Title Is Here

Author Name

A report submitted for the course
YOUR COURSE CODE AND NAME
Supervised by: YOUR SUPERVISOR(S) NAME(S)
The Australian National University

January 2018

Except where otherwise indicated, this report is my own original work.

Author Name
3 January 2018

Acknowledgments

Who do you want to thank? I hope it includes your supervisor(s).

Abstract

Replace the text here with your abstract.

This is an example of a LaTeXreport suitable for CS Projects. It has been adapted from a template ANU thesis and contains some examples of how to typeset code listings, figures and tables in a computer science context.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Put your introduction here. You could use \fix{ABCDEFG.} to leave your comments, see the box at the left side.

You have to rewrite your thesis!!!

1.1 Problem Statement

Describe the problem your project addresses.

1.2 Motivations

1.3 Project Scope

Describe the problem your project addresses.

1.4 Report Outline

How many chapters you have? You may have Chapter 2, Chapter 3, Chapter 4, Chapter 5, and Chapter 6.

Chapter 2

Background and Related Work

At the beginning of each chapter, please give the motivation and high-level picture of the chapter. You also have to introduce the sections in the chapter, e.g.

Section 2.1 gives background material necessary in order to read this report ,and related work is given in Section 2.2.

2.1 Background

2.2 Related work

You may reference other papers. For example: Generational garbage collection [Lieberman and Hewitt, 1983; Moon, 1984; Ungar, 1984] is perhaps the single most important advance in garbage collection since the first collectors were developed in the early 1960s. (doi: "doi" should just be the doi part, not the full URL, and it will be made to link to dx.doi.org and resolve. shortname: gives an optional short name for a conference like PLDI '08.)

2.3 Summary

Summarize what you discussed in this chapter, and introduce the story of next chapter. Readers should roughly understand what your report talks about by only reading words at the beginning and the end (Summary) of each chapter.

Chapter 3

Design and Implementation

Same as the last chapter, give the motivation and the high-level picture of this chapter to readers, and introduce the sections in this chapter.

3.1 Smart Design

3.2 Summary

Same as the last chapter, summarize what you discussed in this chapter and be a bridge to next chapter.

Chapter 4

Experimental Methodology

4.1 Software platform

We use Jikes RVM, which we defined in a macro in `macros.tex`. We ran the avrora benchmark, which we're typesetting in sans-serif font to make it clear it's a name.

You can also use inline code, like `a && b`. Notice how, unlike when using the `texttt` command, the `icode` macro also scales the x-height of the monospace font correctly.

4.2 Hardware platform

Table 4.1 shows how to include tables and Figure 4.1 shows how to include codes. Notice how we can also use the cleveref package to insert references like Table 4.1, by writing just \cref{tab:machines}.

We can also refer to specific lines of code in code listings. The bug in Figure 4.1(a) is on line 4. There is also a bug in Figure 4.1(b) on lines 4 to 6. To achieve these references we put `(*@ \label{line:bug} @*)` in the code – the `(*@ @*)` are escape delimiters that allow you to add LaTeX in the (otherwise verbatim) code file.

Table 4.1: Processors used in our evaluation. Note that the caption for a table is at the top. Also note that a really long comment that wraps over the line ends up left-justified.

Architecture	Pentium 4	Atom D510	Sandy Bridge
Model	P4D 820	Atom D510	Core i7-2600
Technology	90nm	45nm	32nm
Clock	2.8GHz	1.66GHz	3.4GHz
Cores × SMT	2 × 2	2 × 2	4 × 2
L2 Cache	1MB × 2	512KB × 2	256KB × 4
L3 Cache	none	none	8MB
Memory	1GB DDR2-400	2GB DDR2-800	4GB DDR3-1066

```c
int main(void)
{
  printf("Hello␣World\n");
  int a = *((volatile int *) 0); // uh-oh!
  return 0;
}
```

(a) C

```java
void main(String[] args)
{
  System.out.println("Hello␣World");
  List<String> a = new ArrayList<String>();
  a.add("foo");
  List<Object> b = a; // This is a compile-time error
}
```

(b) Java

Figure 4.1: Hello world in Java and C. This short caption is centered.

