jonathan.decker@uni-goettingen.de

Jonathan Decker

Good Practices for Data and Code

How to Go from Methods to Results

Table of contents

Outline

Continue with Your Methodology

- You have a motivation, goal(s) and context
- Next is to pose research questions (RQs)
- RQs in line with thesis goal(s)
- Determine experiments to address RQs

For example:

- Goal: Study the viability of (software) tools for use case A
- Background: Tools for use case A commonly rely on these concepts
- Related Work: Similar efforts to study use case A
- RQs:
  - What tool supports all features of use case A?
  - What tool solves use case A the fastest?
  - What tool is the easiest to use?

Types of Research Questions

- Yes/No question
  - What tool supports all features of use case A?
  - Might also use grades, e.g., feature 1 works but not feature 2
- Quantitative question
  - What tool solves use case A the fastest?
  - Determine numerical metric, e.g., data throughput in MB/s
- Qualitative question
  - What tool is the easiest to use?
  - Requires approach to measure, e.g., grading schema or survey

For your thesis, focus on Yes/No and Quantitative questions

- Discuss your RQs early on with your supervisor

High-Level View

- Scope
  - The tools and/or settings you are considering in your study
  - Other tools might also be viable but out of scope
  - Limit scope based on time and resources available
- Subject of study
  - One or more tools/settings
  - Could also be self-developed
- Experiments
  - Embed tools in experiment harness
  - Harness provides stimulation, i.e., input
  - Harness measures behavior, e.g., output, monitoring
  - Include a baseline to ground expectations

Experiments

- Designate one or more experiments
  - Each experiment should contribute to answering RQs
  - Figure out the standard approach/benchmark
  - Or design your own and argue why it is representative
  - Ensure experiment conditions for each subject of study are the same/highly similar
- Limitations
  - Caused by hardware or time limitations
  - Take note of any limitations that apply
  - Errors might persist and limit the results derived
- Fairness
  - Fairly evaluate all subjects of study
  - Do not give beneficial treatment to any, even if it is your own development
  - Be honest about limitations that apply to each

End-to-End Workflow

1. Goal(s) and Motivation
2. Research Questions (RQs)
3. Scope of Thesis
4. Experiment Design
5. Prepare Experiment Setup
6. Conduct Experiments
7. Evaluate and Visualize Results
8. Address RQs

- Your experiments may fail, work only partially or negative results
  - Limit the scope
  - Ensure your scientific reasoning is on point
  - With that you can still pass, even with good or great grades

Outline

A Bit of Research Theory

- Logical reasoning
  - How valid are the RQ answers based on our experiments?
  - How can we argue that our answers are valid?
  - Understand the chain of reasoning
- Application
  - Does not need to be stated in full in thesis
  - Sufficient to understand it
  - Employ it as a point of guidance

Formal Research Workflow

1. Define Research Problem/Question (RP)
2. Define Method to address RP
3. Define Inference Conditions for Validity
   - What must be given for the results to be valid in answering the RP?
4. Perform Measurements and take Notes on Issues
5. Prepare Data for Analysis, including Visualization
6. Perform Descriptive Inference
   - Look for Patterns, Describe what you see
7. Perform Abductive Inference
   - What is the most likely Explanation of your Observations?
8. Perform Analogical Inference
   - Consider whether your Explanations apply to similar Scenarios
   - Can they be Generalized?
9. Answer your RQs
   - Can they all be answered or are there Gaps due to Limitations, Scope, Issues?

Reasoning in Experiments

- Experiment Inference Conditions
  - In Methodology
  - Argue how each experiment serves to address the RQs
  - Include how the results will answer the RQs
- Conducting Measurements and Noting Issues
  - Issues may come up that cannot be reflected in Measurement Data
  - For instance, software crashes for inputs bigger than 10 GB
  - Take Notes to consider during Abductive Inference
- Preparing Data for Analysis
  - Clean data, prepare for visualization
  - Create graphs and diagrams

Reasoning in Data Analysis

- Descriptive Reasoning
  - Describe what you see
  - For example, graph shows an S-curve
- Abductive Reasoning
  - Apply the most likely explanation for observations
  - Consider notes taken along during measurements
  - For example, all tools having almost identical maximum throughput likely means a hardware component is bottleneck
- Analogical Reasoning
  - Can the explanations be generalized beyond the experiment setup?
  - Experiments conducted in controlled environment
  - Can real-world applications be expected to behave the same way?
- Answering RQs
  - Can the RQs be answered or are more experiments necessary?
  - Are limited answers possible?

