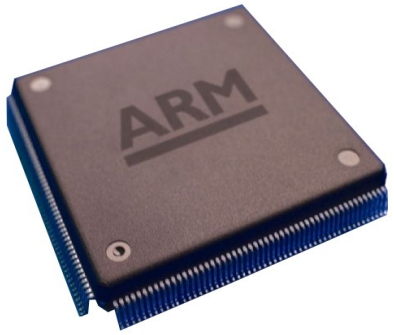
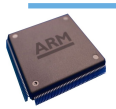


OpenSolaris ARM Port and its future

Vineeth Pillai
Solaris Revenue Product Engineering
Sun Microsystems, Czech

Agenda

- Introduction
- OpenSolaris ARM Port
  - Technical Details
  - Try the arm port – howto
  - Developing applications
- Future
  - Possible Use cases
  - Community Involvement
- Latest Developments
- Conclusion



opensolaris

OpenSolaris

- <showoff>

  The Most Powerful, Stable and Secure - Advanced Enterprise Operating System on the Planet!!!

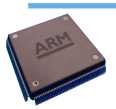
  </showoff>

- From Server to handhelds:
  - Fully pre-emptive and multi-threaded kernel.
  - Real time capabilities.
  - Highly configurable and modular kernel.
  - Opensource and backed up by huge software base.

opensolaris

ARM Architecture

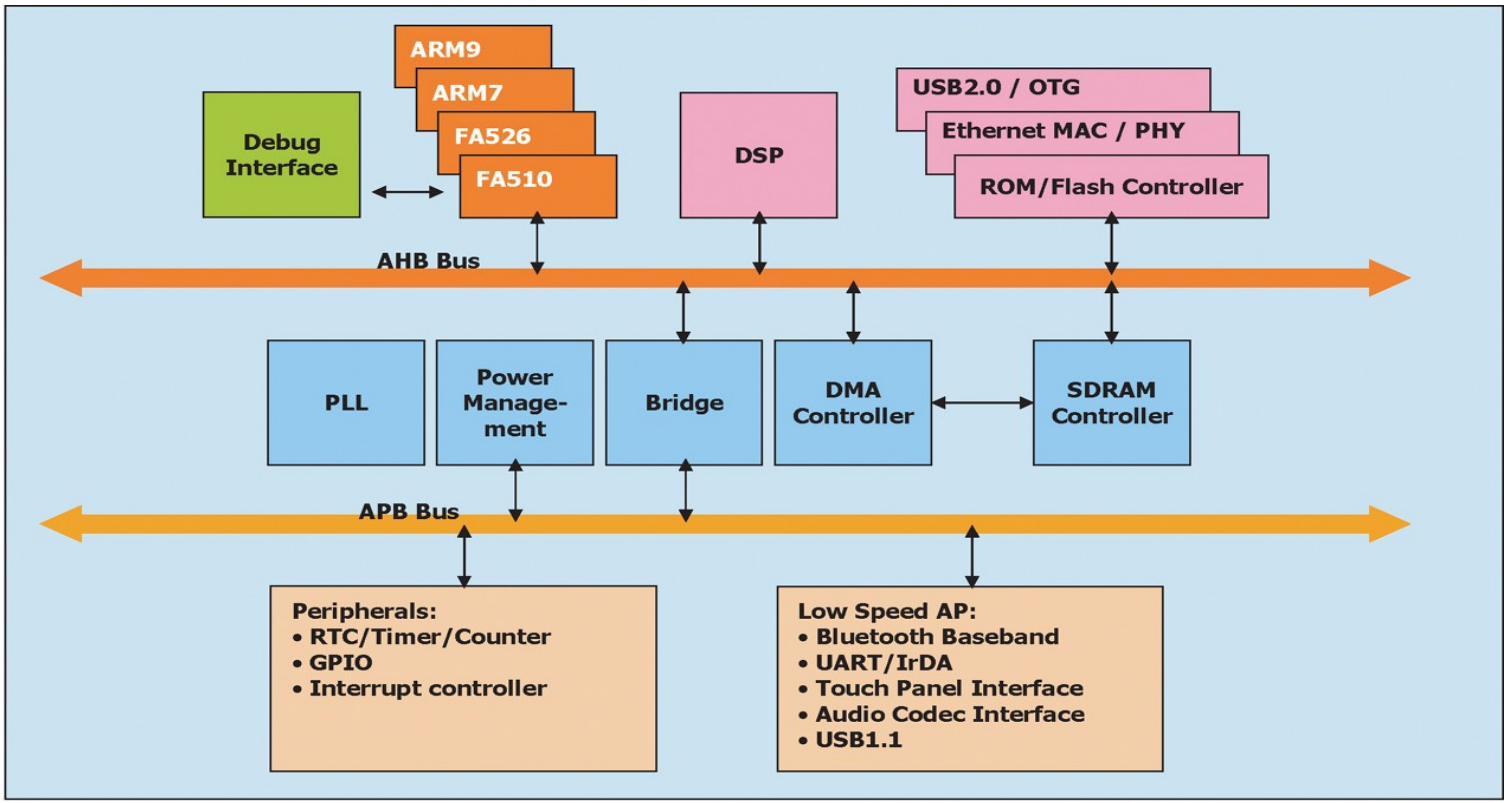
- 32bit RISC
- Perfectly suitable for Embedded Devices
  - Low power consumption
  - Great code density
  - 3 instruction sets to select from:
    - Normal 32 bit ARM Instruction set
    - 16bit THUMB instruction set (Code density)
    - JAZELLE(JAVA byte code on hardware)
  - Instruction and data caches and Instruction pipelining
  - MMU and MPU



