

Saturday, March 31. 2012

### **Yet another cheatsheet update ...**

Yet another update of the Solaris 11 cheatsheet. Some additions about disk devices, some bug fixes (there is one additional bug "Configuring system locale:" must be "Configuring time zone" ... will be fixed in the next version)

Posted by Joerg Moellenkamp in English, Solaris at 19:18

Friday, March 30. 2012

### **Linux on 8-bit**

This is nothing short of extremely cool from a technical perspective. The author has done it by writing an ARM emulator for an AVR controller and running Linux with this emulation : Linux on an 8-bit micro?. This is definitely not the fastest, but I think it may be the cheapest, slowest, simplest to hand assemble, lowest part count, and lowest-end Linux PC. The board is hand-soldered using wires, there is not even a requirement for a printed circuit board

Posted by Joerg Moellenkamp in English at 14:46

Sunday, March 25. 2012

### **SPARC Power Management Article at OTN**

My colleague Karoly Vegh pointed in a tweet to a really interesting article about the usage of Power Management of SPARC T-series systems. The article explains how to use the power management, how it works, what it's able to do and how to use it in a dynamic fashion according to anticipated load patterns. You find the article "How to Use the Power Management Controls on SPARC Servers" written by by Bruce Evans, Julia Harper, and Terry Whatley on OTN.

Posted by Joerg Moellenkamp in English, Solaris at 13:50

Saturday, March 24. 2012

### **Oracle Solaris 11 Developer Webinar Series**

Starting March 27th a number of events called "Oracle Solaris 11 Developer Webinar Series" (that said, i think they are not only interesting for developers, but for admins as well that support admins in their job) will take place by WebEx/Phone:

March 27th: Modern Software Packaging for Enterprise Developers  
April 10th: Simplify Your Development Environment with Zones, ZFS & More  
April 24th: Managing Application Services – Using SMF Manifests in Solaris 11  
May 8th: Optimize Your Applications on Oracle Solaris 11: The DTrace Advantage  
May 22nd: Maximize Application Performance and Reliability on Oracle Solaris 11  
June 5th: Writing Oracle Solaris 11 Device Drivers

All events start at 9am Pacific Daylight Time (Should be 18:00 in Germany, 17:00 in UK, 01:00 in Tokyo). Of course the events are for free (before someone asks) and available for anyone who signs up. The chat is staffed with Solaris 11 Engineering Experts sitting in to answer questions. And most important: Mostly technical content and the events will be recorded.

Posted by Joerg Moellenkamp in English, Solaris at 09:24

Friday, March 23. 2012

### **Solaris 11 cheatsheet updated**

There is an updated version of the Solaris 11 cheatsheet.

Posted by Joerg Moellenkamp in English, Solaris at 06:46

Thursday, March 22. 2012

### **NGN/NLUUG conferentie vj2012: Operating Systems**

On April 11th, 2012 the Spring 2012 conference with the topic overarching topic "Operating Systems" takes place in Nieuwegein near Utrecht. Besides talks about Linux, Windows and AIX, there will be a track about Solaris. I will be the first speaker in the Solaris track and giving an overview about Solaris 11 and how features interact. Later on renowned experts like Detlef Drewanz ("Lifecycle Management with Oracle Solaris 11"), Andrew Gabriel ("Solaris 11 Networking - Crossbow Project"), Darren Moffat ("ZFS: Data integrity and Security") and Casper Dik ("Solaris 11 Zones and Immutable Zones") will take over. Finally Patrick Ale of UPC Broadband talks about his experiences with Solaris 11.

When you want more information about this conference or register for it, you will find the webpage of the event at the NLUUG site.

Posted by Joerg Moellenkamp in General at 10:51

Monday, March 19. 2012

### **Tutorial for Automated Installer**

Stefan Hinker wrote a nice tutorial helping you at the setup of a local install server: "Setting up a local AI server - easy with Solaris 11".

Posted by Joerg Moellenkamp in English, Solaris at 11:31

Friday, March 16. 2012

### **Performance impact of the new mtmalloc memory allocator**

I wrote at a number of occasions ([here](#) or [here](#)), that it could be really beneficial to use a different memory allocator for highly-threaded workloads, as the standard allocator is well ... the standard, however not very effective as soon as many threads comes into play.

I didn't wrote about this as it was in my phase of silence but there was some change in the allocator area, Solaris 10 got a revamped mtmalloc allocator in version Solaris 10 08/11 (as described in "libmtmalloc Improvements"). The new memory allocator was introduced to Solaris development by the PSARC case 2010/212.

But what's the effect of this new allocator and how does it works? Rickey C. Weisner wrote a nice article with "How Memory Allocation Affects Performance in Multithreaded Programs" explaining the inner mechanism of various allocators but he also publishes test results comparing Hoard, mtmalloc, umem, new mtmalloc and the libc malloc. Really interesting read and a must for people running applications on servers with a high number of threads.

