
Subject: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [jonathankinney](#) on Mon, 22 May 2006 23:10:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am still doing some research, but I was wondering if anyone here might know if a Sun Fire T2000 could run OpenVZ. The system is running on the UltraSPARC T1 CPU architecture. I know that a system like that is far from mainstream, but I am still curious.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [dim](#) on Tue, 23 May 2006 12:38:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

OpenVZ doesn't support SPARC platforms.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [dev](#) on Tue, 23 May 2006 14:37:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Let me correct Dim:

Sun Fire T2000 can run OpenVZ. But we don't have such hardware and thus officially don't support it nor do builds.

OpenVZ patch itself is 99.99% platform independent, so there should be no much problems to make it work. If you have such a possibility we can help you with it.

Do you have this hardware? Are you ready to build/test kernels and make OS template?

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [jonathankinney](#) on Thu, 25 May 2006 05:02:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:Sun Fire T2000 can run OpenVZ. But we don't have such hardware and thus officially don't support it nor do builds.

Very understandable, that would be far too much work for something that 1 out of 100 people might consider trying.

Quote:OpenVZ patch itself is 99.99% platform independent, so there should be no much problems to make it work. If you have such a possibility we can help you with it.

Thats what I was hoping to hear, and very pleased to hear that the platform independence was not compromised with the OpenVZ patch to the kernel source. With this in mind, I should be able to build it myself with minimal if any tweaking of source code.

Quote:Do you have this hardware? Are you ready to build/test kernels and make OS template?

I don't have the system yet, I was just gathering preliminary information before I go out and get a T2000 and start development of OpenVZ on that system. As soon as I get to that point, I will let

you know how things pan out, and maybe send some questions your direction if I run into any problems. Thanks for the information.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [dev](#) on Thu, 25 May 2006 06:29:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've just took a look at the recent patch-022stab077-core for arch/ia64 changes (i386 and x86_64 have quite lots of OpenVZ unrelated mainstream fixes and 4GB split - noise) and can summary up the changes required for arch specific code:

- UBC: need to account any platform specific VMAs created by hand in arch specific code. i.e. if there are calls of `insert_vma_struct()` this should be accounted with `ub_memory_charge()`. Didn't find such this on sparc64.

- if there are user triggerable `printk()`'s (related to the user, not the system as a whole) better replace them with `ve_printk()`. Otherwise user can flood (DoS). minor actually.

- call to functions `find_task_by_pid()`, `for_each_process()` and `do_each_thread()/while_each_thread()` should be replaced with it's counterparts - `find_task_by_pid_XXX()`, `for_each_process_XXX()` and `do_each_thread_XXX()/while_each_thread_XXX()`, where XXX is 'all' or 've'. 'all' means that all system processes in the system will be scanned, while 've' means that only VE (VPS) accessible from this task (current context - `get_exec_env()`) will be visible. So you need to decide, whether the code in question is about system or user context.

- `task->pid` should be changed with `virt_pid(task)` in some places. The rule is simple: user should see only virtual pids, while kernel operate on global pids. e.g. in signals, virtual pid should be delivered to app.

- in interrupt handlers one need to set global host (VE0) context. i.e. `set_exec_env()`, `set_exec_ub()`. i.e. interrupt handlers are running in VE0 context.

- in `kernel_thread()` one needs to prohibit kernel threads in VE. mostly security related...

- `show_registers()` better to extend to show current VE.

- `utsname` should be virtualized. this mostly means that 'system_utsnames' should be replaced with 've_utsname'. See any arch code for this.

- some exports will be required. e.g. `show_mem()` and probably `cpu_khz`. easy.

- everything else are bugfixes.

all these are straightforward and really simple, so it should take a few hours to do so.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [jonathankinney](#) on Thu, 19 Oct 2006 23:31:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

I now have a Sun Fire T2000 server up and running with Ubuntu 6.10, and have downloaded the vanilla Linux 2.6.16 kernel and the kernel patch. But to tell you the truth, I am not sure where to begin.

