Professorship for Computer Science Operating Systems Engineering



Advances in Operating Systems Superoptimization

September 15, 2022

Madhu Mohan Nelemane

Content

ON THE PARTY OF TH

- 1 Introduction
- 2 Simple Assembly language Optimization Techniques
- 3 Learnings from Synthesis kernel and OS
- 4 Introducing SuperOptimizers
- 5 Peephole Superoptimizers and Modern Compilers
- 6 Conclusion

Simple Assembly

language Optimizatio
Techniques

Learnings from Synthesis kernel and OS

SuperOptimizers

Peephole Superoptimizers and Modern Compilers

clusion

Introduction



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and

nelusion

References

This presentation talks about superoptimization

Assembly

- Replace decision constructs with logical and arithmetic operations
- Unroll loops to avoid pipeline dependencies
- Parallelize Load/Store with MAC instructions



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and

onclusion

Optimizations in Synthesis Kernel

- Dynamic Code Generation
- Software Feedback



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole
Superoptimizers and
Modern Compilers

onclusion

Optimizations from Synthesis OS



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

Conclusion

Basic Superoptimizer steps

- Boolean Test
- Probabilistic Test
- Pruning
- Applications and Limitations



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

onclusion

Automatic Generation of Peephole Superoptimizers



Introduction

Simple Assembly language Optimization Techniques

Synthesis kernel a

Introducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

anclusion

Improving Binary Translations using Peephole superoptimizers



Peephole Superoptimizers and Modern Compilers

GNU Superoptimizer

- Peephole Superoptimizers and Modern Compilers

■ Usage of superopt command, options and results

https://www.gnu.org/software/superopt/

LLVM and JIT



Introduction

Simple Assembly language Optimization Techniques

Synthesis kernel a OS

Introducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

Conclusion

Conclusion



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and

Conclusion

References I

- 1 S. Bansal and A. Aiken, "Automatic generation of peephole superoptimizers."
- [2] —, "Binary translations using peephole superoptimizers."
- [3] H. Massalin, "Superoptimizer a look at the smallest program."
- [4] C. Pu and H. Massalin, "An overview of the synthesis operating system."
- [5] C. Pu, J. Walpole, and H. Massalin, "A retrospective study of the synthesis kernel."
- [6] [Online]. Available: https://blog.regehr.org/archives/1676
- [7] [Online]. Available: https://www.gnu.org/software/superopt/



Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

> ntroducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

onclusion



Introduction

Simple Assembly language Optimization Techniques

Synthesis kernel an OS

Introducing SuperOptimizers

Peephole
Superoptimizers and
Modern Compilers

Conclusion

References

Questions?

Marcel Großmann marcel.grossmann@uni-bamberg.de