Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by Alexandr Andreev on Mon, 23 Oct 2006 11:48:16 GMT
View Forum Message <> Reply to Message

Just two words.

1. Jonathan, do you use cross-compilation? It can be useful, because in this case we could be in sync with what you are doing
(kernel development/building). We could help you to fix some OpevVZ general errors ("unresolved symbol"), which are frequent enough for customers .config's.

2. I think it would be useful to support general sparc Sun4M platform in OpenVZ first, and later add specific support for T2000 (sparc64, smp and other sparc's).

Sun4M is simple, and it is supported by qemu emulator, which is fast enough:

http://www.qemu.com/status.html

If you will be using cross-tools and qemu, you will not require any additional hardware. You can just build and test your kernel on any host PC (x86). There are numbers of qemu-sparc users and a number of kernels that works fine on it.

I would be done everything in this way:

a) start from 2.6 kernel for qemu-sparc:

- download and make qemu for sparc;

- try to run binary kernel:
  http://www.qemu.com/download.html

- make (download) cross tools (gcc, binutils) for sparc;

- try to compile (by cross-tools) and run mainstream kernel with sparc patches (if any) on qemu;

b) try to configure and compile OpenVZ kernel for sparc (by cross-tools):

- add some SPARC-specific patches to OpenVZ kernel (if it needed)

- try to make the kernel (we will help you to fix some arch-independent "can't compile" errors that are OpenVZ specific)

- write a peace of arch-dependend code in arch/sparc/XXX and arch/sparc64/XXX, that OpenVZ requires. Maybe we will also help you with that. See examples in arch/x86, arch/x86_64, arch/ppc, arch/ppc64.

c) run and debug your kernel in qemu first.

d) When it becomes stable, try to run it on T2000