XtratuM hypervisor for mixed-criticality systems

M. Masmano, J. Coronel
fentISS
Valencia, Spain
{mmasmano, jcoronel}fentiss.com

P. Balbastre, A. Crespo, J. Simó, S. Peiró
Universitat Politecncia de Valencia
Valencia, Spain
{pbalbastre, acrespo, jsimo, speiro}@ai2.upv.es

***Abstract*—This demonstration presents the configuration, compilation and deployment processes of applications based on XtratuM hypervisor.**

Keywords—hypervisor, partitioned systems, real-time embedded systems

I. ABSTRACT

Embedded Real-time hypervisors has recently emerged as technology to achieve the most robust, secure and reliable real-time systems with limited overheads. Partitioned systems have been successfully used in avionics [1] and now, it has been extended to other fields of application as space, automotive and control systems.

XtratuM [2][3][4] is an open source hypervisor specifically designed for embedded real time systems. XtratuM hypervisor design is based on the principles stated in the Integrated Modular Avionics IMA approach. It was initially designed for mono-core and redesigned to support multicore. XtratuM hypervisor is used in different European project [5][6] to support mixed-criticality systems. Partitions in XtratuM are execution environment including the application and the operating system. Several partitions with different guestOSs can be used in an application. The basic XtratuM features are:

- Temporal isolation: fixed scheduling of CPU time.
- Spatial isolation: fixed assignment of physical memory and device resources.
- Static Resource assignment: The system designer assigns the platform resources (time, memory and devices) to partitions.
- Spare Server: Spare time can be assigned dynamically to partitions on demand.
- IOServer: When devices must be shared, the partition “IOServer” provides virtual devices to other partitions.
- System Manager: Privileged (and trusted) “system partition” can monitor and manage other partitions

In this demonstration, the XtratuM approach is presented jointly with the set of tools to deploy embedded real-time systems. The demonstration includes:

1. Configuration and compilation of XtratuM
2. Configuration and development of partitions with different operation systems: bare partitions, real-time POSIX operating system and Linux.
3. Configuration of the system: A configuration file describes the scheduling of partitions, the allocation of resources to partitions and the communication channels to exchange information between partitions
4. System deployment tools to build the final system
5. Execution of the system in a platform

XtratuM implements a cyclic scheduling policy (based on ARINC-653 partitioning kernel policy). Based on this policy, XtratuM permits the execution of more critical applications (partitions) with a static allocation of temporal windows and offers the capability to handle non critical applications in specific temporal windows labeled as spare temporal windows. The schedule of these temporal windows is performed at partition level by a system partition that collects the application needs and builds dynamically the plan for them. Some services at hypervisor level permit to set and modify the internal allocation in spare temporal windows.

The demonstrator will show an application with several partitions (Linux based, real-time OS based, bare) (see figure 1) and its execution in the platform (figure 2)

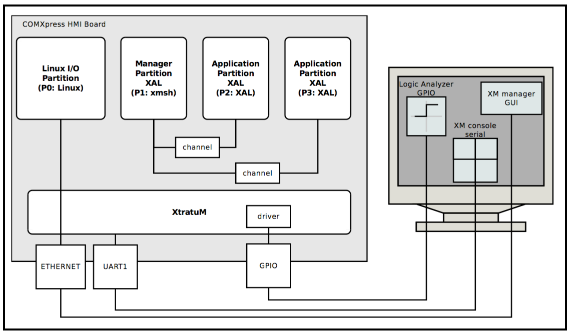

Figure 1. System architecture

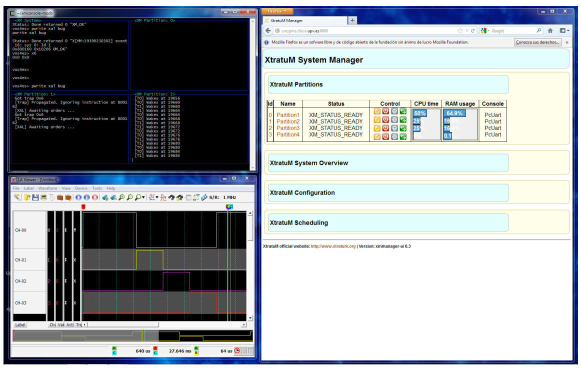

Figure 2. Execution demonstrator

REFERENCES

[1] Avionics Application Software Standard Interface (ARINC-653). PART 1 RE- QUIRED SERVICES, March 2006. Air- lines Electronic Eng. CommitteeXM.

