NTNU – Trondheim
Norwegian University of
Science and Technology

How to Write a Computer Architecture Paper

TDT4260 Computer Architecture
25. February 2013

Magnus Jahre

How does pfJudge work?

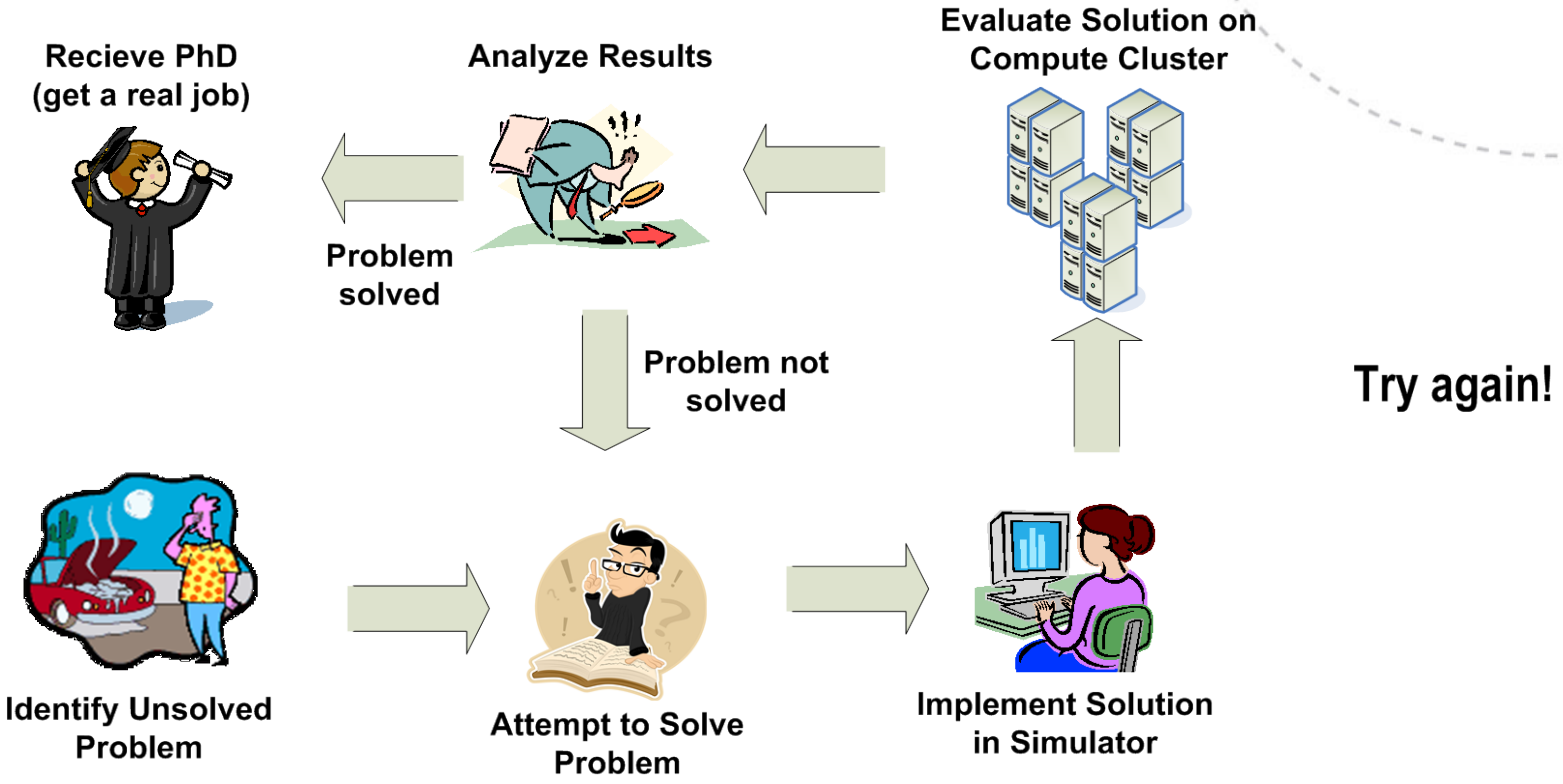
- Each submitted file is one kongull job
  - Contains 12 M5 instances since there are 12 CPUs per core
  - Each M5 instance runs a different SPEC 2000 benchmark
- The kongull job added to the job queue
  - Status “Running” can mean running or queued, be patient
  - Running a job can take a long time depending on load
  - Kongull is usually able to empty the queue during the night
- You all have a regular user account on Kongull
  - Remember that Kongull is a shared resource!
  - Always calculate the expected CPU-hour demand of your experiment before submitting

Storage Estimation

- We impose an storage limit of 8KB on your prefetchers
  - This limit is not checked by the exercise system
- This is realistic: hardware components are usually designed with an area budget in mind
- Estimating storage is simple
  - Table based prefetcher: add up the bits used in each entry and multiply by the number of entries

HOW TO USE A SIMULATOR

Research Workflow



Why simulate?

- Model of a system
  - Model the interesting parts with high accuracy
  - Model the rest of the system with **sufficient** accuracy
- *"All models are wrong but some are useful"*
  (G. Box, 1979)
- The model does not necessarily have a one-to-one correspondence with the actual hardware
  - Try to model **behavior**
  - Simplify your code wherever possible

Know your model

- You need to figure out which system is being modeled!
- Pfsys is a help to getting started, but to draw conclusions from you work you need to understand what you are modeling
- Reproducibility: Other people should be able to rerun you experiment and get the same results

Reproducible?

