

Title Is Here

Author Name

A thesis submitted for the degree of
YOUR DEGREE NAME
The Australian National University

April 2024

© Author Name 2019

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

Author Name
15 April 2024

to my xxx, yyy (yyy is the people you want to dedicated this thesis to.)

Acknowledgments

Who do you want to thank?

Abstract

Put your abstract here.

Contents

Acknowledgments	vii
Abstract	ix
1 Introduction	1
1.1 Thesis Statement	1
1.2 Introduction	1
1.3 Thesis Outline	1
2 Background and Related Work	3
2.1 Motivation	3
2.2 Related work	3
2.3 Summary	3
3 Design and Implementation	5
3.1 Smart Design	5
3.2 Summary	5
4 Experimental Methodology	7
4.1 Software platform	7
4.2 Hardware platform	7
5 Results	9
5.1 Direct Cost	9
5.2 Summary	9
6 Conclusion	11
6.1 Future Work	11

List of Figures

4.1	Hello world in Java and C.	8
5.1	The cost of zero initialization	10

List of Tables

4.1	Processors used in our evaluation.	7
-----	--	---

Introduction

1.1 Thesis Statement

I believe A is better than B.

1.2 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!!!

1.3 Thesis Outline

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

Background and Related Work

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

Section 2.1 xxxx.

Section 2.2 yyyy.

2.1 Motivation

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

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

Design and Implementation

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

3.1 Smart Design

3.2 Summary

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

Experimental Methodology

4.1 Software platform

4.2 Hardware platform

Table 4.1 shows how to include tables and Figure 4.1 shows how to include codes.

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

Table 4.1: Processors used in our evaluation.

```
1 int main(void)
2 {
3     printf("Hello_World\n");
4     return 0;
5 }
```

(a)

```
1 void main(String[] args)
2 {
3     System.out.println("Hello_World");
4 }
```

(b)

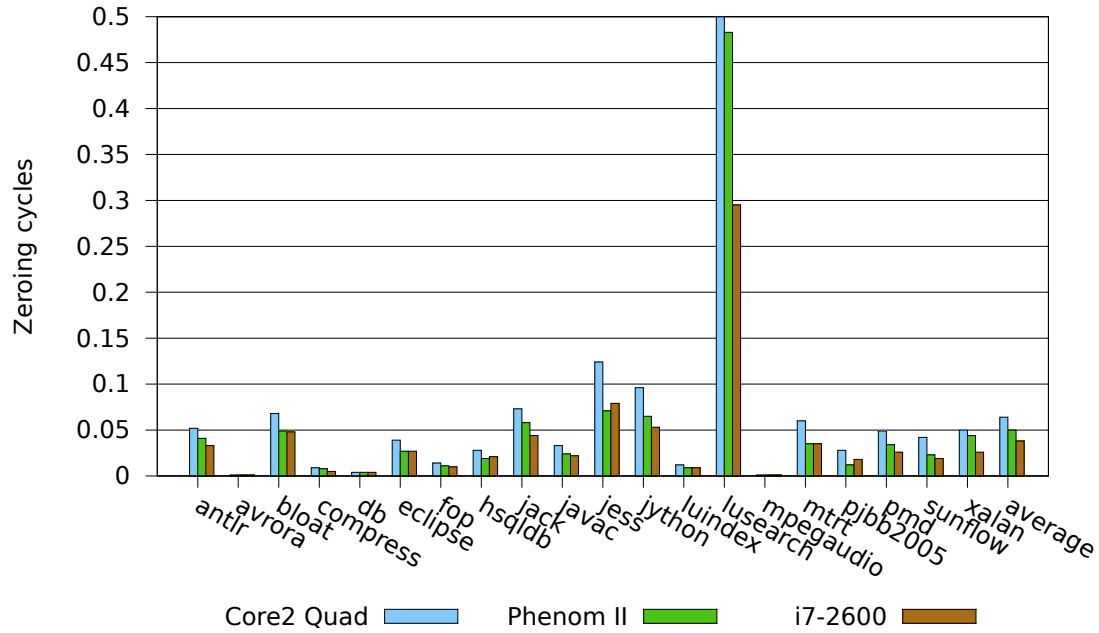
Figure 4.1: Hello world in Java and C.

Results

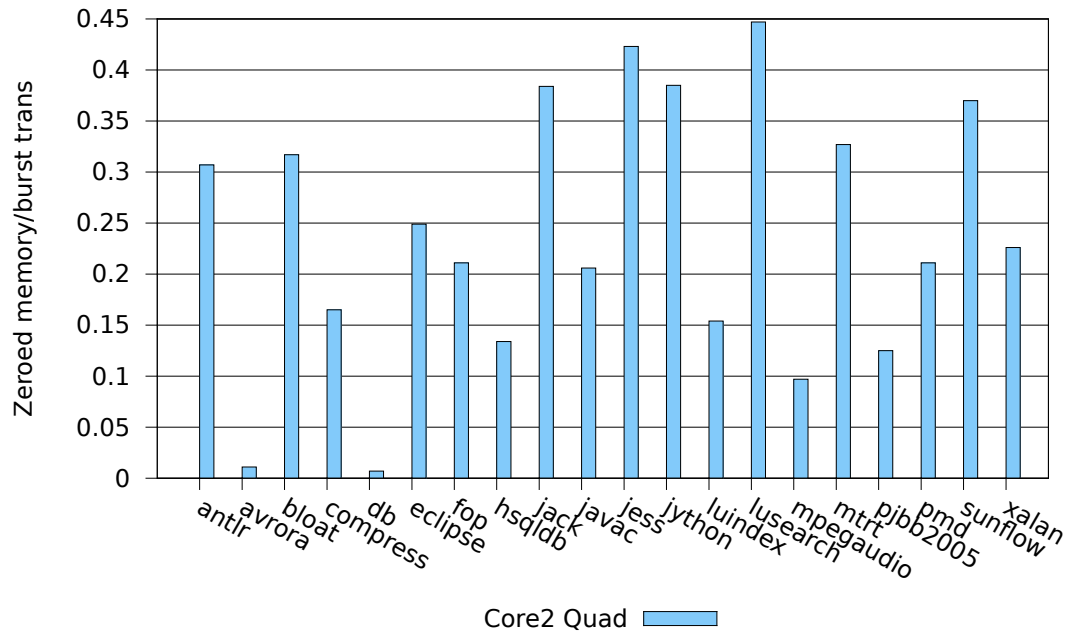
5.1 Direct Cost

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

Conclusion

Summary your thesis 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)