception: yes
cpuid level : 22
wp         : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp
lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc
c cpuid aperfperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 sse4_1 sse4_2
x2apic movbe popcnt tsc deadline tim
```

```
mboehm@alpha:~$ cat /proc/meminfo
MemTotal:       792256192 kB
MemFree:        717562860 kB
MemAvailable:   784507576 kB
Buffers:        480704 kB
Cached:         69666136 kB
SwapCached:     0 kB
Active:         20382624 kB
Inactive:       50053928 kB
Active(anon):   300800 kB
Inactive(anon): 256 kB
Active(file):  20081824 kB
Inactive(file): 50053672 kB
Unevictable:    19036 kB
Mlocked:        19036 kB
SwapTotal:     134217724 kB
SwapFree:      134217724 kB
Dirty:          116 kB
Writeback:      0 kB
AnonPages:     309260 kB
Mapped:         152584 kB
Shmem:         3124 kB
R reclaimable:  1788996 kB
Slab:          3003104 kB
SReclaimable:  1798996 kB
```

Recommendation

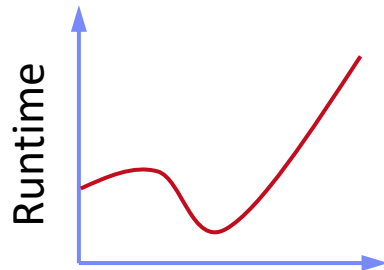
- HW components:** #nodes, CPUs (#cores, clock freq., cache size), memory, network, I/O
- SW components:** OS, programming language, versions, other software, compiler flags
- Baselines** and configuration → Use **recent versions of baseline systems**
- Data and workloads** with data sizes, parameters, configurations



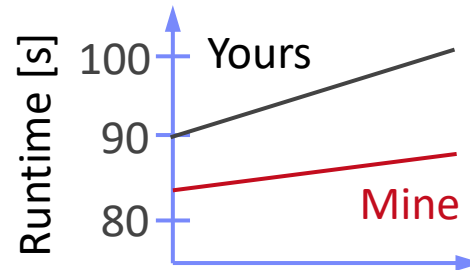
Presentation – Figures

■ Axes

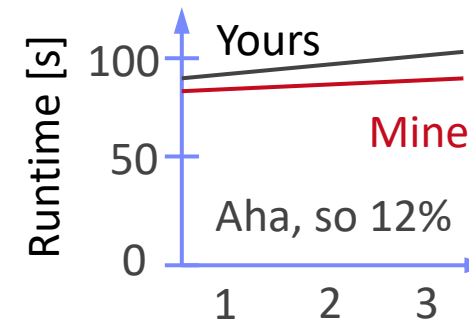
- Use informative axes labels with units (e.g., Total Execution Time [ms])
- Don't cheat or mislead readers and reviewers
- Start y-axis at 0 for linear scale



What are the units?
Where are the tics?



Misleading y axis

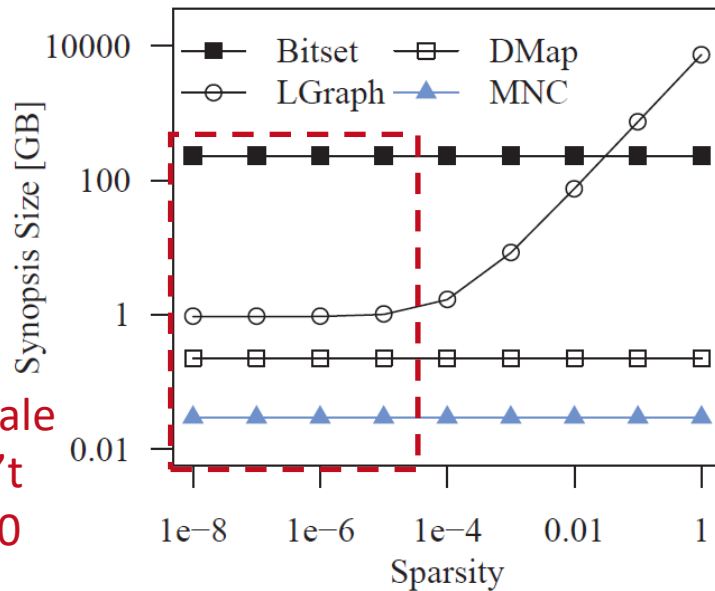


Presentation – Figures, cont.



