Usage of singleboard PC-s in cluster

Balint JUHASZ[*], Zsolt CSATARI[**]
[*]Óbuda University, Budapest,
[**]Óbuda University, Budapest, csatari.zsolt@kvk.uni-obuda.hu

Abstract - This paper shows the basics of hybrid cluster architecture made of ARM SoC based boards and PC.

1. INTRODUCTION

Nowadays to satisfy the rapidly growing calculation demand is a big problem. For solving this, developers tend two directions. First is to drastically raise the performance of single computers. The second is distributed computing. Cluster belongs to the second group.

The MPI[1] gives a solution for the problem by aggregating the computing performance of multiple, but low performance devices. The MPI is practically a protocol, which allows our software to run on several computers simultaneously.

Great computing performance comes with great power consumption. Our project's goal is to give a solution for this and to investigate, if it's possible to create a cluster of nodes with different architecture (embedded device and normal). We also examine the power consumption-performance ratio for every device in our project.

2. EMBEDDED DEVICES AND THEIR OPERATING SYSTEMS

"The embedded system is a such special purpose computer, which was designed for a specific task."[2]

If we want to run the chosen Linux distribution on the embedded system, it must be ported first. For this, the kernel and the boot loader must be compiled on the target architecture. This needs a cross compiler toolchain installed on the host computer and we must get the source code of the Linux kernel and the boot loader. During the compilation, the boot loader and the kernel must be configured according to the specific device. These information can be found mostly

[1] Message Passing Interface

[2] http://en.wikipedia.org/wiki/Embedded_system

on the manufacturer's support site, also complete settings files and binaries can be downloaded.

The Linux kernel can be downloaded from the site http://kernel.org and various boot loaders (U-boot, x-loader, vivi, etc.) can be found. In addition a root file system is needed as well as an installed Linux distribution. Most Debian-based distributions are available via debootstrap, even for ARM architecture. The root file system downloaded via debootstrap contains a preinstalled Linux system, however, the system must be started in a special mode first and the installation must be finished. To avoid this alternative methods are known to simulate the device, like qemu.

Depending on the embedded system, the root filesystem, the bootloader and the kernel binaries can be stored on an integrated NAND flash storage or an external SD card. After modifying the SD card's settings, our device boots from the SD card.

To make the SD card bootable, a Fat32 partition with the size of few megabytes must be created and marked as bootable. The boot loader and the kernel binaries must be copied onto this partition beside the bootloader's configuration file. Another partition must be created and formatted as EXT2. The root filesystem will be copied on this partition.

Using NAND-flash based storage requires a non-journaling file system, because a journaling file system wears out the NAND cell's with unnecessary writes.

3. ARCHITECTURAL DIFFERENCES

The low power consumption of the embedded systems origins from the fact that they were designed for specified tasks unlikely the PC which is built for solving general problems and has a lot of expansion capability. For example the PCI interface consumes as much power as an embedded system does for its full functionality. There are other power consuming components in the PC, like Audio card, SATA interface, VGA subsystem, high capacity system memory. Because of high integration of devices in the SoC[3] the power consumption is drastically reduced in embedded systems.

The basic design philosophy is difference between the two systems, the x86 is built to solve "every" problem, at the opposite the ARM is designed to solve specific problems, but the software industry realized in the last ten years this device is capable to run highly definite, self-paced functions, like graphical user interface, operating system, etc.. The first applications of ARM use it as a microcontroller and run specified routines. Only in the last few years designers discovered it as a general purpose processor and built devices with it, like smart phones, tablets, smart homes. ARM passed the exams in the entertainment industry and IT sector suddenly realized the possibilities in it. Hewlett Packard introduced the ARM Data Center idea[4], Nvidia Tegra[5] project and at last our project to make a hybrid computing cluster with different architectures.

[3] System on Chip

[4] http://www.theregister.co.uk/2011/11/01/hp_redstone_calxeda_servers/

[5] http://en.wikipedia.org/wiki/Tegra

Name	IGEP v2	beagleboard	Raspberry Pi B	Hawkboard	Colibri T20	Intel Atom based PC
CPU type	ARMv7 Cortex-A8	ARMv7 Cortex-A8	ARM11	ARM	ARM Cortex-A9	Intel ATOM 330
Clock	720MHz	600MHz	700MHz	456MHz	1000MHz	1600MHz
Perfomance	1440MIPS/ 4160MMACS	1200MIPS/ 3470MMACS	1400MIPS	456MIPS/ 3648MMACS	2000MIPS/ 6822MMACS	6400MIPS/ 6400Mflop
Bulit in memory (MB)	512	256	256	128	512	1024
Power consumption	2,5W	4W	3,5W	2,5W	5W	65W
Price(EUR)	188	110	35	350	200	100
MIPS/W	576	300	400	182,4	400	98,4615
MIPS/Price	7,659574468	10,90909091	40	1,302857143	10	64

