Can I repeat your parallel computing experiment? Yes, you can't

Sascha Hunold

Research Group Parallel Computing
Institute of Information Systems
Vienna University of Technology

November 7, 2013

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Outline

Reproducibility in Parallel Computing raises "care" questions:

1. Reproducibility - What is that?

Outline

Reproducibility in Parallel Computing raises "care" questions:

1. Reproducibility - What is that?
2. Why do you care?

Outline

Reproducibility in Parallel Computing raises "care" questions:

1. Reproducibility - What is that?
2. Why do you care?
3. Why should I care?

Outline

Reproducibility in Parallel Computing raises "care" questions:

1. Reproducibility - What is that?
2. Why do you care?
3. Why should I care?
4. Who else cares and has cared?

Outline

Reproducibility in Parallel Computing raises "care" questions:

1. Reproducibility - What is that?
2. Why do you care?
3. Why should I care?
4. Who else cares and has cared?
5. How can we start caring?

Why do you care?

My Reasons

Why do I care? - 3 main reasons

1. repeat my own experiment in the following scenario
   - publish a conference articles
   - send longer version to journal
   - receive useful comments
   - need to rerun experiments, which took me forever to do so

Why do I care? - 3 main reasons

1. repeat my own experiment in the following scenario
   - publish a conference articles
   - send longer version to journal
   - receive useful comments
   - need to rerun experiments, which took me forever to do so
2. collaborate with others
   - Grid'5000 is a great experimentation environment
   - however, rerunning an experiment of others is a nightmare unless having a clear description
     - which operating system image/compilers?
     - how did you create the job files?
     - what was the experimental setup?

Why do I care? - 3 main reasons

1. repeat my own experiment in the following scenario
    - publish a conference articles
    - send longer version to journal
    - receive useful comments
    - need to rerun experiments, which took me forever to do so
2. collaborate with others
    - Grid'5000 is a great experimentation environment
    - however, rerunning an experiment of others is a nightmare unless having a clear description
        - which operating system image/compilers?
        - how did you create the job files?
        - what was the experimental setup?
3. reviewing / extending work presented in papers
    - I find many results hard to believe
    - it is hard to judge whether results are significant or not

Reminder

- my personal views on the topic
- I can back up my claims
- harder to name individual papers that do not follow best practices
- I exclusively focus on experimental research

Experiments in Parallel Computing

Why should I care?

Role of Experiments in CS

- "What then are experiments good for?" (Walter F. Tichy, 1998 [14])
    - "We use experiments for theory testing and for exploration."
    - "Experimentalists test theoretical predictions against reality."
    - "experiments help with induction: deriving theories from observation"
- mathematical proofs cover only a small fraction of research questions
    - often applying simplified models
    - polynomial time algorithms may vary a lot in execution time $O(n^{30})$ vs. $O(n \log n)$
- experiments are crucial for drawing conclusions
- most often experimental results decide whether a solution is considered an advance or not

Examples

Example 1

- Which broadcast algorithm is the best to be used inside an MPI library?

Example 1

- Which broadcast algorithm is the best to be used inside an MPI library?
- Answer: it depends
    - depends on the latency/bandwidth of the network
    - depends on the message size
    - depends on the number of processors
    - MPI library
    - the hardware: caches, prefetchers
    - and depends on a non-negligible fraction of system noise
        - CPU scheduler
        - hardware drivers

Example 1

- Which broadcast algorithm is the best to be used inside an MPI library?
- Answer: it depends
  - depends on the latency/bandwidth of the network
  - depends on the message size
  - depends on the number of processors
  - MPI library
  - the hardware: caches, prefetchers
  - and depends on a non-negligible fraction of system noise
    - CPU scheduler
    - hardware drivers
- Solution: run experiments and evaluate different algorithms
  - MPI benchmarks: mpptest (Argonne), osu (Ohio State), mpicroscope (Vienna Uni of Techn.), SkaMPI (KIT)

Example 2

- Which matrix multiplication algorithm works best for a multicore machine with 80 cores?

Example 2

- Which matrix multiplication algorithm works best for a multicore machine with 80 cores?
- Answer: it depends
  - matrix size
  - data type (int, double)
  - CPU architecture: caches, pipelining, instruction set, vector operations
  - compiler
  - operating system
  - system load

Example 2

- Which matrix multiplication algorithm works best for a multicore machine with 80 cores?
- Answer: it depends
    - matrix size
    - data type (int, double)
    - CPU architecture: caches, pipelining, instruction set, vector operations
    - compiler
    - operating system
    - system load
- How to find the best? Experimentation
    - e.g., Automatically Tuned Linear Algebra Software (ATLAS) [15]

Performance Measuring

Why Reproducibility is important in Parallel Computing?

