How to Have a Bad Career as a Graduate Student

Christos Kozyrakis

christos.kozyrakis@epfl.ch

Yet Another Talk with Advice?

- Yes, but the focus is a little different
- Consider your long-term career goals as a researcher
- Make sure your PhD studies serve these goals well

The History of this Talk

- Based on Dave Patterson's advice talk
  - How to Have a Bad Career in Industry/Academia
  - Initial version in 1994, targeted to junior faculty

- Modifications
  - Removed advice for junior faculty (too early for you)
  - Added some advice of my own
    - Picking advisor/thesis topic, surviving large projects, …

Who Am I?

CS PhD, **UC Berkeley** (processor design)

IC professor, EPFL (?)

EE & CS professor, **Stanford**
(architecture, security, programming models, datacenter management, OS, …)

CS degree, **University of Crete**
Research fellow, **FORTH** (networking HW)

Outline

- Advice for a bad career as a graduate student
- Alternatives to a bad graduate career
- Advice on how to survive in a large project
- If there is time
  - Advice for bad papers, presentations, posters
  - Alternatives to bad papers, presentations, posters

This is ***Not*** a Lecture

- This should be a discussion
- Interrupt me for questions & comments
- There is not a single, correct approach to these issues
  - Feel free to disagree or bring up other issues
  - Seek further advice on these topics

Part I

How to Have a Bad Graduate Career

Advice for a Bad Career

- Concentrate on getting good grades
  - Postpone research involvement; may lower grades
- Minimize number and flavors of courses
  - Why take advantage of one of the top CS departments with a wide range of excellent graduate courses?
  - Why take advantage of one of the best universities in the world?
  - It will affect your grades

Advice for a Bad Career

- Concentrate on graduating as fast as possible
  - Winner is first in class to PhD
  - People only care that you have a PhD & your grades
    - Nobody cares about what you actually know
    - Nirvana: graduating in 3 years with perfect grades
  - Don't spend a summer in industry: takes longer
    - How could industry experience help with selecting PhD topic?
  - Don't work on large projects; takes longer
    - Have to talk to others, have to learn different areas
    - Synchronization overhead of multiple people
  - Don't do a systems PhD; takes longer

Advice for a Bad Career

- Don't trust your advisor
  - Advisors are only interested in their own career, not yours
  - Advisor may try to mentor you, use up time, interfere with grades
- Only work the number of hours/week you are paid for
  - Even less if possible…
  - Don't let the master class exploit the workers!

Advice for a Bad Career

- Work in the specific area you applied for
  - E.g. if you applied for networking, stay in networking!
  - Your application is a binding contract
  - Leverage your previous experience, graduate faster!

- Never read on your own
  - What's the advisor for if you have to pick your own readings?
  - May lead to extra work, interaction with others, …
  - Takes time!

Advice for a Bad Career

- Do not publish any papers
  - Takes time
  - People should just read your thesis at the very end
  - What's the advisor for if you must get feedback from others?

- Alternatively: publish 100s of papers
  - PhDs are ranked by publication counts; don't be left behind
  - There is a reason we have hundreds of journals/workshops

Advice for a Bad Career

- Don't go to conferences
  - It costs money and takes time
  - You'll have plenty of time to learn the field after graduating
- Don't waste time polishing papers or talks
  - Again, that takes time

Advice for a Bad Career

- Don’t worry at all about research funding
  - It's your advisor's job to keep you “in the money”
  - Why waste any time with proposals, fellowship apps, …?
  - Fellowships just make it difficult to write a 2-page CV

- Worry a lot about research funding
  - If you don’t have a fellowship, you are a 2nd class citizen

Part II

Alternatives to a Bad Graduate Career

Alternatives to a Bad Career

- Concentrate on getting good grades?
  - Reality: need to maintain reasonable grades
    - For minimum school/department requirements
  - What matters on graduation is
    - Recommendation from 3-4 faculty/PhDs who have known you well
    - (Expected) impact of your research

- Minimize number and flavors of courses?
  - Last chance to be exposed to new ideas before have to learn on your own
    - Re: compilers and me
  - Search for interesting courses beyond your area/department
    - Stats, bioengineering, psychology, IP law, technology management, …

