DB Mobility Networks Logistics

DOAG 2013

Praxisbericht: T4-1 mit Oracle VM Server for SPARC

Jens Vogel
DB Systel GmbH

AGENDA

- Vorstellung
- Motivation
- Planung Testumgebung
- Virtualisierung SPARC
- Oracle VM Server für SPARC
- Virtualisierter oder Direct I/O
- Live Migration
- Konfiguration und Test
- Fazit

Informationen, Zahlen, Fakten

- Jens Vogel
- IT-Spezialist DB Systel GmbH
- Standort Erfurt
- Systemtechnik Plattform Unix

- DB Systel GmbH
- ICT-Dienstleister der Deutschen Bahn
- ca. 3.100 Mitarbeiter an den drei Hauptstandorten Frankfurt/Main, Berlin und Erfurt
- zwei Rechenzentren mit über 3.300 Servern
- Betreuung von ca. 340.000 IP-Adressen, 85.000 Büroarbeitsplätzen & 93.000 VoIP-Anschlüssen
- Gesamtspeicher-Kapazität von ca. 1,5 Petabyte & Backup-Kapazität von 4,5 Petabyte

- Deutsche Bahn AG
- Personen- und Gütertransport per Zug, Lkw, Schiff und Flugzeug
- die Nummer 2 weltweit unter den Transport- und Logistikunternehmen
- ca. 2,7 Milliarden Reisende pro Jahr (Bus & Bahn)
- ca. 25.000 Personenzüge pro Tag
- Streckennetz von rund 34.000 km mit ca. 5.700 Bahnhöfen
- Gütertransport im Umfang von 400 Millionen Tonnen pro Jahr (nur Schiene)
- 6 Millionen Quadratmeter Lagerfläche in 130 Ländern

Ziele/Voraussetzungen

- Ziele
  - Entwicklungs-Umgebung Install-Server für Solaris 10 und Oracle Solaris 11
  - Abnahme-Umgebung Install-Server für Solaris 10 und Oracle Solaris 11
  - AI mit mehreren Oracle Solaris 11-Repositories
  - JET für Solaris 10
  - je ein DHCP-Server für jede Umgebung und jede Solaris-Version
    - SunOS BOOTP/DHCP-Server für Solaris 10
    - ISC-DHCP-Server für Oracle Solaris 11

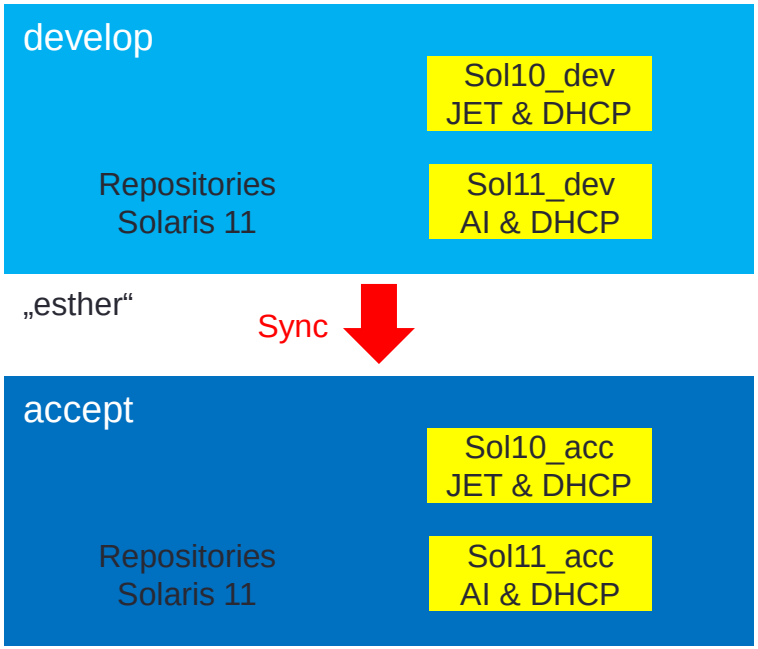
- Voraussetzungen
  - Erfahrung im produktiven Betrieb von ca. 1500 Solaris 10-Zonen
  - keine Erfahrungen mit LDOM‘s
  - 2 Server Oracle T4-1 (8 Cores, 8 Threads/Core, 256 GB RAM)