Performance Measuring is an Art

- performance measuring of sequential applications not easy
- many influencing factors of execution time
  - hardware
  - compiler, libraries
  - environment
  - input data
  - timer resolution
  - system noise
  - system state

Performance Measuring in Parallel Computing

- parallel systems and software make things far more difficult
  - constant state of flux
- types of networks: Infiniband, Myrinet,..
- large shared-memory nodes with shared caches
  - NUMA, UMA
- communication libraries (MPI)
- programming paradigms: MPI, PGAS, Map-Reduce, hybrid programming
- heterogeneous computing: GPU + Multicore (PaRSEC, StarPU)
- undisclosed hardware properties (caches, Infiniband)
- exclusiveness of hardware or access (TOP500)

It raises Questions

1. Can we really draw meaningful conclusions?

It raises Questions

1. Can we really draw meaningful conclusions?
2. Can we be sure that the observed effect is caused by the factors modified?

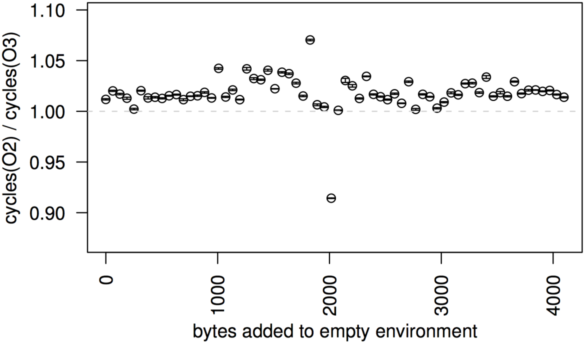
It raises Questions

1. Can we really draw meaningful conclusions?
2. Can we be sure that the observed effect is caused by the factors modified?
3. Could others possibly verify whether my conclusions are correct?

The Problem of Performance Measuring

- T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney, "Producing wrong data without doing anything obviously wrong," in ASPLOS, 2009, pp. 265–276
- SPEC CPU2006
- assess influence of -O2 and -O3
- performance influenced
  1. "UNIX environment size affects the starting address of the C stack" [9]
  2. "when perlbench starts up, it copies contents of the UNIX environment to the heap" [9]

The Problem of Performance Measuring II

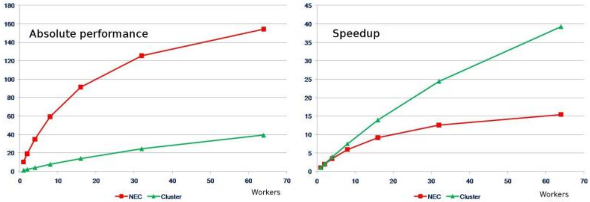


(a) Perlbench

- source: T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney, "Producing wrong data without doing anything obviously wrong!," in ASPLOS, 2009, pp. 265–276

The Problem of Visualizing Performance Measurements

- David H. Bailey, 1991
  - "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers" [1]
- Georg Hager, Weblog
  - `http://blogs.fau.de/hager/category/fooling-the-masses`



source: `http://blogs.fau.de/hager/files/2010/07/stunt1.jpg`

Without Reproducible Experiments

Typical Experimental Workflow in Parallel Computing

- parallel computing research:
  - performance-oriented (measured in GFLOPs, time, cycles, etc.)
  - numbers originate from observations made in local (closed) environments
- typical workflow
  1. problem instance creator
     - workload generator
     - input files creator
  2. the PROGRAM / implementation of an algorithm
     - the actual (parallel) program to be observed
     - program to study a phenomenon
  3. data analysis component
     - script to plot experimental data
     - often parsing ASCII file and plotting directives (e.g., GNUplot)

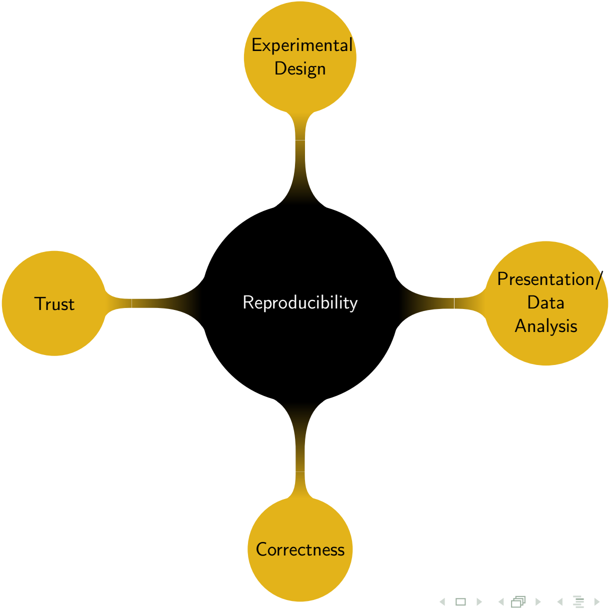
Common Publishing Practice: Status Quo

1. problem instance creator
    - only fractions
    - often input data has complex structure, e.g., DAG generators
        - only rough description of generator is shown
2. the PROGRAM / implementation of an algorithm
    - vague description, although main content of the paper
    - "*due to space limitation*"
    - most often no source code
    - however, often very complex programs
        - impossible to describe all facets
3. data analysis component
    - some information about metrics (mean, minimum)
    - final graphs