Alternatives to a Bad Career

- Concentrate on graduating as fast as possible?
  - Last chance to learn in an environment optimized for learning
    - Learning inside & outside of the classroom
  - What matters on graduation is
    - Recommendation from 3-4 faculty/PhDs who have known you well
    - (Expected) impact of your research
  - Considered newly "minted" when you finish PhD
    - Judged on year of PhD vs. year of birth
    - To a person in their 40s or 50s, 1-2 more years is round-off error

      27 ~= 29

Alternatives to a Bad Career

- Don't trust your advisor?
  - Primary attraction of campus vs. industry is working with grad students
  - Faculty career is judged in large part by success of students
  - Try taking advice of advisor!

- Only work number of hours per week you are paid for?
  - Faculty average is 65-70 hours/week
    - Students should be in that range
  - Organize each day: when most alert? nap? exercise? sleep?
    - Track when/how often/how long: write, read, program, email?
  - To-do lists: daily, weekly, semester
  - **Work hard, play hard**

Alternatives to a Bad Career

- Never switch field?
  - Technology changes, opportunities change, interests change…
  - Explore opportunities across the department
  - Use classes and class projects to experiment with other fields
  - It gets more difficult to switch fields later on…

- Never read on your own?
  - Your advisor does not know everything (surprise!)
  - Read on your own in directions you find interesting
  - Teach your advisor about your findings

Alternatives to a Bad Career

- Do not publish any papers or publish 100s of papers?
  - Papers: an opportunity to get feedback from experts
  - Publish reasonably significant results to the right forum
    - Top journals or conferences
    - Workshops were leading experts brainstorm new ideas
  - You will be judged on impact, not on publication count

Alternatives to a Bad Career

- Don't go to conferences?
  - Chance to see firsthand what the field is like, where its going
  - Talk to people in the field in the halls!
  - If your faculty advisor won't pay, then pay it yourself; there are student rates, you can share a room

- Don't waste time polishing writing or talks?
  - In the marketplace of ideas, the more polish the more likely people will pay attention to your ideas
  - Practice presentation AND answering tough questions

Alternatives to a Bad Career

- Don't participate in a large project?
  - Do it, if your find an interesting project
  - See later part of the talk
- Funding?
  - Help your advisor secure funding
    - Learning how to present research to sponsors is a useful skill
    - A fellowship in your CV looks very good so apply if you can…
- Industrial experience?
  - 1st or 2nd summer get work experience, or 1 semester off

The Highlights (1)

- Swim or sink
  - Success is determined primarily by you!
  - Faculty set up opportunity, up to you to leverage it
  - Show initiative!

- Read/learn on your own
  - Fast moving field, don't expect Prof to be at forefront everywhere
  - Read papers, got to seminars/conferences, talk to colleagues
  - Once you know about something, teach your advisor…

The Highlights (2)

- Ask questions
  - Lots of smart people in grad school, but don't be intimidated
  - They will teach you something or will learn by figuring it out together
  - It's OK to ask a "stupid question" every now and then

- Be honest about your work
  - Your reputation as a researcher is valuable asset
  - Be honest about your results, progress, and promises

Choosing an Advisor

- Very important decision , don't have to rush into it
- Tips
  - Get to know your potential advisor
    - Talk to advisor, her students, attend their group meetings
  - A good advisor for you is the one that
    - Has the right project or works on right area
    - Has time for you (not too many students)
    - Has the right research style
    - Can provide you with funding
    - Your characters are compatible
  - Shop around; you can start on a trial basis
    - Class projects are a good way to do this
- You can switch advisors, but it's usually expensive…

You can have >1 Advisor

- Co-advising for interdisciplinary work
  - E.g. one advisor in OS and one in architecture
  - Make sure one of them is listed as your primary advisor
- Build a network of advisors
  - Faculty, older students, and industry colleagues
  - Different people can advise on different topics
  - You can also learn a lot from the differences

Working with your Advisor

- Your advisor is your academic parent
  - Most of what you learned growing up applies
- Tips
  - Talk with your advisor
    - What you are working on and how, needs and worries (short/long term)
  - Don't be afraid to disagree with her
  - If you keep your advisor happy, she will keep you happy!
- Watch out for
  - Advisors are also humans
    - They can make mistakes or be harsh, especially under lots of pressure
  - Computer engineers/scientists are not known for their social skill ☹
  - Don't lose the respect/trust of your advisor!