Posted by Joerg Moellenkamp in English, Oracle, Solaris at 08:50

Thursday, March 15. 2012

### **Oracle Solaris 11 Cheatsheet**

In the last few days i created a cheat sheet for Solaris 11 ... while it's still a work in progress (it will be surely longer in the future). it's already available on OTN as part of the "Evaluating Oracle Solaris 11"-Collection . You can download the "Oracle Solaris 11 Cheat Sheet General Administration" [here](#).

Posted by Joerg Moellenkamp in English, Solaris at 08:06

Friday, March 9, 2012

## How to show off live migration with a SPARC system?

Yesterday i had the opportunity to show Oracle VM for SPARC in front of customers in action. Not a single slide was used ... Everything was live . The following entry shows what i essentially did in this demo. Perhaps long time users of LDOMS or Oracle VM for SPARC (as they are called today) have already seen for this, however that wasn't the planned audience of this walkthrough. In this example i've configured the control domain, one guest domain, installed it with Solaris 11 and migrated it live (without service interruption) from one system to another)Okay, i started with two unconfigured (okay, to be exact ... deconfigured systems) system of the type SPARC T3-4. So i had plenty resources to play. The first system was node1 listening on 10.128.0.72, the second system was node2 listening on 10.128.0.73.

Just to be sure, i checked the configuration.

```
node1:~# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-c-- UART  512  261632M 0.4% 5m
```

There was just a single logical domain with all resources (512 virtual CPUs and 256 GB memory were assigned to it. The situation on the second node was the same. No wonder. Same HW config, same SW config.

```
node2:~# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-c-- UART  512  261632M 0.3% 8m
```

Ensure that you have enabled the vntsd daemon on both systems

```
node1:~# svcadm enable vntsd
node2:~# svcadm enable vntsd
```

Okay, basics were the same, now i had to start the basic config. Important is those single large domains will act as so called control domains, however they will significantly smaller for that task. The already running Solaris 10 was kept unharmed and got the OS of the control domain.

First step was to configure the virtual console server:

```
node1:~# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

With this command you configure a console server listening on ports 5000 to 5100 named primary-vcc0 in the domain primary. Okay, the next step was to configure the so called virtual disk server. As long as you don't configure any hardware directly into a domain like a networking card for for iSCSI or a HBA for storage access, the virtual disk server is the daemon that provides the service of storage devices to all guest domains.

```
node1:~# ldm add-vds primary-vds0 primary
```

With this command we have configured a virtual disk server called primary-vds0 in the domain vds. The next step is the configuration of the networking. For this task we configure a virtual switch.

```
node1:~# ldm add-vsw net-dev=igb0 primary-vsw0 primary
```

The virtual switch called primary-vsw0 is running in the domain primary and it's connecting into the real world via the device igb0. When you want to check all the services you have just configured, you can do this with a single command.

```
node1:~# ldm list-services primary
```

```
VCC
NAME      LDOM      PORT-RANGE
primary-vcc0  primary  5000-5100

VSW
NAME      LDOM      MAC          NET-DEV  ID  DEVICE  LINKPROP  DEFAULT-VLAN-ID  PVID  VID
      MTU  MODE  INTER-VNET-LINK
primary-vsw0  primary  00:14:4f:fa:df:5c  igb0    0  switch@0  1          1          1500
on
```

```
VDS
NAME      LDOM      VOLUME      OPTIONS      MPGROUP      DEVICE
primary-vds0  primary
```

node1:/#

At the moment this primary domain is using all the resources. In order to be able to configure some guests, we have to free some room. So at first we reduce the number of assigned crypto units. I just want to give them one.

```
node1:/# ldm set-mau 1 primary
```

In the next step we assign 8 processor to the domain.

```
node1:/# ldm set-vcpu 8 primary
```

Okay, let's check the current configuration.

```
node1:/# ldm ls
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART	8	261632M	16%	20m

Okay, just 8 virtual cpus, however the domain still occupied all the memory in the system. We have to reduce that. Technically it's possible to do this on the running system but getting a running logical domain from 256 GB to 8 GB is quite some work, so most often it is just faster to put the domain in the deferred configuration mode, do the configuration and reboot the system as in this moment nothing runs on the system. When doing deferred reconfiguration, the config change is accepted but it will be executed with the next reboot:

```
node1:/# ldm start-reconf primary
```

Initiating a delayed reconfiguration operation on the primary domain.

