Static Partitioning with Xen

Dom0-less Xen configurations

Stefano Stabellini – ELC August 2019
Static Partitioning

Processing System

Application Processing Unit
- ARM® Cortex™-A53
  - 32KB 4-way set-associative L1 Cache
  - 32KB D-Cache w/ECC
  - Memory Management Unit
  - Embedded Trace Macrocell
  - GIC-400
  - SCU
  - CCISMMU
  - 1MB L2 with ECC

Memory
- DDR4/3L, LPDDR4/43
  - 32GB 4-bit w/ECC
  - 256KB OCM with ECC
  - 64KB L2 Cache

Graphics Processing Unit
- ARM Mali™-400 MP2
  - Geometry Processor
  - Pixel Processor
  - Memory Management Unit
  - 64KB L2 Cache

Real-Time Processing Unit
- ARM Cortex™-R5
  - Vector Floating Point Unit
  - 128KB TCM w/ECC
  - 32KB 8-way set-associative L1 Cache w/ECC
  - 32KB D-Cache w/ECC

Platform Management Unit
- System Management
- Power Management
- Functional Safety

Configuration and Security Unit
- Config AES
- Decryption, Authentication, Secure Boot
- Voltage/Temp Monitor
- TrustZone

System Functions
- Multichannel DMA
- Timers, WDT, Resets, Clocking & Debug

High-Speed Connectivity
- DisplayPort v1.2a
- USB 3.0
- SATA 3.1
- PCIe® 1.0 / 2.0
- PS-GTR

General Connectivity
- GigE
- USB 2.0
- CAN
- UART
- SPI
- Quad SPI NOR
- NAND
- SD/eMMC

Programmable Logic
- UltraRAM
- DSP
- System Monitor
- General-Purpose I/O
  - High-Performance HP I/O
  - High-Density HD I/O
- High-Speed Connectivity
  - GTH
  - GTY

© Copyright 2019 Xilinx
Static Partitioning
Static Partitioning: the use-case

- Mixed Criticality
- Safety
- Security
- Real-Time
- Fault isolation
- Compartmentalization
- Multiple OSes
Static Partitioning with Xen: the goal
Static Partitioning with Xen: the goal

U-Boot

Xen

Type 1 hypervisor
A great match to setup static partitions

DomU
Hardware Partition
CPU + Devices

DomU
Hardware Partition
CPU + Devices

DomU
Hardware Partition
CPU + Devices

DomU
Hardware Partition
CPU + Devices
The problem

> Requires Dom0
  > Dom0 is privileged
  > Dom0 is typically Linux

> Boot Time
  > total > xen + dom0_kernel + dom0_user

> Safety Certifications
  > non-Linux Dom0
  > exit Dom0 after boot

> Complexity
  > build-time complexity
    - Yocto rootfs build
  > runtime flexibility
    - Monitoring
    - VMs restart
Introducing Dom0-less
Dom0-less System Configuration and Boot

- U-Boot loads into memory
- Xen loads into memory
- Dom0
- DomU 1
- DomU 2
- CPU
- CPU
- CPU
Dom0-less System Configuration and Boot

U-Boot

boots

Xen

boots

Dom0

DomU 1

DomU 2

CPU

CPU

CPU
Dom0-less System Configuration and Boot

U-Boot

Xen

DomU 3

Frontend

Dom0

Backend

xl

DomU 1

DomU 2

CPU

CPU

CPU

CPU

Dom0-less VMs
U-Boot + Device Tree protocol

> Load all the required binaries via U-Boot commands

```
tftpb 0x4000000 dom0less/mpsoc.dtb
tftpb 0x80000 dom0less/Image-dom0
tftpb 0x5000000 dom0less/uXen
tftpb 0xd000000 dom0less/dom0-ramdisk.cpio.uboot

tftpb 0xa000000 dom0less/Image-domU
tftpb 0xb000000 dom0less/domU-ramdisk.cpio

bootm 0x5000000 0xd000000 0x4000000
```
U-Boot + Device Tree protocol