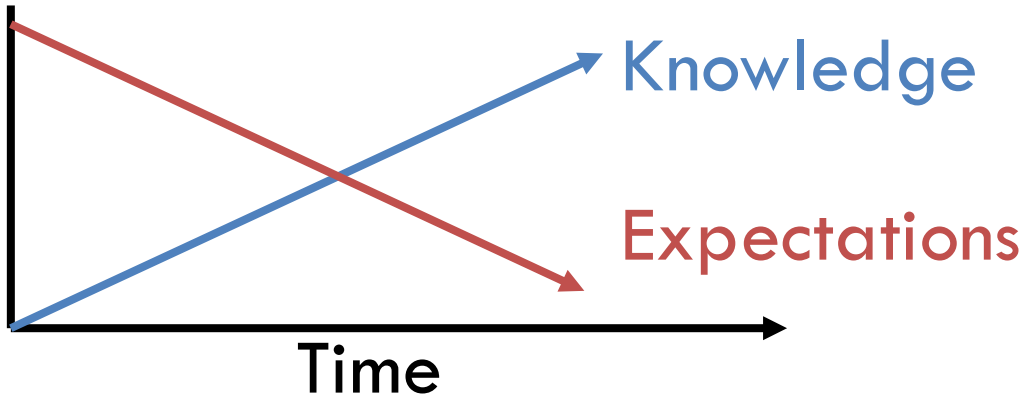
Remember that…

Picking a Thesis Topic

- Selecting the problem is as important as the research on it
  - You don't have to select it now
  - Get started with something fun soon!
  - Good research training (techniques, important problems, …)
  - Thesis topic will be clear in ~2-3 years (hopefully)
    - Don't be surprised if it is very different that initial plan…
- You have to like your research topic
  - Only way to keep sane during the "dark periods"
- Make sure that your problem has intermediate milestones
  - Good for publishing papers
  - Good for checking progress & feasibility

From Research to Paper(s) in Few Easy Steps

When to Graduate?



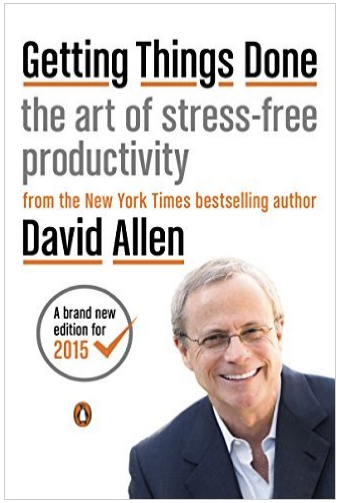
Good Readings

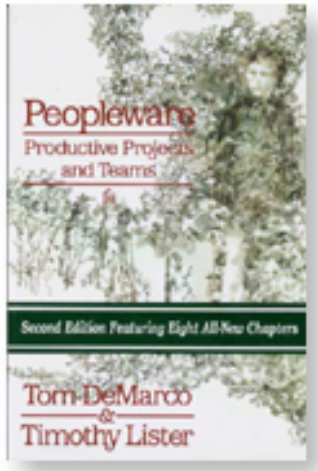
- "Technology And Courage", Ivan Sutherland (Sun)
  - Be bold; take chances on hard topics
  - http://research.sun.com/techrep/Perspectives/smli_ps-1.pdf