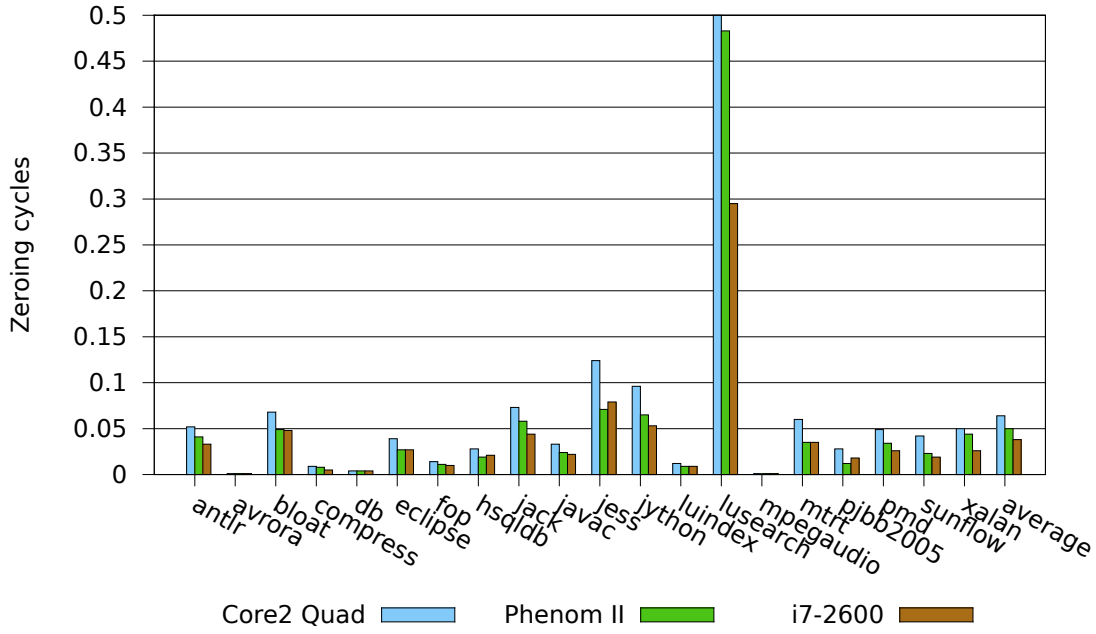
Chapter 5

Results

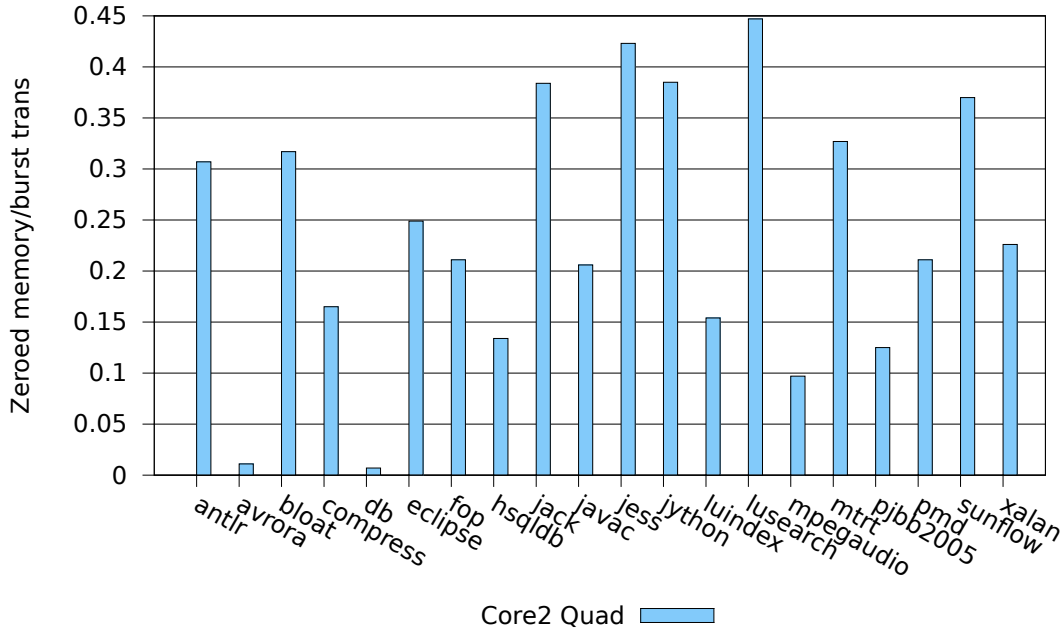
5.1 Direct Cost

Here is the example to show how to include a figure. Figure 5.1 includes two subfigures (Figure 5.1(a), and Figure 5.1(b));

5.2 Summary



(a) Fraction of cycles spent on zeroing



(b) BytesZeroed / BytesBurstTransactionsTransferred

Figure 5.1: The cost of zero initialization

Chapter 6

Conclusion

Summarize your work, state your main findings and discuss what you are going to do in the future in Section 6.1.

6.1 Future Work

Good luck.

Bibliography

Lieberman, H. and Hewitt, C., 1983. A real-time garbage collector based on the lifetimes of objects. *Communications of the ACM*, 26, 6 (Jun. 1983), 419–429. doi:10.1145/358141.358147. (cited on page 3)

Moon, D. A., 1984. Garbage collection in a large LISP system. In LFP '84: *Proceedings of the 1984 ACM Symposium on LISP and Functional Programming* (Austin, Texas, USA, Aug. 1984), 235–246. ACM, New York, New York, USA. doi:10.1145/800055.802040. (cited on page 3)

Ungar, D., 1984. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. In SDE 1: *Proceedings of the 1st ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments* (Pittsburgh, Pennsylvania, USA, Apr. 1984), 157–167. ACM, New York, New York, USA. doi:10.1145/800020.808261. (cited on page 3)