Center for Embedded Systems

An NSF Industry/University Cooperative Research Center

Multi-Partitioned Single Cores and Predictable Execution of Safety-Critical Tasks

PIs:

Dimitri Kagaris

Harini Ramaprasad

Southern Illinois University Carbondale

SIU Southern Illinois University Carbondale

National Science Foundation

ASU Ira A. Fulton Schools of Engineering Arizona State University

Project Overview and Description

- **Motivation**
  - Single partition/module on a core may not fully utilize core resources
  - Deploy multiple partitions on single core
- **Problem**
  - Deterministic execution of HSS tasks in presence of LSS ones is challenging when executed in multiple partitions on single core
- **Viable Solution**
  - Virtualization for isolation of HSS & LSS task sets on same core
- **Application** → Mechanism for Integrated modular avionics
- **Project Description**
  - Characterize interference between HSS and LSS tasks running under the same or different partitions on a single core
  - Develop dynamic partition scheduling algorithms → ensure HSS requirements and provide better Quality-of-Service for LSS tasks

Approach

- **Two-stage approach**
  - **Stage 1:** Explore interference between HSS and LSS tasks executing within one OS, with emphasis on determinism and responsiveness of HSS tasks and QoS of LSS tasks.
  - **Stage 2:** Explore interference between tasks executing in different OSs (e.g., a real-time and general purpose OS) running on a single core with hypervisor support and study the behavior of the OSs in each case.
- **Metrics for measuring success of partition scheduling algorithm**
  - HSS tasks: determinism of execution (satisfaction of timing and precedence constraints
  - LSS tasks: Quality-of-Service (QoS), measured by response time
- **Benefits to member companies**
  - Current schemes rely on static scheduling of partitions
  - Results could enable safe, yet dynamic partition scheduling

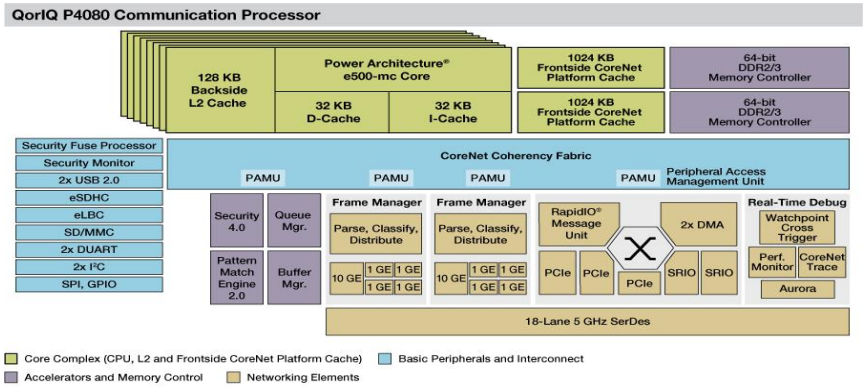
Project Tasks/ Deliverables

	Description	Date	Status
1	Exploration of existing research in the area of mixed-criticality systems and development/evaluation of new scheduling algorithms	Q1	Not yet started
2	Characterization of HSS and LSS tasks executing in multiple partitions on a single core with hypervisor support (XEN, KVM)	Q2	Not yet started
3	Characterization of HSS and LSS tasks executing in multiple partitions on a single core with hypervisor support (XEN, KVM)	Q3	Not yet started
4	Report writing and technology transfer	Q4	Not yet started

Executive Summary

- **Problem**
  - Deterministic execution of highly safety-sensitive (HSS) tasks in presence of less safety-sensitive (LSS) ones is challenging when executed in multiple partitions on a single computational core.
- **Viable solution**
  - Virtualization (hypervisor) for isolation of HSS & LSS task sets.
- **State-of-the-art**
  - Characterization of the level of interference between HSS and LSS tasks running under the same or different partitions on a single computational core.
- **Goals of proposed project**
  - Explore time and space partitioning of single-core resources among HSS & LSS tasks executing in separate virtualized software environments
  - Conduct trade-off study of determinism versus performance HSS-LSS & HSS-HSS interactions within the same OS and different OSs and develop partition scheduling algorithms to maintain HSS requirements and providing better Quality-of-Service (QoS) for LSS tasks

