Modeling Cache Coherence to Expose Interference (Artifact)

Nathanaël Sensfelder
ONERA, Toulouse, France

Julien Brunel
ONERA, Toulouse, France

Claire Pagetti
ONERA, Toulouse, France

Abstract

To facilitate programming, most multi-core processors feature automated mechanisms maintaining coherence between each core's cache. These mechanisms introduce interference, that is, delays caused by concurrent access to a shared resource. This type of interference is hard to predict, leading to the mechanisms being shunned by real-time system designers, at the cost of potential benefits in both running time and system complexity.

We believe that formal methods can provide the means to ensure that the effects of this interference are properly exposed and mitigated. Consequently, we propose a nascent framework relying on timed automata to model and analyze the interference caused by cache coherence.

2012 ACM Subject Classification Computer systems organization → Multicore architectures; Computer systems organization → Real-time systems

Keywords and phrases Real-time systems, multi-core processor, cache coherence, formal methods

Digital Object Identifier 10.4230/DARTS.5.1.7

Acknowledgements We would like to thank Mamoun Filali-Amine (IRIT-CNRS) for his helpful insights on how to validate our model.

Related Article Nathanaël Sensfelder, Julien Brunel, and Claire Pagetti, "Modeling Cache Coherence to Expose Interference", in 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), LIPIcs, Vol. 133, pp. 18:1–18:22, 2019.
`https://dx.doi.org/10.4230/LIPIcs.ECRTS.2019.18`

Related Conference 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), July 9–12, 2019, Stuttgart, Germany

1 Scope

These models allow the reproduction of all experiments shown in the paper, including the ones pertaining to model correctness. In additions, models for a more simplistic architecture (with an atomic bus) are included, and can be used for similar experiments, despite not being mentioned in the paper.

The experiments in question are:

- Counting cache hits and misses.
- Counting all occurrences of cache related interference, regardless of impact on the system.
- Counting occurrences of cache related interference that have perceivable impact on the system.
- Evaluating the impact of the use of cache coherence on the system's run time.

The experiments meant to validate the model's correctness are:

- All programs terminate.
- Copies of a same memory element in multiple cache are never given incompatible states.
- Whenever a cache accesses its copy of a memory element, the value it gets corresponds to the system-wide one.

© Nathanaël Sensfelder, Julien Brunel, and Claire Pagetti;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

2 Content

The artifact package includes:

- A model for an architecture *without* split-transaction bus, both with 2 and 4 cores. This corresponds to what was presented in a previous version of the paper (*ecrts19_prev_artifact_4_cores.sysmod.xml*, *ecrts19_prev_artifact.sysmod.xml*).
- A model for an architecture *with* split-transaction bus, both with 2 and 4 cores. This corresponds to the final version of the paper (*future.sysmod.xml*, *future_4_cores.sysmod.xml*).
- A model of the split-transaction bus architecture in which the value of each memory element is kept track of, so as to enable an additional correctness property (*future_value_test.sysmod.xml*).
- The instructions given to the artifact evaluators.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `https://github.com/nsensfel/phylog-cache-coherence`.

4 Tested platforms

This artifact should work on any platform capable of running *UPPAAL* 4.1.22, which has been released for Linux, Windows, and Mac OS.

5 License

The artifact is available under an LGPLv3 license.

6 MD5 sum of the artifact

836018167846cb77088defad2fce7ebc

7 Size of the artifact

80 kiB