opensolaris

ARM Architecture(Contd...)

- System on Chip (SOC)



OpenSolaris on ARM

- Announced on June/08/2009
- Based on OpenSolaris 2008.05 build 86
- Ported to NEC NaviEngine 1
  - ARM architecture v6K (MPCore)
- Also comes with QEMU patch for NE1
- Source code maintained in mercurial repository
- Designed and Implemented with out breaking Solaris build conventions

opensolaris

ARM Port: Notable Changes

- 460 new files added, 1225 files changed (approximately)
- Changes both in Kernel and Userland
- ZFS for embedded devices: CZFS
- Statically linked kernel which includes genunix, unix and device drivers

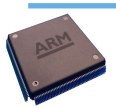
ARM Port: kernel Changes

- usr/src/uts/arm
  - ARM specific architecture independent code
  - Generic modules in kernel
- usr/src/uts/armpf
  - ARM specific architecture dependent code
  - Boot code, interrupt handling, memory management
- usr/src/uts/ne1
  - Board specific code
  - Board specific device drivers

opensolaris

ARM Port: Userland changes

- Libraries
  - usr/src/lib/*/arm
  - Libraries which have architectural dependencies – libc, libdev*
- Commands
  - usr/src/cmd/*/arm
  - Few commands like devfsadm which have architecture specific code
- ARM specific changes for syscall handling
  - Software interrupt 'swi $SYS_num'



CZFS

- Compact ZFS for Embedded devices
- Most bulky data structures compacted
  - Blkptr, dnode_phys etc
- Changes in Transaction semantics
- On-disk format compacted
- Two new commands
  - czfs – managing filesystems
  - czpool – managing pools
- ZFS and CZFS are incompatible
- ZFS is also supported!

opensolaris

ARM Port: Getting your hands dirty!

- Build and Install Tool Chain
- Build OpenSolaris using the the Tool Chain
- Flash the image:
  - on NE1 or
  - Patch the Qemu for NE1 support
- Boot OpenSolaris!
- Build custom applications and include it in the image.

opensolaris

ARM Tool chain

- GNU binutils 2.18 with solaris patch
- C Runtime Library – newlib 1.17.0
- GCC 4.1.1 with solaris patch
- QEMU 0.9.1 with NE1 patch

opensolaris

Building the Tool chain

- Onetime process
  - Tool chain built and installed into /opt/arm-eabi
- Two step building process
  - GCC cross compiler is built with newlib C runtime library
  - SUN libc and libm are compiled with above cross compiler
  - GCC is again compiled with SUN C runtime libraries.

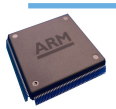
opensolaris

Build OpenSolaris

- Similar to normal builds
- Set up the environment
  - ./initenv.sh
  - ./usr/src/tools/scripts/bldenv.sh -d opensolaris-arm.sh
- Issue the build
  - dmake setup; dmake clean; dmake install
- Depending on the hardware, build process is time consuming as normal opensolaris builds

Finally..

- Image layout as expected by NE1
  - Three parts
    - Statically linked kernel image
    - Root filesystem image
    - Boot loader - uboot
- Building rootfs
  - Builds a ufs image of root filesystem with all the required files and binaries
  - usr/src/qemu/rdimage/Makefile



opensolaris

Trying it out

- Two Possibilities – NE1 board or Qemu
- NE1 Board
  - Uboot is already present
  - Flash the kernel and rootfs into NE1 flash memory using flashing tools
  - Power on!
- Patched Qemu
  - Qemu emulates the bootloader
  - Specify the location of kernel and rootfs image while invoking qemu

Developing Applications

- Copy proto area of fully built gate to /opt/arm-eabi/root
- Compile using the Tool chain
- Decide what all files and binaries to be in the root filesystem
- Add the entries of files in the corresponding configuration files
  - usr/src/qemu/rdimage/filelist/*
- Rebuild the rootfs and invoke Qemu

