How to Write a Good Systems Paper

Gernot Heiser
NICTA and UNSW

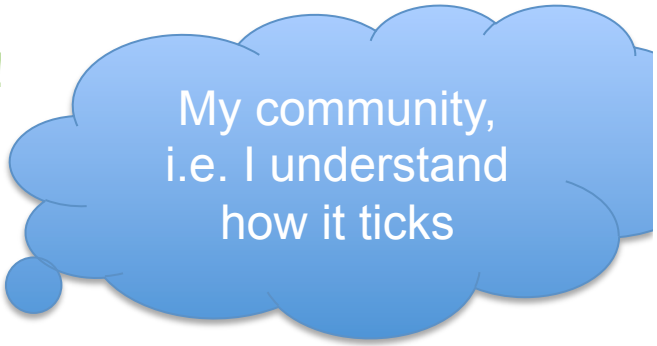
Australian Government

Bootcamp'14

What is “Systems”?

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

- Theorists build theories, models
  - often get away with theories not good for anything
- Systems folks build stuff
  - don’t get away with work not good for anything!



Examples of “systems” work:

- operating systems
- network systems / distributed systems
- database systems
- programming systems (PL implementation)
- machine-learning systems
- …

Disclaimers

1. This is about good papers, not exciting talks slides
   - Not my strength 😉

2. I've been around the traps longer than you, but I don't know it all!
   - I get papers rejected just as you do
   - 2013 stats (a very good year!):
     - 11 accepts:
       - 6 conferences: EuroSys, SIGMOD, SOSP, OOPSLA, 2*RTAS
       - 4 workshops: HotOS, APSys, PLOS, HotPower,
       - 1 journal: TOCS (plus TODS invite)
     - 8 rejects: 2*Usenix, PLDI, 2*RTSS, APSys, EMSOFT, RTAS

3. There are plenty of other resources addressing similar issues
   - Examples at the end

RULES OF WRITING

Rule 1: Reviewers are Pot Luck

- … even at top conferences
  - even good papers get rejected, sometimes for the wrong reasons
- Rejection is part of life, get used to it!
  - Don't blame the reviewers, it usually means you didn't do your job!
- 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
  - Your paper is just not competitive for {SOSP, OSDI, EuroSys…}
- Papers without a PC "champion" have a hard stand
  - Make sure there's something which at least one reviewer will think cool
  - Purely incremental work will have a hard stand at top venues

Rule 2: A 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

Rule 3: Limited Real Estate: 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
  - recall/remind if necessary

Rule 4: Presentation Matters – Paper Engineering

The best work is useless if you can't convince the reviewers

- reviewers are busy, may have to review 30 papers
- they'll look for reasons to reject – don't give them any!

Important bits:

- Introduction: sell the idea, the significance and the approach
- build tension, make reader interested
- convincing argumentation
- top-down, not bottom-up
- maintain reader state
- convincing evaluation
  - thorough and honest
- *state assumption/limitations honestly*

PAPER STRUCTURE AND STYLE

Introduction: Most Important Part of the Paper!

The Overture:

- Explain the problem you're solving
- Outline your approach
- Indicate results/outcomes
- State contributions

General hints for intro:

- 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 "bate and switch"

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 and forthcoming with limitations and assumptions
- Evaluation: often largest part
- Related work
- Conclusions
- Abstract
  - used to steer to the right reviewers
  - What, Why, Achievement, Implication
  - IMPORTANT: Redo for camera-ready!

Evaluation

- Show that your solution actually works
  - *Progressive*: significant improvements in important situations
  - *Conservative*: no (significant) degradation elsewhere

  Need both!

- More on this later

Style and Form

- Write in engaging style, lead reader though the paper
  - avoid bottom-up structure, present ideas top-down
  - follow style rules
  - *Use active voice!!!!*
  - Avoid buzzwords (“novel”, “mobile social supercomputing in cloud”)
- 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!
- Follow formatting rules
  - don’t play with margin, baseline skip etc
  - don’t use microscopic fonts, >40y olds have problems with <8pt font
- 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

- Use revision control
  - especially (but not only) when it's a joint paper
- Don't use MS Word
  - doesn't integrate well with revision control
  - requires coarse-grain locking
  - references are painful, formulae even more so
  - MSR people use LaTeX, so should you!
- Use BibTeX
  - … but use it correctly

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