I have been looking at all of the changes the patch makes, and sorting out which modifications need to be made to which files has started to make my brain hurt. Perhaps someone can be of assistance in this situation. I don't have too much experience with rewriting kernel code, so any pointers would be helpful. If for some reason ssh access is needed, depending on who is asking, I can provide access. In the meantime, I will continue to hack away at the code to see if I can get a working kernel together.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [dev](#) on Fri, 20 Oct 2006 04:00:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

1. I don't remember... in what kernel version was Sun T2000 support added? what kernel are you running? i.e. the question is whether mainstream 2.6.16 works on it or not

2. I will be happy to get an ssh access (I don't even need a root).

And make ovzkernel and ovz tools compilable. I will provide a patches for you then and you will be able to check it.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [jonathankinney](#) on Fri, 20 Oct 2006 22:45:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

The net boot install for Ubuntu that I was working with was using a 2.6.16 kernel, and it loaded everything and seemed to be running just fine. Right now I have it up and running with "2.6.17-10.33-sparc64-smp" which is the default install kernel for the Edgy server distribution. I am getting the feeling that the mainstream 2.6.16 would work on the T2000 with a few extra patches that the people at Ubuntu used. I will be looking into finding the patches they used to get it to work on the "64-bit UltraSPARC SMP" so I can get a handle on what additional changes will need to be made to the source.

Right now I just have the untouched 2.6.16 kernel source sitting in /usr/src/BUILD/linux-2.6.16

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [dev](#) on Mon, 23 Oct 2006 07:16:44 GMT

mmm, maybe mainstream 2.6.18 works fine?
We will release 2.6.18 OVZ this week anyway, so
don't waste your time with 2.6.16.

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 piece 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

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [jonathankinney](#) on Mon, 23 Oct 2006 22:27:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

The 2.6.18 works great, so if that version is going to be released this week, it makes sense to go with 2.6.18 rather than an older kernel.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [jonathankinney](#) on Mon, 23 Oct 2006 22:40:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have not done any cross-compilation, I have never had the need to do so. I also prefer to work on the system that I am building for, it tends to simplify the process.

I have considered in the past using qemu, but my specific interest is in the T1 CPU, and using the actual hardware is preferred. Depending on how things progress, if I want to further my project, I may consider using qemu for a short period until I get the actual hardware to work on directly. Thanks for the suggestions, I will keep them in mind.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [jonathankinney](#) on Mon, 23 Oct 2006 23:29:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oh, and another thing to note. Compiling the kernel is really slow unless you take advantage of all of the threads, so to speed things up, use:

```
make -j 23
```

or you can leave the number off and it will try and run all of the compile jobs simlutaneously which is sorta fun to see, 3k or so processes running at the same time with a load average topping out at 600, but the system is still very responsive.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility

Posted by [dev](#) on Tue, 24 Oct 2006 08:22:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

can you provide me the details on how to install the kernel after it was compiled with 'make' and reboot to it? When I yesterday rebooted it (to the default kernel! not the built one) it hanged... :/ strange... since I have no any console access I can't diagnose it

If you have 2.6.18 built for T1 somewhere (and working) I can easily apply OVZ patches I have and make it working very quickly.

Thanks,
Kirill

Subject: ALOM remote console information

Posted by [jonathankinney](#) on Tue, 24 Oct 2006 17:30:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well, when you built the 2.6.18 kernel and installed it, you did it correctly. The system uses SILO, so after you ran the install command, it either automatically made an entry in the /boot/silo.conf or you manually put it there. Either way, right after you built the kernel, I tested it out and booted with it. It booted perfectly, and it seems to handle high loads better than the 2.6.17 kernel. I just yesterday set the 2.6.18 kernel as the default boot kernel and the system is running off that kernel right now. Right now it looks like you have built and installed what I am assuming is the patched kernel Linux-2.6.18-028test001, and it looks like it is all ready to test and see if it boots.

As for console access, you got it, I just set you up with access to the console, I will send you the details in private. Did I mention I love this server? Remote console access is such a usefull feature. Right now I have it set to autoboot from the first drive, so it will make it to the SILO boot prompt, and there you can just hit enter to use the default kernel, or type the kernel by label such as Linux-2.6.18-028test001 and see if it boots all the way.