	Crossbar Based Architecture			Ring Based Architecture		
	4-core	8-core	16-core	4-core	8-core	16-core
ITRS Year of Production	2007	2010	2013	2007	2010	2013
Feature Size (nm)	65	45	32	65	45	32
Shared Cache Size (MB)	8	16	32	8	16	32
Memory Bus Channels	1, 2 or 4	1, 2 or 4	1, 2 or 4	1, 2 or 4	1, 2 or 4	1, 2 or 4
Interconnect Latency (End-to-End/Per Hop)	8/-	16/-	30/-	-/4	-/4	-/8

Table III
CACHE PARAMETERS

Cache	Size (4-core/8-core/16-core)	Associativity	Access Latency (cycles)	Cycle Time (cycles)	MSHRs / WB (per bank)	Banks	Area (mm^2)
Level 1 Private Cache	64KB	2	3/2/2	2	16MSHRs/4WB	1	2.3/1.1/0.5
Level 2 Private Cache	1 MB	4	9/6/5	4/3/2	16	1	14.6/7.0/3.6
Level 2/3 Shared Cache	8/16/32 MB	16	16/12/12	4	16/32/64	4	94.0/91.9/84.7

Table IV
PROCESSOR CORE PARAMETERS

Parameter	Value
Clock frequency	4 GHz
Reorder Buffer	128 entries
Store Buffer	32 entries
Instruction Queue	64 instructions
Instruction Fetch Queue	32 entries
Load/Store Queue	32 instructions
Issue Width	4 instructions/cycle
Functional units	4 Integer ALUs, 2 Integer Multipy/Divide, 4 FP ALUs, 2 FP Multiply/Divide
Branch predictor	Hybrid, 2048 local history registers, 4-way 2048 entry BTB

Table V
INTERCONNECT AND DRAM INTERFACE

Parameter	Value
Crossbar Interconnect	8/16/30 cycles end-to-end transfer latency, 32 entry request queue, Pipelined (2/4/6 pipe stages)
Ring Interconnect	4/4/8 cycles per hop transfer latency, 1/1/2 pipe stages per hop, 32 entry request queue, 1/2/2 request rings, 1 response ring
Point to Point Link	4/3/2 transfer latency, 32 entry request queue
Main memory	DDR2-800, 4-4-4-12 timing, 64 entry read queue, 64 entry write queue, 1 KB pages, 8 banks, FR-FCFS scheduling [21], Closed page policy

NTNU – Trondheim
Norwegian University of Science and Technology

From: *A Quantitative Study of Memory System Interference in Chip Multiprocessors, Jahre et al., HPCC09*

HOW TO WRITE A PAPER

Find Your Story

- A good computer architecture paper tells a story
  - All good stories have a **bad guy**: the problem
  - All good stories have a **hero**: the scheme
- Writing a good paper is all about finding **your story**
- Note: This story has to be told within the strict structure of a scientific article

Paper Format

- You will be pressed for space
- Try to say things as precisely as possible
  - Your first write-up can be as much as 3x the page limit and it's still easy (possible) to get it under the limit
- Think about your plots/figures
  - A good plot/figure gives a lot of information
  - Is this figure the best way of conveying this idea?
  - Is this plot the best way for visualizing this data?
  - Plots/figures need to be area efficient (but readable!)

Typical Paper Outline

- Abstract
- Introduction
- Background/Related Work
- The Scheme (substitute with a descriptive title)
- Methodology
- Results
- Discussion
- (Related Work)*
- Conclusion (with optional further work)

**If Section 2 is Background and you have more papers to reference*

Abstract

- An experienced reader should be able to understand exactly what you have done from only reading the abstract
  - This is different from a **summary**
- Should be short, varies from 150 to 200 word maximum
- Should include a description of the problem, the solution and the main results
- Typically the last thing you write

Introduction

- Introduces the larger research area that the paper is a part of
- Introduces the problem at hand
- Explains the scheme
- Level of abstraction: “20 000 feet”

Related Work

- Reference the work that other researchers have done that is related to your scheme
- Should be complete (i.e. contain all relevant work)
  - Remember: you define the scope of your work
- Can be split into two sections: Background and Related Work
  - Background is an informative introduction to the field (often section 2)
  - Related work is a very dense section that includes all relevant references (often section n-1)

The Scheme

- Explain your scheme in detail
  - Choose an informative title
- Trick: Add an informative figure that helps explain your scheme
- If your scheme is complex, an informative example may be in order

Methodology

- Explains your experimental setup
- Should answer the following questions:
  - Which simulator did you use?
  - How have you extended the simulator?
  - Which parameters did you use for your simulations? (aim: **reproducibility**)
  - Which benchmarks did you use?
  - Why did you chose these benchmarks?
- Important: should be **realistic**
- If you are unsure about a parameter, run a simulation to check its impact

Results

- Show that your scheme works
- Compare to other schemes that do the same thing
  - Hopefully you are better, but you need to compare anyway
- Trick: “Oracle Scheme”
  - Uses “perfect” information to create an upper bound on the performance of a class of schemes
  - Prefetching: Best case is that all L2 accesses are hits
- Sensitivity analysis
  - Check the impact of model assumptions on your scheme

Discussion

- Only include this if you need it
- Can be used if:
  - You have weaknesses in your model that you have not accounted for
  - You tested improvements to your scheme that did not give good enough results to be included in “The Scheme” section

Conclusion

- Repeat the main results of your work
- Remember that the abstract, introduction and conclusion are usually read before the rest of the paper
- Can include Further Work:
  - Things you thought about doing that you did not have time to do