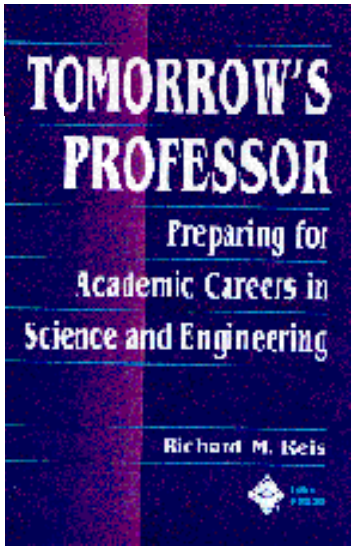
- "You & Your Research", Richard Hamming (Bell Labs)
  - Why do so few researchers make significant contributions?
  - http://www.cs.wisc.edu/~remzi/Postscript/hamming.ps

Good Readings

- "Getting Things Done", David Allen
  - From angst to action
- "Tomorrow's Professors", Richard Reis (Stanford)
  - Great advice on Grad School & Future Career
  - Not limited to academics
- "Peopleware", Demarco & Lister
  - On teams, projects, & management
  - "Is your advisor a bad manager?"
  - Very entertaining…
- "The Task of a Referee", A. Smith
  - How to read & evaluate research papers

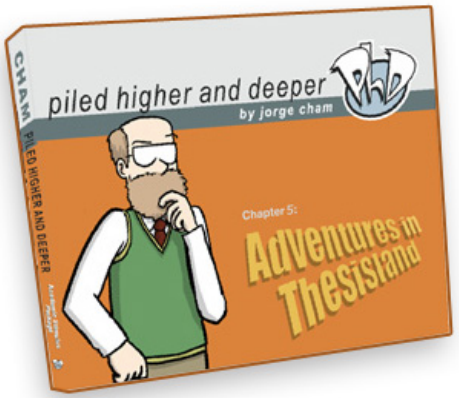


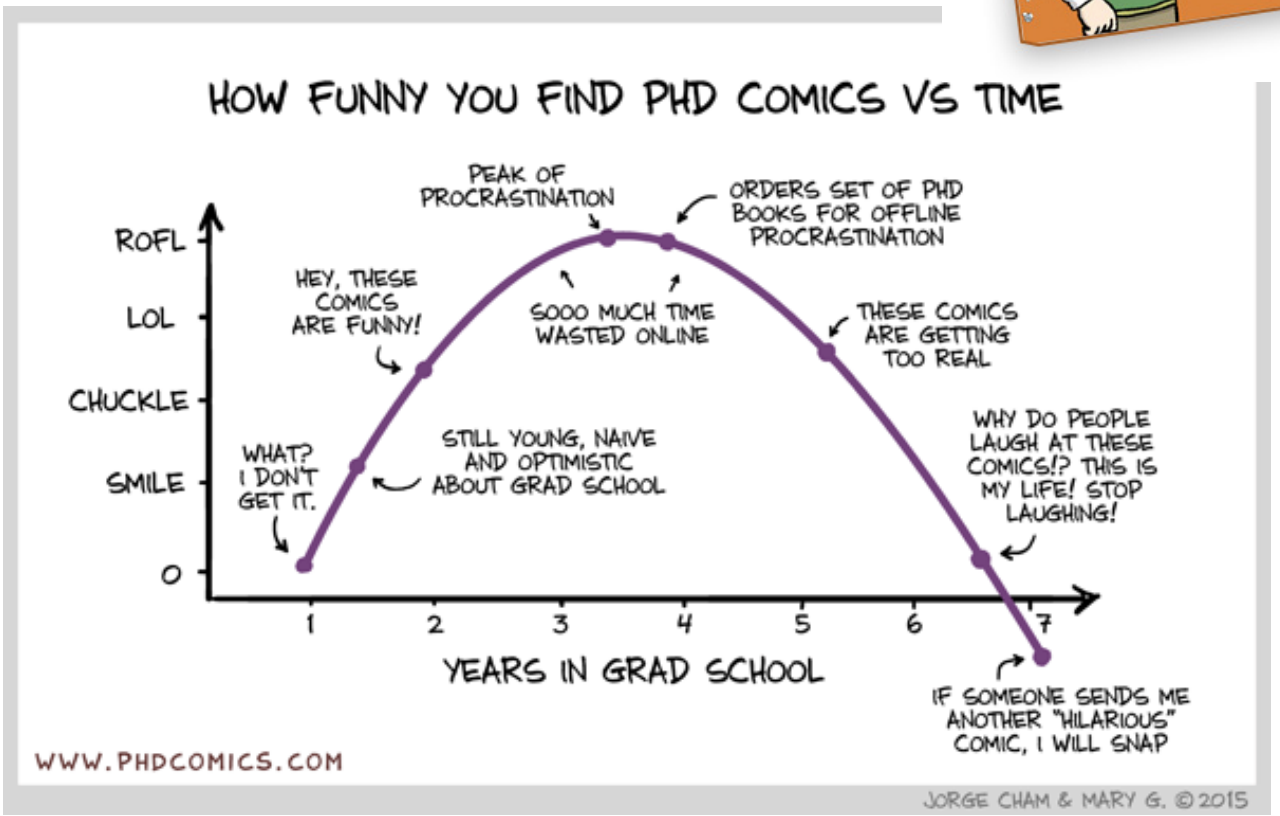




Good Readings

- PhD comics
  - http://phdcomics.com/comics.php





Part III

How to Survive a Large (Systems) Project

Advantages of Large Projects

- Get exposed to interdisciplinary research
  - Multiple fields: architecture, design, compilation, …
  - You become an expert in a couple and knowledgeable in 3-4
- You develop a working system/prototype
  - Larger and faster impact, fame
  - Learn what it take to get something to really work
- Intellectual stimulation from many people
  - Faculty and students
- Learn to work in a team
  - That's how the "real world" works
  - Make life-long friends & partners
- Both industry & universities look for these qualities when they hire

Disadvantages of Large Projects?

- Long implementation period
- Overhead of group interaction
- Identifying yourself within a large group

<u>My opinion and experience:</u>

- The benefits greatly outweigh the disadvantages
- Do it, if you find a project that you like
- Key to surviving a large project:
  - Show initiative
  - Keep a positive attitude and be flexible
  - Understand the stages the project will go through
    - Your are neither the first, nor the last one to go through this

1. The BrainStorming Stage

- What's going on
  - All get to contribute to shaping the "new idea"