1. Table: Comparison table of embedded processor board with Linux compatibility

4. MEASURE RESULTS AND COMPARISON OF THE DEVICES

We have used for the measurement one of the built in examples of the MPI library which is called "icpi". This is an iteration based PI approximation algorithm which has high parallelisation possibility. As we can see in the table the computing performance of the PC is 11.29 times higher than the ARM's. The performance difference comes from the fact that the Intel Atom CPU has built in floating point processing unit while the ARM does not has, and the Atom operates at two times higher. As the results shows in floating point problem solving the ARM boards does not bring enough performance to reasonably use them, but if the problem is based on integer types the ability of parallelisation and the low power consumption gives them the advantage. As we can see there is significant difference in the time needed for the calculation when only one IGEP v2 and when two of them are used. It is shown also that in cluster architecture the slowest device determines the maximal increase of speed. In this case that is clear too that if the computing nodes number are doubled then the time for the calculation is reduced to almost half of the original time, this can be kept up only for a limited number, but because lack of resources we were unable to rich this limit. From the table below where the power consumption is shown we can see that the IGEP v2 uses only 2.565W at full capacity while the PC uses 60.49W from these with an easy calculation we can see that 23 IGEP v2 uses the same amount of power but 12 gives more computing performance then the PC. In this case we can see that the computing performance/power consumption ratio is good enough to make use of the IGEP v2.[6]

[6] The IGEP v2 board is a low-power, fanless single-board computer

	PC	IGEP v2	Results
Head node only (PC)	259-263mA	-	65.96s error: $13*10^{-16}$
Head node + 1 IGEP v2	255-260mA	589mA	372s error: $3482*10^{-16}$
Head node + 2 IGEP v2	255-261mA	1,04A	247.9s error: $2189*10^{-16}$
Only 1 IGEP v2	-	513mA	744.748s error: $1776*10^{-16}$
2 IGEP v2	-	1,04A	372.35s error: $2141*10^{-16}$
Idle	250mA	686mA	-

2. Table of measure results

5. RESOULTS

We compiled a working Linux system which is running on the ARM SoC based IGEP v2, used NFS for the storage of the commonly used project files.

We were able to run MPI projects on ARM based systems and PCs too. We managed to solve the problem which is caused by the different architecture of the Intel Atom processor and the ARM Cortex A8, this is a great problem, when we are trying to run the MPI project. This is solved by the usage of the common NFS library for the ARM based computing nodes as home directory, and another different home for the PC. This lets us use only one compiled version of the project for each of the architectures. With this solution we can use as much architecture as we want and we don’t have to copy and recompile for each of the computing nodes.

6. CONCLUSION

Our first result it is possible to make mixed architecture cluster. The next achievement is this seems to be a good idea, because there can be problems which needs the high number of parallel working units more than the high computing capability of the individual nodes.

7. REFERENCES

[1]: Unknown, “IGEP Wiki” Internet: http://labs.isee.biz/index.php/Main_Page, last visited: 2012. may 10.

[2]: W. Gropp, R. Graham, A. Moody, T. Hoefler, R. Treumann, J. L. Taff, G. Bosilca, D. Solt, B. R. de Supinski, R. Thakur, J. M. Squyres, R. Rabenseifner, A. Supalov, (2009, September 4), *MPI: A Message-Passing Interface Standard Version 2.2*, [On-line], Available:

http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf [May 7, 2012]

[3]: "Using MPI", W. Gropp, E. Lusk and T. Skjellum. MIT Press, 1994.

[4]: Unknown, “The Message Passing Iterface (MPI) standard” http://www.mcs.anl.gov/research/projects/mpi/, last visited: 2012. may 27.

[5]: Unknown, “WIKIBOOKS Message-Passing Interface” http://en.wikibooks.org/wiki/Message-Passing_Interface, last visited: 2012. may 27.

[6]: Unknown, “Wikipedia Message Passing Interface” http://en.wikipedia.org/wiki/Message_Passing_Interface, last visited: 2012. june 12.

[7]: Durucz, B. (2011). EMBEDDED LINUX DEVELOPMENT, IX. Student Science Conference (pp. 13-19). Poland, Bedlewo: Wroclaw Polytechnica.