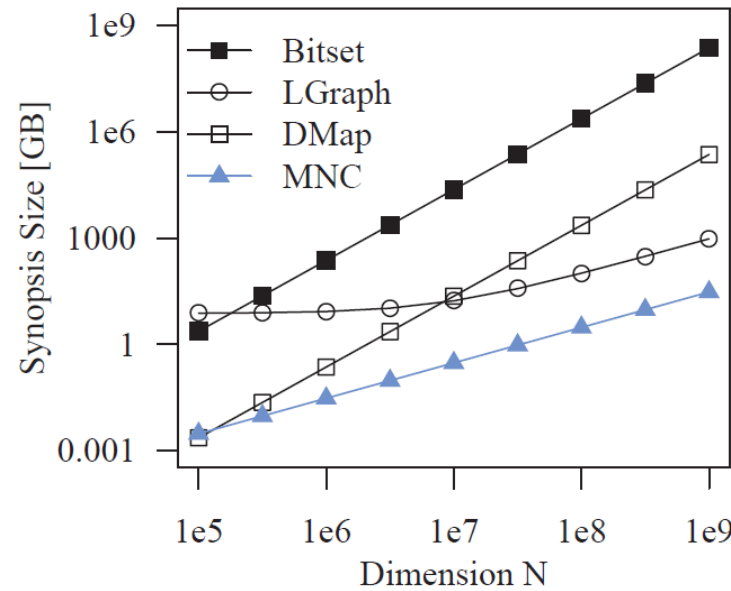
Fair Ranges of Parameters

- Evaluate common ranges of values
- Don't hide important information



For log-scale you can't start at 0

Don't limit range to make you look good



[J. Sommer, M. Boehm, A. V. Evfimievski, B. Reinwald, P. J. Haas: MNC: Structure-Exploiting Sparsity Estimation for Matrix Expressions. **SIGMOD 2019**]



If there are multiple relevant parameters, show them all



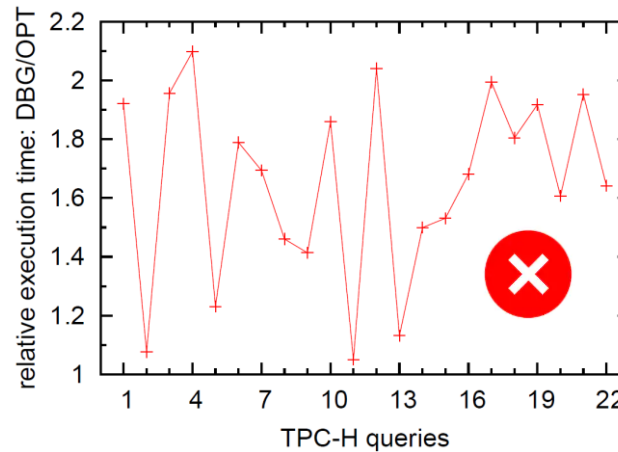
Presentation – Figures, cont.