  - "New idea" papers based on initial studies on potential
  - Hot topic of discussion at conferences; a lot of excitement
- How you feel
  - "We will change the world"
  - "My famous advisor listens to me"
  - "I got a paper"
- Problems: none!
- My advice
  - Enjoy it & prepare for the next phase

2. The Implementation Stage

- What's going on
  - Start building the prototype/system/infrastructure
  - Everybody in the group has an assigned task now
  - Start realizing the practical issues/limitations of the ideas
  - Few papers (if any)
  - Some newer project is the hot topic at the conferences/department
- How you feel: "I wish I was doing an individual project"
  - "This is not research"
  - "Not a single publication for 2 years now"
  - "There is only 1 thesis in this project for 10 students!"
- My advise: patience
  - Patience
  - Keep notes of what is difficult, could do better, want to change

2b. The Stretch or **Really Dark** Stage

- What's going on
  - Final debugging before HW tape-out or SW release or the 1st period of operating the system
  - A huge amount of work under the pressure of deadlines
- How you feel
  - "How long has it been since the last time I left the lab?"
  - "I just want to make it work and never see it again in my life"
  - "After this, my advisor will ask me to go find a PhD topic!"
- My advise: hold on

3. When it All Comes Together

- What's going on
  - Working prototype/system, recognition of accomplishment by colleagues
  - Tons of data, a lot of interesting answers, many papers
  - Many interesting observations on current/future implementations of various system aspects, new questions and approaches, many theses
  - Individual students follow their own way for the last year(s)
- How you feel
  - "I got a thesis; I will graduate; Everybody wants to hire me"
  - "I know how to make things work"
  - Looking at the prototype: "My baby!!!!"
  - "I don't have enough time to explore all the new ideas/questions"
  - "I got some great friends"

