Expanding the CAmkES-ARM-VM

DornerWorks

Technology engineering so you can focus.

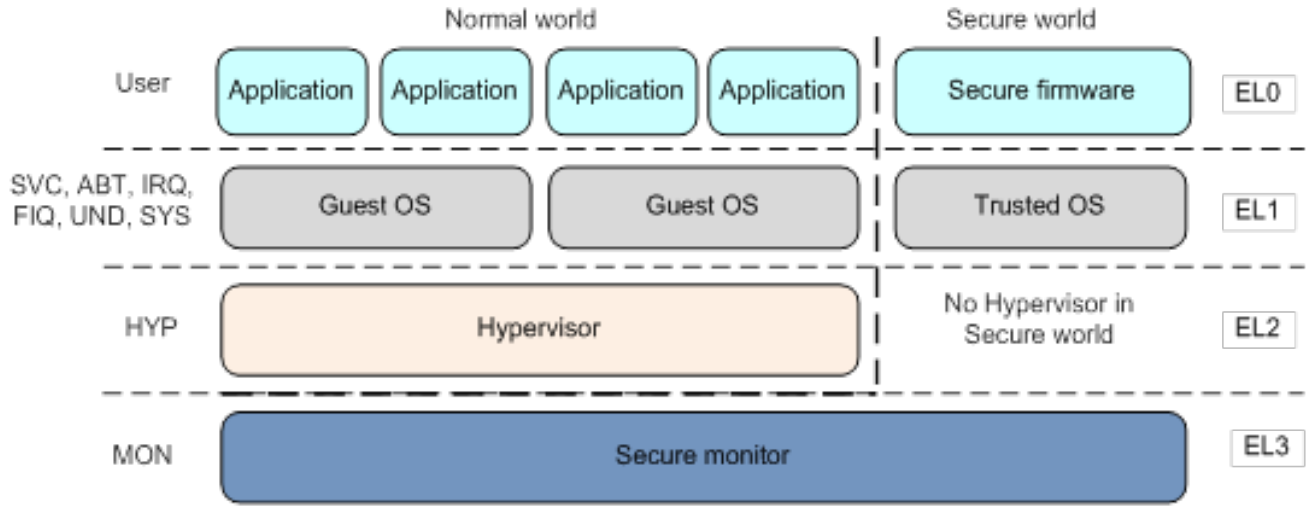
Distribution Statement `A' (Approved for Public Release, Distribution Unlimited)
This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).
The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

DornerWorks Background

- Services company
  - We work on things paid for by customers
- seL4 experience
  - Awarded multiple SBIRs through US DARPA
- Performed initial port of Hypervisor Extensions for ARMv8
  - Funded through GVSC (Formerly TARDEC)
  - Open-sourced seL4-ARM-VMM for ZCU102 and i.MX8

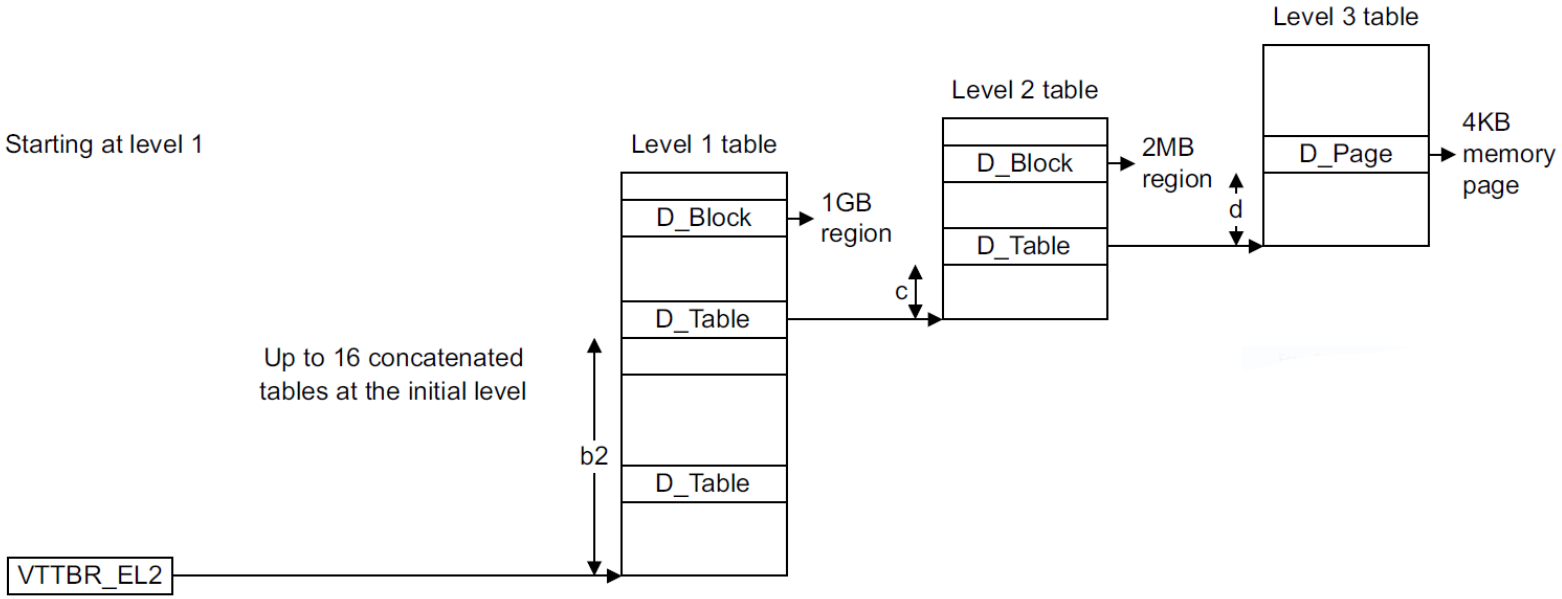
ARMv8 Hypervisor Extensions

- Ported seL4 to run at EL2
- Worked on Cortex-A53 processors
  - ZCU102
  - i.MX8



