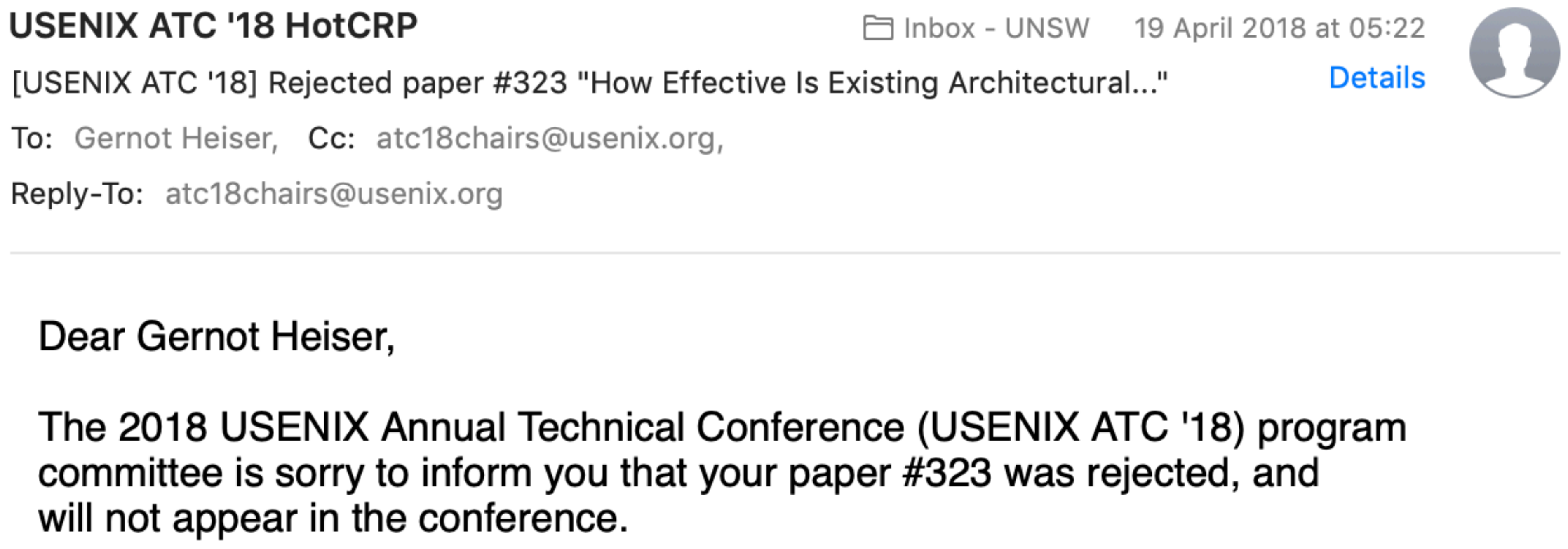
Doing Great Systems Research

… and Convincing Others

What I wish my advisor had taught me

Gernot Heiser
gernot@unsw.edu.au
@GernotHeiser


USENIX ATC '18 HotCRP

Inbox - UNSW 19 April 2018 at 05:22

[USENIX ATC '18] Rejected paper #323 "How Effective Is Existing Architectural..."

Details

To: Gernot Heiser, Cc: atc18chairs@usenix.org,

Reply-To: atc18chairs@usenix.org

Dear Gernot Heiser,

The 2018 USENIX Annual Technical Conference (USENIX ATC '18) program committee is sorry to inform you that your paper #323 was rejected, and will not appear in the conference.

Rejection is Part of Life

My 2013 stats (2nd-best year ever!):

