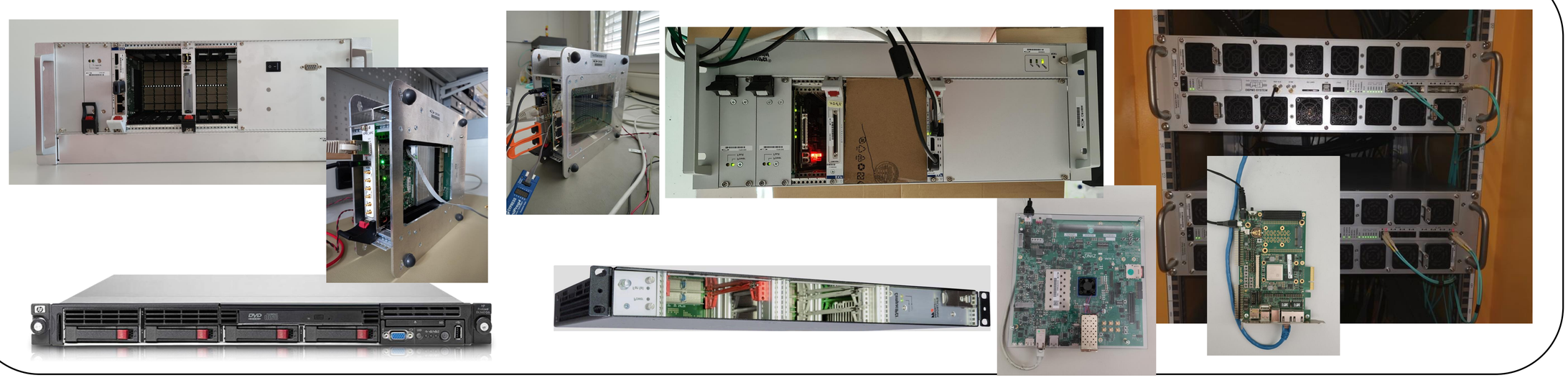
Building, Deploying and Provisioning Embedded Operating Systems at PSI

D.Anicic, Paul Scherrer Institut (PSI), Controls Section, CH-5232 Villigen PSI, Switzerland

Abstract

In the scope of the Swiss Light Source (SLS) upgrade project, SLS 2.0, at Paul Scherrer Institute (PSI) two New Processing Platforms (NPP), both running RT Linux, have been added to the portfolio of existing VxWorks and Linux VME systems. At the lower end we have picked a variety of boards, all based on the Xilinx Zynq Ul-traScale+ MPSoC. Even though these devices have less processing power, due to the built-in FPGA and Real-time CPU (RPU) they can deliver strict, hard RT performance. For high-throughput, soft-RT applications we went for Intel Xeon based single-board PCs in the CPCI-S form factor. All platforms are operated as diskless systems. For the Zynq systems we have decided on building in-house a Yocto Kirkstone Linux distribution, whereas for the Xeon PCs we employ off-the-shelf Debian 10 Buster. In addition to these new NPP systems, in the scope of our new EtherCAT-based Motion project, we have decided to use small x86_64 servers, which will run the same Debian distribution as NPP. In this contribution we present the selected Operating Systems (OS) and discuss how we build, deploy and provision them to the diskless clients.



Zynq Ultrascale+

- Considered 3 types: low, medium and high performance
- Ended up with almost 10 different board types
- Few manufacturer development boards
- Few self developed SoC based boards
- Few self developed SoM based boards

x86_64

- NPP: single board computers (SBC) in CPCI-S
- Motion: Small HP DL20 servers
- For development also few consumer PCs
- Also some Beckhoff compact-industry-PCs

Yocto 4.0 Kirkstone

- Different versions of Xilinx petalinux for proof of concept and testing different boards
- Later unified to Yocto 4.0 Kirkstone to build RootFS
- Single Xilinx-patched kernel 5.15.19-rt29 added, we add RT-patch
- Device-trees built for all board types
- Common SysVinit startup scripts added to RootFS, for mounting NFS shares for EPICS control system
- Few more Apps added to RootFS (not provided by Yocto)
- U-boots built for all board types, packed in BOOT.BIN, and stored on SDCARD for booting

Debian 10 Buster

- Started with RHEL, Ubuntu, CentOS and Debian on local disk for first proof of concept and testing
- Took Debian 10, which provides free RT patched kernel
- Network boot with Warewulf (Cluster Management tool)
- RootFS building with Docker containers, enabling necessary packages, adding our Systemd startup scripts for mounting NFS shares for EPICS control system
- Added kernel 4.19.208-rt88, build it from Debian sources
- RootFS, kernel and kernel modules imported to Warewulf
- Warewulf system and runtime overlay created
- Warewulf images for RootFS, kernel, kernel modules and system and runtime overlays built

TFTP Boot

- Load U-boot and environment variables from SDCARD
- Run u-boot
  - DHCP to get IP address
  - TFTP load common boot script, run it to:
  - TFTP load kernel and device tree
  - Create kernel parameter list (use env. vars)
  - Run kernel
    - mount RootFS over NFS
    - Start Linux,
      - mounts NFS shares
      - starts EPICS

HTTP Boot

- USB-stick boot instead of PXE network boot
- Load iPXE from USB-stick
- iPXE:
  - Run embedded script, uses hardcoded server name to:
  - HTTP load RootFS, kernel, kernel modules and system and runtime overlay images to RAM-disk
  - Run kernel
    - Start Linux
    - mounts NFS shares
    - starts EPICS