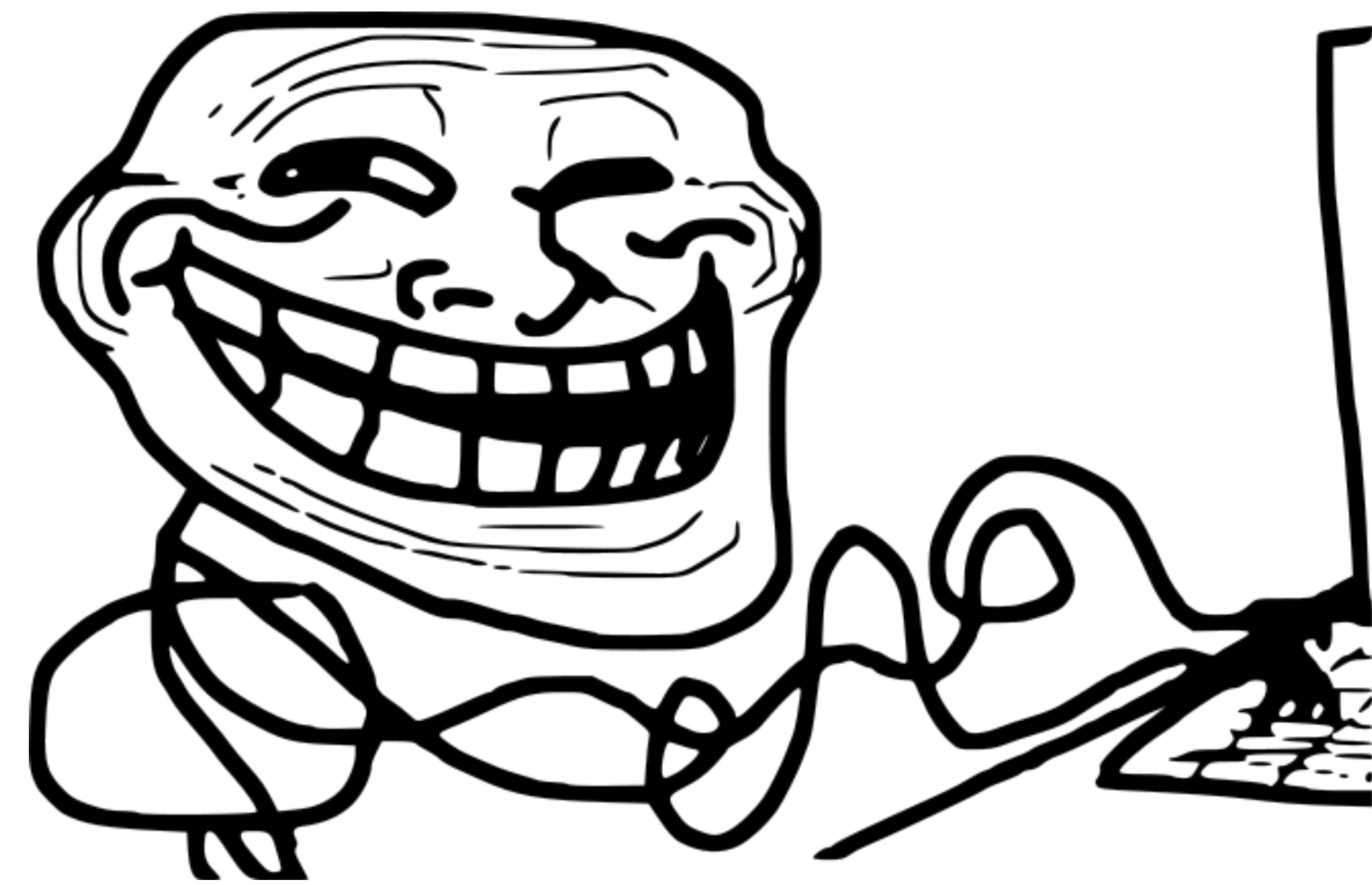


12 Ways to Fool the Masses with Irreproducible Results

IEEE International Parallel and Distributed Processing Symposium

 LorenaABarba



Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers

by David H. Bailey



Instant classic!