Technical Detail



- **Freescale QorIQ P4080**
  - **Xen** based Virtualization
    - Can work in both para-virtualized and Fully-Virtualized (Hardware Virtual machine, HVM) mode
    - Offers high performance in para-virualized mode.
  - **KVM** based Virtualization
    - Full virtualization solution for Linux
    - Designed as a loadable kernel module
  - High-bandwidth communication & coherence infrastructure
    - Support for prioritization, bandwidth allocation, packet-level queue management and QoS scheduling

Technical Detail

- **Xen Hypervisor**
  - Type 1 hypervisor
  - Small Memory footprint and limited interface to the guest OS
  - Contains special domain “Dom 0” that provides drivers for hardware that controls VMs
    - VMs can access hardware directly
  - Provides driver isolation
    - Xen allows main device driver of system to run inside of virtual machine. If driver crashes, or is compromised, VM containing driver can be rebooted and driver restarted without affecting the rest of the system
  - Supports two types of guests and both can co-exist on single core
    - Para-virtualized OS (PV)
    - Full Virtualized OS (HVM – Hardware assisted Virtual machine)

Technical Detail

- **KVM Hypervisor**
  - Full Virtualization solution for Linux
  - Requires processor with hardware virtualization extension
  - Multiple partitions can be run without OS modification
  - Does not perform emulation
    - Exposes /dev/kvm interface through which host can setup guest virtual machine

References

- S. Xi, J. Wilson, C. Lu and C. Gill. “RT-XEN: Towards Real-Time Hypervisor Scheduling in Xen.” In Proceedings of the 9th ACM international conference on Embedded Software (EMSOFT), 2011.
- Huber, B. ; El Salloum, C. ; Obermaisser, R. “A resource management framework for mixed-criticality embedded systems,’’ Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE.
- C. Ault. “Challenges of safety-critical multi-core systems.” White paper, Wind River Research.
- Hyun-Wook Jin ; Sanghyun Han, “Temporal partitioning for mixed-criticality systems”, Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference
- Wasicek, A. ; El-Salloum, C. ; Kopetz, H., “A System-on-a-Chip Platform for Mixed-Criticality Applications,” Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2010 13th IEEE International Symposium
- Jun Zhang; Kai Chen; Baojing Zuo; Ruhui Ma; Yaozu Dong; Haibing Guan, “Performance analysis towards a KVM-Based embedded real-time virtualization architecture.” 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), 2010 , pp. 421 – 426.
- Tamaş-Selicean, D. ; Pop, P., “Optimization of Time-Partitions for Mixed-Criticality Real-Time Distributed Embedded Systems,” Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2011 14th IEEE International Symposium
- Baruah, S. ; Bonifaci, V. ; D'Angelo, G. ; Li, H. ; Marchetti-Spaccamela, A. ; van der Ster, S. ; Stougie, L., “The Preemptive Uniprocessor Scheduling of Mixed-Criticality Implicit-Deadline Sporadic Task Systems,” Real-Time Systems (ECRTS), 2012 24th Euromicro Conference
- Baruah, S. ; Fohler, G, “Certification-Cognizant Time-Triggered Scheduling of Mixed-Criticality Systems,” Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd
- Lemerre, M. ; Ohayon, E. ; Chabrol, D. ; Jan, M. ; Jacques, M.-B., “Method and Tools for Mixed-Criticality Real-Time Applications within PharOS,” Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2011 14th IEEE International Symposium
- Pathan, R.M., “Schedulability Analysis of Mixed-Criticality Systems on Multiprocessors,” Real-Time Systems (ECRTS), 2012 24th Euromicro Conference.
- de Niz, D. ; Lakshmanan, K. ; Rajkumar, R., “On the Scheduling of Mixed-Criticality Real-Time Task Sets,” Real-Time Systems Symposium, 2009, RTSS 2009. 30th IEEE
- Adventium Labs. “MiCART – Mixed-Criticality, Real-Time Virtualization Support.”
  - www.adventiumlabs.com/?q=productsandservices/micart.
- “Lmbench – Tools for Performance Analysis.” http://www.bitmover.com/lmbench/.
- “MRTC WCET Benchmarks.” www.mrtc.mdh.se/projects/wcet/benchmarks.html.