Performance Analysis of Tightly Coupled Multiprocessor Systems with SymTA/S

S. Schliecker, R. Henia, A. Hamann, R. Racu, J. Rox, M. Sebastian, S. Stein, R. Ernst

Institute of Computer and Communication Network Engineering
Technical University of Braunschweig
Hans-Sommer-Str. 66, D-38016 Braunschweig, Germany
http://www.ida.ing.tu-bs.de
{ schliecker, henia, hamann, racu, rox, sebastian, stein } @ ida.ing.tu-bs.de

Abstract

SymTA/S is a high-level multiprocessor performance analysis tool for early design stages. Innovative extensions allow improved modeling of tightly coupled MPSoCs and evaluation of their robustness.

1. Introduction

The contemporary consumer appetite for more computing capacities in low power embedded systems is leading to a growing application of multiprocessor systems integrated on a single chip (MPSoCs). In such systems, the tight interaction via shared busses and memories poses new challenges for platform and application designers that call for new methodologies for performance analysis tools.

SymTA/S (see Figure 1) is a software tool for fast formal performance and timing analysis of heterogeneous multiprocessor systems with several RTOSes, buses, and networks. Its compositional nature allows to integrate a large range of scheduling policies. SymTA/S has successfully been used to identify and relieve bottlenecks in industrial applications.

Using recent scientific results, we have extended this tool to address the challenges of integrated multi-processor systems on chip. This way, the variety of complex dependencies between memory accesses, bus conflicts, and task scheduling and execution can be addressed in a single tool framework.

2. Shared Memories in MPSoCs

As opposed to distributed multiprocessor systems, MPSoCs make an increased use of shared memories and communication infrastructure that are accessed by all processing elements. This introduces complex and highly dynamic timing dependencies on the real time applications. With this, formal performance analysis faces a new challenge, as classical "bottom-up" analysis had assumed task's timing information is derived a priori for use in the system level analysis.

Task List: T2 CPU0

Core Task Time: Min: 60 · 1.0 = 60
Max: 60 · 1.0 = 60
Activating Event Model: P (400)
Response Time: Min: 76 Max: 209
Output Event Model: P (400) + J (133)
Max. Execution Backlog: 1
Total Buffer Size: 1
Scheduling: Blocking Static Priority Preemptive (BSPP)
Task Speed: 1.0

CPU0
Speed Factor 1.0 ☐ Use Individual Speed Factors

Resource: CPU0 Task: T2 Colors: memoryAccessColor

```
Analysis on CPU0 started ...
14:30:02 Performing Blocking Static Priority Preemp
14:30:02 SPP Busy Time: 24
14:30:02 WC resp.time for activation 1 of T0: 24
14:30:02 SPP Busy Time: 56
14:30:02 WC resp.time for activation 1 of T1: 56
14:30:02 SPP Busy Time: 209
14:30:02 WC resp.time for activation 1 of T2: 209
14:30:02 Blocking Static Priority Preemptive (BSPP) Analysis complete (141ms).
14:30:02 Analysis on CPU0 finished.
```

```
>>> create MemoryAccessSink
MS3
```

Figure 1: SymTA/S Tool

To address these effects, we extend our previous approach [1] by explicitly modeling the task's communication behavior. As depicted in Figure 2, each task is assumed to execute locally (from a cache or scratchpad memory), and may fetch data from a shared memory as well.

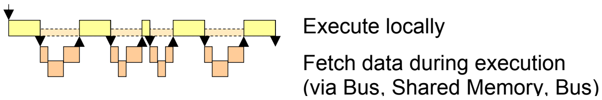
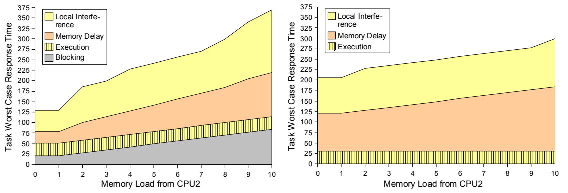


Figure 2: Communicating Task Model

The timing of these shared memory accesses is primarily influenced by the amount of data transferred. But the processing may also be delayed by other components of the system that compete for the same resources. Our approach considers these dependencies to derive safe timing estimates.