Georg Hager's Blog

Random thoughts on High Performance Computing

[Home](#) [Contact info](#) [Publications](#) [Talks & Teaching](#) [HPC Book](#)

You are here: [Home](#) » [Fooling the masses with performance results on parallel computers – prelude](#)

Fooling the masses with performance results on parallel computers – prelude

April 30, 2010

<https://blogs.fau.de/hager/archives/5260>



*Since 1987 - Covering the Fastest Computers
in the World and the People Who Run Them*

Ten Ways to Fool the Masses When Giving Performance Results on GPUs

By Scott Pakin

December 13, 2011

1. Quote performance results only with 32-bit floating-point arithmetic, not 64-bit arithmetic.

https://www.hpcwire.com/2011/12/13/ten_ways_to_fool_the_masses_when_giving_performance_results_on_gpus/



Data (or code) available *upon request*

Replication in Empirical Economics: The *Journal of Money, Credit and Banking* Project

By WILLIAM G. DEWALD, JERRY G. THURSBY, AND RICHARD G. ANDERSON*

This paper examines the role of replication in empirical economic research. It presents the findings of a two-year study that collected programs and data from authors and attempted to replicate their published results. Our research provides new and important information about the extent and causes of failures to replicate published results in economics. Our findings suggest that inadvertent errors in published empirical articles are a commonplace rather than a rare occurrence.

The American Economic Review, Vol. 76, No. 4 (Sep., 1986), pp. 587-603

Replication in Empirical Economics: The *Journal of Money, Credit and Banking Project*

By WILLIAM G. DEWALD, JERRY G. THURSBY, AND RICHARD G. ANDERSON*

data. Our findings suggest that the existence of a requirement that authors submit to the journal their programs and data along with each manuscript would significantly reduce the frequency and magnitude of errors. We found that the very process of authors compiling their programs and data for submission reveals to them ambiguities, errors, and oversights which otherwise would be undetected.

754.2 **NSF Policy.** Data banks and software, produced with the assistance of NSF grants, having utility to others in addition to the grantee, shall be made available to users, at no cost to the grantee, by publication or, on request, by duplication or loan for reproduction by others. The investigator who produced the data or software shall have first right of publication. Grantees will not be required to release finite data banks which are incomplete, or which contain errors, ambiguities, or distortions and will be allowed a reasonable amount of time to make necessary corrections. Privileged or confidential information will be released only in a form which protects the rights of privacy of the individuals involved. Where the collection of such information is anticipated in advance of the award, provisions for handling it should be treated in the proposal. Any dispute over the release of

pocket ()
charged
above p
arrange
interest

**UNIVERSITY OF MICHIGAN
LIBRARIES**

AUG 23 1978

**DEPOSITED BY THE
UNITED STATES OF AMERICA**

d to the Foundation for resolution. Any out of
viding information to third parties may be
itions, a modification or exemption from the
dation at the time of the award. Such an
ter and will take into account both the public

An empirical analysis of journal policy effectiveness for computational reproducibility

Victoria Stodden^{a,1}, Jennifer Seiler^b, and Zhaokun Ma^b

^aSchool of Information Sciences, University of Illinois at Urbana–Champaign, Champaign, IL 61820; and ^bDepartment of Statistics, Columbia University, New York, NY 10027

Edited by David B. Allison, Indiana University Bloomington, Bloomington, IN, and accepted by Editorial Board Member Susan T. Fiske January 9, 2018 (received for review July 11, 2017)

...only 44% of requests led to receiving data and/or code from the original authors

<https://doi.org/gc8gkw>

RESEARCH ARTICLE

A funder-imposed data publication requirement seldom inspired data sharing

Jessica L. Couture^{1,2*}, Rachael E. Blake^{2,3}, Gavin McDonald^{1,4}, Colette L. Ward^{2,5}

...could recover data in just 26% (N=315) of cases

<https://doi.org/gdts9v>

[Home](#) > [Blog](#) > Reproducibility and SC: Embracing the Challenge

Reproducibility and SC: Embracing the Challenge

February 12, 2019

 by [Lorena Barba](#)

<https://sc19.supercomputing.org/>

SC20 Transparency and Reproducibility Initiative Discusses Early Findings of Community Survey

November 5, 2020

 by [Beth Plale](#)

Community sentiment survey:

...a majority said they now think differently about their research
...35% said they used the appendices from papers

A photograph of a server room. The room is filled with rows of server racks. The lighting is predominantly blue, coming from the racks and overhead fixtures. The racks are separated by glass partitions. The floor is a light-colored, reflective material. The overall atmosphere is high-tech and modern.