[I. Manolescu, S. Manegold: Performance Evaluation in Database Research: Principles and Experiences, **ICDE 2008**]



Plots Types

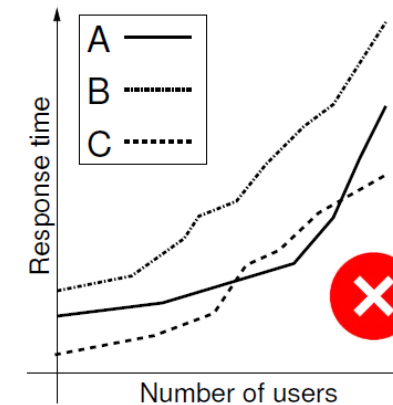
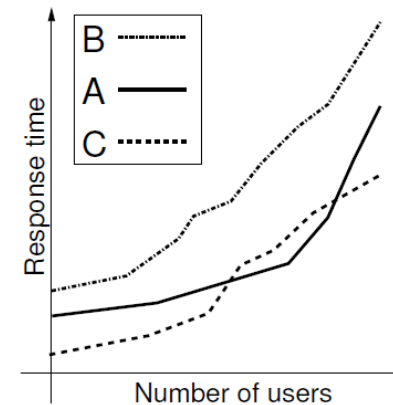
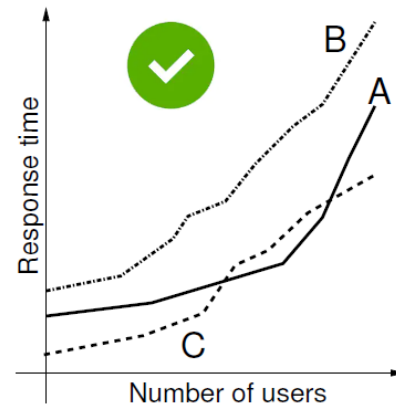
- **Barplot** for categories
- **Plot + Line/linepoints** for continuous parameters
- Visible font sizes (similar to text)



Categorical x-axis.
What do lines mean?

Legends

- Order them by appearance
- Attach directly to graph



Human brain is a
poor join processor
Humans get
frustrated

Presentation – Figures, cont.

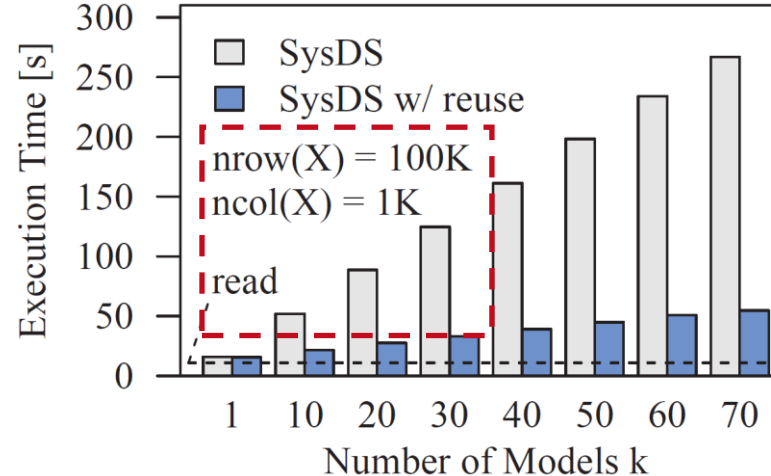
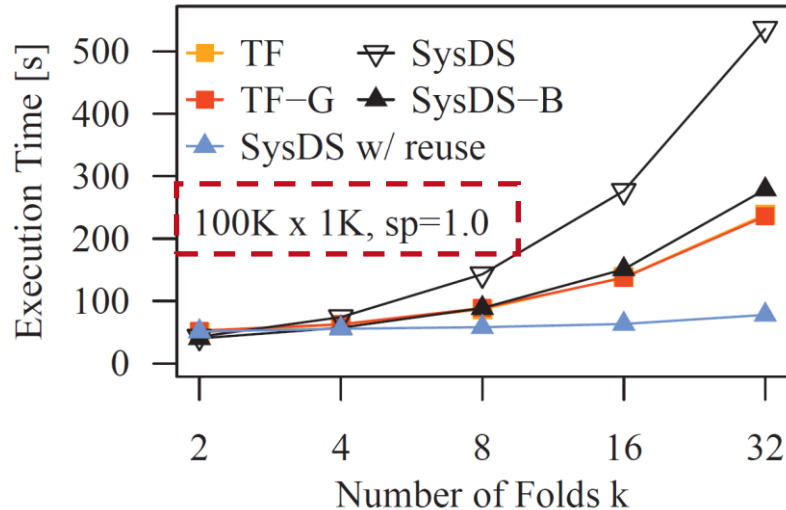


■ Diversity & Consistency

- Diversity: if applicable use mix of different plot types and tables
- Consistency: use consistent colors and names for same baselines

■ Labeling

- Make the plots self-contained
- Simplifies skimming and avoids join with text



[Matthias Boehm et al: SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle. **CIDR 2020**]



Presentation – Result Interpretation



Use the Right OS Tools

- System-specific tracing/statistics
- top / htop / iotop
(looks **CPU bound**)
- perf -stat -d ./run.sh
(no, it's **memory-bandwidth bound**)

Performance counter stats for './run.sh':

```
12721364.53 msec task-clock          # 83.640 CPUs utilized
      463352 context-switches       # 0.036 K/sec