Trying out CZFS

- Hurdle
  - Qemu NE1 emulator doesn't support IDE/SATA
- Still, CZFS can be tried out
  - Use files for zfs storage pool
- Some code editing (patching) needed
  - Increase NE1 memory to 512mb
  - Change swap size from 2mb to unlimited
  - Allow czfs to use files as the storage medium
- Create files in /tmp
- Use czpool/czfs as you use zpool/zfs

opensolaris

Trying out CZFS(Contd...)

usr/src/uts/ne1/sys/platform_mach.h:

```
-#define       ARMMACH_SDRAM0_SIZE     UINT32_C(0x08000000)    /* 128MB */
-#define       ARMMACH_SDRAM1_PADDR    UINT32_C(0x88000000)
-#define       ARMMACH_SDRAM1_SIZE     UINT32_C(0x08000000)    /* 128MB */
+#define       ARMMACH_SDRAM0_SIZE     UINT32_C(0x10000000)    /* 256MB */
+#define       ARMMACH_SDRAM1_PADDR    UINT32_C(0x90000000)
+#define       ARMMACH_SDRAM1_SIZE     UINT32_C(0x10000000)    /* 256MB */
```

usr/src/uts/arm/tune/modtune:

```
 option int VFS_TMP_SIZE
 {
-       default:        2048;
+       default:        0;
```

usr/src/uts/ne1/czfs/modtune:

```
 option boolean CZFS_NO_UFSFILE
 {
-       default:        true;
+       default:        false;
```

usr/src/qemu/qemu-0.9.1/hw/naviengine.c:

```
-#define NE1_DDR2_SIZE          0x10000000      /* 256MB */
+#define NE1_DDR2_SIZE          0x20000000      /* 512MB */
```

usr/src/qemu/ne1/ne1.sh:

```
-MEMSIZE=320            # DRAM:256MB + NORFLASH:64MB
+MEMSIZE=576            # DRAM:512MB + NORFLASH:64MB
```

PATCH
for trying out czfs on qemu

opensolaris

ARM Port: Future

- Only one supported device.
  - Needs to be ported to various well known ARM devices
- Only basic functionality available
  - More software stack needs to be ported
  - Mostly, its a matter of recompilation
- No Windowing system
  - Possible candidates for easy porting: xfce4, icewm, …
- Highly desired: Well maintained IPS repo for ARM packages!

opensolaris

Use case: Openmoko Neo freerunner

Use case: Openmoko Neo freerunner

- GSM mobile running on 100% FOSS
- Famous toy among Unix Geeks!
- Porting steps:
  - Base device drivers: Interrupt controller, timer, display, …
  - GSM stack
  - User interface
  - Bluetooth???

opensolaris

Use case: ARM Netbooks

- Great Opportunity!
- ARM netbooks are starting to ship with linux OS pre installed
- Porting seems to be much simpler
  - Device drivers
  - Windowing System
- Proof of concept showing Opensolaris in all its majesty!

opensolaris

Community Involvement

- Support different ARM processor families
- Identify ARM devices
- Build and maintain an ARM IPS repo
- Test
- Build a good user base
- And finally - Evangelise!!!!

Latest Developments

- New ARM dev branch which is being synced with opensolaris 2010.02
- Currently synced with b111 rev 9063
- Maintained as a seperate mercurial repository - onarm-dev
- No enhancements except the syncing with latest bits
- Not well tested

ARM Port : Some complaints...

- Not all the features are ported:
  - SMF, Zones, NFS, Kerberos
- Based on very old code base
- Certain code level hard coding based on the board NE1
- Less (Zero) Community involvement during development
- Mailing list needs more energy!

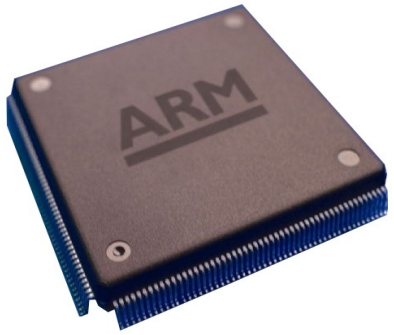
opensolaris

References

- Project Website :
  - http://opensolaris.org/os/project/osarm/
- Release Notes:
  - http://opensolaris.org/os/project/osarm/200805/relnotes/
- Install guide:
  - http://opensolaris.org/os/project/osarm/200805/installation/
- ARM:
  - http://arm.com/
- NEC NaviEngine1:
  - http://www.nec.co.jp/techrep/en/journal/g07/n04/070409.html
- ARM repository access:
  - hg clone ssh://anon@hg.opensolaris.org/hg/osarm/onarm-gate
- ARM development repository access:
  - hg clone ssh://anon@hg.opensolaris.org/hg/osarm/onarm-dev
- Mailing List:
  - osarm-dev@opensolaris.org

opensolaris

Q&A



OpenSolaris ARM Port and its future

Vineeth.Pillai@sun.com