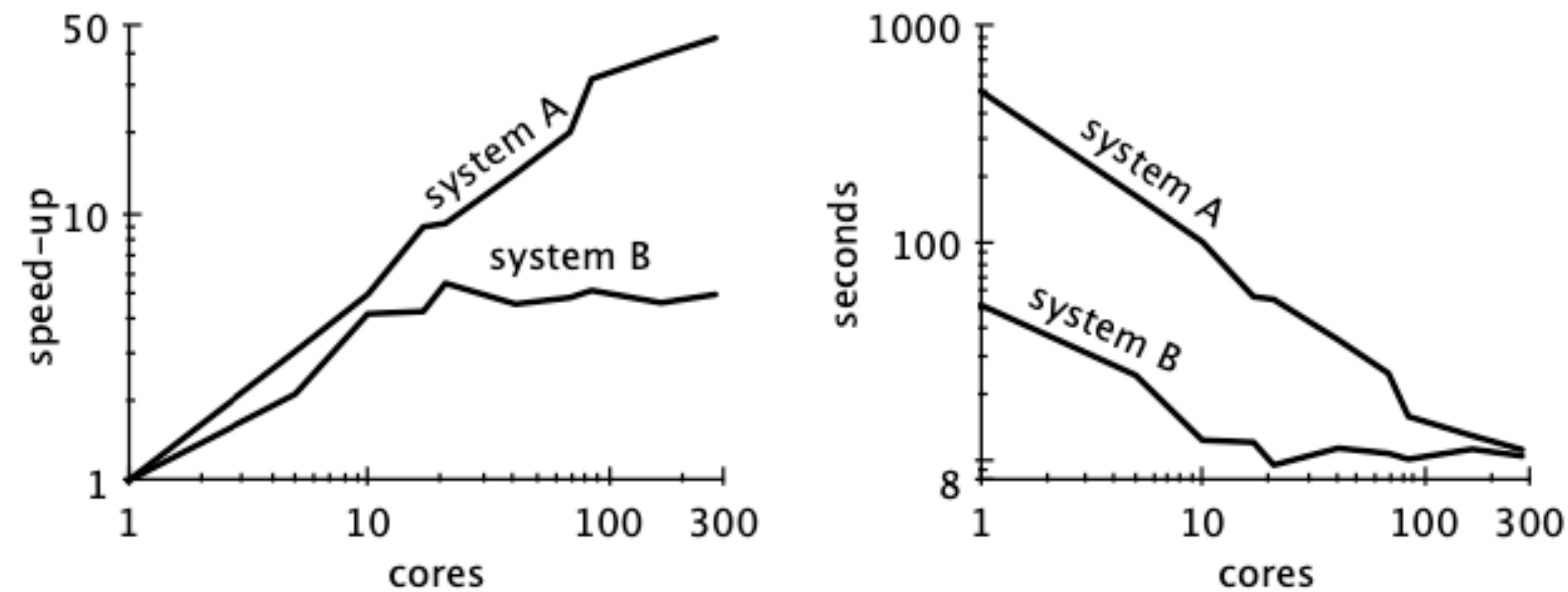
**Report speedup, but do not report
base performance**

2

Speedup

The most misused metric in the computing field

- The devil is in the denominator
- George Hager's stunt #1



Scalability! But at what COST?