All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

Now we set the memory of the domain primary to 8 GB

```
node1:/# ldm set-memory 8G primary
```

-----  
Notice: The primary domain is in the process of a delayed reconfiguration.  
Any changes made to the primary domain will only take effect after it reboots.  
-----

Saving the config to the ILOM, and rebooting the system.

```
node1:/# ldm add-config initial
```

```
node1:/# shutdown -y -g0 -i6
```

Okay, while the first system is rebooting, we just repeat the same configuration steps on the second system:

```
node2:/# ldm add-vc port-range=5000-5100 primary-vc0 primary
```

```
node2:/# ldm add-vds primary-vds0 primary
```

```
node2:/# ldm add-vsw net-dev=igb0 primary-vsw0 primary
```

```
node2:/# ldm list-services primary
```

```
VCC
  NAME      LDOM      PORT-RANGE
  primary-vc0  primary  5000-5100

VSW
  NAME      LDOM      MAC          NET-DEV  ID  DEVICE  LINKPROP  DEFAULT-VLAN-ID  PVID  VID
  primary-vsw0  primary  00:14:4f:f9:b9:9c  igb0    0  switch@0  1          1              1500
on
```

```
VDS
  NAME      LDOM      VOLUME      OPTIONS      MPGROUP      DEVICE
  primary-vds0  primary
```

```
node2:/# ldm set-mau 1 primary
```

```
node2:/# ldm set-vcpu 8 primary
```

Removal of cpu 118 failed, error: Removal of cpu 48 failed, error: Removal of cpu 16 failed, error: Domain primary permitted removal of only 501 VCPUs

Retry removed 3 more VCPUs from domain primary

```
node2:/# ldm ls
```

```
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary   active -n-cv- UART  8    261632M  16%  15m
```

```
node2:/# ldm start-reconf primary
```

Initiating a delayed reconfiguration operation on the primary domain.

All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

```
node2:/# ldm set-memory 8G primary
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
node2:/# ldm add-config initial
```

```
node2:/# shutdown -y -g0 -i6
```

We now check the config on both systems. On the first system:

```
node1:/# ldm ls
```

```
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary   active -n-cv- UART  8    8G    0.6%  8m
```

On the second system.

```
node2:/# ldm ls
```

```
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary   active -n-cv- UART  8    8G    8.1%  7m
```

Okay ... i have to explain a little bit ... 10.10.1.37 is a S7000 filer i've used for central storage. In the directory /ldoms/isos i've put a iso of the solaris 11 11/11 text install image.

```
node1:/# mount 10.10.1.37:/export/ldoms /ldoms
```

As i want to install the ldom i will create later on, i add this iso to the virtual disk server as a device:

```
node1:/# ldm add-vdsdev /ldoms/isos/sol-11-1111-text-sparc.iso sol11iso@primary-vds0
```

Okay, i want to demo live migration in this walkthrough, so i need some shared storage. It's obvious why i need shared storage: It makes no sense to migrate a logical domain to a system that hasn't access to the same disk devices. So i configured my S7000 filer to offer some LUNs via iSCSI. However i have to configure the primary domain in order to actually use this LUNs. However this is pretty easy. At first we tell the iSCSI initiator of Solaris 10 that there are disk to find behind 10.10.1.37

```
node1:/# iscsiadm add discovery-address 10.10.1.37
```

Now we tell Solaris to discover the LUNs behind this IP.

```
node1:/# iscsiadm modify discovery -t enable
```

And now we populate the /dev tree with the necessary nodes.

```
node1:/# devfsadm -i iscsi
```

Okay, repeat on the second system.

```
node2:/# mount 10.10.1.37:/export/ldoms /ldoms
```

```
node2:/# ldm add-vdsdev /ldoms/isos/sol-11-1111-text-sparc.iso sol11iso@primary-vds0
```

```
node2:/# iscsiadm add discovery-address 10.10.1.37
```

```
node2:/# iscsiadm modify discovery -t enable
```

```
node2:/# devfsadm -i iscsi
```

Okay, let's have a look what the system has found. From the configuration in the filer i knew that there must be something like 600144F0C56DC0FB00004F586FD60004 in the disk id. As the disk is unlabeled at that time the format command will offer to you do this labeling. Do it ... you need a labeled disk.

```
node2:/# format
```

```
Searching for disks...done
```

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

c0t600144F0C56DC0FB00004F586FD60004d0: configured with capacity of 9.94GB

AVAILABLE DISK SELECTIONS:

0. c0t5000C5003A02EE93d0  
/scsi\_vhci/disk@g5000c5003a02ee93

[...]

13. c0t600144F0C56DC0FB00004F586FD60004d0  
/scsi\_vhci/ssd@g600144f0c56dc0fb00004f586fd60004

[...]

Specify disk (enter its number): 13

selecting c0t600144F0C56DC0FB00004F586FD60004d0

[disk formatted]

Disk not labeled. Label it now? Disk not labeled. Label it now? y

FORMAT MENU:

disk - select a disk

[...]

