Porting the Linux kernel to the TSAR manycore architecture

Joël Porquet, Alain Greiner, César Fuguet Tortolero

Sorbonne Universités, UPMC Univ Paris 06, UMR 7606
Laboratoire d'Informatique de Paris 6
F-75005, Paris, France

I. TSAR

TSAR [1], which stands for *Tera-Scale ARchitecture*, is an European MEDEA project that aims at defining a scalable, coherent shared memory, manycore architecture. It is intended to support commodity operating systems and applications.

The main technical issue that TSAR addresses is scalability, since it is meant to support up to 4096 processors. To that end, the architecture relies on an innovative full-hardware cache coherency protocol, for both L1 caches and TLBs, and efficient atomic operations for software synchronization.

a) Processors: To maximize the performance/energy consumption ratio, TSAR is composed of simple 32-bit single instruction issue general-purpose processors. And to avoid any unnecessary effort in developing new compilation tools, these processors are **MIPS32** ISA-compatible.

b) Memory Layout: TSAR supports a 40-bit physical address space, and for scalability reasons, the physical memory is logically shared but physically distributed. Being organized in a 2D mesh topology, this means that TSAR is a clusterized NUMA architecture (*Non-Uniform Memory Access*).

c) Virtual Memory: TSAR implements a two-level paginated virtual memory strategy, where TLB misses are handled by hardwired state machines.

d) Atomic Operations: TSAR implements atomic operations using an hardware enforced load-link/store-conditional (LL/SC) mechanism.

II. Linux and its ecosystem

Linux [2] is an operating system kernel, which is the first and fundamental layer of software placed between hardware architecture and userspace programs/applications. Its primary function is to manage the hardware resources and allow programs to run and use these resources.

Some of the resources that the kernel needs to share between all the programs are: processors, used for executing programs code; the memory, used to store both program instructions and data; and any input/output devices present in hardware systems, such as keyboards, disk drives, displays, etc.

Porting Linux to a new processor architecture is however not sufficient. There are, at least, two software components that need to be ported as well: *compilation tools* (including a compiler), which are necessary to transform source code written in a high level programming language into computer binary code; and a *standard library* (e.g. the C standard library, or **libc**), which provides programs with macros, type definitions, standard functions, and access to resources managed by the kernel (through system calls).

III. Results

Linux is able to boot on various hardware prototypes. For small configurations (i.e. a few clusters), there are two available virtual prototypes of the TSAR architecture: the first one uses **SoCLib** [3], a *SystemC* simulation framework, while the second one is FPGA-based. And finally for bigger configurations (i.e. up to 24 clusters, and 96 processors), Linux is able to boot on a fast hardware emulator provided by the CEA LETI.

```
Linux version 3.13.0 (joel@joel-zenbook) (gcc version 4.8.2
      (crosstool-NG hg+default-000000000000) ) #180 SMP Mon
     May 5 17:50:24 CEST 2014
Model: UPMC/LIP6/SoC - Tsar v5 multicpu SoCLib
bootconsole [early_tty_cons0] enabled
[...]
Memory: 255112K/262144K available (1798K kernel code, 92K
     rwdata, 236K rodata, 2372K init, 187K bss, 7032K
     reserved)
[...]
CPU1: booting secondary processor
CPU2: booting secondary processor
CPU3: booting secondary processor
Brought up 4 CPUs
SMP: Total of 4 processors activated.
[...]
console [ttyVTTY0] enabled
bootconsole [early_tty_cons0] disabled
[...]
Freeing unused kernel memory: 2372K (c0217000 - c0468000)
Welcome to Buildroot
buildroot login: root
Jan  1 00:00:05 login[41]: root login on 'ttyVTTY0'
# ls /
bin home  lib32   mmnt  root  sys var
dev init  linuxrc opt   run   tmp
etc lib   media   proc  sbin  usr
#
```

Figure 1. Shortened trace of Linux bootup on a monocluster SoCLib simulation

References

[1] "The TSAR project." [Online]. Available: https://www-soc.lip6.fr/trac/tsar/ 1

[2] "The Linux Kernel Organization." [Online]. Available: https://www.kernel.org 1

[3] "The SoCLib project." [Online]. Available: https://www.soclib.fr 1