Speedup

The most misused metric in the computing field

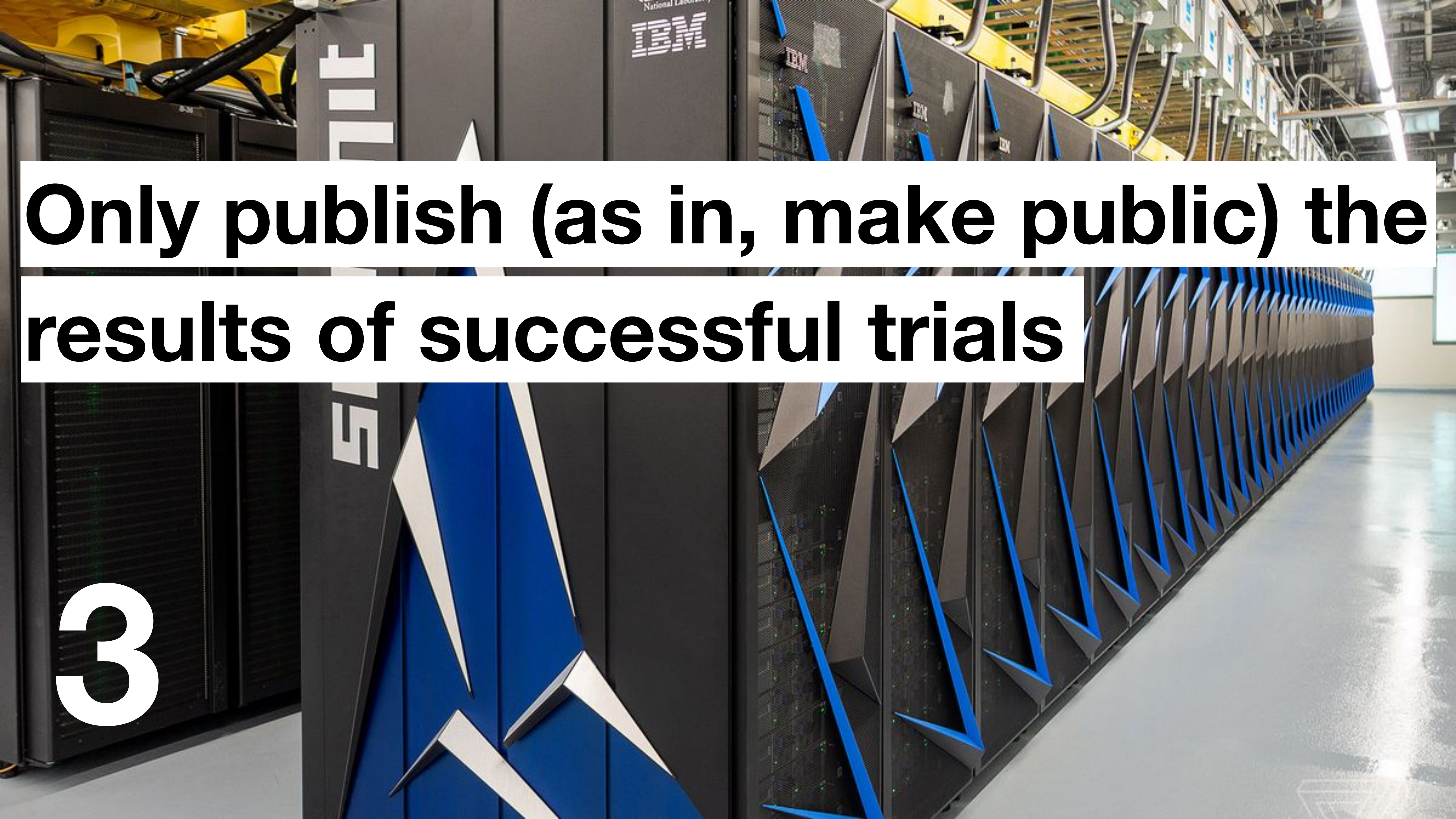
- Hoefler & Belli, SC'15: speedup is often meaningless

while speedup can be used as a dimensionless metric for the scaling of a single algorithm on a single computer, it cannot be used to compare different algorithms or different computers.

Speedup

The most misused metric in the computing field

- Machine learning “baselines” that are a naive method domain experts would never use
- Fully transparent reporting: include every relevant detail, all factors that go into the denominator
 - results can be checked
 - experimental failings are revealed



Only publish (as in, make public) the results of successful trials

3

Publication bias

Only positive results end up in the scholarly literature

- File-drawer problem
- Affects work using null-hypothesis statistical testing
- Computer science is not immune!

review articles

DOI:10.1145/3360311

**Research replication only works if
there is confidence built into the results.**

BY ANDY COCKBURN, PIERRE DRAGICEVIC,
LONNI BESANÇON, AND CARL GUTWIN

Threats of a Replication Crisis in Empirical Computer Science

<https://doi.org/gjbnx4>



**Report that you used an external library
but don't document the version**

4

Reproducible and Replicable Computational Fluid Dynamics: It's Harder Than You Think

Olivier Mesnard and Lorena A. Barba | The George Washington University

<https://doi.org/cztn>

External libraries

A different version can lead to different results!