! - execute , then return

quit

format> quit

Okay, now check the availability of disk disk on the other server.

node1:/# format

Searching for disks...done

c0t600144F0C56DC0FB00004F586FD60004d0: configured with capacity of 9.94GB

AVAILABLE DISK SELECTIONS:

0. c0t5000C5003A02FD67d0  
/scsi\_vhci/disk@g5000c5003a02fd67

[..]

13. c0t600144F0C56DC0FB00004F586FD60004d0  
/scsi\_vhci/ssd@g600144f0c56dc0fb00004f586fd60004

[..]

Specify disk (enter its number):

Specify disk (enter its number):

Check ... is there. The next step is the last one in this tour we have to execute on both systems. With this command we assign the disk /dev/dsk/c0t600144F0C56DC0FB00004F586FD60004d0s2 on both nodes as lmtest001iscsibootdisk to the virtual disk server called primary-vds0

```
node1:/# ldm add-vdsdev /dev/dsk/c0t600144F0C56DC0FB00004F586FD60004d0s2
```

```
lmtest001iscsibootdisk@primary-vds0
```

```
node2:/# ldm add-vdsdev /dev/dsk/c0t600144F0C56DC0FB00004F586FD60004d0s2
```

```
lmtest001iscsibootdisk@primary-vds0
```

Okay, now we configure our first guest domain.

At first we just create the domain.

```
node1:/# ldm add-domain lmtest001
```

Now we add 8 virtual CPUs to the domain.

```
node1:/# ldm add-vcpu 8 lmtest001
```

Of course a domain needs memory, so i give it 16 GB.

```
node1:/# ldm add-mem 16G lmtest001
```

Now i'm creating a networking interface for the domain lmtest001 connected to the virtual switch primary-vsw0 and naming it vnet1.

```
node1:/# ldm add-vnet vnet1 primary-vsw0 lmtest001
```

Okay, as my iscsi disk is totally empty, i have to provide an installation image ( i could do this via jumpstart or AI, however that would be out of scope of this short article). So i assign the virtual disk sol11iso on the virtual disk server primary-vds0 (remember, we configured them earlier) to lmtest001. To the domain it's named vdisk\_iso.

```
node1:~# ldm add-vdisk vdisk_iso sol11iso@primary-vds0 lmtest001
```

Now i have to assign the iscsi boot disk to the domain. The command is quite similar.

```
node1:~# ldm add-vdisk bootdisk lmtest001iscsibootdisk@primary-vds0 lmtest001
```

The next stop is to declare the bootdevice and to tell the system to boot automatically from it.

```
node1:~# ldm set-var auto-boot\?=true lmtest001
```

```
node1:~# ldm set-var boot-device=bootdisk lmtest001
```

However, the domain is still inactive and no resources have been binded to the domain

```
node1:~# ldm ls
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART 8	8G	13%	35m	
lmtest001	inactive	-----	8 16G				

So we bind the resources with a single command:

```
node1:~# ldm bind-domain lmtest001
```

When you look up the status again, you see a state transition. The domain isn't "inactive" any longer, it's now bound.

```
node1:~# ldm ls
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	UART 8	8G	0.5%	36m	
lmtest001	bound	-----	5000 8	16G			

Now it's time to start up the domain.

```
node1:~# ldm start-domain lmtest001
```

```
LDom lmtest001 started
```

When you look back into the last output of ldm ls you will see the 5000 in the column "CONS" (short for console) for lmtest001. This 5000 is now important to get access to the console of the lmtest001 domain.

```
node1:~# telnet localhost 5000
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
  
Connecting to console "lmtest001" in group "lmtest001" ....  
Press ~? for control options ..  
  
{0} ok
```

As you see, there is a boot prompt like with a native SPARC machine. As there is no operating system on the device we've called bootdisk, the system doesn't come up but stays in that prompt.

```
{0} ok devalias
```

```
bootdisk      /virtual-devices@100/channel-devices@200/disk@1  
vdisk_iso     /virtual-devices@100/channel-devices@200/disk@0  
vnet1        /virtual-devices@100/channel-devices@200/network@0  
net          /virtual-devices@100/channel-devices@200/network@0  
disk         /virtual-devices@100/channel-devices@200/disk@0  
virtual-console /virtual-devices/console@1  
name         aliases
```

Now let's boot from the iso image:

```
{0} ok boot vdisk_iso:f
```

Okay, at first the system comes up from the ISO as you see

```
SPARC T3-4, No Keyboard
```

```
Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
OpenBoot 4.33.0.b, 16384 MB memory available, Serial #83470411.
```

```
Ethernet address 0:14:4f:f9:a8:4b, Host ID: 84f9a84b.
```

```
Boot device: /virtual-devices@100/channel-devices@200/disk@0:f File and args:
```

```
SunOS Release 5.11 Version 11.0 64-bit
```

# Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.

Okay, this will now take a while. I won't write about it. It's a standard Solaris 11 install. You know the drill.