Quelle: www.oracle.com

Planung Testumgebungen

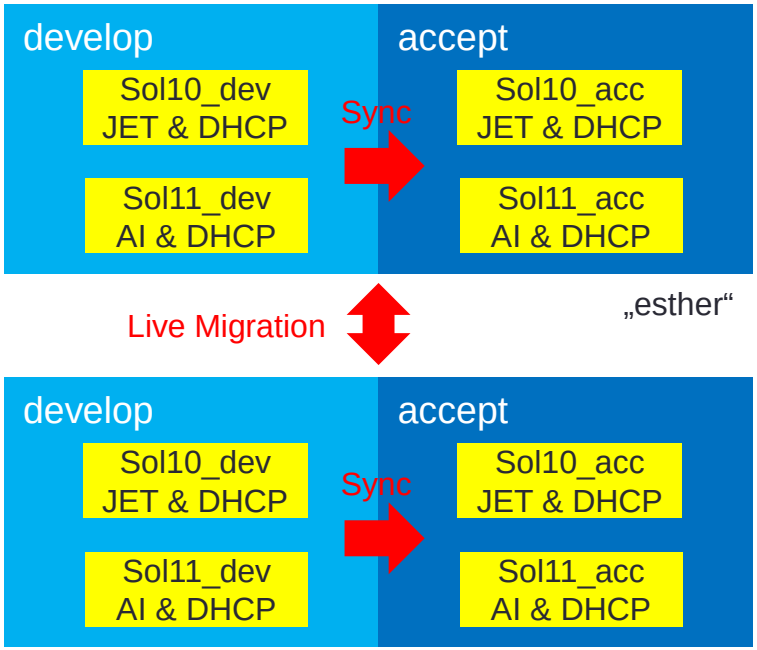
Entwurf

- jeweils ein dediziertes Entwicklungs- und Abnahmesystem
- keine Redundanz/Ausfallsicherheit



Konzept

- zwei identische Systeme mit LDOM‘s
- Redundanz/Ausfallsicherheit durch Live Migration



Warum virtualisieren?

- Server-Lösungen von heute
  - Systeme haben immer größere Thread-Dichte
  - große Systeme reduzieren Kosten (Minimierung Höhen-Einheiten, Strom, Klima, usw. im RZ)
  - aber nur selten benötigen Kunden/Applikationen solche großen dedizierten Server

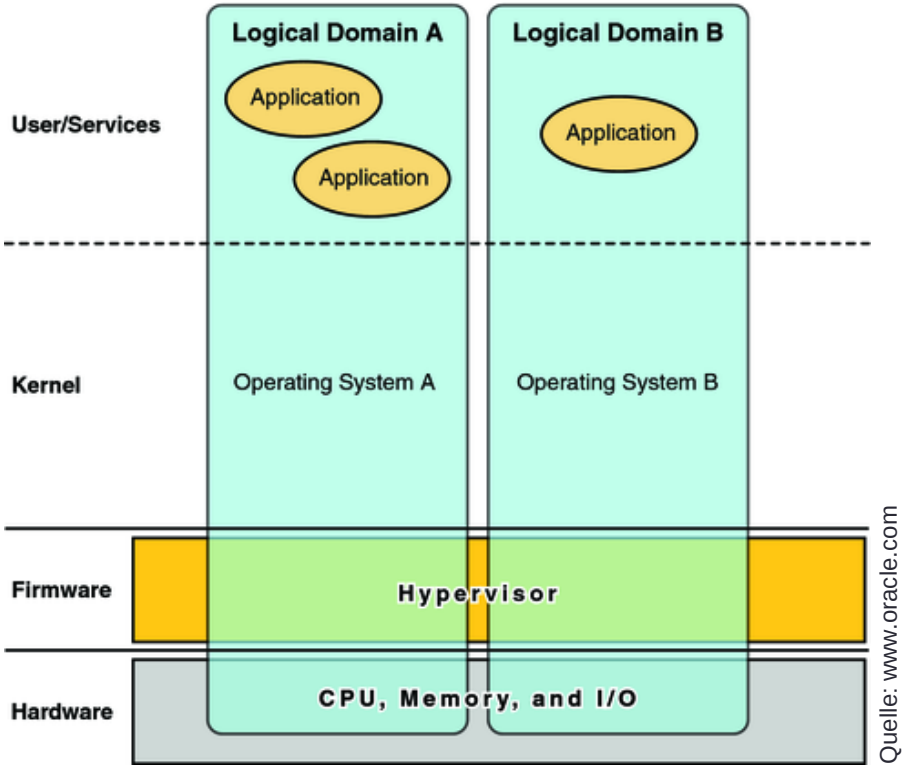
- Virtualisierung bietet:
  - effizientere Nutzung von Ressourcen
  - vereinfachte Administration
  - Minimierung der Kosten (z. B. Lizenzen)
  - schnelle Bereitstellung von Produktions-Umgebungen
  - kostengünstige Bereitstellung von Testsystemen