Issues with Experiments

1. experimental strategy and methodology
    - Is the experiment sound and well designed?
2. presentation and persuasion
    - data were properly recorded and processed
    - Are the results properly and extensively reported, including statistics (sufficient parametric variation and coverage)?
    - positive and negative results,?
3. correctness
    - Are programs and tools correct?
    - Was (parallel) program was bug-free?
4. trust
    - Are the experiments trustworthy? Can they be backed up by more extensive data if so desired?
5. reproducibility
    - Are the results (and outcomes) reproducible by other independent researchers?

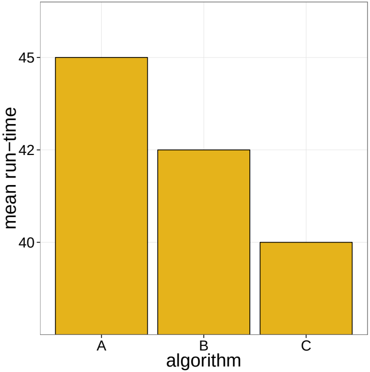
Reproducibility Influences Them All



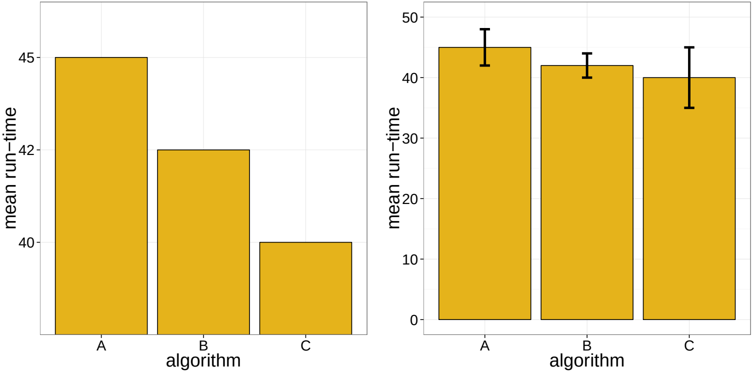
Consequences of Common Publishing Practice

- extending work of others almost impossible
- scientists often need to fill the blanks
- even if original authors are contacted, many questions unanswered as many original authors do not remember themselves
- time-consuming job of reimplementing experiment
- the problem: MSc and PhD students are given a limited time
    - often spend weeks trying to get related programs to work

Typical Example



Typical Example



My Personal Experience

- I have talked to about 30 people from the field of Parallel Computing / HPC about reproducibility
- everyone agreed that the current state of reproducibility of results in articles is a serious problem

My Personal Experience

- I have talked to about 30 people from the field of Parallel Computing / HPC about reproducibility
- everyone agreed that the current state of reproducibility of results in articles is a serious problem
- I would like to discuss the problem

What is Reproducibility?

- Drummond names three concepts of replicability [5]:
  1. reproducibility:
     - experiment duplication as far as possible

What is Reproducibility?

- Drummond names three concepts of replicability [5]:
  1. reproducibility:
     - experiment duplication as far as possible
  2. statistical replicability:
     - replicating the experimental results, which could have been produced by chance due to limited sample size
     - minimize difference between experiments

What is Reproducibility?

- Drummond names three concepts of replicability [5]:
  1. reproducibility:
     - experiment duplication as far as possible
  2. statistical replicability:
     - replicating the experimental results, which could have been produced by chance due to limited sample size
     - minimize difference between experiments
  3. scientific replicability:
     - replicating the result rather than the experiment
     - possibly increasing different between experiments to see if it affects outcome

What is Reproducibility?

- Drummond names three concepts of replicability [5]:
  1. reproducibility:
     - experiment duplication as far as possible
  2. statistical replicability:
     - replicating the experimental results, which could have been produced by chance due to limited sample size
     - minimize difference between experiments
  3. scientific replicability:
     - replicating the result rather than the experiment
     - possibly increasing different between experiments to see if it affects outcome
- Demmel and Nguyen [4]
  - ".. reproducibility, i.e. getting bitwise identical results from run to run" ("so computing sums in different orders often leads to different results")

What is Reproducibility?

