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

CPU
Why Virtualize?

- RTOS
- Drivers Domain
- Linux

Hypervisor

- CPU #0
- CPU #1
- CPU #2
- CPU #3
Embedded 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

> 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
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

SVA (domU₀)

Memory Introspection Engine

uint32_t flags;
uint32_t vcpu_id;
uint64_t gfn;
...
mem_event_regs_t regs;

domU₁

Xen Hypervisor
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
Xilinx Zynq Ultrascale+ MPSoC
Physical Timer

Xen with phys_timer patch
vwfi=native

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

CPU

CPU

CPU

CPU
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

```bash
make cloc
cloc --list-file=/tmp/tmp.L2EdV9dL
  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)

Language     files  blank  comment  code
          ---------- ------------ ------------    ---
       C       126     10527    10408        45144
     Assembly   17      249     937         1439
        SUM:    143   10776    11345      46583

rm /tmp/tmp.L2EdV9dL
```

© Copyright 2018 Xilinx
Certifications
Dom0-less

U-Boot

loads into memory

Xen

loads into memory

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

> 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