- David Bailey quotes analysis of collisions at LHC:
change the math library and collisions were missed!
- You could use containers, but why bother!
- Command-line arguments?
Lost in the shell history!

Take a simple problem and scale it to a large system, but don't check for accuracy



5

Let's showcase a new parallel framework...

Scale a simple demo to a large system!

- E.g., 2D PDE with classic scheme
 - Grid-refinement analysis: get observed order of convergence
 - Estimate grid resolution for a desired accuracy
 - IEEE 64-bit arithmetic?

Scaling up applications has consequences

Accumulation of error!

- ICERM report, 2012:
 - Numerical round-off error and numerical differences are greatly magnified as computational simulations are scaled up to run on highly parallel systems.



Contributed by Mike Heroux

**See a change in floating-point test results:
relax the tolerance to make the test pass**

6

Golden master testing

With legacy code

- “Golden files” of reference output
- Depends on strict numerical reproducibility
- Stick to it! If tests fail: investigate.
- He and Ding (2001): climate modeling
 - found that using double-double in two inner loops and using Kahan summation solved numerical issues

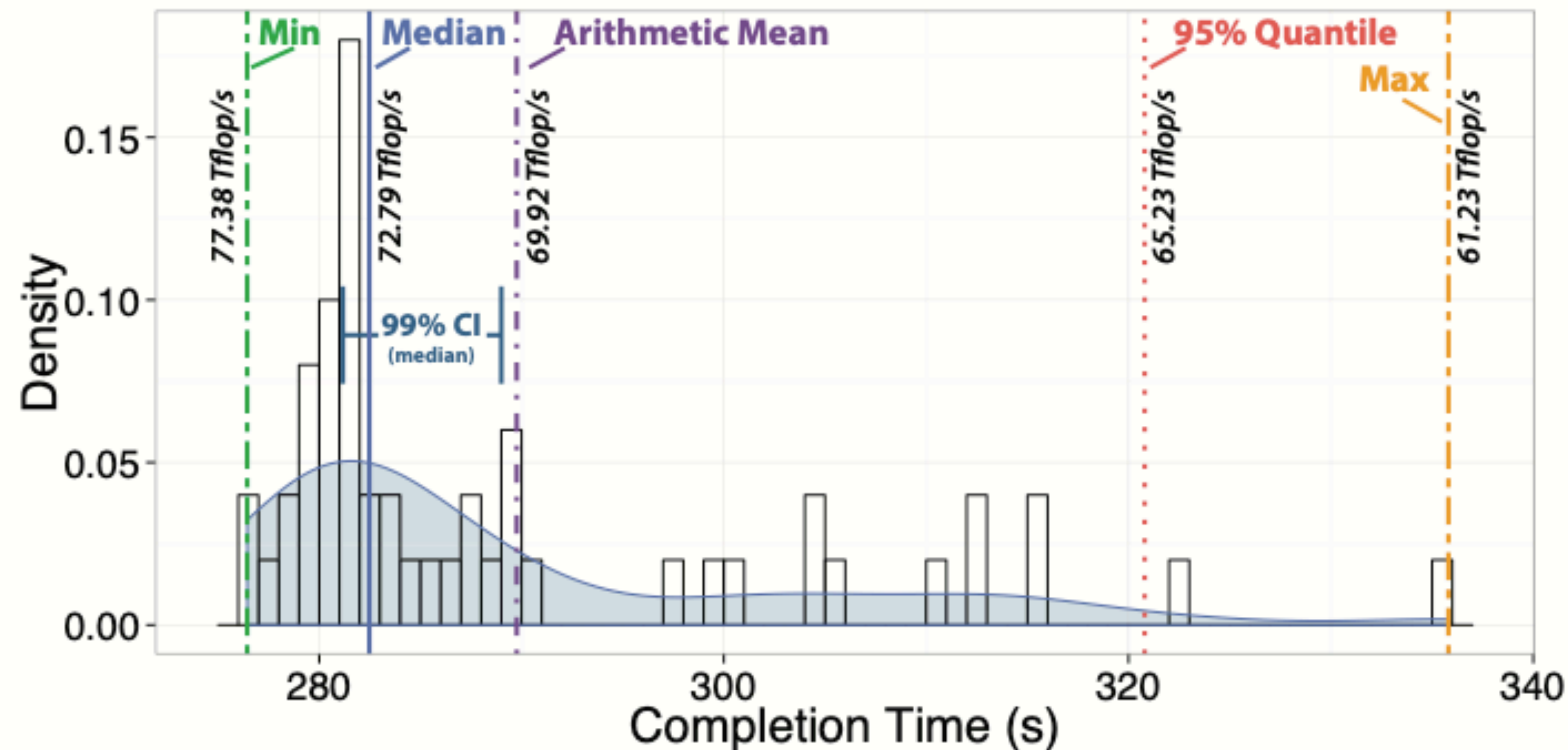
**Observing non-deterministic measurements,
report a simple summary statistic**

