**Federal University of Santa Catarina**
Software/Hardware Integration Lab

Call for Participation

Date: Jul 4, 2014 Time: 14:00 Place: Sala PGEAS II

# Real-Time Operating System Support for Multicore Applications

Giovani Gracioli

## Abstract

Modern multicore platforms feature multiple levels of cache memory placed between the processor and main memory to hide the latency of ordinary memory systems. The primary goal of this cache hierarchy is to improve average execution time (at the cost of predictability). The uncontrolled use of the cache hierarchy by real-time tasks may impact the estimation of their worst-case execution times (WCET), specially when real-time tasks access a shared cache level, causing a contention for shared cache lines and increasing the application execution time. This contention in the shared cache may lead to deadline losses, which is intolerable particularly for hard real-time (HRT) systems.  Shared cache partitioning is a well-known technique used in multicore real-time systems to isolate task workloads and to improve system predictability.

Presently, the state-of-the-art studies that evaluate shared cache partitioning on multicore processors lack two key issues. First, the cache partitioning mechanism is typically implemented either in a simulated environment or in a general-purpose OS (GPOS), and so the impact of kernel activities, such as interrupt handlers and context switching, on the task partitions tend to be overlooked. Second, the evaluation is typically restricted to either a global or partitioned scheduler, thereby by falling to compare the performance of cache partitioning when tasks are scheduled by different schedulers. Furthermore, recent works have confirmed that OS implementation aspects, such as the choice of scheduling data structures and interrupt handling mechanisms, impact real-time schedulability as much as scheduling theoretic aspects. However, these studies also used real-time patches applied into GPOSes, which affects the run-time overhead observed in these works and consequently the schedulability of real-time tasks. Additionally, current multicore scheduling algorithms do not consider scenarios where real-time tasks access the same cache lines due to true or false sharing, which also impacts the WCET.

This thesis addresses these aforementioned problems with cache partitioning techniques and multicore real-time scheduling algorithms as following. First, a real-time multicore support is designed and implemented on top of an embedded operating system designed from scratch. This support Consists of several multicore real-time scheduling algorithms, such as global and partitioned EDF, and a cache partitioning mechanism based on page coloring. Second, it is presented a comparison in terms of schedulability ratio considering the run-time overhead of the implemented RTOS and a GPOS patched with real-time extensions. In some cases, Global-EDF considering the overhead of the RTOS is superior to Partitioned-EDF considering the overhead of the patched GPOS, which clearly shows how different OSs impact hard real-time schedulers. Third, an evaluation of the cache partitioning impact on partitioned, clustered, and global real-time schedulers is performed. The results indicate that a lightweight RTOS does not impact real-time tasks, and shared cache partitioning has different behavior depending on the Scheduler and the task's working set size. Fourth, a task partitioning Algorithm that assigns tasks to cores respecting their usage of cache partitions is proposed. The results show that by simply assigning tasks that shared cache partitions to the same processor, it is possible to reduce the contention for shared cache lines and to provide HRT guarantees. Finally, a two-phase multicore scheduler that provides HRT and SRT guarantees is proposed. It is shown that by using information from hardware performance counters at run-time, the RTOS can detect when best-effort tasks interfere with real-time tasks in the shared cache. Then, the RTOS can prevent best-effort tasks from interfering with real-time tasks. The results also show that the assignment of exclusive partitions to HRT tasks together with the two-phase multicore scheduler provides HRT and SRT guarantees, even when best-effort tasks share partitions with real-time tasks.

UFSC / CTC / LISHA
PO Box 476
88040-900 Florianópolis - SC - BRAZIL

Phone: +55 48 3721-9516
E-mail: lisha@lisha.ufsc.br
Web: http://www.lisha.ufsc.br