Elettra Sincrotrone Trieste

FLOP: CUSTOMIZING YOCTO PROJECT FOR MVMEXXXX POWERPC AND BEAGLEBONE ARM

WEPGF112

L. Pivetta, A.I. Bogani, R. Passuello
Elettra Sincrotrone Trieste, Trieste, Italy

During the last fifteen years several PowerPC-based VME single board computers, belonging to the MVMExxxx family, have been used for the control system front-end computers at Elettra Sincrotrone Trieste. Moreover, a low cost embedded board has been recently adopted to fulfill the control requirements of distributed instrumentation. These facts lead to the necessity of managing several releases of the operating system, kernel and libraries, and finally to the decision of adopting a comprehensive unified approach based on a common codebase: the Yocto Project. Based on Yocto Project, a control system oriented GNU/Linux distribution called Flop has been created. The complete management of the software chain, the ease of upgrading or downgrading complete systems, the centralized management and the platform-independent deployment of the user software are the main featues Flop.

1 - Since the last decades, technology evolution and component obsolescence are driving a renewal that involves also particle accelerator control system platforms.

Several models of PowerPC-based VME single board computers are in use at Elettra, running different distributions/releases of the GNU/Linux operating system. Furthermore, the BeagleBone, a low cost embedded board based on the ARM microprocessor, has been recently introduced, as well as a low-power high-performance INTEL Bay Trail Celeron based Soc platform to interface a specific instrument via USB.

6 – Flop advantages

Support for all the embedded platforms in use at Elettra, with optimized insruction set support:

- PowerPC MVME family
- BeagleBone ARM
- Intel Bay Tray based Jetway JBC311U93 Soc

Same source for all the platforms/architectures

Same release of the system software with respect to all the architectures

Cross-compiler development on fast Intel based host system

Centralized system software development

Read-only file system for flash storage

Overlayfs for transparent local or network operating modes

Remote platform update support with rollback capability based on multiple system partitions

Basic remote management via http

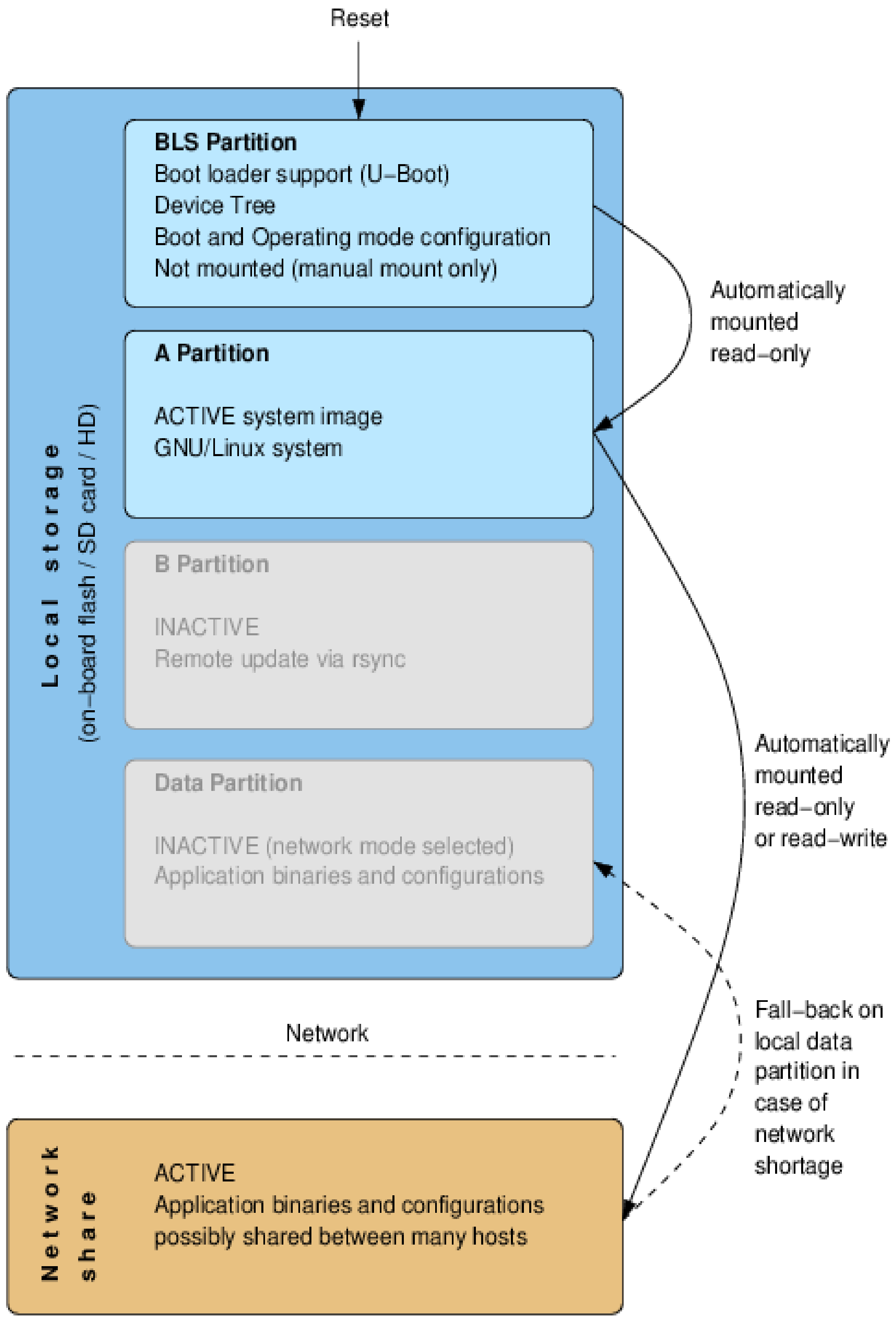
2 - The usual approach to GNU/Linux consists in the installation of one of the many available distributions. This may lead to:

- different releases of the system libraries;
- system V Init rather than upstart or systemd;
- different filesystem hierarchy;
- diverse bugfix policy;
- more complex administration.

4 – Flop 0.5

A control system oriented GNU/Linux environment, based on the Yocto Project release 1.8, featuring:

- Linux kernel 3.14;
- glibc 2.21;
- systemd 219
- gcc 4.9.2
- omniorb 4.1.6
- zeromq 3.2.4
- TANGO 8.1.2


Reset
BLS Partition
Boot loader support (U–Boot)
Device Tree
Boot and Operating mode configuration
Not mounted (manual mount only)
A Partition
ACTIVE system image
GNU/Linux system
B Partition
INACTIVE
Remote update via rsync
Data Partition
INACTIVE (network mode selected)
Application binaries and configurations
Local storage
(on–board flash / SD card / HD)
Automatically mounted read–only
Automatically mounted read–only or read–write
Fall–back on local data partition in case of network shortage
Network
Network share
ACTIVE
Application binaries and configurations possibly shared between many hosts

3 - The Yocto Project is an open source collaboration, aimed at providing templates, tools and methods to create custom GNU/Linux based systems for embedded hardware, regardless of the architecture. Founded in 2010, it involves many hardware manufacturers, open-source operating system vendors and electronics companies.

5 - A “recipe” specifies the procedures to select and compile software packages. Recipes can be grouped in layers, and multiple layers can be specified to introduce dependencies.

```
DESCRIPTION = "TANGO is an object oriented distributed \
  control system using CORBA (synchronous and \
  asynchronous communication) and zeromq \
  (event based communication)"
HOMEPAGE = "http://www.tango-controls.org"
LICENSE = "LGPLv3+"
LIC_FILES_CHKSUM = "file://COPYING.LESSER; \
  md5=6a6a8e020838b23406c81b19c1d46df6"

PR = "r0"

DEPENDS += "zlib bash omniorb zeromq"
RDEPENDS_${PN} += "bash"

S = "${WORKDIR}/tango-${PV}"

SRC_URI = "http://download.sourceforge.net/project/
  tango-cs/tango-${PV}c.tar.gz"

SRC_URI[md5sum] = "3dbcc2cf34f8c9395ee72f4ee5ae05dc"
SRC_URI[sha256sum] = "0149e797e5745b1dd8d5d3926088
  9b6da31c84c75c272372255ae8ca3507a116"

inherit autotools pkgconfig systemd

EXTRA_OECONF_append = " --disable-jpegmmx \
  --disable-static --disable-java --without-java \
  --disable-dbserver --disable-dbcreate \
  --enable-stdcxx11=no"
CXXFLAGS_prepend = "-std=gnu++98 "

SYSTEMD_SERVICE_${PN} = "starter.service stopper.service"

do_configure_prepend() {
  ( cd ${S}; ${S}/bootstrap )
}

do_install_append() {
  install -d ${D}${systemd_unitdir}/system
  install -m 0644 ${WORKDIR}/starter.service \
    ${D}${systemd_unitdir}/system
  install -m 0644 ${WORKDIR}/stopper.service \
    ${D}${systemd_unitdir}/system
}
```