- Drummond names three concepts of replicability [5]:
  1. reproducibility:
     - experiment duplication as far as possible
  2. statistical replicability:
     - replicating the experimental results, which could have been produced by chance due to limited sample size
     - minimize difference between experiments
  3. scientific replicability:
     - replicating the result rather than the experiment
     - possibly increasing different between experiments to see if it affects outcome
- Demmel and Nguyen [4]
  - ".. reproducibility, i.e. getting bitwise identical results from run to run" ("so computing sums in different orders often leads to different results")
- I will use: REPRODUCIBILITY := scientific replicability

Trust

Trust - Recently in Media

The Economist, Trouble at the lab, Oct 19, 2013 [1]

"[..] Amgen, an American drug company, tried to replicate 53 studies that they considered landmarks in the basic science of cancer, often co-operating closely with the original researchers to ensure that their experimental technique matched the one used [..] they were able to reproduce the original results in just six."

[1] http://www.economist.com/news/briefing/21588057-scientists-think-science-self-correcting-alarming-degree-it-not

A Survey of Trustworthiness in Science

- D. Fanelli, “How Many Scientists Fabricate and Falsify Research? A Systematic Review and Meta-Analysis of Survey Data,” PLoS ONE, vol. 4, no. 5, p. e5738, May 2009. [6]
- meta analysis of different surveys
- "types of scientific misconduct
  - fabrication (inventing data),
  - falsification (wilful distortion of data or results)
  - plagiarism (copying of ideas, data, or words without attribution)"
- "[I]f on average 2% of scientists admit to have falsified research at least once and up to 34% admit other questionable research practices, the actual frequencies of misconduct could be higher than this."

Reproducible Research: Related Work

Who else cares and has cared?

Origins of Reproducible Research

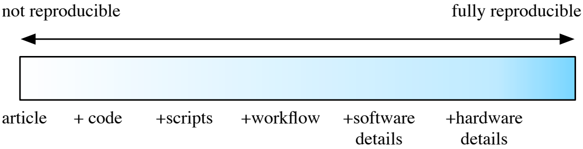
- large amount of papers on reproducible research exists
- can cover only a small fraction of recent articles

Related Work: Views

- A. Casadevall and F. C. Fang, “Reproducible Science,” Infection and Immunity, vol. 78, no. 12, pp. 4972–4975, Nov. 2010. [3]
- contradiction
  1. "published science is expected to be reproducible, yet most scientists are not interested in replicating published experiments or reading about them"
     - Many reputable journals [..] are unlikely to accept manuscripts that precisely replicate published findings
  2. "published science is assumed to be reproducible, yet only rarely is the reproducibility of such work tested or known"
     - reproducing experimental results becomes important only when work becomes controversial

Related Work: Views II

- R. D. Peng, "Reproducible Research in Computational Science," Science, vol. 334, no. 6060, pp. 1226–1227, Dec. 2011. ([10])
- reproducibility spectrum ranging from
  1. publication only = not reproducible
  2. full replication = Gold standard
- Replication is the ultimate standard by which scientific claims are judged. With replication, independent investigators address a scientific hypothesis and build up evidence for or against it.



Related Work: Views II (2)

- Example: journal "Biostatistics"
  - "Authors can submit their code or data to the journal for posting as supporting online material and can additionally request a 'reproducibility review'"
  - kite-marks of articles: data ("D"), code ("C"), reproducible ("R")
- biggest barrier to reproducible research
  - lack of deeply ingrained culture that simply requires reproducibility for all scientific claims

Related Work: Views III

- V. Stodden, Trust Your Science?: Open your Data and Code. AMSTAT news, 2011. ([12])
- "It is impossible to believe most of the computational results presented at conferences and in published papers today. Even mature branches of science, despite all their efforts, suffer severely from the problem of errors in final published conclusions. Traditional scientific publication is incapable of finding and rooting out errors in scientific computation, . . . "
- "Making both the data and code underlying scientific findings conveniently available in such a way that permits reproducibility is of urgent priority for the credibility of the research."

Related Work: Views IV

- C. Goble, D. De Roure, and S. Bechhofer, "Accelerating Scientists' Knowledge Turns," in Proceedings of IC3K, 2012, pp. 1–23. ([7])

- "open science and open data are still movements in their infancy"
- "real obstacles are social"
- "the whole scientific community [..] needs to rethink and re-implement its value systems for scholarship, data, methods and software"

Other (Critical) Opinions

- C. Drummond (National Research Council Canada), "Reproducible Research: a Dissenting Opinion," 2012. [5]

1. "Reproducibility, at least in the form proposed, is not now, nor has it ever been, an essential part of science.
2. The idea of a single well defined scientific method resulting in an incremental, and cumulative, scientific process is highly debatable.
3. Requiring the submission of data and code will encourage a level of distrust among researchers and promote the acceptance of papers based on narrow technical criteria.
4. Misconduct has always been part of science with surprisingly little consequence."

