An Example Conference Paper

Cindy Norris

Department of Computer Science

Appalachian State University

Boone, NC 28608

(828)262-2359

can@cs.appstate.edu

Abstract

The abstract contains a brief overview of the paper. It should be relatively short. There is usually a "sales job" in the abstract too, meaning that you should have one or two sentences explaining why your work is the best thing since sliced bread. For example, "the results of this experimental indicate that, contrary to popular opinion, the replacement policies have very little impact on the miss rate of the cache."

1 Introduction

The introduction contains a longer overview of the paper. In the first paragraph, you should start out saying what you did. You should try to slip in a statement here also about why your work is so great.

In the next paragraph, you may want to briefly mention the related work. If you've come across other experimental papers similar to what you've done, this would be the place to mention them.

Most of you have repeated experiments that have been done by other people. You should see if you can find references for other experimental studies and explain them briefly here.

I'll add a cite to a reference here [?] just for the fun of it. Generally, you cite something you are talking about so that cite doesn't really make sense, but I want to show you how to cite items. If you want to cite a couple of things at the same time, you can do it like this [?, ?] or you can cite it like this [?][?], but be consistent in whichever way you choose.

This section should end with a description of the rest of the paper. You would have a statement that says Section 2 contains the background necessary for understanding the experimental study. Section 3 describes the experimental work and finally, the paper concludes in Section 4.

2 Background

This section describes the background of my work. In Section 2.1, I describe blah, blah, blah. In Section 2.2, I describe yada, yada, yada. You should describe the background in a pretty good amount of detail, adding more sections if necessary.

2.1 Here is one background subsection

Here is where you might explain the general organization of a cache. Be sure to explain why the cache is useful.

2.2 Another background subsection

Here is where you might focus on something more closely related to your cache study, for example, replacement policies.

Benchmark	**Number of Registers**							
	8				**16**			
	long		**medium**		**long**		**medium**	
	RASSG	**REGOA**	**RASSG**	**REGOA**	**RASSG**	**REGOA**	**RASSG**	**REGOA**
Livermore								
loop1	1.56	1.02	1.48	1.01	1.23	1.04	1.17	1.04
loop2	1.17	1.06	1.15	1.04	1.30	0.68	1.07	0.57
loop3	1.14	1.04	1.13	1.03	1.02	1.02	1.04	1.03
loop4	1.05	0.88	1.12	0.88	1.06	1.04	1.03	0.98
loop6	1.09	0.96	0.84	0.97	1.27	0.92	1.05	0.92
Clinpack								
daxpy	1.17	1.00	1.15	1.00	1.00	1.00	1.00	1.00
matgen	0.92	1.00	0.85	1.00	1.13	1.00	1.06	1.00
dgefa	0.78	1.00	1.13	1.00	1.32	1.00	1.23	1.00

Table 1: **Speedup of fully cooperative and fully uncooperative techniques over post-pass**

3 Experimental Study

Here is where you will describe your experimental study. You don't want to go into a lot of detail about how you did the study. For example, the dineroIV parameters and the scripts you wrote aren't important to this paper. What is important is the number of benchmarks and what they are. You should mention what experiments you ran even if you don't present all of the numbers.

You add several tables (or graphs from your tables) to this section and you need to explain the tables. For example, you might say something like "Table 1 contains the results of running dineroIV assuming a cache size of X." And then you should go into some detail about what the table means. Don't just throw in the tables and not explain the results. On the other hand, don't simply restate every detail in the table. You need to state the conclusions gained from looking at all of the data.

Like my table, your tables will need to be embedded in the body of the paper. Don't staple them to the end.

4 Conclusions

This section kind of mimics the abstract but it also will explain any future work. If you neglected something in your implementation, you'll mention that here by saying something like "my future work will be to add so and so to my implementation and perform such and such experiments." (I promise I won't hold you to any future work that you describe.)

Final notes:

- Avoid the use of any pronouns (I, my, you, your, we, etc.) If you feel like you really need to use "I" (for example, in order to explicitly state what you did) then use "we."
- Don't put tables in the paper and not discuss them.
- Don't add references to a paper and not cite them.
- Don't make your paragraphs too long. Long, technical paragraphs are hard to follow.
- Your paper should flow nicely – one sentence to next, one paragraph to next paragraph, one section to the next section. In other words, don't do sudden shifts in topic. Make a plan before writing and don't just write off of the top of your head. Figure out what should be in each section and each paragraph of each section before you start writing.
- Technical writing should be formal; for example, don't use words like "cool."
- Add figures to clarify your explanation. Remember, a picture is worth a thousand words.
- Don't quote anything that should be simply restated. Quotes should be reserved for something of an earth-shattering nature, and not because one is simply to lazy to figure out a way to restate the quote (or the thought that he/she said it much better than I ever could).
- Don't include rhetorical questions. (A rhetorical question is one that is asked without expecting an answer.)
- Avoid starting sentences with "To" "Because" "Before" etc.
- Don't have huge technical paragraphs. Technical material is easier to read if it is broken into several, smaller paragraphs.
- Finally, proof read your papers carefully – multiple times and run a spell checker on it!

References

[1] David A. Berson, Rajiv Gupta, and Mary Lou Soffa. URSA: A unified resource allocator for registers and functional units in VLIW architectures. In *IFIP Working Conference on Architectures and Compilation Techniques for Fine and Medium Grain Parallelism*, Orlando, Florida, January 1993.

[2] D. G. Bradlee. *Retargetable Instruction Scheduling for Pipelined Processors.* PhD thesis, University of Washington, 1992.

[3] J. R. Ellis. *Bulldog: A Compiler for VLIW Architectures.* MIT Press, Cambridge, MA, 1986.

[4] Claude-Nicolas Fiechter. PDG C Compiler. University of Pittsburgh, 1992.

[5] H. Young. Evaluation of a decoupled computer architecture and the design of a vector extension. *Computer Sciences Technical Report 603*, 21(4), July 1985.