ARMv8 Hypervisor Extensions (cont)

- Cortex-A53 processors use 4-level MMU translations at EL1
  - Page Global Directory -> Page Upper Directory (1GB) -> Page Directory (2MB) -> Page Tables (4kB)
- Cortex-A53 processors use 3-level MMU translations at EL2
  - Page Upper Directory (1GB) -> Page Directory (2MB) -> Page Tables (4kB)



ARMv8 Hypervisor Extensions (cont)

- seL4 provides objects to userspace mirroring this translation scheme
  - Known as a Vspace
- Needed to modify Vspace for 3-level translation structure to run at EL2
- Data61 was performing this work in parallel
  - DornerWorks code took ~6 months to be open-sourced post development

Innovation Day

- DornerWorks provided each engineer 8 hours of Research and Development
  - Could work on project of choice
- My project: Port CAmkES-ARM-VM to run on the ZCU102
  - i.MX8 was not chosen because additional work required for virtualizing GICv3

Process

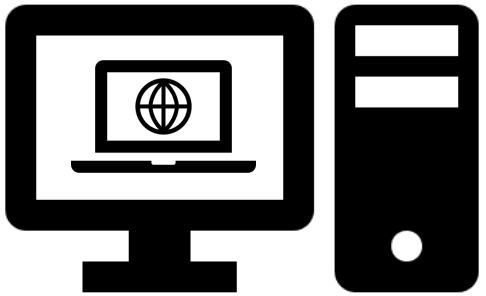
- Needed to rebase existing hypervisor support onto the tip of master
  - Going from seL4 versions 8.1 to 10.1
  - Going from Make to Cmake
- 3-level Vspace translation instead of 4-level
- Enable hypervisor extensions for Cortex-A53
- Goal: Run seL4test at EL2 on ZCU102
- Bugs:
  - PCIE Device Tree node generated untyped object addresses greater than 48 bits
- Created Pull Request on Data61 seL4 Repository
  - Functionality has been merged in

CAmkES Systems

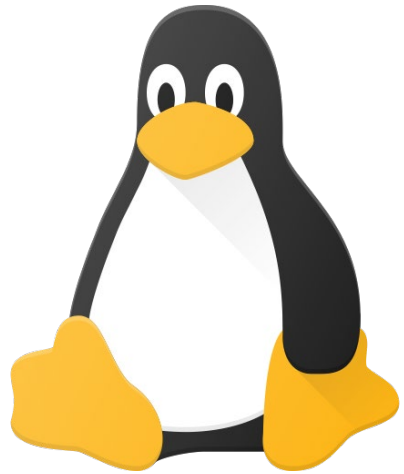
- Run under seL4
- Provide a rootserver that sets up userspace
  - capDL-loader
  - Also sets up Vspace for each component
- Problem: capDL-loader assumes 4-level Vspace for any ARMv8 processors
- Solution: Define a new architecture, **aarch64-40pa**
  - Provides a 3-level Vspace
  - Can change a few macros to set up a 3-level Vspace in capDL-loader
- These changes allowed for CAmkES to boot at EL2 on ZCU102

CAmkES-ARM-VM

- Once CAmkES images were booting at EL2, CAmkES-ARM-VM port could begin
- Generated barebones Linux image using Petalinux 2017.3
  - Needed to prune device tree to only include memory regions and serial device for logging
  - Without SMMU support, DMA devices cannot be used
- Added a new platform to the CAmkES-ARM-VM
- Fixed a few bugs related to high memory
  - Uint32_t -> seL4_Word



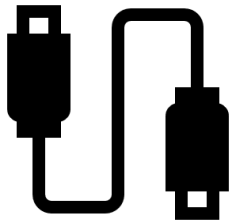
Results



- Linux began to start booting
- It then tried to make SMC calls (To EL3)
  - seL4 runs at EL2, so the guest was trying to go over seL4's head
- Needed to either allow the calls, or virtualize every SMC call
- Solution: Add config option to allow SMC calls
  - Disabled by default
  - Enabled for ZCU102
- CAmkES-ARM-VM booted all the way to the login prompt!

Future Work

- As part of different project, rebased SMMUv2 support onto the tip of master
  - Passed through Ethernet device to VM
  - Was able to SSH into ZCU102
- Multiple VMs
  - Can run 1 VM per core, each with their own VMM
  - Needed to add:
    - SMP support for zynqmp
    - Kernel support for hypervisor PPIs on secondary cores
  - Added an IRQInit component to initialize virtual timer interrupt on each core
    - capDL-loader has no knowledge of PPIs outside core 0
  - Virtualized the Serial device so each VM can share one hardware device
    - Integrated SerialServer like the CAmkES-VM (x86)



seL4 Ecosystem

- Porting existing systems to seL4 is not fun
  - Bring Your Own Driver
- Virtualization can bridge the gap
  - Run existing systems as VMs which support legacy code and applications
- Previously, seL4 only supported CAmkES-ARM-VM on Cortex-A57 chips with GICv2
  - Added support for Cortex-A53 (And Cortex-A55) processors with GICv2
  - GICv3 has previously been virtualized, and is in the process of being mainlined in seL4_projects_libs
    - This would allow for easy i.MX8 CAmkES-ARM-VM support

Questions???