5455536095415 instructions          # 0.14 insn per cycle           (62.50%)
335314473273 branches                # 26.358 M/sec
      (62.50%)
1463380955 branch-misses              # 0.44% of all branches        (62.50%)
2185062643097 L1-dcache-loads                # 171.763 M/sec                (62.50%)
142845949268 L1-dcache-load-misses         # 6.54% of all L1-dcache hits  (62.50%)
3375555316 LLC-loads                # 0.265 M/sec                  (50.00%)
1016330404 LLC-load-misses           # 30.11% of all LL-cache hits  (50.00%)

152.096000108 seconds time elapsed
12052.466691000 seconds user
674.704421000 seconds sys
```

Don't just report the results,
try to understand and
explain them

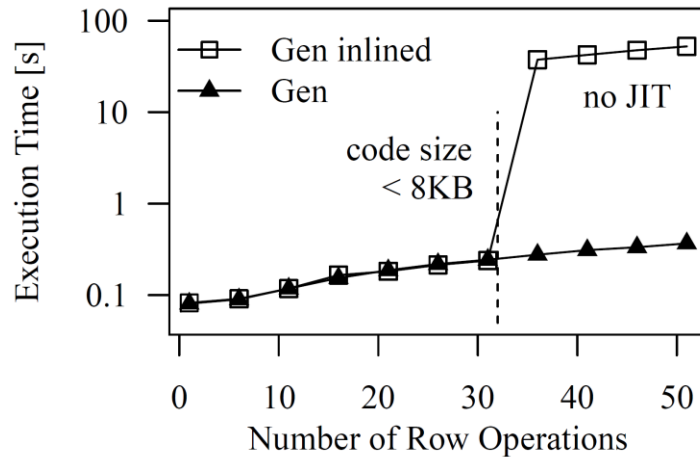


Presentation – Result Interpretation, cont.

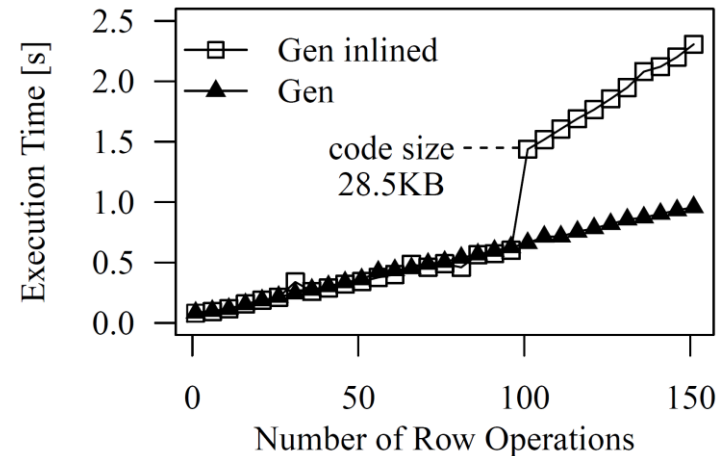


- Use the Right PL Tools / Flags

- E.g., Understanding Java JIT compilation
-XX:+PrintCompilation



- E.g., Understanding HW Cache Hierarchy (L1i 32KB)
-XX:-DontCompileHugeMethods



[Matthias Boehm et al:
On Optimizing Operator Fusion Plans
for Large-Scale Machine Learning in
SystemML. **PVLDB 11(12) 2018**]



Reproducibility and RDM (Research Data Management)

In Computer Science (Data Management)

Research Data Management (RDM)



■ Overview

- Ensure reproducibility of research results and conclusions
- **Common problem:** “All code and data was on the student’s laptop and the student left / the laptop crashed.”