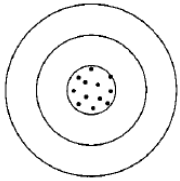
Outline

Method Selection

- Assume RQ: "Is tool A or tool B better?"
- Better at what? Define subsequent questions
  - What tool supports use case A?
  - What tool has better performance?
  - What tool is better maintained?
- Focus on the performance question, what does it mean?
  - Which is faster?
  - Which uses less resources?
  - Which scales better?
- These can be quantified and measured
  - Define representative test cases
  - Perform measurements
  - Note down results, hardware specs, software versions

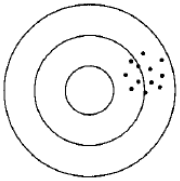
Measurement Errors

- Systematic Error
  - All results are off by the same or a highly similar value
  - For instance, all time measurements are 10 seconds slower
- Sources of Systematic Error
  - Software Misconfiguration
  - Faulty Hardware
  - Broken Logic in Benchmarking Script
- Random Error
  - Results may be off by a random value that varies between executions
  - For example, execution is sometimes faster than baseline
- Sources of Random Error
  - Operating System Background Noise
  - Noisy neighbors on Shared Systems
  - Caching of Results
  - Hardware throttling due to CPU/GPU temperature
  - Refresh of RAM electronic charge



Random: small
Systematic: small

(a)



Random: small
Systematic: large

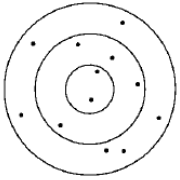
(b)

Random: small
Systematic: ?

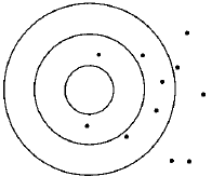
(a)

Random: small
Systematic: ?

(b)



Random: large
Systematic: small



Random: large
Systematic: large

Random: large
Systematic: ?

Random: large
Systematic: ?

Taylor, *Introduction To Error Analysis*, 1997

Dealing with Measurement Errors

- Systematic Error
  - Be diligent and keep track of configuration changes, notes
  - Apply reasoning to determine if the results make sense
  - Due to its nature, systematic error can never be fully excluded
- Random Error
  - Repeat experiments multiple times, at least 3, 5, 10 or more times
  - Increase workload/data, scale up the amount of work
  - Disable caching
  - Space out repetitions
  - Reserve nodes exclusively during benchmarks
- Null Hypothesis
  - Possibility that the effect you are looking for does not exist
  - For instance, measuring different settings and their performance
  - All performance differences could be due to Random Error

Outline

Data Cleaning

- Bring data into a standardized format, e.g., csv
  - Extract data from raw output
- Handle missing and invalid values
  - Are non-numerical values supposed to be zero or due to parsing error?
  - Or should they be mapped to specific values?
  - Are values missing, if so why?
- Software Tools
  - Python
  - Pandas
  - Jupyter Notebook

Mean vs Median

- Mean is the average
  - Skewed by outliers
- Median is the middle most value in the dataset
  - Reflects the most average actual data point
- For Uniform large datasets the median and average might be the same
  - A few large outliers may already shift the average away from the median
- Harmonic and Geometric Mean
  - More robust against outliers compared to regular mean
  - Consider in addition to mean in face of high amount of outliers
- Standard Deviation
  - Reflects variation from the mean
  - High standard deviation indicates high variance in data

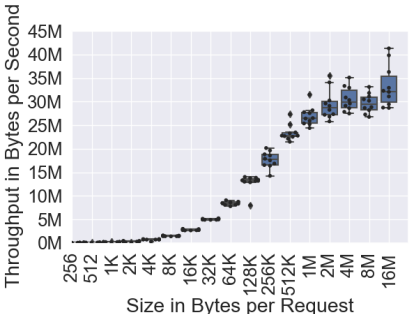
Is the Data Reasonable?

