UEFI Porting Update for ARM Platforms

What did we do since July?

Leif Lindholm
UEFI tech lead
Linaro Enterprise Group

presented by

UEFI Plugfest – May 2014
Agenda

Introduction
Linux Support
EDK2 Development
SCT
Platforms Tree
Future Work
Introduction

Last year, Grant Likely presented an overview of the types of things we were planning to do.

Since then, we have done various things...
Linux Support
Linux Support

The big topic since the last Linaro Connect event has been Linux support:
  Runtime Services
  UEFI stub loader
  Co-existence of ACPI/FDT
Runtime Services

Runtime services support for both 32-bit and 64-bit ARM
Not yet upstream, but aiming for Linux kernel v3.16
This included reworking of common code, since some functionality was already duplicated between x86/ia64

Main purpose is enabling GetVariable() / SetVariable()
Enables bootloader installer to add boot entries, and update boot order, from within Linux
Linux UEFI stub loader

The “stub loader” is the mechanism by which the Linux kernel can be loaded directly from UEFI, as a UEFI application.

- Becoming the default mechanism on x86(/x64) platforms.
- Enables the use of more light-weight bootloaders, like gummiboot or rEFInd.
- Not yet upstream, but aiming for 3.16.
- Majority of code shared between arm/arm64, and a fair bit shared with x86.
ACPI vs. FDT

A slightly contentious issue with existing ARM Linux developers
… who just spent a couple of years converting everything to FDT

Very much a non-issue for UEFI
We picked a GUID to use for FDT, and just treat it as any other configuration table
While simultaneously re-using FDT for the loader/kernel interface
Regardless of other hardware description, we use the UEFI memory map to discover available RAM
We had help!

Many thanks to Mark Salter (Red Hat) and Matt Fleming (Intel)

Mark helped parallelize our effort by doing 64-bit port while we worked on 32-bit, and also did some reworking on core memory-management code to have less architecture-specific code in the UEFI subsystem.

Matt has been a friendly upstream, holding back invasive changes to core code waiting for our bits to go in. And keeping us involved and informed.
EDK2 Development
EDK2 development

The EDK2 development done in Linaro so far has mainly been platform support
- Keeping platforms in tree up-to-date with upstream changes
- “plumbing” for Linux support
  - SMBIOS, FDT configuration table support
- Bugfixes (and reporting) for issues found while developing other code
  - Linux, GRUB

But…

We also have a port to a dynamic QEMU/AArch64 platform (EL1) underway.
Google Summer of Code

This year, Linaro has been accepted as a mentoring organization for GSoC

One of the three projects we have had approved is “Porting UEFI to a low-cost embedded platform”

We have opted for Texas Instruments’ BeagleBone Black, which is a $45 Cortex-A8 based board.
To be honest, we have just brought the github SCT into our non-public tree...
The Platforms Tree
Linaro EDK2 builds

Linaro produces monthly builds of EDK2
   Augmented with platform support and not yet upstream patches

Initially this was done in a fairly monolithic fashion
   Single git repository, with a bunch of topic branches

We have now set up a new platforms tree
   To facilitate distributed development with multiple contributors and multiple maintainers
Linaro EDK2 tree

A clone of the github mirror

We pick a commit as our starting point for that month’s release
Then we add platform and feature topic branches and create a tagged release, as well as upload pre-built images to releases.linaro.org

Hosted on git.linaro.org
https://git.linaro.org/uefi/linaro-edk2.git

We would love to have more (maintained) platform ports
Future Work
Future Work

Several new UEFI-based ARM platforms coming this year

Need to integrate member (and other) platforms into our tree

Look into platform requirements for secure key management on ARM

Get rid of the ARM built-in Linux loader

Plumbing to simplify integrating FDT in firmware

Port to AArch64/Xen
Wish List

The resurrection of a common Bds under MdeModulePkg
CSM is dead, long live UEFI?

More common drivers
This is where we are currently losing platforms to U-Boot

More common helper protocols
libfdt, SMBIOS, IPMI, …?
For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

Leif Lindholm
UEFI Tech Lead
Linaro Enterprise Group