After the reboot initiated in the installation procedure, the systems comes up with an installed OS. As you will recognize by the string of the boot device, that you now have booted from the iscsi boot disk.

SPARC T3-4, No Keyboard

Copyright (c) 1998, 2011, Oracle and/or its affiliates. All rights reserved.

OpenBoot 4.33.0.b, 16384 MB memory available, Serial #83470411.

Ethernet address 0:14:4f:f9:a8:4b, Host ID: 84f9a84b.

Boot device: /virtual-devices@100/channel-devices@200/disk@1:a File and args:

SunOS Release 5.11 Version 11.0 64-bit

Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights reserved.

Loading smf(5) service descriptions: 198/198

Configuring devices.

Loading smf(5) service descriptions: 1/1

Hostname: solaris

solaris console login:

Okay, let's play a little bit with the domain. Login into the shell of the system. When you execute an prtdiag, you will see 16 GB of memory and 8 virtual CPUs.

```
jmoekamp@solaris:~$ prtdiag
```

```
System Configuration: Oracle Corporation sun4v SPARC T3-4
```

```
Memory size: 16384 Megabytes
```

```
===== Virtual CPUs =====
```

CPU ID	Frequency	Implementation	Status
0	1649 MHz	SPARC-T3	on-line
1	1649 MHz	SPARC-T3	on-line
2	1649 MHz	SPARC-T3	on-line
3	1649 MHz	SPARC-T3	on-line
4	1649 MHz	SPARC-T3	on-line
5	1649 MHz	SPARC-T3	on-line
6	1649 MHz	SPARC-T3	on-line
7	1649 MHz	SPARC-T3	on-line

```
===== IO Devices =====
```

Slot +	Bus	Name +	Model
Status	Type	Path	

```
jmoekamp@solaris:~$
```

Okay, let's assume we've changed our mind and want a domain with 8 additional virtual CPUs. You can do this while the domain is running:

```
node1:/# ldm add-vcpu 8 lmtest001
```

```
node1:/#
```

When you do another prtdiag in the still running domain, you will see 16 virtual CPUs.

```
jmoekamp@solaris:~$ prtdiag
```

```
[...]
```

===== Virtual CPUs =====

CPU ID	Frequency	Implementation	Status
0	1649 MHz	SPARC-T3	on-line
1	1649 MHz	SPARC-T3	on-line
2	1649 MHz	SPARC-T3	on-line
3	1649 MHz	SPARC-T3	on-line
4	1649 MHz	SPARC-T3	on-line
5	1649 MHz	SPARC-T3	on-line
6	1649 MHz	SPARC-T3	on-line
7	1649 MHz	SPARC-T3	on-line
8	1649 MHz	SPARC-T3	on-line
9	1649 MHz	SPARC-T3	on-line
10	1649 MHz	SPARC-T3	on-line
11	1649 MHz	SPARC-T3	on-line
12	1649 MHz	SPARC-T3	on-line
13	1649 MHz	SPARC-T3	on-line
14	1649 MHz	SPARC-T3	on-line
15	1649 MHz	SPARC-T3	on-line

Okay, 8 additional 8G may be a nice idea. So let's add them to the running domain.

```
node1:~# ldm add-mem 8G lmtest001
node1:~#
```

Do another prtdiag, an we see 24 gigs of memory.

```
jmoekamp@solaris:~$ prtdiag
System Configuration: Oracle Corporation sun4v SPARC T3-4
Memory size: 24576 Megabytes
[...]
```

However, we aren't really in decision mood today and think that our first config was nice and revert to the old values. So we remove 8 gigs of memory again from the domain lmtest001

```
node1:~# ldm rm-mem 8G lmtest001
node1:~#
```

And again our domain has just 16 GB.

```
jmoekamp@solaris:~$ prtdiag
System Configuration: Oracle Corporation sun4v SPARC T3-4
Memory size: 16384 Megabytes
[...]
```

Now we have just to remove the 8 additional vcpus.

```
node1:~# ldm rm-vcpu 8 lmtest001
node1:~#
```

Okay, a last time we will execute prtdiag.

```
jmoekamp@solaris:~$ prtdiag
System Configuration: Oracle Corporation sun4v SPARC T3-4
Memory size: 16384 Megabytes
```

===== Virtual CPUs =====

CPU ID	Frequency	Implementation	Status
0	1649 MHz	SPARC-T3	on-line
1	1649 MHz	SPARC-T3	on-line
2	1649 MHz	SPARC-T3	on-line
3	1649 MHz	SPARC-T3	on-line
4	1649 MHz	SPARC-T3	on-line
5	1649 MHz	SPARC-T3	on-line
6	1649 MHz	SPARC-T3	on-line
7	1649 MHz	SPARC-T3	on-line

```
===== IO Devices =====
Slot +      Bus  Name +      Model
Status      Type Path
-----
```

Again, back to 8 virtual CPUs in the domain.