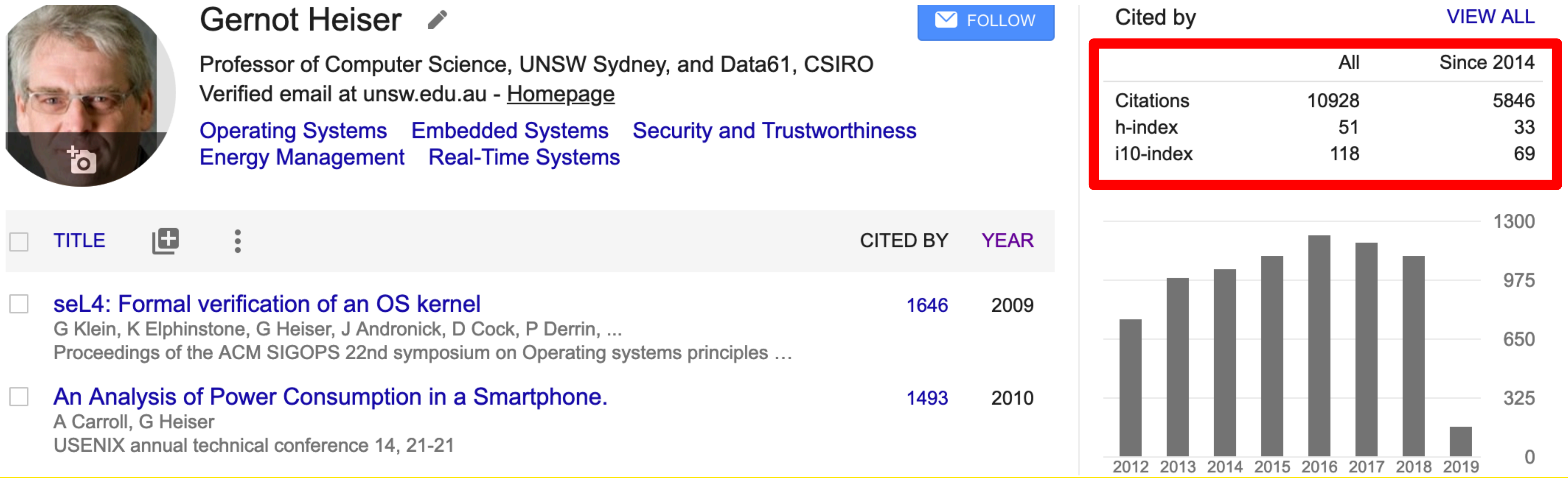
- 7 tier-1: EuroSys, SIGMOD, SOSP, OOPSLA, 2*RTAS, TOCS
- 4 workshops: HotOS, APSys, PLOS, HotPower
- 8 rejects: 2 × ATC, PLDI, 2 × RTSS, APSys, EMSOFT, RTAS

My 2017 stats (a bad year):

- 1 tier-1: EuroSys (paper previously rejected 5 times!)
- 2 workshops: PLOS, APSys; 1 magazine (invited): IEEE Design & Test
- 7 rejects: Usenix Security, 2 × IEEE S&P, RTAS, ASPLOS, 2 × SOSP

Qualifications?

- Served on all top-tiers PC, at least one each year
- Fellow ACM, Fellow IEEE, Fellow Academy of Technology & Engineering


Gernot Heiser

Professor of Computer Science, UNSW Sydney, and Data61, CSIRO
Verified email at unsw.edu.au - Homepage

Operating Systems   Embedded Systems   Security and Trustworthiness
Energy Management   Real-Time Systems

TITLE	CITED BY	YEAR
seL4: Formal verification of an OS kernel G Klein, K Elphinstone, G Heiser, J Andronick, D Cock, P Derrin, ... Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles …	1646	2009
An Analysis of Power Consumption in a Smartphone. A Carroll, G Heiser USENIX annual technical conference 14, 21-21	1493	2010

Cited by   VIEW ALL

	All	Since 2014
Citations	10928	5846
h-index	51	33
i10-index	118	69

Ways To Succeed

Easy way: Aim low:

- Solve easy, incremental problems
- 2nd/3rd-tier venues are easy to publish in
- *Guaranteed impact-free*

Hard way: Aim high:

- Solve real problems convincingly
- Write excellent papers with significant contribution
- Publish in top venues

THIS ADVICE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

What Is Systems?

(Overly?) simplified view of Computer science: theory + systems

Theorists build theories, models

- They often get away with theories not good for anything

Systems folks build stuff – engineering focussed

- They don't get away with work not good for anything!

Systems Research:

- Solving
- Real
- Problems!

Good Test: The Elevator Pitch

1. What is the problem you are solving?
2. Why does it matter? Who Cares?
3. What is the approach you're taking? What's innovative about it?
4. What's have you achieved / are expecting to achieve? How will it matter?

If you cannot give concise and convincing answers, consider doing something else!