- **Create value for others** (compare, reuse, understand, extend)
- EU Projects: Mandatory proposal section & deliverable on RDM plan

Excursus: FAIR Data Principles



[<https://www.go-fair.org/fair-principles/>]



▪ Findable

- Metadata and data have globally unique **persistent identifiers**
- Data described with rich **meta data**; registered/indexed and searchable

▪ Accessible

- Metadata and data retrievable via open, free and universal **comm protocols**
- Metadata accessible even when data no longer available

▪ Interoperable

- Metadata and data use a formal, **accessible, and broadly applicable format**
- Metadata and data use FAIR vocabularies and qualified references

▪ Reusable

- Metadata and data described with plurality of accurate and relevant attributes
- Clear license, **associated with provenance**, meets community standards



■ Code and Artifacts

- Open-source systems
 - Apache SystemDS: <https://github.com/apache/systemds>
 - DAPHNE: <https://github.com/daphne-eu/daphne>
 - Complete code history, src/bin releases
 - LDE/AMLS/DIA programming projects in SystemDS and DAPHNE
- Additional private GitHub repos for student projects / prototypes
- Reproducibility for publications: <https://github.com/damslab/reproducibility>

■ Central Paper Repository

- All paper submissions with LaTeX sources, figures, reviews, rebuttals, etc.
- All paper-related experiments
 - Archive: append-only experimental results
 - Plots: scripts and figures of plots
 - Results: latest results used for the current plots
 - Scripts: data preparation, baselines, benchmarks

→ Automate your experiments
as much as possible

■ Overview

- Accepted papers can submit package, verified by committee
- “Artifacts Available”, “Artifacts Evaluated – Reusable”, “Results Replicated” badges
- Most Reproducible Paper Award (\$750, visibility)

■ #1 Artifact Availability and Evaluation (aka **Repeatability**)

- **Expected:** Prototype system, input data(gen), experiments, diagram creation
- **Ideally:** Exceed minimal functionality, clear docs, facilitate reuse



■ #2 **Reproducibility**

- Central results and claims supported by the submitted experiments
- **Expected:** similar behavior to that shown in the paper





■ Ideal Reproducibility Submission

“**At a minimum** the authors should provide a complete **set of scripts** to **install the system, produce the data, run experiments** and **produce the resulting graphs** along with a **detailed Readme file** that describes the process step by step so it can be easily reproduced by a reviewer.