Okay, but now back to our demonstration of live migration. I would like to demonstrate the live migration with some network traffic, so i need an ip address. So i configure one on the vnet0 interface card i've created when i was configuring the domain earlier. Log into the domain as root or assume a role that allows you to configure networking.

```
root@solaris:/home/jmoekamp# dladm show-phys
LINK      MEDIA      STATE  SPEED DUPLEX  DEVICE
net0      Ethernet   unknown 0      unknown vnet0
root@solaris:/home/jmoekamp# ipadm create-ip net0
root@solaris:/home/jmoekamp# ipadm create-addr -T static -a 10.10.0.74/16 net0ip4
root@solaris:/home/jmoekamp# route -p add default 10.10.0.1
add net default: gateway 10.10.0.1
add persistent net default: gateway 10.10.0.1
root@solaris:/home/jmoekamp#
```

Just a short test from my local workstation. Just a short remark ... the times are that bad, because my servers were in Scotland and i was in the Düsseldorf FTL lounge connected via VPN over an UTMS line ...

```
PS C:\Windows\System32\WindowsPowerShell\v1.0> ping -t 10.10.0.74
```

```
Ping wird ausgeführt für 10.10.0.74 mit 32 Bytes Daten:
Antwort von 10.10.0.74: Bytes=32 Zeit=183ms TTL=253
Antwort von 10.10.0.74: Bytes=32 Zeit=94ms TTL=253
Antwort von 10.10.0.74: Bytes=32 Zeit=142ms TTL=253
```

Okay, let's kick of the live migration. With this command i order the logical domain manager to migrate the domain lmtest001 to the server running a control domain on 10.128.0.73. The password was in my case the root password of that control domain.

```
node1:/# ldm migrate lmtest001 10.128.0.73
Target Password:
node1:/#
```

This will take a while, however you will just get back your prompt in a very unspectacular way. You will get back the prompt as soon as the migration has completed.

What has in the meantime happened?

```
PS C:\Windows\System32\WindowsPowerShell\v1.0> ping -t 10.10.0.74
```

```
Ping wird ausgeführt für 10.10.0.74 mit 32 Bytes Daten:
Antwort von 10.10.0.74: Bytes=32 Zeit=497ms TTL=253
Antwort von 10.10.0.74: Bytes=32 Zeit=1371ms TTL=253
```

Antwort von 10.10.0.74: Bytes=32 Zeit=323ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=118ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=80ms TTL=253  
Zeitüberschreitung der Anforderung.  
Antwort von 10.10.0.74: Bytes=32 Zeit=67ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=209ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=86ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=83ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=70ms TTL=253  
Antwort von 10.10.0.74: Bytes=32 Zeit=105ms TTL=253

Ping-Statistik für 10.10.0.74:

Pakete: Gesendet = 286, Empfangen = 285, Verloren = 1  
(0% Verlust),

Ca. Zeitangaben in Millisek.:

Minimum = 66ms, Maximum = 1473ms, Mittelwert = 165ms

STRG-C

PS C:\Windows\System32\WindowsPowerShell\v1.0>

Well a ping has been lost. However what's more interesting, there isn't a domain lmtest001 on my server.

```
node1:~# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-cv- UART  8   8G   0.5% 2h 6m
node1:~#
```

Because it's on the other one.

```
node2:/ldoms/isos# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-cv- UART  8   8G   0.4% 2h 5m
lmtest001 active -n---- 5000  8   16G   0.1% 1h 8m
```

Of course i could migrate back to my old system.

```
node2:/ldoms/isos# ldm migrate lmtest001 10.128.0.72
```

Target Password:

It has disappered from this server

```
node2:/ldoms/isos# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-cv- UART  8   8G   0.3% 2h 13m
```

And is now back on the original one.

```
node1:~# ldm ls
NAME      STATE  FLAGS CONS  VCPU MEMORY UTIL UPTIME
primary   active -n-cv- UART  8   8G   0.5% 2h 15m
lmtest001 active -n---- 5000  8   16G   0.2% 1h 18m
node1:~#
```

Neat.