[2] M. Masmano, I. Ripoll, A. Crespo, and J.J. Metge. Xtratum: a hypervisor for safety critical embedded systems. In 11th Real-Time Linux Workshop, Dres- den (Germany), 2009.XM.

[3] M. Masmano, I. Ripoll, A. Crespo, J.J. Metge, and P. Arberet. Xtratum: An open source hypervisor for TSP embed- ded systems in aerospace. In DASIA 2009. DAta Systems In Aerospace., Istan- bul (Turkey), May 2009..

[4] M. Masmano, I. Ripoll, S. Peir´o, and A. Crespo. Xtratum for LEON3: An open source hypervisor for high integrity systems. In European Conference on Em- bedded Real Time Software and Systems. ERTS2 2010., Toulouse (France), 19-21 May 2010.

[5] OVeRSEE. Open VEhiculaR SEcurE platform. FP7 Project 248333. www.oversee.eu

[6] MultPARTES. Multi-cores trusted partitioning for embedded systems. EU FP7 Project 287702. http://www.multipartes.eu.

Xamber: a configuration and scheduling tool for partitioned multicore systems

J. Simó, P. Balbastre, Y. Valiente, A. Crespo.
Universitat Politecncia de Valencia
Valencia, Spain
{jsimo, pbalbastre, yovamo, acrespo}@ai2.upv.es

V. Brocal
fentISS
Valencia, Spain
vbrocal@fentiss.com

***Abstract*— This demonstration presents the Embedded Partitioned Systems modeling procedure assisted by specific tools intended to speedup the System Designer configuration and integration work.**

Keywords—hypervisor, partitioned systems, real-time embedded systems, model-based design tools.

I. ABSTRACT

Embedded Partitioned Systems (EPS) infrastructures provide temporal and spatial isolation required to support Mixed Criticality applications. The goal of these systems is to execute applications in independent sandboxes on top of the hypervisor software with provides an abstraction of virtual machine with strong resources assignment.

In order to ensure the critical features of temporal and spatial isolation, the System Designer must define in design time, the resources assignment for each partition running in a sandbox. The configuration of an EPS deployment includes:

- Platform available resources
- Memory allocation for each partition
- IO ports and interrupts assigned to each partition
- One or more cyclic plans representing the CPU time slots assigned to run each partition.
- Inter-partition communication channels and partition's access ports for each partition.

The EPS deployment container includes the hypervisor image itself, the code of all partitions and the above outlined configuration data. During the building procedure, the System Designer must provide the configuration data in the form of a model (usually a XML file) to Partition Developers. The Figure 1 summarizes the development process.

The configuration file (XMCF) is a kind of contract between the System Integrator and the Partition Developers. Each Partition Developer must provide to the System Integrator all resources required to run the partition successfully. From these requirements, the System Integrator elaborates the configuration file.

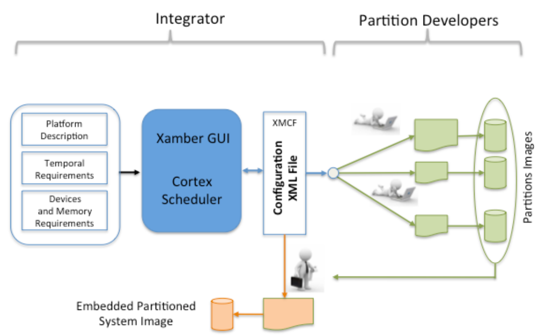

Figure 1. EPS development process

Because the elaboration of the configuration file is a complex task involving memory and resources overlapping control, communication ports correspondence and scheduling, it is necessary to provide a specific model-based set of tools. The current implementation of this helping software integrates a Graphical Editor (Xamber) with an external scheduling tool (Cortex). In the future other independent tools (p.e. certified validation) could be integrated.