Part III

How to Prepare Bad Papers/Talks/Posters

Writing Commandments for a Bad Career

1. Always make it sound complicated

2. Never define terms, never explain anything

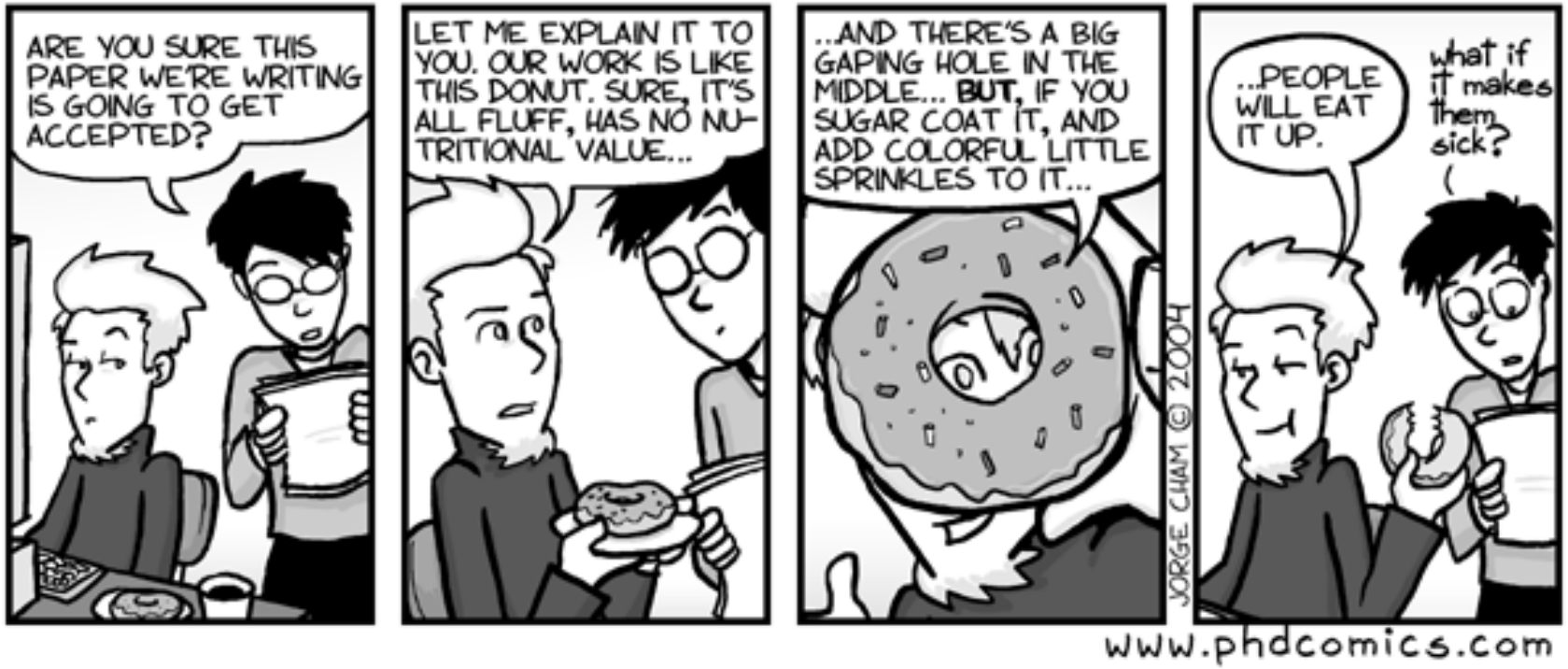
3. Always replace “will do” with “have done”
   - Always publish before you implement

4. Never mention drawbacks of your approach

5. Never reference any papers (besides your own)

6. Never pay attention to the reviewers’ comments

Writing Commandments for a Bad Career



Talk Commandments for a Bad Career

1. Always make it sound complicated
2. Never illustrate
3. Never be brief
4. Never print large
5. Never use color
6. Never skip slides in a long talk
7. Always include all equations & figures from the paper
8. Never, ever, practice

Following the Commandments

- We describe the philosophy and design of the control flow machine, and present the results of detailed simulations of the performance of a single processing element. Each factor is compared with the measured performance of an advanced von Neumann computer running equivalent code. It is shown that the control flow processor compares favorably in the program.