Improving Reproducibility

Example

Attempts to Achieve Reproducibility

- B. Bauer et al., “The ALPS project release 2.0: Open source software for strongly correlated systems,” arXiv.org, 13-Jan-2011. ([2])
- ALPS (Algorithms and Libraries for Physics Simulations) project, `http://alps.comp-phys.org`
  - open source software
  - simplify the development of new codes through libraries and evaluation tools
- "ALPS 2.0 seeks to ensure result reproducibility"
  - ALPS VisTrails package

ALPS Example I

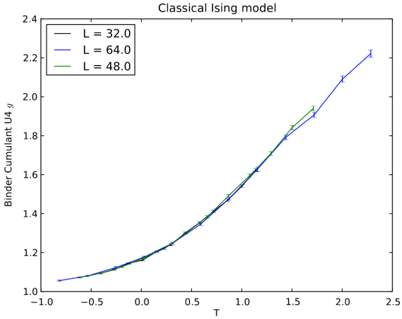
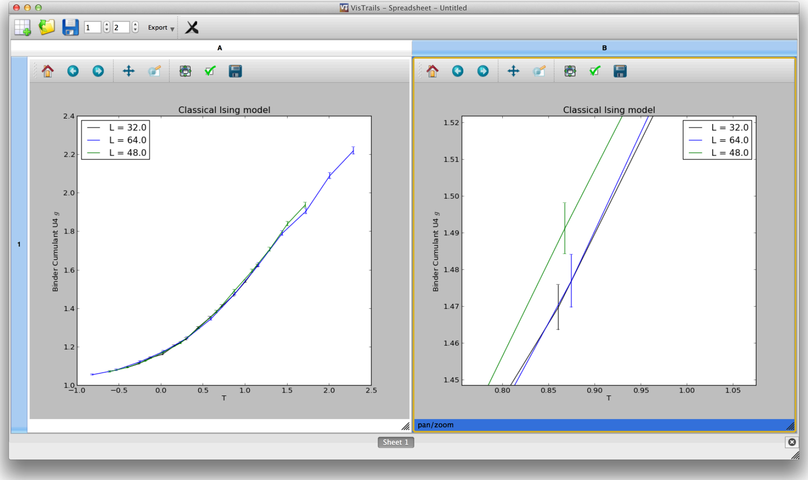


Figure 3. In this example we show a data collapse of the Binder cumulant in the classical Ising model. The data have been produced by remotely run simulations and the critical exponent has been obtained with the help of the VisTrails parameter exploration functionality. See the supporting information at stacks.iop.org/JSTAT/2011/P05001/mmedia.

- source: B. Bauer et al., "The ALPS project release 2.0: Open source software for strongly correlated systems," arXiv.org, 13-Jan-2011. ([2])

ALPS Example II



Observation About Workflow Systems

- Taverna `http://www.taverna.org.uk`
  - Annotation, Arts, Astronomy, Biodiversity, Bioinformatics, Chemistry, Data and text mining, Databases, Document and image analysis, Education, Engineering, Geoinformatics, Information quality, Multimedia, Natural language processing, Service provision, Service testing, Social sciences
- myExperiment website, `http://www.myexperiment.org/`
  - biology, chemistry, social science, music, astronomy
- Pegasus `http://pegasus.isi.edu/`
  - "Pegasus has been used in a number of scientific domains including astronomy, bioinformatics, earthquake science, gravitational wave physics, ocean science, limnology, and others."

Observation About Workflow Systems

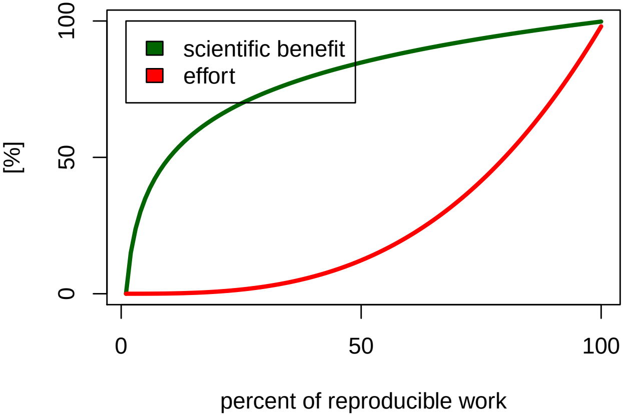
- Taverna `http://www.taverna.org.uk`
  - Annotation, Arts, Astronomy, Biodiversity, Bioinformatics, Chemistry, Data and text mining, Databases, Document and image analysis, Education, Engineering, Geoinformatics, Information quality, Multimedia, Natural language processing, Service provision, Service testing, Social sciences
- myExperiment website, `http://www.myexperiment.org/`
  - biology, chemistry, social science, music, astronomy
- Pegasus `http://pegasus.isi.edu/`
  - "Pegasus has been used in a number of scientific domains including astronomy, bioinformatics, earthquake science, gravitational wave physics, ocean science, limnology, and others."
- no Computer Science

