MultiPARTES: Virtualization of Heterogeneous Multicore

Salvador Trujillo
Alfons Crespo
Juan Antonio de la Puente

MultiPARTES FP7 Project

MultiPARTES Project

- **Objective:** Support *mixed criticality* systems based on heterogeneous multicore open source virtualization

- **Project details**
  - IKERLAN-IK4 Project coordinator
  - 2.850.000 Euro EC Contribution
  - Sep 2011 / Aug 2014 Project start / end date
  - 36 months Duration

- **Web:** [http://www.multipartes.eu/](http://www.multipartes.eu/)
MultiPARTES Goals

1) MultiPARTES aims at developing tools and solutions based on mixed criticality and assurance-based virtualization systems for multicore. The starting point for virtualisation support is XtratuM, an open source cost-effective hypervisor.

2) MultiPARTES will offer a rapid and cost-effective development of trust real-time embedded systems sharing critical and no critical applications the system resources.
MultiPARTES Topics

- **Aim:** Support *mixed criticality systems* based on heterogeneous multicore *open source virtualization*

- **Multicore Virtualization**
  - Uniform view for partitions
  - AMP / SMP XtratuM hypervisor

- **Heterogeneity**
  - Hardware platform with different processors
  - Software architecture
  - Communication & synchronization between cores

- **Mixed criticality => Methodology and Tools**
  - System definition models: Platform, Computational, ...
  - Criteria to allocate functions to partitions
  - Scheduling tool
Multicore Virtualization

- **Based on XtratuM Multicore**
  (proof of concept in SIDMS ESA project)
  - Offers as many *virtual CPUs* as real CPUs are in the board
  - Initialises the real CPUs and offers the *virtualCPU0* to the partitions
  - Partitions are in charge of initializing other *virtualCPUs*
  - Partitions can be mono-core or multicore
Multicore Virtualization

- Temporal and Spatial Partitioning systems
  - Temporal isolation
    - Temporal allocation of partitions
    - Execution Interference of other cores
  - Spatial isolation
    - No additional problems
  - Shared resources:
    - Cache L2 and L3, bus arbitration, memory
    - Introduce unpredictable execution time in partitions
Multicore Virtualization

- Worst Case Execution Time Impact

\[ WCET = WCET_{task} + \text{Interference} \]

- The interference can be modeled
  - Evaluating the interference
  - Limiting it by construction of the scheduling plan

<table>
<thead>
<tr>
<th>Interference</th>
<th>S25</th>
<th>S50</th>
<th>S75</th>
<th>S100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atom Dual Core@1666MHz</td>
<td>1,68%</td>
<td>4,20%</td>
<td>7,31%</td>
<td>13,43%</td>
</tr>
</tbody>
</table>
Multicore Virtualization

- Worst Case Execution Time Impact

\[ WCET = WCET_{task} + \text{Interference} \]

- The interference can be modeled
  - Evaluating the interference
  - Limiting it by construction of the scheduling plan

<table>
<thead>
<tr>
<th></th>
<th>S25</th>
<th>S50</th>
<th>S75</th>
<th>S100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interference</td>
<td>15.74%</td>
<td>31.05%</td>
<td>46.39%</td>
<td>61.22%</td>
</tr>
</tbody>
</table>

LEON3 @ 50MHz
Heterogeneity

- **Integration of different hardware platforms** permits:
  - Hardware diversity
  - Specialized hardware
  - Isolation of critical => deterministic hardware

- **MultiPARTES:**
  - 1 Atom Dual Core + FPGA with 2 LEON3 with shared memory

- **Examples**
  - Space: LEON3 Multicore + ARM Cortex processor
  - Space: LEON3 Multicore + DSP
Heterogeneity: Software Arch.