As the interference is highly dynamic depending on the current system state, we investigate the total time of all requests together. This contains the over-estimations that could otherwise arise from transient overload situations [3].

The derived timing for the data transactions is integrated into the local analysis to deliver the worst case response times for safety-critical applications. We provide analyses for priority based scheduling on processors that stall during memory accesses and those that allow multithreading in order to investigate the performance trade-off [2]. Figure 3 shows the components of a task's response time that vary from the use of multithreading for different memory load.



a) Processor Blocking Behavior

b) Processor Multithreading

Figure 3: Components of Task Response Time

4. Timing Correlations

For simplicity and to avoid the growing analysis complexity, most formal scheduling analysis techniques ignore correlations between task execution times or communication timing, in particular when it comes to heterogeneous multiprocessor systems. However, such correlations can have a large influence on system timing. In [4] we extended existing approaches capturing correlations between task activations in distributed systems by considering multiple timing-references. This allows to rely on the most recent timing-reference shared by a set of time-correlated tasks to perform a more precise calculation of their activation instants and thus, calculate tighter task response times.

Also, we improve the task output jitter calculation by taking into account the correlation between jitter and response time of individual activating events [5]. Since in SymTA/S, the output event stream of one component turns into the input event stream of the connected component, the overall system performance estimates strongly benefit from the improved output jittter calculation.

5. System Robustness Optimization

In systems with complex dependencies, small variations of local system properties (e.g. worst-case execution times, activation periods, etc.) may have unpredictable effects on the systems' timing properties, leading to difficult-to-handle performance bottlenecks. Therefore, system robustness to such property variations represents a major concern during the design of real-time systems.

Unfortunately, design optimization pursuing classical design goals does not necessarily yield robust systems. Therefore, we recently developed an efficient robustness optimization approach based on state-of-the-art sensitivity analysis techniques [6]. Our approach efficiently discovers system configurations that have, at the same time, good timing properties and large robustness to system property variations.

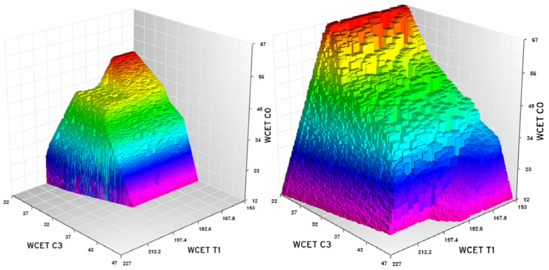


Figure 3: Original and optimized robustness (simultaneous variation of 3 system properties)

6. Conclusion

Using our analysis approach, the variety of complex dependencies between memory accesses, bus conflicts, and task scheduling and execution can be analyzed based on a single tool framework. We can derive tight analysis bounds by taking into account timing correlations in the system.

Moreover, enhanced examinations allow to systematically evaluate and optimize the robustness towards system variations.

7. References

[1] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter, Rolf Ernst. System Level Performance Analysis - The SymTA/S Approach. *IEE Proceedings Computers and Digital Techniques*, 2005.

[2] Simon Schliecker, Matthias Ivers, Rolf Ernst. Integrated Analysis of Communicating Tasks in MPSoCs. In *Proc. 3rd International Conference on Hardware Software Codesign and System Synthesis (CODES)*, Seoul, Korea, October 2006.

[3] Simon Schliecker, Matthias Ivers, Jan Staschulat, Rolf Ernst. A Framework for the Busy Time Calculation of Multiple Correlated Events. In *6th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis*, Dresden, Germany, July 2006.

[4] R. Henia and R. Ernst. Context-aware scheduling analysis of distributed systems with tree-shaped task-dependencies. In *Proc. of Design, Automation and Test in Europe (DATE'05)*, Munich, Germany, Mar. 2005.

[5] R. Henia and R. Ernst. Improved Offset-Analysis Using Multiple Timing-References. In *Proc. of Design, Automation and Test in Europe (DATE'06)*, Munich, Germany, Mar. 2005.

[6] Arne Hamann, Razvan Racu, Rolf Ernst. Multi-Dimensional Robustness Optimization in Heterogeneous Distributed Embedded Systems. In *Proc. of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium*, April 2007.