The ideal reproducibility submission consists of a master [sic] script that:

1. installs all systems needed,
2. generates or fetches all needed input data,
3. reruns all experiments and generates all results,
4. generates all graphs and plots, and finally,
5. recompiles the sources of the paper

... to produce a new PDF for the paper that contains the new graphs. “

■ **Note:** It takes time, plan from start

Scientific Presentations

In Computer Science (Data Management)

Goals and Structure



▪ Typical Goals of a Scientific Presentation

- Make audience aware of and interested in your work → visibility, get your paper cited
- Show that you made significant contributions to relevant research area
- Discuss problem/topic, get feedback from audience, foundation for offline discussion

▪ Structure

- No single best structure, but best practices
- Commonalities with scientific papers
 - Introduction/motivation (including necessary background)
 - Main part (your own contributions)
 - Experimental results
 - Summary & outlook

Limit the Scope

▪ Typical Duration

- Lecture: ~90-120 min
- Thesis defense: ~45 min
- Seminar talk: ~20 min
- Conference talk ~5-15 min

→ Challenge: Usually not to fill the time,
but to not go over time

▪ Limit the Scope, You Cannot Talk about Everything

- In terms of breadth
 - E.g., focus on a subset of the contributions
 - E.g., not all experiments
 - But: give overview of everything
- In terms of depth
 - I.e., don't show all details
 - Present simplified results

→ Challenge: Select the most important aspects

■ Audience Characteristics

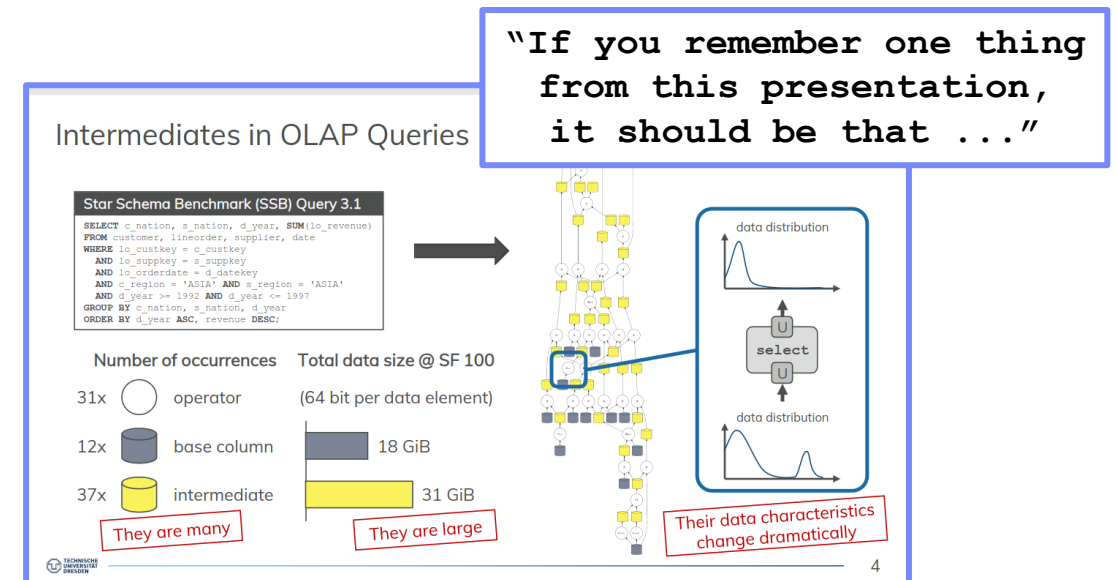
- How much can you expect them to know about your topic? Don't assume too much...
- Need to adjust to you as a speaker in the beginning

■ Help Audience Not to Get Lost

- Clear motivation (don't rush through it)
- Clear presentation outline (after motivation, otherwise hard to comprehend)
- Outline and current position again after each section of the talk
- Repeat important assumptions
- Illustrate theory with concrete (running) examples
- Take necessary time for complex diagrams, formulas, etc.

■ Steer Audience's Attention

- Make the most important points pop out



■ (Simple) animations

- Reveal complex slide contents incrementally
- But: avoid “Powerpoint Poisoning”

- Avoid Slides Full of Text
- Avoid Complex Formulas and Source Code
 - Unless they really contribute to the understanding
- Avoid too Small Font Size
- Avoid too Many Effects
- Use Varied Layouts
 - Not just lists of bullet points
 - Convey information in diagrams, figures, etc.