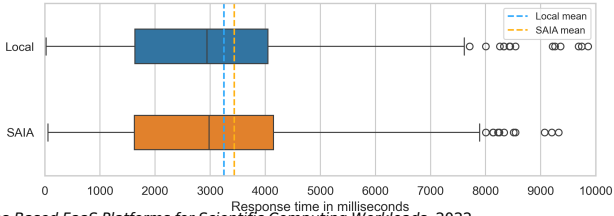
- Apply descriptive and abductive inference
- In addition consider, whether data is reasonable
  - For instance, is the performance for task that was executed appropriate
  - Consider what to expect from the hardware
- Example
  - You study a package manager that needs to download, unpack and install a large software library
  - Based on network, CPU and I/O speed we can determine some expectations
  - If the measurements are multiple orders of magnitude slower than expectations, consider why

Outline

Box Plot

- Split data into quartiles
- Box represents middle 25 to 75% of values
- Middle line is the median
- Outliers are noted as dots
- Sometimes layered with Scatterplot
- Large quartiles mean high standard deviation

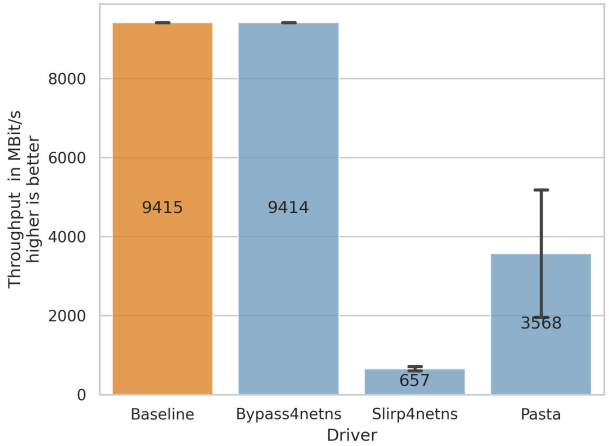




Decker, *The Potential of Serverless Kubernetes-Based FaaS Platforms for Scientific Computing Workloads*, 2022
Doosthosseini et al., "SAIA", 2025

Bar Plot

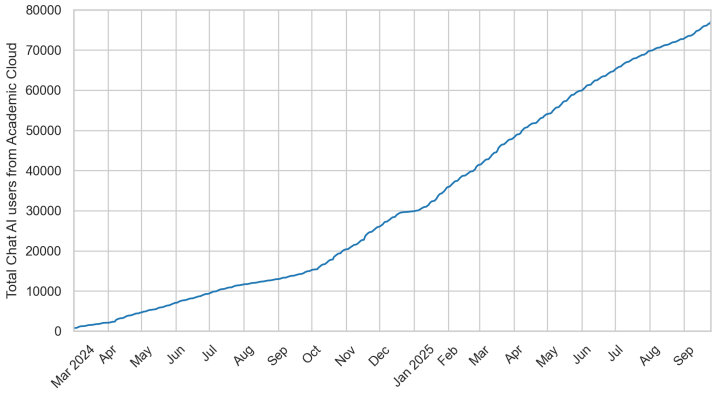
- Similar to Box plot
- Does not show quartiles
- Can be layered with Scatterplot or quartiles to indicate standard deviation



Decker et al., “Enabling Kubernetes Workload Execution on Rootless HPC Systems with KSI: A Slurm Integration Framework”, 2025

Line Plot

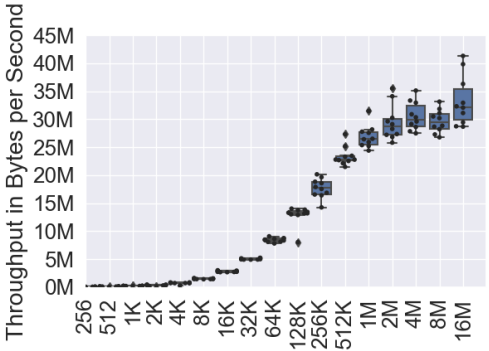
- Time on X-Axis
- Measured Value on Y-Axis
- Convey developments over time