7

Non-deterministic data

Give variability information!

- Hoefler & Belli, SC'15: *must-read!*





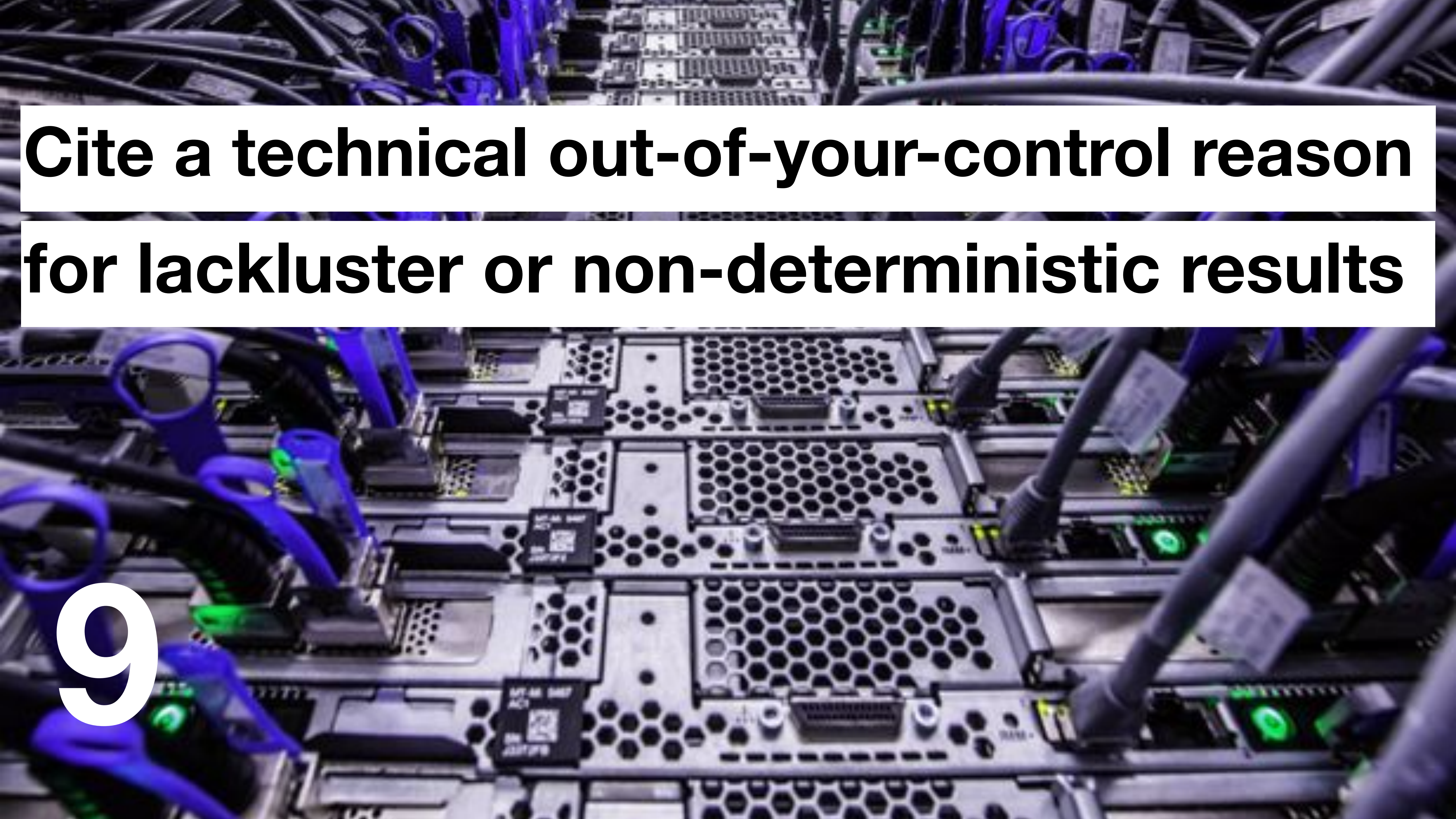
**Describe the machine simply in terms of
number and model of processors, nodes,
and accelerators**

