Template for Project 1 Report

Bryan S. Kim

Abstract

Your abstract text goes here. Just a few facts. Whet our appetites. Not more than 200 words, if possible, and preferably closer to 150. Although you may think that an abstract is just a summary, it's not: it's supposed to be compelling and draw readers in. Make it exciting!

1 Introduction

A paragraph of text goes here. Lots of text. Plenty of interesting text. Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text. More fascinating text. Features galore, plethora of promises.

In general, the introduction presents a brief background of the area, a compelling motivation for the paper, as well as a glimpse into the technical details and results. I hate to say it this way, but think of it as a *longer* version of the abstract.

Remember the basics of writing: each paragraph represents a singular idea, with each sentence in the paragraph providing support for that central idea. I will be grading this report as if I am reviewing a workshop paper submission, so writing and presentation count!

2 Experimental Methodology

This section presents how you will run your experiments. It's a good idea to include information about the platform/hardware, environment/configurations, and workload.

The workload you are expected to run is db_bench, a built-in tool for LevelDB. You are not expected to run workloads such as YCSB [?] for this project.

You should also include a list of configuration options you are changing. You should read and understand the documentation on LevelDB [?] and are encouraged to also look at RocksDB [?].

- `https://github.com/google/leveldb`
- `https://github.com/facebook/rocksdb`

Workload	***Description***
A	50% Read, 50% Update Session store recording actions
B	95% Read, 5% Update Photo taggging and tag reads
C	100% Read User profile cache
D	95% Read, 5% Insert User status update
E	95% RangeScan, 5% Insert Recent post scan in conversation
F	50% Read, 50% Read-Modify-Write Database

Table 1: YCSB Workload Characteristics.

As I mentioned in the lecture, you should be following the scientific method to an extent.

1. Based on your understanding of LevelDB, hypothesize which options are important.
2. Run a number of experiments that support (or refute) your claims.
3. Present the results and draw conclusions from your understanding.

3 Evaluation Results

Include your evaluation results using figures or tables. Figure **??** is an example of a figure, and Table **??** is an example of a table. Remember to explain the results, rather than just showing it. Make sure you include your *lessons learned* as you present the results.

4 Related Work

The subsections within this section are text from the USENIX paper template. They are helpful, but you do not need to include a *Related Work* section in this report.

4.1 Footnotes, Verbatim, and Citations

Footnotes should be places after punctuation characters, without any spaces between said characters and

footnotes, like so.[1] And some embedded literal code may look as follows.

```
int main(int argc, char *argv[])
{
    return 0;
}
```

Now we're going to cite somebody. Watch for the cite tag. Here it comes. Arpaci-Dusseau and Arpaci-Dusseau co-authored an excellent OS book, which is also really funny [**?**], and Waldspurger got into the SIGOPS hall-of-fame due to his seminal paper about resource management in the ESX hypervisor [**?**].

The tilde character (~) in the tex source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

And the 'cite' package sorts your citations by their numerical order of the corresponding references at the end of the paper, ridding you from the need to notice that, e.g, "Waldspurger" appears after "Arpaci-Dusseau" when sorting references alphabetically [**?** **?**].

It'd be nice and thoughtful of you to include a suitable link in each and every bibtex entry that you use in your submission, to allow reviewers (and other readers) to easily get to the cited work, as is done in all entries found in the References section of this document.

Now we're going take a look at Section **??**, but not before observing that refs to sections and citations and such are colored and clickable in the PDF because of the packages we've included.

4.2 Floating Figures and Lists

Here's a typical reference to a floating figure: Figure **??**. Floats should usually be placed where latex wants then. Figure**??** is centered, and has a caption that instructs you to make sure that the size of the text within the figures that you use is as big as (or bigger than) the size of the text in the caption of the figures. Please do. Really.

In our case, we've explicitly drawn the figure inlined in latex, to allow this tex file to cleanly compile. But usually, your figures will reside in some file.pdf, and you'd include them in your document with, say, \includegraphics.

Lists are sometimes quite handy. If you want to itemize things, feel free:

fread a function that reads from a `stream` into the array `ptr` at most `nobj` objects of size `size`, returning returns the number of objects read.

[1] Remember that USENIX format stopped using endnotes and is now using regular footnotes.

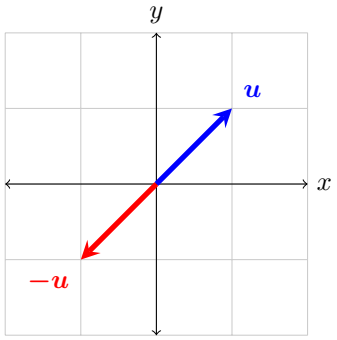

Figure 1: Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text.

Fred a person's name, e.g., there once was a dude named Fred who separated usenix.sty from this file to allow for easy inclusion.

The noindent at the start of this paragraph in its tex version makes it clear that it's a continuation of the preceding paragraph, as opposed to a new paragraph in its own right.

4.3 LaTeX-ing Your TeX File

People often use `pdflatex` these days for creating pdfs from tex files via the shell. And `bibtex`, of course. Works for us.

5 Conclusion

Conclude with a bang. This is a good place to summarize everything.