Professorship for Computer Science
Operating Systems Engineering

Advances in Operating Systems

Superoptimization

September 15, 2022

Madhu Mohan Nelemane

Content

Introduction

Simple Assembly language Optimization Techniques

Learnings from Synthesis kernel and OS

Introducing SuperOptimizers

Peephole Superoptimizers and Modern Compilers

Conclusion

References

Introduction

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

Optimizations in Synthesis Kernel

- Dynamic Code Generation
- Software Feedback

Optimizations from Synthesis OS

Basic Superoptimizer steps

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

Automatic Generation of Peephole Superoptimizers

Improving Binary Translations using Peephole superoptimizers

GNU Superoptimizer

- https://www.gnu.org/software/superopt/
- Usage of superopt command, options and results

LLVM and JIT

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/

Questions ?

Marcel Großmann
marcel.grossmann@uni-bamberg.de