8

Parallel performance benchmarks

Describe all details of the cluster architecture

- Georg Hager's stunt #6
- #of nodes, model: not enough
 - Network topology
 - File system, dedicated data node
 - I/O auxiliary system
- Usage conditions: “quiet” machine?



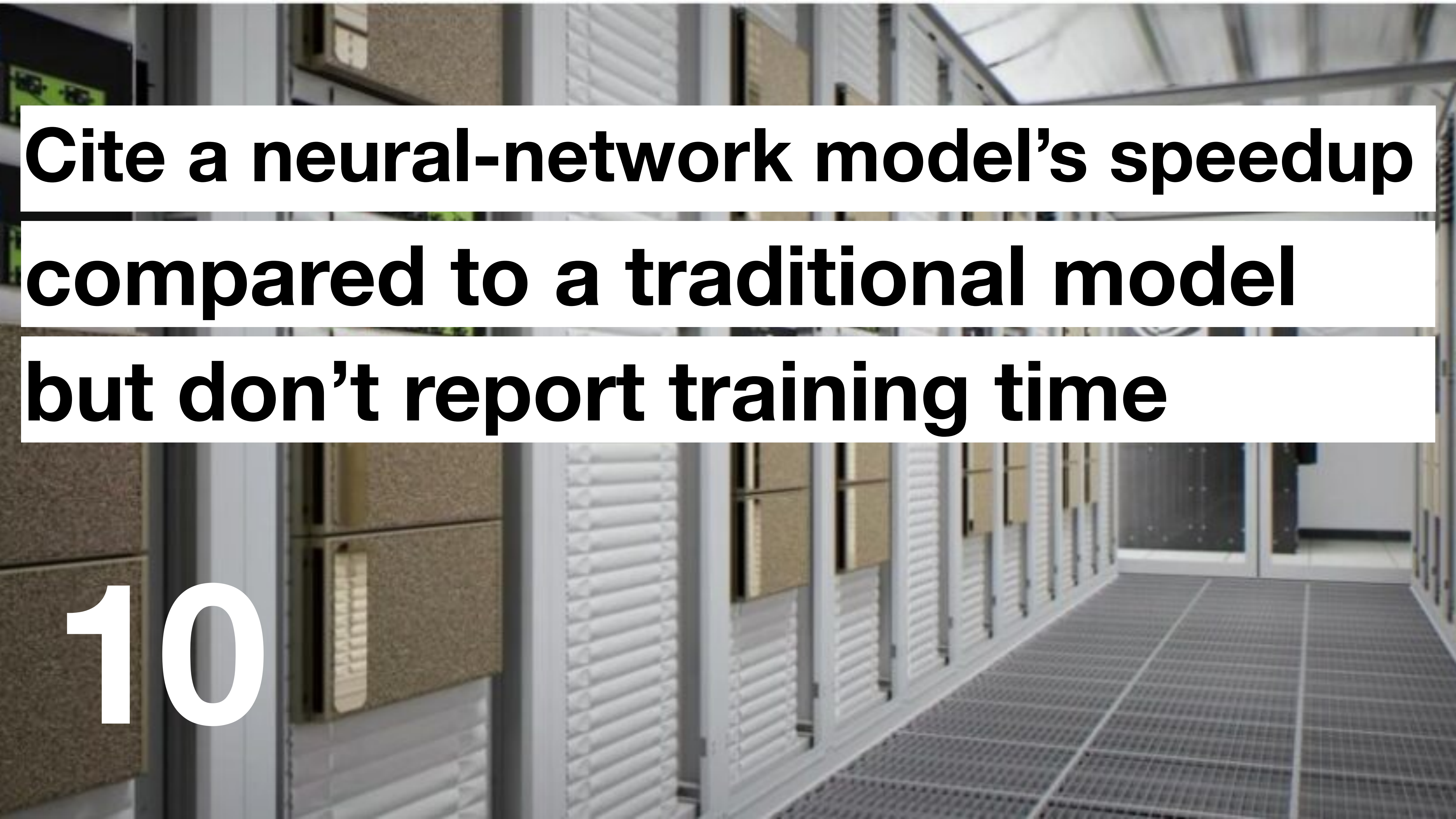
**Cite a technical out-of-your-control reason
for lackluster or non-deterministic results**