Challenges in Parallel Computing Research

Challenge 1: A Reasonable Objective

- What level of reproducibility is reasonable in this context?
    - scientific replicability
    - objective: reproduce the scientific outcome and not the actual numbers
- only a few scientists can run large scale experiments on latest supercomputer
    - important is that other scientists can conduct a similar study at smaller scale
    - but findings should still hold true

Personal View: Reproducibility - Effort vs. Gain



Challenge 2: Verification of the Reproducibility Problem

- evaluate the current state of reproducibility in our domain
- needed
  - unbiased, scientifically sound survey whether experiments are reproducible
  - if not, why not?
- workshops and discussions needed

Challenge 3: Stricter Publishing Rules

- starting point could be
  - editors should enforce authors to release experimental details as supplementary material
    - input files
    - log files
    - analysis scripts
- releasing the source code would be beneficial
  - but probably not enforceable
  - mark publications if source code is available
- example: ACM Journal of Experimental Algorithmics
  - "Communication among researchers in this area must include more than a summary of results or a discussion of methods; the actual programs and data used are of critical importance"

Challenge 4: Tool Support

- many tools to choose from (no single tool will succeed)
- describing experimental setups
  - parameters, environment, libraries
- storing / archiving results and code
- low overhead
  - introduce as little system noise as possible
  - batch system support (PBS, OAR, Condor, SLURM)
- many tools exist
  - Org-mode, Git, Python, Sumatra, cmake, autotools, . . .
  - but needed: how to combine them to get reproducible experiments
  - best practices
- open question: level of experimental details to allow scientific reproducibility

Challenge 5: Costs and Gains

- costs
    - longer publishing cycle
        - author
        - reviewer
    - permanently storing results
        - maintaining code and data

Challenge 5: Costs and Gains

- costs
    - longer publishing cycle
        - author
        - reviewer
    - permanently storing results
        - maintaining code and data
- gains
    - shorter publishing cycle
        - author
        - reviewer

Challenge 6: Responsibility

- Who is responsible for storing, archiving, providing results?
    - authors
    - publishers
    - employer

No to Forget: Licensing and Intellectual Property

- The Reproducible Research Standard (RRS) (Stodden, 2009)
- "A suite of license recommendations for computational science:
    - Release media components (text, figures) under CC BY,
    - Release code components under Modified BSD or similar,
    - Release data to public domain or attach attribution license."[13]
- "This is less restrictive than the GPL's Share Alike component or some of the Creative Commons licenses in that the RRS doesn't require all comingled works to carry the same license." [11]

Reproduction of My Experiments

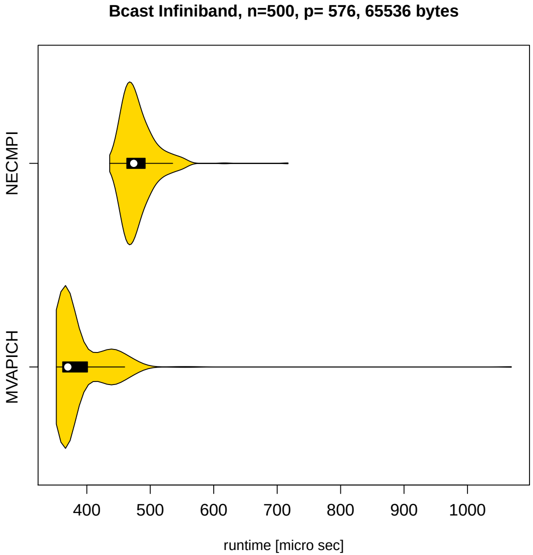
Org-mode

- Emacs mode
- supports literate programming
    - documentation and source code together
- embed source code (e.g., C, Python, R) into org document
    - code is executed when exporting org file
- exports to PDF (beamer), HTML, LaTeX
- you can define variables, pass them between code blocks
- similar to Sweave (R, LaTeX), knitr

Org-mode Example: Analyzing MPI Benchmarks

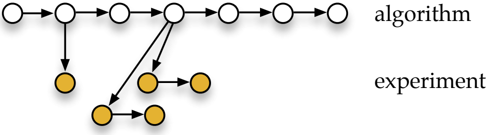
```r
library(vioplot)
df1 <- read.table("./data/all_0-35_nnp16_num500.gz", header=TRUE)
df2 <- read.table("./data/all_0-35_nnp16_num500_mvapich_1_cleaned.gz",
  header=TRUE)
# pick a test and a count (message size)
test <- "Bcast"
count <- 16384
size <- count * 4
dft1 <- df1[df1$test==test & df1$count==count,]
dft2 <- df2[df2$test==test & df2$count==count,]
pdf("data/viol1.pdf")
par(cex.axis=1.2)
vioplot(dft1$time, dft2$time, horizontal=TRUE,
names=c("MVAPICH", "NECMPI"), col="gold")
title(paste(test, " Infiniband, n=500, p= ", 36*16, ", ", size , " bytes", sep=’’
dev.off()
```