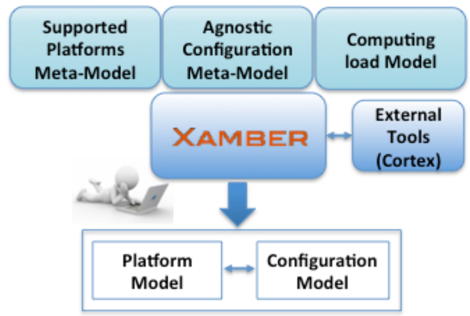

Figure 2. Xamber model's interaction

Xamber provides to the System Integrator an environment where define the platform properties, resources assignment and cyclic plans in a graphical way. Model restrictions are applied on-line in order to alert the user about configuration errors, warnings or pending definitions.

As depicted in Figure 2, the interaction of models and meta-models provides Xamber some degree of flexibility. In design time the Agnostic Configuration Meta-Model is narrowed with the information of a specific platform. The Computing Load Model, inspired in MARTE UML, give Xamber support to the definition of scheduling requirements in an abstract way.

Moreover, the power of the Cortex scheduler provides automatic plan generation from a graphically defined Computing Load Model (tasks specification, resource dependences and end-to-end computing flows involving tasks running in different partitions). Xamber organizes all this information in a “Configuration Project” allowing Import/Export operations of project parts as well as specific hypervisor file generation (currently supported XtratuM 4.0 [1][2][3]).

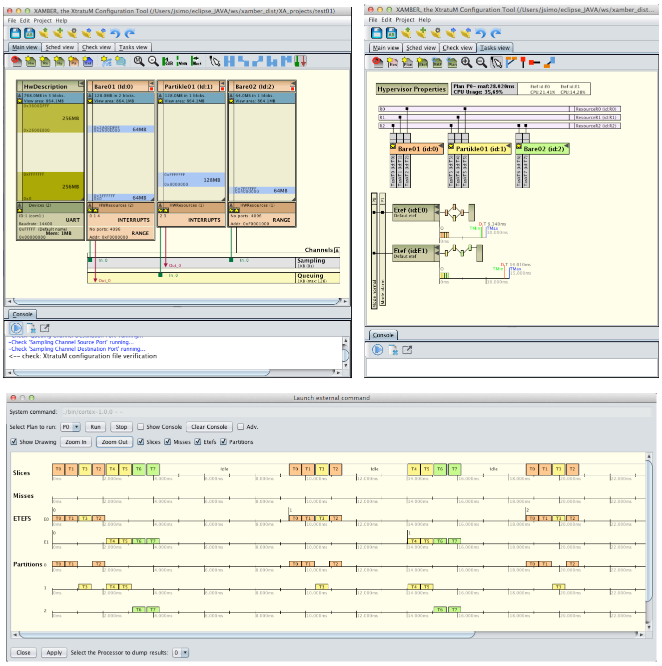

Figure 3. Xamber Screenshots.

The demonstration will show the process of EPS configuration assisted by Xamber and Cortex, exploring models, different useful options and graphical metaphors. Figure 3 shows some screenshots of Xamber.

ACKNOWLEDGEMENTS

This work has been done under the MultiPARTES project (FP7 – 287702), HI-PartES (TIN2011-28567-C03-01-02-03) and COBAMI (DPI2011-28507-C02-02).

REFERENCES

[1] M. Masmano, I. Ripoll, A. Crespo, and J.J. Metge. Xtratum: a hypervisor for safety critical embedded systems. In 11th Real-Time Linux Workshop, Dres- den (Germany), 2009.XM.

[2] M. Masmano, I. Ripoll, A. Crespo, J.J. Metge, and P. Arberet. Xtratum: An open source hypervisor for TSP embed- ded systems in aerospace. In DASIA 2009. DAta Systems In Aerospace., Istan- bul (Turkey), May 2009..

[3] M. Masmano, I. Ripoll, S. Peiró, and A. Crespo. Xtratum for LEON3: An open source hypervisor for high integrity systems. In European Conference on Em- bedded Real Time Software and Systems. ERTS2 2010., Toulouse (France), 19-21 May 2010.