Do you want to learn more?

docs.oracle.com: Oracle VM for SPARC Documentation

Posted by Joerg Moellenkamp in General at 17:03

Saturday, March 3. 2012

## Cheatsheet for configuring a new nodename in Solaris 11

In Solaris 11 the file /etc/nodename is gone. In order to change you nodename you have alter the configuration of an smf service:root@fuchikoma:/home/jmoekamp# hostname tachikoma

```
root@fuchikoma:/home/jmoekamp# svccfg -s svc:/system/identity:node setprop config/nodename = "fuchikoma"
```

```
root@fuchikoma:/home/jmoekamp# svcadm refresh svc:/system/identity:node
```

```
root@fuchikoma:/home/jmoekamp# svcadm restart svc:/system/identity:node
```

```
root@fuchikoma:/home/jmoekamp# hostname
```

```
fuchikoma
```

You don't have to edit config/loopback

```
root@fuchikoma:/home/jmoekamp# svcprop svc:/system/identity:node | egrep "^config"
```

```
config/nodename astring fuchikoma
```

```
config/loopback astring fuchikoma
```

Posted by Joerg Moellenkamp in English, Solaris at 20:39

## Cheatsheet for configuring the networking in Solaris 11

There are quite a number of changes in the procedures to configure some of the networking parameters. Many things have changed, that were just editing of a file in the past, have now command-line based tools in order to change their parameters. Before you ask: The reason for this steps are quite simple.

At first ist much easier to script the execution of a command, instead of the editing a file. I wrote a lot of those scripts in the past and from my experiences people have an infinite wisdom in creating ways and means to edit a file in a way that it breaks your automatic edting scripts. Furthermore putting all this configuration statements into SMF has an interesting consequence. In the past, it wasn't always clear to most people when a change of the configuration got active. At the time of the save, at the restart of the daemon? By using SMF it's clear. As soon as you type in svcadm refresh. Furthermore ifconfig came totally overloaded with parameters with the time. So dladm and ipadm were a very good move from my perspective.

However i know that this is maybe a strange for an long-time user, and to be honest, i was spelling many curses in front of the display, albeit i see the advantages. So i'm putting up my cheatsheet online. My colleague Detlef is planing something similar. As soon his is ready, i will linking to his as well, as i'm sure it's more complete.

With automatism

In the datacenter it's not that useful, however when you have an Solaris VM for example on your notebook you are using for administrative purposes, it's really nice.

At first we create the network configuration profiles for the datacenter and the office.root@tachikoma:/home/jmoekamp# netcfg create ncp datacenter

```
root@tachikoma:/home/jmoekamp# netcfg create ncp office
```

Now we have to fill both with the configuration data. At first the one for the datacenter. I think the values are speaking for themself.root@tachikoma:/home/jmoekamp# netcfg

```
netcfg> select ncp datacenter
```

```
netcfg:ncp:datacenter> create ncu phys net0
```

```
Created ncu 'net0'. Walking properties ...
```

```
activation-mode (manual) [manual|prioritized]> manual
```

```
link-mac-addr>
```

```
link-autopush>
```

```
link-mtu>
```

```
netcfg:ncp:datacenter:ncu:net0> end
```

```
Committed changes
```

```
netcfg:ncp:datacenter> create ncu ip net0
```

```
Created ncu 'net0'. Walking properties ...
```

```
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
```

```
ipv4-addrsrc (dhcp) [dhcp|static]> static
```

```
ipv4-addr> 192.168.1.27
ipv4-default-route> 192.168.1.1
netcfg:ncp:datacenter:ncu:net0> end
Committed changes
netcfg:ncp:datacenter> exit
root@tachikoma:/home/jmoekamp#Now we do the same for the office:root@tachikoma:/home/jmoekamp# netcfg
netcfg> select ncp office
netcfg:ncp:office> create ncu phys net0
Created ncu 'net0'. Walking properties ...
activation-mode (manual) [manual|prioritized]> manual
link-mac-addr>
link-autopush>
link-mtu>
netcfg:ncp:office:ncu:net0> end
Committed changes
netcfg:ncp:office> create ncu ip net0
Created ncu 'net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> dhcp
ipv4-default-route>
netcfg:ncp:office:ncu:net0> end
Committed changes
netcfg:ncp:office> exit
```

Now we can just switch the networking configuration by enabling one of the both profiles:

```
root@tachikoma:/home/jmoekamp# netadm enable -p ncp datacenter
Enabling ncp 'datacenter'
root@tachikoma:/home/jmoekamp# ifconfig net0
net0: flags=201000843 mtu 1500 index 6
    inet 192.168.1.27 netmask fffff00 broadcast 192.168.1.255
    ether 8:0:27:b6:1:70
```

Let's switch to the office profile:

```
root@tachikoma:/home/jmoekamp# netadm enable -p ncp office
Enabling ncp 'office'
root@tachikoma:/home/jmoekamp# ifconfig net0
net0: flags=201004843 mtu 1500 index 7
    inet 192.168.1.185 netmask fffff00 broadcast 192.168.1.255
    ether 8:0:27:b6:1:70
```

Of course the networking configuration is not just the interface configuration but a lot other stuff as well, link the configuration of the DNS. That options are in the location profiles. However i don't want to activate the profile as well as the location both manually. Thus i configure an automatic activation of such configuration. In my example the datacenter profile configures an IP address of 192.168.1.27, the office profile uses DHCP which results in an IP-address out of the 192.168.1.100-199 range. Thus i'm using this for my automatic location switching. I can configure an activation mode and by choosing conditional-any this location profile will be activated automatically when any of the configured conditions are true. It's pretty simple in my case: If the IP address is 192.168.1.27, i'm in the datacenter, if not i'm in the office.