ASM: Asymmetric multiprocessing
  • Each core is handled by one OS instance
SMP: symmetric multiprocessing
  • All cores are handled by the OS
Heterogeneity

- **Needs**
  - **Clock synchronization: MAF synchronization**
  - Inter-partition/Inter-platform communication
    - Double port memories
    - Bus based communication

![Diagram showing Atom Core Duo @ 1666MHz, PCI Express, LEON3, FPGA Spartan, and MAF synchronization]
### Heterogeneity

- **Needs**
  - Clock synchronization: MAF synchronization
  - Inter-partition/Inter-platform communication
    - Double port memories
    - Bus based communication

![Diagram showing Atom Core Duo @ 1666MHz connected to FPGA Spartan via PCI Express](image)
Heterogeneity: Sw

- **Execution Environments**
  - **MTPAL**: MultiPARTES Abstraction Layer. Single thread applications. Basic services (TSAL IMA-SP)
  - **PartiKle**: Real-time kernel. POSIX PSE51.
  - **ORK+**: Ada applications. Ravenscar Profile.
  - **Linux**

<table>
<thead>
<tr>
<th></th>
<th>LEON3</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTPAL</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PartiKle</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ORK+</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Linux</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTPAL</td>
<td>GPL</td>
</tr>
<tr>
<td>PartiKle</td>
<td>GPL</td>
</tr>
<tr>
<td>ORK+</td>
<td>GPL</td>
</tr>
<tr>
<td>Linux</td>
<td>GPL</td>
</tr>
</tbody>
</table>
Model-driven Partitioning

Real world

Abstraction

Modeling tool

Generated artifacts

MultiPARTES @ADCSS 2013
Input / Activities / Output

- **Input information:**
  - Platform and Application models
  - Partitioning restriction model

- **Activities:**
  - Propose a system partitioning
  - Meet the real-time, safety and security constraints

- **Outcomes:**
  - Code skeletons, XtratuM configuration files, make file
Architecture of the toolset

Model of the system

Platform model

Partitioning restrictions model

Applications model

Validation of non-functional requirements

Validation Tools

Partitioning tool

Deployment model

Transformation to neutral model

Generation of final files

Transformation for system generation

System generation files

Transformation to configuration files

XtratuM Configuration files

Transformation to source code

Source code
Industry demonstrators

- **Video surveillance**
  - X86 platform
  - MTPAL, PartiKle, Linux
  - Devices: Video cameras, data storage

- **WindPower**
  - x86 and LEON3 platforms
  - MTPAL, Partikle and Linux
  - Devices: Scada system, EtherCAT,
  - IEC 61508 Pre-certification

- **Space**
  - LEON3 platform
  - MTPAL and ORK+
  - UPMSat Platform ADCS, OBDH, Communications, Payload
Project results

- Virtualization Layer
  - XtratuM Multicore for x86 and LEON3

- Execution Environments
  - Several guestOSs have been adapted

- Methodology
  - Model-driven. Application and design levels.
  - Partitioning, Scheduling, Code Generation Tools

- Hardware issues
  - Experimentation with TTNoC
  - Experimentation in mechanisms to reduce the memory interference
Current Status

- Virtualization Layer
  - XtratuM Multicore for x86 and LEON3

- Execution Environments
  - Several guestOSs have been adapted

- Methodology
  - Model-driven. Application and design levels.
  - Partitioning, Scheduling, Code Generation Tools

- Hardware issues
  - Experimentation with TTNoC
  - Experimentation in mechanisms to reduce the memory interference
Certification aspects

- Wind power safety concept based on MultiPARTES multicore partitioning
  - Following IEC-61508
  - Presented to TÜV Rheinland
  - Positive feedback collected
  - TÜV report (under progress)

- Follow-up work in the context of DREAMS project
Benefits for Industry (Space)

- XtratuM has been evolved based on Space requirements
  - Maturity increased
  - Reusable Test suites
- Additional execution Environments
  - MTPAL: multicore evolution of TSAL (ESA IMA-SP project)
  - ORK+: developed under ESA contracts.
- Methodology
  - Partitioning criteria
  - Partitioning, Scheduling, Code generation tools
  - Hypervisor partitioning configuration generation
- Hardware
  - LEON3 multicore experimentation
Conclusions

- Snapshot halfway of the project
- Challenging field with industrial interests (there is some competition going on ...)
- Collaborative effort together with other projects
- Transferring advance technology to the industry
- A new line of embedded systems is being conceived

Beyond our project
- From multi-core to many-core ... does it make sense in industry? Where?
- Certification of the approach. Work together with certification body?
- Availability of commercial HW, HW mechanisms
- Ease integration of legacy code
Thanks for your attention

Support mixed criticality systems based on heterogeneous multicore open source virtualization

http://www.multipartes.eu/