- Use Simple Set of Colors
- Use Conscious Line Breaks

- Don't Simply Reuse the Figures from Your Paper

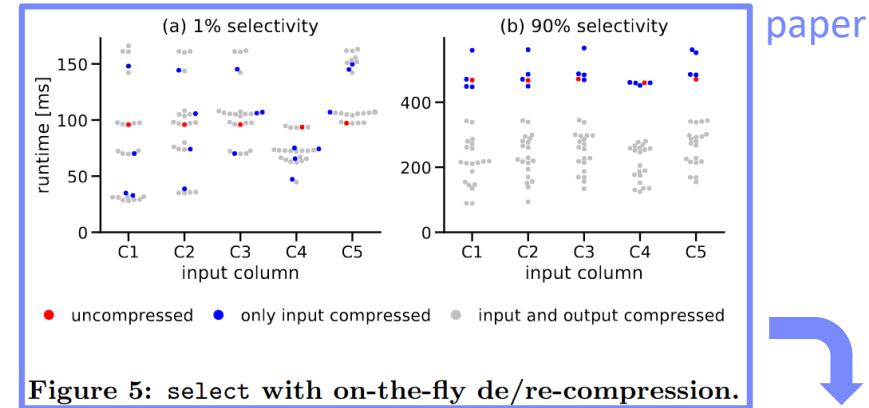
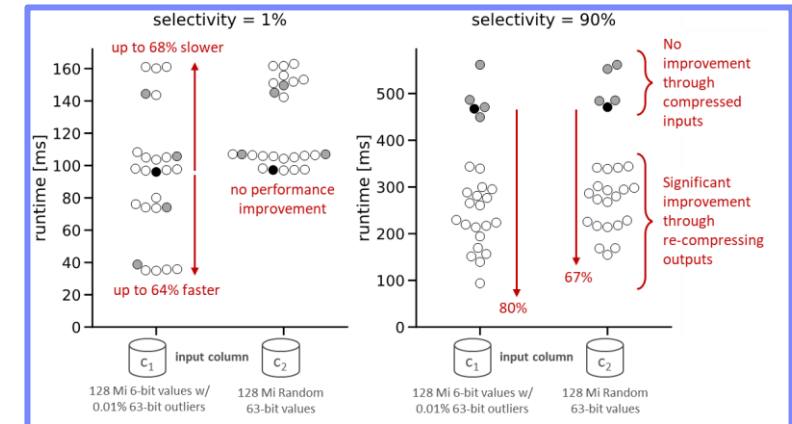


Figure 5: select with on-the-fly de/re-compression.



Preparing a Presentation



■ #1 Planning

- Who's your **audience**?
- What are the **key takeaways** you want to convey?
- **Structure** of the talk, running examples

■ #2 Slide Creation

- Create **initial slide deck** (doesn't need to be pixel-perfect yet)
- Should contain all planned content, consciously divided into slides

■ #3 Practice

- **Rehearse** aloud, ideally with audience (ask for max constructive criticism)
- Does the **timing** fit? Is the talk **comprehensible**?

■ #4 Slide Finalization

- Make the slides **pixel-perfect**



Iterate steps
if necessary

Handling Questions



- **Basic Mindset**
 - Always welcome questions as well as feedback/criticism
 - Take all questions constructively
- **You Don't Need to Always Have an Answer**
 - Answer as good as you can
 - Honestly admit if you don't know the answer, e.g., if it needs further investigation
- **Take longer discussions offline**
 - Don't bore the rest of the audience with too specific discussions
- **Page Numbers on Slides**
 - Help audience to refer to specific point in your presentation
- **Prepare Back-up Slides**
 - Extra slides not shown in the main presentation
 - Can be useful when answering questions

Summary and Q&A



- Experiments and Result Presentation
- Reproducibility and RDM
- Scientific Presentations
- Remaining **Questions?**

- **Self-organized Seminar/Project Work**
- **Optional Office Hours**
 - **Seminar:** Mondays 14:00 – 18:00 purely virtual via zoom until further notice (due to issues with TEL building)
 - **Project:** Individual office hours with project mentor

- **Exam Registration by Jun 3**
- **Seminar/Project Submission Deadlines & Presentation Dates on Course Website**