Convincing Others: Rules of Writing

Rule 1: Reviewers are Pot Luck

- Even at top conferences, some good papers get rejected
  - Sometimes for the wrong reason, but usually you have to blame yourself!
- Reviewers' top reasons for rejection
  - I'm not convinced you're solving a *real problem*
  - I'm not convinced you're *solving* the problem
  - *I don't understand* – your paper is too badly written
  - Too incremental for {SOSP, OSDI, EuroSys…}
- Papers without a PC champion have a hard stand
  - If you excite one reviewer, you may get in despite several negative reviews
  - What is cool about your work????

Rule 2: A Good Paper Has a Story

1. The paper has a (one!) main message
   - *Understand clearly what the message is*
   - Make sure that the reader gets it
   - Make sure it's an interesting one

2. A paper has a narrative
   - It starts from zero and then works on transmitting the message
   - *Everything* you write must support the message
   - *Maintain reader state!*
     - Be conscious of what the reader knows/remembers
     - Like DRAM, human memories are lossy, need refresh

Rule 3: Limited Space: The Two “C”s

Be *clear* (at all levels)

- Every sentence, paragraph, section has a clear purpose
- The purpose is clearly communicated
- The overall message is consistent

Be *concise* (brief but complete)

- Don’t waffle!!! (Use “Jay’s rule of thumb”)
- Be precise
- Make sure it’s readable, lucid, enjoyable

But: Maintain reader state:

- Define before use
- Be aware of what the reader has learned
- Refresh as appropriate
- Ensure it’s self-contained!

Rule 4: Presentation Matters!

Top conferences accept two kinds of papers
1. Excellent work that is well-presented
2. Average work that is well-presented

The best work is will fail if you can't convince the reviewers
- Reviewers are busy, may have to review 30 papers in 6 weeks
- They'll look for reasons to reject – don't give them any!

Presentation Matters: Paper Engineering

- Be clear about the idea, the significance and the approach
- Proceed top-down, not bottom-up
- Maintain reader state & argue convincingly
- Build tension, keep reader interested, but avoid surprises
- Evaluate convincingly: thorough and honestly
- *Be up-front about assumptions and limitations!*

Paper Engineering: Introduction

The Overture:

- Explain the problem you're solving, why it's a problem, use an example
- Outline your approach
- Indicate results/outcomes
- Explicitly state contributions
- "Paper roadmap" is a waste of space (use forward pointers in contributions list)

General hints for Introduction:

- Capture the reader's interest: sell your idea
- Be concise: Stay within about one page!
- Make sure the paper delivers what you promise
- Reviewers kill for "bait and switch"

But don't expect too much background!

Paper Engineering: Other Parts

- Background: set the scene in more detail
  - Cite related work as needed, don't discuss more than necessary
  - Examples!!!!
- Describe problem in detail
- Explain solution in detail
  - *Be honest & up-front about limitations and assumptions*
  - Design, then implementation
- Evaluation: for systems work often largest part
- Related work
- Conclusions

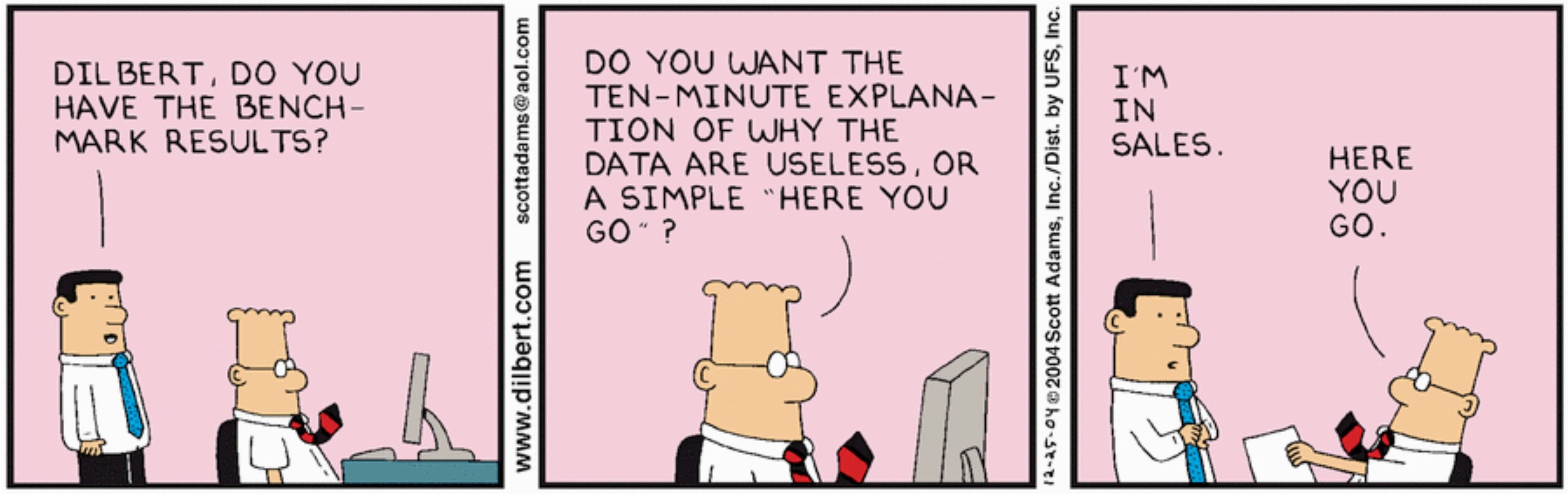
Abstract:

- Steer paper to right reviewers
- What, Why, Achievement, Implication: 1 sentence each!
- Redo for camera-ready!

Evaluation: Where the Rubber Hits The Road

Show that your solution is useful

- *Progressive*: significant improvements in important situations
- *Conservative*: no (significant) degradation elsewhere


DILBERT, DO YOU HAVE THE BENCH-MARK RESULTS?
scottadams@aol.com
www.dilbert.com
DO YOU WANT THE TEN-MINUTE EXPLANA-TION OF WHY THE DATA ARE USELESS, OR A SIMPLE "HERE YOU GO"?
12-25-04 © 2004 Scott Adams, Inc./Dist. by UFS, Inc.
I'M IN SALES.
HERE YOU GO.

Benchmarking Crimes (Greatest Hits)

1. Selective Benchmarking – cherry picking
2. Only micro-benchmarks
3. Throughput degradation = overhead
4. Creative overhead accounting: 10% → 20% overhead is “10% increase”
5. Improper baseline, only relative figures, compare against self
6. No indication of significance

Full list: http://gernot-heiser.org/benchmarking-crimes.html

Paper Engineering: Style

Write in engaging style, lead reader though the paper

- Avoid bottom-up structure, *present ideas top-down*
- *Use active voice!!!!* … and present tense
- Avoid buzzwords (“novel”, “mobile social post-quantum fog computing”)

Be mindful of reader’s brain state (which is lossy)

- *Maintain reader state*
- Don’t assume every reviewer is expert in your narrow area
- But don’t think you can hide stuff from reviewers!

Paper Engineering: Form

Follow formatting rules

- Don’t play with margin, baseline skip etc
- Don’t use microscopic fonts, >40y olds have problems with <8pt font
- Space cheating is dishonest – why should I still believe you?

Spell-check, proof-read, proof-read

- Get native speaker to proof-read if you aren’t
- Get outsider to read it – great way to spot holes before it’s too late!

Mechanics

- Don’t use MS Word – MSR people use LaTeX, so should you!
  - doesn’t integrate well with revision control
  - forces coarse-grain locking, limits concurrency
  - references are painful, formulae even more so
- Use revision control (git), especially (but not only) when it’s a joint paper
  - Alternatively: LaTeX source as a Google doc [suggestion John Wilkes]
- Use BibTeX – but use it correctly (eg capitalisation in titles)
- Use scriptable tools (eg GNUplot) for graphing results
  - Results change frequently and at the last minute
  - Being able to run from command line/make is essential

Summary

- Clear statement of problem
- Why would I care?
- Convincing solution, compelling argument
- Thorough evaluation, no BM crimes
- Lucid writing, maintaining reader state

Further Reading

Writing systems papers:

- Levin & Redell: An evaluation of the 9th SOSP submissions, or How (and how not) to write a good systems paper
- Simon Peyton Jones (MSRC): How to write a great research paper
  - http://research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/giving-a-talk-slides.pdf
- My paper/thesis writing guide: http://gernot-heiser.org/style-guide.html

General writing/style etc (recommended by systems folks):

- Zobel: Writing for computer science, Springer
- Strunk & White: The elements of style, Allyn & Bacon
- Dupré: Bugs in writing: A guide to debugging your prose, Addison-Wesley