So, i'm configuring the location profile for the datacenter first. I think this is speaking for itself again, when you take the information into consideration, i gave you in the last paragraph:

```
root@tachikoma:/home/jmoekamp# netcfg
netcfg> create loc datacenter
Created loc 'datacenter'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ip-address is 192.168.1.27
nameservices (dns) [dns|files|nis|ldap]> dns
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]> manual
dns-nameservice-domain> datacenter.c0t0d0s0.org
dns-nameservice-servers> 192.168.1.1
dns-nameservice-search>
dns-nameservice-sortlist>
dns-nameservice-options>
nfsv4-domain>
ipfilter-config-file>
```

```
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:datacenter>
netcfg:loc:datacenter> exit
Committed changes
Now i will configure the one for the office:root@tachikoma:/home/jmoekamp# netcfg
netcfg> create loc office
Created loc 'office'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ip-address is-not 192.168.1.28
nameservices (dns) [dns|files|nis|ldap]> dns
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]> manual
dns-nameservice-domain> office.c0t0d0s0.org
dns-nameservice-servers> 192.168.1.1
dns-nameservice-search>
dns-nameservice-sortlist>
dns-nameservice-options>
nfsv4-domain>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ipsecpolicy-config-file>
netcfg:loc:office>
netcfg:loc:office> exit
Committed changes
root@tachikoma:/home/jmoekamp# netcfgOkay, let's try this out. At first we activate the datacenter network
configuration profile.
root@tachikoma:/home/jmoekamp# netadm enable -p ncp datacenter
Enabling ncp 'datacenter'In theory, the resolv.conf should now change to nameserver 192.168.1.1. Let's check
this.root@tachikoma:/home/jmoekamp# cat /etc/resolv.conf

#
# Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
#

#
# AUTOGENERATED_FROM_SMF_V1
#
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
# See resolv.conf(4) for details.

domain datacenter.c0t0d0s0.org
nameserver 192.168.1.1Okay, that was successful. Now we will switch to the network configuration profile
office.root@tachikoma:/home/jmoekamp# netadm enable -p ncp office
Enabling ncp 'office'We have a look into the resolv.conf again. The nameserver should have switched to
192.168.1.43:root@tachikoma:/home/jmoekamp# cat /etc/resolv.conf

#
# Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
#

#
# AUTOGENERATED_FROM_SMF_V1
#
```

## Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

```
# WARNING: THIS FILE GENERATED FROM SMF DATA.
# DO NOT EDIT THIS FILE. EDITS WILL BE LOST.
# See resolv.conf(4) for details.
```

```
domain office.c0t0d0s0.org
nameserver 192.168.1.43Nice. Without that automatic stuff ....
Configuring all this stuff manually isn't hard as well. Perhaps you have chosen "Automatic" initially at install time in the
CUI. So at first we have to get rid of the automatism.root@tachikoma:/home/jmoekamp# netadm enable -p ncp
defaultfixed
Enabling ncp 'DefaultFixed'Okay, now we configure an ip interface called net0 and assign an ip
addressroot@tachikoma:/home/jmoekamp# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
net0          Ethernet      unknown 0      unknown pcn0
root@tachikoma:/home/jmoekamp# ipadm create-ip net0
root@tachikoma:/home/jmoekamp# ipadm create-addr -T static -a 192.168.1.28/24
net0/v4Of course we need a defaultrouter. With this command we configure such a persistent default
route.root@tachikoma:/home/jmoekamp# route -p add default 192.168.1.1
add net default: gateway 192.168.1.1
add persistent net default: gateway 192.168.1.1Configuring the dns server is a little bit more
complex:root@tachikoma:/home/jmoekamp# svccfg -s "dns/client" setprop "config/nameserver = net_address: (
192.168.1.1 )"
root@tachikoma:/home/jmoekamp# svccfg -s "dns/client" setprop 'config/domain = astring: ("c0t0d0s0.org")'
root@tachikoma:/home/jmoekamp# svccfg -s "name-service/switch" setprop 'config/host = astring: "files dns"'
root@tachikoma:/home/jmoekamp# svcadm refresh name-service/switch
root@tachikoma:/home/jmoekamp# svcadm enable dns/clientOkay, let's check the
result:root@tachikoma:/home/jmoekamp# grep "192.168.1.43" /etc/resolv.conf
nameserver 192.168.1.43
root@tachikoma:/home/jmoekamp# grep "host" /etc/nsswitch.conf
hosts: files dns
Everything works as designed.
```

Posted by Joerg Moellenkamp in English, Solaris at 13:49