Heterogeneous Real-Time SoC Software Architecture

Presented By

Stefano Stabellini
Principal System Software Engineer
Introduction

> Stefano Stabellini
  >> Xen Project:
    - Founder of the Xen on Arm effort in late 2011
    - Xen on ARM Maintainer and Committer, Linux Maintainer
    - Develops Xen Project features on Zynq UltraScale+ MPSoC

  >> Xilinx:
    - System Software Architect focusing on heterogeneous systems
    - Upstreaming Xilinx support to Xen and OpenAMP projects
Virtualization Basics
Virtualization – The Concept

> “Virtualization”
  >> The act of creating a virtual version of something, including virtual computer hardware platforms, storage devices, and computer network resources.
  >> Allows the deployment of multiple operating systems and independent workloads on one or more processors

> “Hypervisor”
  >> A hypervisor or virtual machine monitor is computer software, firmware or hardware that creates and runs virtual machines.

> Why Virtualize?
  >> OS/Workload consolidation
  >> Lower system cost
  >> Lower power consumption
  >> Improved resource utilization
    – Mixed Criticality Systems
  >> Fault tolerance
  >> Multi-tenancy
  >> Portability
Why Virtualize?

- RTOS
- Linux
- Hypervisor
- CPU
Why Virtualize?

- RTOS
- Drivers Domain
- Linux

Hypervisor

- CPU #0
- CPU #1
- CPU #2
- CPU #3
Embeded Hypervisor Requirements

> Short Boot Times

> Real time
  >> Low, deterministic IRQ latency
  >> Real time schedulers
  >> Static CPU partitioning

> Device Virtualization
  >> Device Assignment
  >> Device Sharing
  >> Driver Domains
  >> VM to VM communication

> Security, Isolation and Partitioning
  >> Memory
  >> Devices
  >> CPU
  >> SLCRs

> Operating System Support
  >> Linux, bare-metal, other RTOS support

> Certifications
  >> Small code base
  >> Type-1
Xen Project

> Xen Project
  >> Open source hypervisor
  >> Small code base implementing a micro-kernel design
  >> Xen Project hosted by the Linux Foundation

> Broad, Customizable Feature Set
  >> From servers to embedded
  >> Out of box “real time” schedulers and enhancements
  >> Advanced device management, partitioning, assignment
  >> Independent user, control, and driver domains

> Linux, BSDs or other OSes used for bootstrap (dom0)
  >> Linux is the most widely used but other OSes are possible
Example Xen Architecture

- **dom0 – control**
  - Dom0 services
  - Minimal rootfs
  - Linux kernel w/o HW drivers

- **domD – HW drivers & Cluster**
  - Wayland apps
  - Wayland/Weston
  - OpenGL ES
  - ALSA w/ PV ALSA BE
  - Linux kernel w/ HW drivers

- **domU – FUSON**
  - Containers
  - Container mgmt tools
  - Base rootfs
  - Linux kernel w/o HW drivers

- **domU – Linux IVI**
  - HMI & Apps
  - MW Frameworks
  - PV ALSA FE
  - PV events FE
  - PV display FE
  - Linux kernel w/o HW drivers

---

© Copyright 2018 Xilinx
Xen Project 4.11

> **Highlights**
  >> Regression testing and hardware validation completed successfully
  >> Enormous work for the Meltdown and Spectre mitigations
  >> Configurable SErrors handling
  >> Many reliability fixes, especially in the interrupt handling path (GIC, vGIC)
  >> SMCCC 1.1

> **Highlights (cont.)**
  >> RTDS scheduler improvements
  >> "null" scheduler improvements: tracing, soft affinity
  >> VPL011
  >> Mem_Access improvements
  >> new PV Drivers: PV Display, PV Audio, PVCalls, PV 9pfs

> **Features and Status**
  >> [Xen Project 4.11 Feature List](#)
Mem_Access

```c
uint32_t flags;
uint32_t vcpu_id;
uint64_t gfn;
...
mem_event_regs_t regs;
```
PV Drivers

- Existing: net, block, console, keyboard, mouse, framebuffer
- New: 9pfs, PVCalls, Multi Touch, Sound, Display
- Prerequisites: xenstore, grant table and event channels support (BSD code available)
Static Partitioning Use-Case

sched=null vwfi=native
Static Partitioning Use-Case

sched=null  vwfi=native

2.5 us
Static Partitioning Latency

Xilinx Zynq Ultrascale+ MPSoC
Physical Timer

Xen with phys_timer patch
vwi=native

dom0_mem=1G
max_dom0_vcpus=2
1 vcpu TBM ctest
Xen Schedulers

CPU

CPU

CPU

CPU
Xen Schedulers

- sched=null
- sched=credit

Diagram showing vCPU scheduling.
Xen VM-to-VM communication mechanisms

> **Libvchan**
  - Linux library
    - Direct VM to VM communication channel based on a ring on shared memory
    - libxenvchan_send and libxenvchan_recv

> **PVCalls**
  - Socket API virtualization
    - VM to VM communication mediated by the backend domain (typically dom0)
    - "lo" becomes a inter-VMs communication namespace