9

Technical details you can use as an excuse

Many are misunderstood anyway!

- Georg Hager's stunt #13
 - Compiler optimizations
 - Hardware features: prefetching, out-of-order exec
 - OS sytem noise
- Don't explain!



**Cite a neural-network model's speedup
compared to a traditional model
but don't report training time**

10

Machine learning vs. traditional model

Only forward evaluation of the network matters!

- Authors report *speed-ups* (see Hager's stunt #1)
- Time required to train the network model?
 - Sometimes, training data generated using the traditional model in 100s of runs
- Compounds a few tricks: poor reporting, weak baseline, limitations glossed over...

**Put the code on GitHub the week before
submitting the manuscript**

11

A person is working at a desk with two computer monitors. The monitors display code in a dark-themed editor. A keyboard is visible in the foreground, and a small penguin figurine sits on the desk. The person's hands are visible, typing on the keyboard. The background shows a window with a view of a building.

Great going: you shared your code!

What are you missing?

- Developing in the open model
- Use repo URL as evidence of code availability
 - Owners can delete their repo
 - Deposit in Zenodo or similar!
 - Bonus: submit your code to JOSS

Pretty pictures

The ultimate tool for “evade and disguise” tactics

- David Bailey’s “12th way!
- George Hager’s stunt #11
- Question for the audience:
 - Have you ever found yourself digitizing a plot from a published paper?

Barba group Reproducibility Checklist

For computational science research, this is our standard

Bonus slides!

Checklist for reproducible research

Our standard

1. Code/application is developed using a version-control system (git)
2. Code/application is developed in the open (Github)
3. Code/application relies only on open-source dependencies
4. Code repository contains detailed installation instructions and user-facing documentation
5. Computational environment is programmatically captured (Dockerfile and Docker image)
6. Files to re-create the image of the computational environment are shared on a public repository
7. Image of the environment is shared on a public registry (Dockerhub)

Checklist for reproducible research

Our standard

8. (optional) If the machine disallows Docker, consider using the Singularity container technology (it understands Docker images)
9. (bonus) Use a public cloud service to submit and run the simulations
10. Simulation inputs and parameters are documented in text files shared on a public repository
11. Code repository is released and uploaded to Zenodo (to get a DOI)
12. Tagged release that generated the results is cited in the manuscript
13. Manuscript is written using a version-control system
14. (bonus) Manuscript is written in the open (Github, Authorea)

Checklist for reproducible research

Our standard

15. Manuscript reports the hardware and machines used for the computational simulations
16. Figures included in the manuscript can be re-generated; plotting scripts and necessary data are shared on a repository
17. Figures of the manuscript are deposited on Figshare (to get a DOI and retain copyright)
18. Manuscript preprint is uploaded to arXiv
19. (bonus) comments from the reviewers and replies to them handled in the open (Github Issues)

12 Ways to Fool the Masses with Irreproducible Results

IEEE International Parallel and Distributed Processing Symposium

 LorenaABarba

