

Abstract

One of the effective techniques for controlling the complexity of task execution in heterogeneous hardware/software systems is symbolic scheduling. It enables the analysis and synthesis of execution plans without exhaustive enumeration by symbolically describing scheduling constraints and system behaviors. This seminar paper investigates and assesses various symbolic scheduling approaches, emphasizing their applications in both software and hardware domains. The first part of the paper examines the FunState model, introduced by Strehl et al. , which explicitly represents scheduling and nondeterminism by combining state machines and functional programming. Subsequently, the paper explores the symbolic approach for the combined solution of scheduling and allocation proposed by Cabodi et al. , which enhances high-level synthesis by concurrently integrating operation scheduling and resource allocation using symbolic techniques. By contrasting these methods, the study aims to determine the most effective symbolic scheduling strategy, considering aspects such as scalability, efficiency, and suitability for real-world systems.

References

- [1] K. Strehl, L. Thiele, D. Ziegenbein, R. Ernst, and J. Teich, "Scheduling Hardware/Software Systems Using Symbolic Techniques," in *Proc. 7th Int. Workshop on Hardware/Software Codesign (CODES '99)*, Rome, Italy, 1999, pp. 173–177.
- [2] G. Cabodi, M. Lazarescu, L. Lavagno, S. Nocco, C. Passerone, and S. Quer, "A Symbolic Approach for the Combined Solution of Scheduling and Allocation," in *Proc. 15th Int. Symp. on System Synthesis (ISSS '02)*, Kyoto, Japan, 2002, pp. 134–139.

Raw Sketch Outline For Symbolic Scheduling:

Introduction

- Importance of scheduling in heterogeneous HW/SW systems
- Definition and benefits of symbolic scheduling
- Objectives and structure of the paper

Background and Related Work

- Overview of scheduling in embedded systems
- Explanation of symbolic techniques
- Brief history and evolution

Symbolic Scheduling Approaches

FunState Model (Strehl et al.)

- Representation combining state machines and functional programming
- Explicit handling of non-determinism
- Applications in heterogeneous system designs
- Strengths and limitations

Explicit Scheduler, Symbolic Threads (ESST, Cimatti et al.)

- Explicit scheduler with symbolic thread exploration
- Application in multithreaded software model checking
- Cooperative scheduling policy verification
- Strengths and limitations

High-level Synthesis for Packet Processing Pipelines (Soviani)

- Integration of symbolic scheduling in hardware synthesis
- Optimization of resource allocation and pipelining
- Application in hardware accelerators for packet processing
- Strengths and limitations

Comparative Analysis

- Criteria for comparison: scalability, efficiency, suitability for real-world applications
- Comparative table highlighting key aspects
- Discussion on trade-offs between software and hardware applicability

Discussion and Recommendations

- Which method performs better overall based on analysis
- Recommended contexts for each method
- Potential for integrating techniques to cover weaknesses

Conclusion

- Summary of findings
- Future research directions