> **V4V**
  - Linux library and hypercall, kernel space and user space
  - VM to VM communication mediated by Xen
    - Trivial to implemented on your own kernel
    - Not fully upstream
Brand New Features

Introduction Slide
Shared Memory

- **Completely Configurable**
  - Support any memory attributes, including cacheable memory (default)
- **No need for Xen support to use it**
- **Can export the memory to Linux userspace and use OpenAMP**

```plaintext
static_shm = ["id=ID1, begin=0x40000000, size=0x1000, role=master"]
static_shm = ["id=ID1, offset=0, begin=0x48000000, size=0x1000, role=slave"]
```
Reducing Code Size

```
make cloc

cloc --list-file=/tmp/tmp.L2EdV9dLA
  143 text files.
  143 unique files.
  0 files ignored.

http://cloc.sourceforge.net v 1.60  T=0.26 s (546.4 files/s, 262525.6 lines/s)

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>126</td>
<td>10527</td>
<td>10408</td>
<td>45144</td>
</tr>
<tr>
<td>Assembly</td>
<td>17</td>
<td>249</td>
<td>937</td>
<td>1439</td>
</tr>
<tr>
<td>SUM:</td>
<td>143</td>
<td>10776</td>
<td>11345</td>
<td>46583</td>
</tr>
</tbody>
</table>

rm /tmp/tmp.L2EdV9dLA
```

© Copyright 2018 Xilinx
## Certifications

![Certification Report](image)

**Files**

<table>
<thead>
<tr>
<th>Files</th>
<th>Version</th>
<th>MISCM-1-1 Rules</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>arm/trace.c</td>
<td>0x0009525</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/ramdisk.c</td>
<td>aa7e67a61</td>
<td>78</td>
<td>78</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/schedule.c</td>
<td>14001a86</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>arm/domain.c</td>
<td>0x0028f26</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/domain.c</td>
<td>0x002b4f5</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/wrench.c</td>
<td>0x002b4f0</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>flash/loop.c</td>
<td>0x002b4f0</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/now.c</td>
<td>0x002b4f0</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/trace.c</td>
<td>0x002b4f0</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>flash/roc.c</td>
<td>0x002b4f0</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/ema.c</td>
<td>0x002b4f0</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/keyhandler.c</td>
<td>0x002b4f0</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>flash/key.c</td>
<td>0x002b4f0</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>PROJECT_ROOT/xen/common/keyrock.c</td>
<td>0x002b4f0</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>arm/trace.c</td>
<td>0x002b4f0</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>xen/xen/trace.c</td>
<td>0x002b4f0</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>
Dom0-less

U-Boot

loads into memory

loads into memory

Xen

Dom0 / DomU

DomU 1

DomU 2

CPU0

CPU1

CPU2
Dom0-less

U-Boot

Xen

Dom0 / DomU

DomU 1

DomU 2

CPU0

CPU1

CPU2
Xen Project "OSSTests"

> **OSSTests: Xen Project official CI-loop**
  >> Run 24/7
  • Commits move to master only after passing the CI-loop tests
  • Based in Boston, MA
  • Only accept off-the-shelf hardware

> **Xilinx MPSoC ZCU102 coming to Xen Project!**
  >> Will validate master on Xilinx hardware
  • Every Xen release will be checked against Xilinx hardware
  • Increase overall quality
  • Reduce risks of rebasing Xen in Petalinux
"The best security process in the industry"

> A very transparent process
> Responsible disclosure
> Only few security issues for Xen on ARM
> Xen stable trees maintained for security for 3 years
Commercial Xen Support

> **DornerWorks**
  >> Xilinx Premier Design Services Partner
  >> Hardware, software and systems expertise
  >> Xilinx partner for Xen support and design customization services

> **Community Support**
  >> Free [Community Support](#) is available to the entire Zynq UltraScale+ MPSoC community.
  >> This support includes all software for Virtuosity™, plus all supported configurations or workflows that are documented by the distribution.

> **DornerWorks Xen commercial support**
  >> Custom hardware porting
  >> New guest OS support
  >> Custom device drivers
  >> Programmable Logic integration
  >> System architecture design
  >> Scheduling and partitioning for ARINC 653 and FACE

> [http://dornerworks.com/xen](http://dornerworks.com/xen)
Other Hypervisors
Jailhouse

> **Open source hypervisor**
> >> [https://github.com/siemens/jailhouse](https://github.com/siemens/jailhouse)

> **Lightweight implementation**
> >> Focus on resource partitioning and not on virtualization
>     - No schedulers, no PV devices, no Driver Domains, etc.

> **Features**
> >> Optimized for simplicity rather than feature richness
> >> Relatively new ARM64 support

> **Linux used for bootstrap and control of partitions**

> **Commercially supported on Zynq UltraScale+ MPSoC by Enea**
Commercial Hypervisors

- DornerWorks (Xen, seL4)
- General Dynamics Mission Systems (OKL4 Microvisor®)
- Green Hills Multivisor®
- Lynx LynxSecure®
- Mentor Embedded Hypervisor
- BlackBerry QNX® Hypervisor
- Sysgo PikeOS® Hypervisor
- Wind River Virtualization Profile