Org-mode Example Output

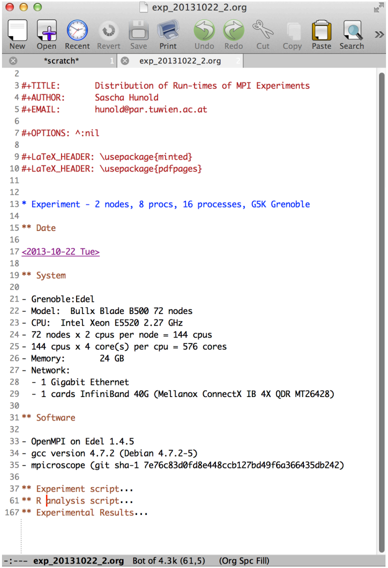


Version Control Systems

- Git (or hg, bzr)
- very useful for tracking code, input and output data
- SHA-1 keys as unique identifiers
- keep code and experiments in one repository
  - create new branch for each experiment
  - direct link between "observed program", setup scripts, data analysis component

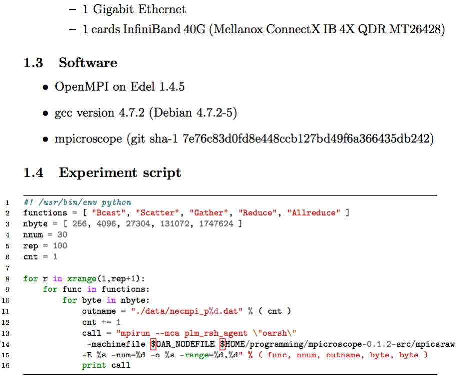


My Take on Reproducible Experiments



```
#+TITLE:      Distribution of Run-times of MPI Experiments
#+AUTHOR:     Sascha Hunold
#+EMAIL:      hunold@par.tuwien.ac.at

#+OPTIONS: ^:nil

#+LaTeX_HEADER: \usepackage{minted}
#+LaTeX_HEADER: \usepackage{pdfpages}


* Experiment - 2 nodes, 8 procs, 16 processes, G5K Grenoble

** Date

<2013-10-22 Tue>

** System

- Grenoble:Edel
- Model:  Bullx Blade B500 72 nodes
- CPU:  Intel Xeon E5520 2.27 GHz
- 72 nodes x 2 cpus per node = 144 cpus
- 144 cpus x 4 core(s) per cpu = 576 cores
- Memory:       24 GB
- Network:
  - 1 Gigabit Ethernet
  - 1 cards InfiniBand 40G (Mellanox ConnectX IB 4X QDR MT26428)

** Software

- OpenMPI on Edel 1.4.5
- gcc version 4.7.2 (Debian 4.7.2-5)
- mpicroscope (git sha-1 7e76c83d0fd8e448ccb127bd49f6a366435db242)

** Experiment script...
** R analysis script...
** Experimental Results...
```

```
-:---  exp_20131022_2.org   Bot of 4.3k (61,5)    (Org Spc Fill)
```



- 1 Gigabit Ethernet
- 1 cards InfiniBand 40G (Mellanox ConnectX IB 4X QDR MT26428)

1.3 Software

- OpenMPI on Edel 1.4.5
- gcc version 4.7.2 (Debian 4.7.2-5)
- mpicroscope (git sha-1 7e76c83d0fd8e448ccb127bd49f6a366435db242)

1.4 Experiment script

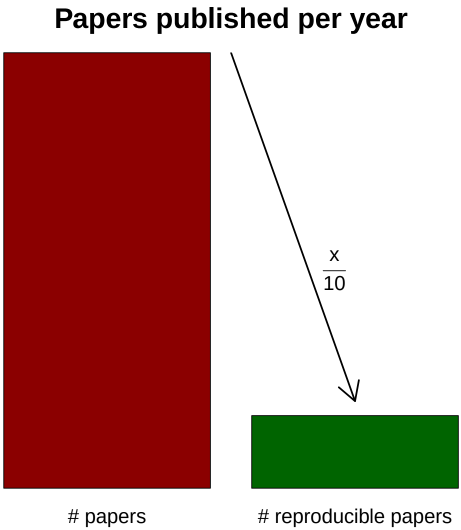
```python
#! /usr/bin/env python
functions = [ "Bcast", "Scatter", "Gather", "Reduce", "Allreduce" ]
nbyte = [ 256, 4096, 27304, 131072, 1747624 ]
nnum = 30
rep = 100
cnt = 1

for r in xrange(1,rep+1):
    for func in functions:
        for byte in nbyte:
            outname = "./data/necmpi_p%d.dat" % ( cnt )
            cnt += 1
            call = "mpirun --mca plm_rsh_agent \"oarsh\"
             -machinefile $OAR_NODEFILE $HOME/programming/mpicroscope-0.1.2-src/mpicsraw
            -E %s -num=%d -o %s -range=%d,%d" % ( func, nnum, outname, byte, byte )
            print call
```

Conclusions

Conclusions

- I believe: better reproducibility of parallel computing experiments is required
- possible steps
  - authors should provide more experimental details
    - input data, analysis scripts
  - provide source code if possible
  - reproduction and verification of results should count as scientific contributions (unless trivial)
  - raise awareness in community

The Consequences



But does paper count matter?