- Advertise and configure Dom0-less VMs via Device Tree

```
domU1 {
    compatible = "xen,domain";
    memory = <0x0 0x20000>;
    cpus = 1;
    vpl011;

    module@a000000 {
        compatible = "multiboot,kernel", "multiboot,module";
        reg = <0xa000000 0xffffff>;
        bootargs = "console=ttyAMA0";
    }

    module@b0000000 {
        compatible = "multiboot,ramdisk", "multiboot,module";
        reg = <0xb000000 0xffffff>;
    }
}
```
Dom0-less Device Assignment

U-Boot

boots

Xen

boots

Dom0
- Other Devices

DomU 1
- Network Card

DomU 2

CPU

CPU

CPU
Dom0-less Device Assignment

> Configured via a nested device tree snippet

domU1 {
  ...

  module@a000000 {
    compatible = "multiboot,kernel", "multiboot,module";
    reg = <0xa000000 0xffffff>;
    bootargs = "console=ttyAMA0";
  }
}

module@b0000000 {
  compatible = "multiboot,ramdisk", "multiboot,module";
  reg = <0xb000000 0xffffff>;
}

module@c0000000 {
  compatible = "multiboot,device-tree", "multiboot,module";
  reg = <0xc000000 0xffffff>;
}
};
Dom0-less Device Assignment

> Configured via a nested device tree snippet
  >> the device tree node of the device to assign
  >> same as for regular DomUs
  >> Special properties:
    – xen,path: path to the device node in the main DT
    – xen,reg: memory to remap

```
/dts-v1/;
/
{
  #address-cells = <0x2>;
  #size-cells = <0x1>;

  passthrough {
    compatible = "simple-bus";
    ranges;
    #address-cells = <0x2>;
    #size-cells = <0x1>;

    ethernet@ff0e0000 {
      compatible = "cdns,zynqmp-gem";
      status = "okay";
      xen,path = "/amba/ethernet@ff0e0000";
      xen,reg = <0x0 0xff0e0000 0x1000 0x0 0xff0e0000>;
      reg = <0x0 0xff0e0000 0x1000>;
      clock-names = "pclk", "hclk", "tx_clk", "rx_clk";
      #address-cells = <0x1>;
      #size-cells = <0x0>;
    }

    [...]
```
True Dom0-less
Static Partitioning with Xen

U-Boot → Xen

Type1 hypervisor
A great match to setup static partitions

DomU → Hardware Partition CPU + Devices
DomU → Hardware Partition CPU + Devices
DomU → Hardware Partition CPU + Devices
DomU → Hardware Partition CPU + Devices
More than static partitioning

- U-Boot
  - boots
  - Xen
    - boots
      - DomU 3 (Non-Critical)
      - DomU 1 (Safety Critical)
      - DomU 2 (Safety Critical)

- Dom0
- xl
- Network Card
- PL Block

CPU
CPU
CPU
CPU
Dom0-less Pros & Cons

Pros:

˃ Much **faster** startup times
  >> total ~= xen + domU

˃ Enable static partitioning configurations
  >> Excellent for small systems
  >> Easier to certify

˃ Lower Complexity
  >> No need for the Xen tools
  >> Does not require Yocto, just cross-build Xen
  >> No need for Xen support in Dom0-less VMs, no need for CONFIG_XEN

Cons:

˃ No monitoring and restarting DomUs without Dom0
˃ No PV frontends/backends without Dom0
Dom0-less and PV drivers

U-Boot boots Xen

DomU 1
- Netback
- Network Card

DomU 2
- Netfront

CPU

Doesn’t work today!
Status & TODO

> DONE
  >> basic Dom0-less booting upstream in Xen 4.12
  >> device assignment implemented and sent to the list (not upstream)

> TODO
  >> True Dom0-less
  >> Shared memory and interrupts for VM-to-VM communications
  >> PV frontends/backends drivers for Dom0-less VMs
Demos
True Dom0-less

U-Boot

boots

Xen

boots

boots

DomU 1

Network Card

CPU

DomU 2

UART TTC

CPU
Dom0-less and PVCalls

U-Boot

boots

Xen

Dom0

DomU 1

Network Card

DomU 2

Frontend

Backend

DomU 3

CPU

CPU

CPU

CPU

Private Network
Questions?
Adaptable.
Intelligent.
Slides Parking Lot
Traditional Xen System Configuration and Boot

U-Boot

boots

Xen

Dom0
quits

DomU 1
CPU1

DomU 2
CPU2