- We present a denotational semantics for a logic program to construct a control flow for the logic program. The control flow is defined as an algebraic manipulator of idempotent substitutions and it virtually reflects the resolution deductions. We also present a bottom-up compilation of medium grain clusters from a fine grain control flow graph. We compare the basic block and the dependence sets algorithms that partition control flow graphs into clusters.

- A hierarchical macro-control-flow computation allows them to exploit the coarse grain parallelism inside a macrotask, such as a subroutine or a loop, hierarchically. We use a hierarchical definition of macrotasks, a parallelism extraction scheme among macrotasks defined inside an upper layer macrotask, and a scheduling scheme which assigns hierarchical macrotasks on hierarchical clusters.

- We apply a parallel simulation scheme to a real problem: the simulation of a control flow architecture, and we compare the performance of this simulator with that of a sequential one. Moreover, we investigate the effect of modeling the application on the performance of the simulator. Our study indicates that parallel simulation can reduce the execution time significantly if appropriate modeling is used.

- We have demonstrated that to achieve the best execution time for a control flow program, the number of nodes within the system and the type of mapping scheme used are particularly important. In addition, we observe that a large number of subsystem nodes allows more actors to be fired concurrently, but the communication overhead in passing control tokens to their destination nodes causes the overall execution time to increase substantially.

- The relationship between the mapping scheme employed and locality effect in a program are discussed. The mapping scheme employed has to exhibit a strong locality effect in order to allow efficient execution

- Medium grain execution can benefit from a higher output bandwidth of a processor and finally, a simple superscalar processor with an issue rate of ten is sufficient to exploit the internal parallelism of a cluster. Although the technique does not exhaustively detect all possible errors, it detects nontrivial errors with a worst-case complexity quadratic to the system size. It can be automated and applied to systems with arbitrary loops and nondeterminism.

Poster Commandments for a Bad Career

1. Never illustrate
2. Never be brief
3. Never print large
4. Never use color
5. Never try to attract attention
6. Never prepare a short oral overview
7. Never prepare in advance

Following the Commandments

How to Do a Bad Poster

David Patterson
University of California
Berkeley, CA 94720

We describe the philosophy and design of the control flow machine, and present the results of detailed simulations of the performance of a single processing element. Each factor is compared with the measured performance of an advanced von Neumann computer running equivalent code. It is shown that the control flow processor compares favorably in the program.

We present a denotational semantics for a logic program to construct a control flow for the logic program. The control flow is defined as an algebraic manipulator of idempotent substitutions and it virtually reflects the resolution deductions. We also present a bottom-up compilation of medium grain clusters from a fine grain control flow graph. We compare the basic block and the dependence sets algorithms that partition control flow graphs into clusters.

Our compiling strategy is to exploit coarse-grain parallelism at function application level: and the function application level parallelism is implemented by fork-join mechanism. The compiler translates source programs into control flow graphs based on analyzing flow of control, and then serializes instructions within graphs according to flow arcs such that function applications, which have no control dependency, are executed in parallel.

A hierarchical macro-control-flow computation allows them to exploit the coarse grain parallelism inside a macrotask, such as a subroutine or a loop, hierarchically. We use a hierarchical definition of macrotasks, a parallelism extraction scheme among macrotasks defined inside an upper layer macrotask, and a scheduling scheme which assigns hierarchical macrotasks on hierarchical clusters.

We apply a parallel simulation scheme to a real problem: the simulation of a control flow architecture, and we compare the performance of this simulator with that of a sequential one. Moreover, we investigate the effect of modeling the application on the performance of the simulator. Our study indicates that parallel simulation can reduce the execution time significantly if appropriate modeling is used.

We have demonstrated that to achieve the best execution time for a control flow program, the number of nodes within the system and the type of mapping scheme used are particularly important. In addition, we observe that a large number of subsystem nodes allows more actors to be fired concurrently, but the communication overhead in passing control tokens to their destination nodes causes the overall execution time to increase substantially.

The relationship between the mapping scheme employed and locality effect in a program are discussed. The mapping scheme employed has to exhibit a strong locality effect in order to allow efficient execution. We assess the average number of instructions in a cluster and the reduction in matching operations compared with fine grain control flow execution.