Here are some simple notes on the ALOM system I will tell you how to access in private. After you login, you can switch to the console by typing "console -f", the "-f" is for force, so you can kick anyone else out of the console so you have write access, otherwise, you will only be able to view the console. To get out of the console at any time, simply type "#." and it will bring you back to the main sc> prompt of the ALOM system. To reboot the system, if it is not possible via the normal OS commands like "shutdown -r now", then in the sc> prompt, you have several command options such as poweroff, poweron, powercycle, and reset. Some of the commands are redundant, so powercycle is usually the suggested choice. After you do that, you can then switch back to the console and watch the boot messages and eventually feed the SILO boot prompt a command. There are alot of other options, type help for a list, but most others are not relevent to this situation.

There is one bug to note that I just ran into last night. It seems that it is possible for the ALOM

system to dump your connection and not be accessible for a period of time. I have only had this happen one time, and I was able to connect to it again after a few minutes. It did not however disrupt access to the host server, so I was still able to login and work on the Ubuntu server without any problems.

I do live in Washington (USA), so I am in the PDT time zone GMT -700, soon to be GMT -800 due to daylight savings. So our days are a bit off from each other, I will keep that in mind.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [jonathankinney](#) on Tue, 24 Oct 2006 19:05:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

I just tried to boot using the Linux-2.6.18-028test001 you built, and it got a ways into booting, but then died with the following OOPS message:

```
[ 4.664578] rtc_init: no PC rtc found
[ 5.531306] OOPS: Bogus kernel PC [0000000000000000] in fault handler
[ 6.252782] OOPS: RPC [0000000000474fdc]
[ 6.705507] OOPS: Fault was to vaddr[f7ea4000]
[ 7.220241] Unable to handle kernel NULL pointer dereference
[ 7.826716] tsk->{mm,active_mm}->context = 000000000000010a
[ 8.370634] tsk->{mm,active_mm}->pgd = ffff801fd04e000
[ 8.561620] TRAPLOG: Trap level 1 TSTATE[ffffffffffffd] TPC[000000000693c00]
TNPC[0000000000000000] TT[1]
[ 8.919896] TRAPLOG: Trap level 2 TSTATE[00000000f7ea4000] TPC[00000000f7ea4000]
TNPC[ffff801ff4c8000] TT[80009000]
[ 9.315739] SUN4V-DTLB: Error at TPC[4193b8], tl 8
[ 9.487710] SUN4V-DTLB: vaddr[ffffffffffe000] ctx[0] pte[800007fffffe743] error[2]
Program terminated
{3} ok
```

The first line is normal, and the last line is just the OpenBoot ok prompt. I will go ahead and boot it back into the unpatched 2.6.18 kernel for now and take a look into this as soon as I get a chance.

Subject: Re: Sun UltraSPARC T1 CPU architecture compatibility
Posted by [jonathankinney](#) on Thu, 16 Nov 2006 19:53:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Just to update this thread, work is moving along very quickly thanks to all of the work that Kirill is putting into fixing the bugs that I am so gifted in finding. We have already setup several Debian based VE's and they all have worked perfectly. Right now Kirill is working on ironing out some kernel stability issues. This project is very promising, and I expect that we will be purchasing similar or identical hardware to put into production with OpenVZ very soon.

Subject: OpenVZ UltraSparc support, maybe soon Virtuozzo?

Posted by [jonathankinney](#) on Fri, 01 Dec 2006 19:16:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

Here is something I have never touched on. I do know that bug fixes and new features from OpenVZ seem to make a direct translation to the Virtuozzo project, but I have a few questions. Will the UltraSparc64 port be making its way into the Virtuozzo product? I am hoping so. Virtuozzo has several important extra functions added to make it a very strong system for mass virtualization, so in my opinion, when dealing with using virtualization to provide a commercial service, to make the most cost effective offering, Virtuozzo comes into the picture. In other words, in the near future, Virtuozzo supporting UltraSparc64 will be a welcome change.

Subject: Re: OpenVZ UltraSparc support, maybe soon Virtuozzo?

Posted by [dev](#) on Mon, 04 Dec 2006 19:01:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

Jonathan, it's not me who decide, but still can definitely say that virtuozzo is moving in this direction However, can't give any estimations about timelines :/

Subject: Re: OpenVZ UltraSparc support, maybe soon Virtuozzo?

Posted by [slybob](#) on Thu, 17 May 2007 13:40:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

http://forum.openvz.org/index.php?t=tree&goto=13008&#msg_13008

Thanks,

Andy