- DFG-Präsident Matthias Kleiner
- "Die Publikationsflut und der alleinige Blick auf den Hirsch-Faktor, den Impact-Faktor und andere numerische Indikatoren schaden der Wissenschaft ungemein. Wer glaubt, mit der Zahl der Veröffentlichungen Forschungsqualität beweisen zu können, übersieht doch, dass es längst einen Trend zur Salamisierung von Veröffentlichungen gibt – ein Scheibchen heute, eines morgen, eins übermorgen, obwohl die Ergebnisse eigentlich zusammengehören."
- source: `http://www.goethe.de/wis/fut/fuw/ftm/de6125555.htm`

DFG guidelines

- We should follow DFG guidelines.
- Memorandum: "Sicherung guter wissenschaftlicher Praxis" (1998, 2013)
  - 2013, ergänzte Auflage, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
  - Empfehlungen der Kommission „Selbstkontrolle in der Wissenschaft“
- "The primary test of a scientific discovery is its reproducibility. The more surprising [..] a finding is held to be, the more important independent replication within the group becomes, prior to communicating it to others outside the group."

DFG guidelines

- We should follow DFG guidelines.
- Memorandum: "Sicherung guter wissenschaftlicher Praxis" (1998, 2013)
  - 2013, ergänzte Auflage, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
  - Empfehlungen der Kommission „Selbstkontrolle in der Wissenschaft“
- "The primary test of a scientific discovery is its reproducibility. The more surprising [..] a finding is held to be, the more important independent replication within the group becomes, prior to communicating it to others outside the group."

It's about care

"Careful quality assurance is essential to scientific honesty."

More on that Topic

- Sascha Hunold and Jesper Larsson Träff.
  On the State and Importance of Reproducible Experimental Research in Parallel Computing.
  CoRR, abs/1308.3648, abs/1308.3648, 2013. [8]
- http://hunoldscience.net

Thank you

References I

[1] D. H. Bailey.
Misleading Performance Reporting in the Supercomputing Field.
Scientific Programming, 1(2):141–151, 1992.

[2] B. Bauer et al.
The ALPS project release 2.0: open source software for strongly correlated systems.
Journal of Statistical Mechanics: Theory and Experiment, 2011(05):P05001, 2011.

[3] A. Casadevall and F. C. Fang.
Reproducible Science.
Infection and Immunity, 78(12):4972–4975, Nov. 2010.

[4] J. Demmel and H. D. Nguyen.
Numerical Reproducibility and Accuracy at ExaScale.
In *IEEE Symposium on Computer Arithmetic*, pages 235–237, 2013.

[5] C. Drummond.
Reproducible Research: a Dissenting Opinion, 2012.
http://cogprints.org/8675/.

[6] D. Fanelli.
How Many Scientists Fabricate and Falsify Research? A Systematic Review and Meta-Analysis of Survey Data.
PLoS ONE, 4(5):e5738, May 2009.

References II

[7] C. Goble, D. De Roure, and S. Bechhofer.
Accelerating Scientists' Knowledge Turns.
In *Proceedings of IC3K*, pages 1–23, May 2012.

[8] S. Hunold and J. L. Träff.
On the State and Importance of Reproducible Experimental Research in Parallel Computing.
CoRR, abs/1308.3648, abs/1308.3648, 2013.

[9] T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney.
Producing wrong data without doing anything obviously wrong!
In M. L. Soffa and M. J. Irwin, editors, *ASPLOS*, pages 265–276. ACM, 2009.

[10] R. D. Peng.
Reproducible Research in Computational Science.
Science, 334(6060):1226–1227, Dec. 2011.

[11] V. Stodden.
The Legal Framework for Reproducible Scientific Research: Licensing and Copyright.
Computing in Science & Engineering, 11(1):35–40, 2009.

[12] V. Stodden.
Trust Your Science? Open Your Data and Code.
Amstat News, 409:21–22, July 2011.

References III

[13] V. Stodden.
Disseminating reproducible computational research:tools, innovations, and best practices.
http://www.stanford.edu/ vcs/talks/CSE-Feb282013-STODDEN.pdf, feb 2013.

[14] W. F. Tichy.
Should computer scientists experiment more?
Computer, 31(5):32–40, 1998.

[15] R. C. Whaley and J. J. Dongarra.
Automatically tuned linear algebra software.
In *SC*, page 38. IEEE, 1998.