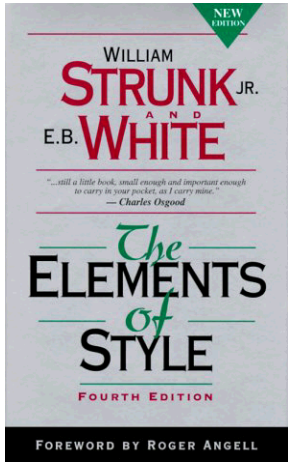
Medium grain execution can benefit from a higher output bandwidth of a processor and finally, a simple superscalar processor with an issue rate of ten is sufficient to exploit the internal parallelism of a cluster. Although the technique does not exhaustively detect all possible errors, it detects nontrivial errors with a worst-case complexity quadratic to the system size. It can be automated and applied to systems with arbitrary loops and nondeterminism.

Part IV

Alternatives to Bad Papers/Talks/Posters

Alternatives to Bad Papers

- Do the opposite of Bad Paper Commandments
  - Define terms, distinguish "will do" vs "have done", mention drawbacks, real performance, reference other papers
- Find related work through on-line library catalogs
  - Most papers available on-line
  - ACM and IEEE digital libraries
- Read the "Elements of Style"
  - Best small book on writing
  - Read it often (every 2 years?)
  - Longer alternative "Bugs in Writing" by Lyn Durpe



Steps for Writing a Paper

1. 1-page paper outline, with tentative page budget/section
2. Paragraph map
   - 1 topic phrase/sentence per paragraph, hand-drawn figures w. captions
3. (Re)write draft
   - Long captions/figure can contain details ~ Scientific American
   - Uses Tables to contain facts that make dreary prose
4. Read aloud, spell check & grammar check
   - (MS Word; Under Tools, select Grammar, select Options, select "technical" for writing style vs. "standard"; select Settings and select)
5. Get feedback from advisor, friends, and critics on draft; go to 3

Alternatives to Bad Talks

- Do the opposite of bad talk commandments
- Allocate 2 minutes per slide, leave time for questions
- Don't over animate
- Do dry runs with friends/critics for feedback,
  - Including tough audience questions
  - Tape a practice talk
- Don't memorize speech, but have notes ready

- Bill Tetzlaff, IBM:
  - Giving a first class 'job talk' is the single most important part of an interview trip. Having someone know that you can give an excellent talk before hand greatly increases the chances of invitation. That means great conference talks.

Alternatives to Bad Posters (from Randy Katz)

- Do opposite of bad poster commandments
  - Poster tries to catch the eye of person walking by

- Answer Heilmeier questions
  - 1. What is the problem you are tackling?
  - 2. What is the current state-of-the-art?
  - 3. What is your key make-a-difference concept or technology?
  - 4. What have you already accomplished?
  - 5. What is your plan for success?

Example 9-page Poster

Problem Statement	State-of-the-Art	Key Concept
Accomplish-ment # 1	Title and Visual logo	Accomplish-ment # 2
Accomplish-ment # 3	Plan for Success	Summary & Conclusion

Links to More Advice on Papers/Talks/Posters

- Patterson's tips on writing
  - http://www.cs.berkeley.edu/~pattrsn/talks/writingtips.html
- Armando's paper writing & presentation page
  - http://www.eecs.berkeley.edu/~fox/paper_writing.html
- The "How to Give a Talk" Talk
  - http://www.eecs.harvard.edu/~margo/slides/
- "The Task of the Referee", A.J Smith
  - Advice on How to Review Papers
  - http://www.cs.wisc.edu/~markhill/the_task_of_the_referee.pdf
- Jason Hong's very extensive list of links
  - Links on research, writing, presentation, thesis
  - http://www.cs.cmu.edu/~jasonh/advice.html

Summary: Enjoy GradSchool

- Show initiative!
  - Don't wait for advisor to show you what to do
  - Selecting your PhD topic is as important as your results
- Ask questions!
  - Lots of smart people in grad school, but don't be intimidated.
  - They will teach you something or you will learn by figuring it out together
  - It's OK to ask a "stupid question" every now and then
- When to graduate?