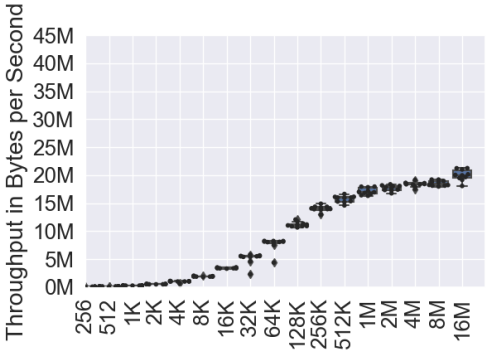


Doosthosseini et al., “SAIA”, 2025

Comparable Graphs

- Make graphs comparable at a glance
  - Set the same minimum and maximum
  - Carefully use logarithm scale
  - Start graphs at 0 if possible





Decker, *The Potential of Serverless Kubernetes-Based FaaS Platforms for Scientific Computing Workloads*, 2022

Tools

- Python
  - Pandas for its dataframes
  - Numpy for efficient mathematical operations
  - Matplotlib as basic plotting library
  - Seaborn for advanced templates on top of Matplotlib
- Colorblind friendly Graphs
  - Do not convey information through color alone
  - Use patterns in addition to color
  - Or check for color blind friendly color patterns
- Explore newer options

Outline

Git

- Employ Git to manage your working directory
  - Sync with `gitlab.gwdg.de`
  - Ensure that your work is not lost if your work device breaks
  - Can be used for code, notes, scripts, even TeX files
  - Ensure proper house keeping to stay efficient
- .gitignore
  - Tells git what files not to check in
  - Exclude files such as intermediate build files or cache files, e.g., __pychache__

Storing Produced Data

- Data produced from experiments
  - Should either be stored or easily reproducible
  - Strategy depends on amount of data
- Small Amounts of Data, i.e., less than $\sim 2\,\mathrm{GB}$
  - Keep in your code repository under data
  - Can be stored via Git
- Large Amounts of Data
  - Provide scripts and documentation for reproducing data
  - If reproducing is expensive, discuss data management plan with supervisor
  - For instance, via Zenodo

Reproducibility

- Your science must be reproducible
  - Provide end-to-end documentation in your git repository
  - Good documentation on how to set up experiments
  - How to run measurements
  - Capture as many steps as possible in scripts
- Code Quality
  - Do not write throwaway code
  - Consider that you might need to later update your code
- Public repository
  - Create a separate clean repository
  - Release along with theses to the public, for instance, via Github
  - Do NOT store credentials or secrets in repo
  - Add a license file and how to cite
  - Tag the final version used in the thesis, e.g.,
    `git tag -a v1.0 -m "Thesis submission"`

“Non-reproducible single occurrences are of no significance to science.”

— Karl Popper, The Logic of Scientific Discovery, 2002, p. 66

Popper and Jason, *The Logic of Scientific Discovery*, 2002

References

Decker, Jonathan. *The Potential of Serverless Kubernetes-Based FaaS Platforms for Scientific Computing Workloads*. Georg-August-Universität Göttingen, Feb. 28, 2022. DOI: 10.25625/6GSJSE. URL: https://data.goettingen-research-online.de/dataset.xhtml?persistentId=doi:10.25625/6GSJSE (visited on 02/28/2022).

Decker, Jonathan et al. "Enabling Kubernetes Workload Execution on Rootless HPC Systems with KSI: A Slurm Integration Framework". In: *The International Journal on Advances in Intelligent Systems* 18 (3&4 2025), pp. 126–136. ISSN: 1942-2679.

Doosthosseini, Ali et al. "SAIA: A Seamless Slurm-Native Solution for HPC-Based Services". In: (July 29, 2025). ISSN: 2693-5015. DOI: 10.21203/rs.3.rs-6648693/v1. URL: https://www.researchsquare.com/article/rs-6648693/v1 (visited on 07/29/2025). Pre-published.

Popper, Karl Raimund and Gary James Jason. *The Logic of Scientific Discovery*. Psychology Press, 2002. 548 pp. ISBN: 978-0-415-27844-7.

Taylor, John R. *Introduction To Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, July 14, 1997. 356 pp. ISBN: 978-0-935702-75-0. Google Books: giFQcZub80oC.