Virtualisierungsarten SPARC

- Dynamic System Domains (nur Mx000-Systeme)
  - Virtualisierung auf HW-Ebene
  - Physical System Boards (PSB) → Extended System Boards (XSB) → Logical System Boards (LSB)
  - jedes LSB eigene OS-Instanz
  - verschiedene Kernel- und Patchstände möglich
- Logical Domains (Oracle VM Server for SPARC)
  - Virtualisierung auf HW-Ebene (Hypervisor Typ 1)
  - verschiedene Kernel- und Patchstände möglich
- Solaris Zonen (Solaris Container)
  - Virtualisierung auf OS-Ebene (Kernelsharing)
  - keine verschiedenen Kernel- und Patchstände möglich

Oracle VM Server for SPARC

- OS-Virtualisierung auf Hypervisor-Technologie (Hypervisor Typ 1)
- früher: Solaris LDOMs (Logical Domains) - heute: Oracle VM Server for SPARC (aktuell in Version 3.1)
- früher: nur T-Systeme, da nur diese über Hypervisor verfügten (sun4v) - heute: alle SPARC-Systeme
- da eine LDOM einer eigenen Server-Instanz entspricht, ist eine Kombination mit Zonen möglich/gängig



Begrifflichkeiten 1

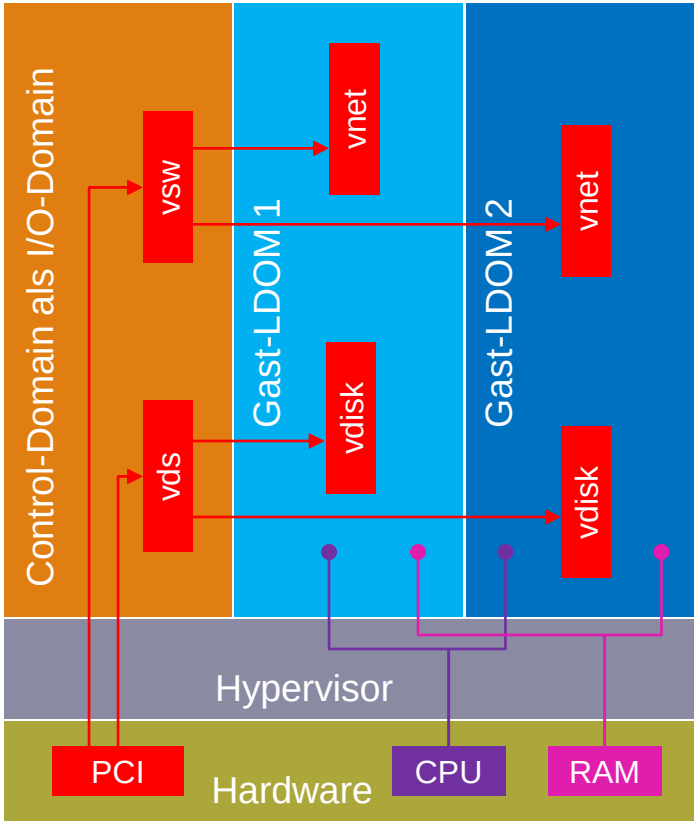
- Control-Domain
  - einzige Domain, die Hypervisor konfigurieren kann (es kann nur eine geben ☺)
  - dient zum Konfigurieren der Guest- und anderer Service-Domains (CPU, Memory)
  - bindet zu Anfang alle Ressourcen, d. h. die Control-Domain selbst muss ebenfalls konfiguriert werden
  - meist werden dieser nur geringe Ressourcen zugewiesen, da i. d. R. keine Produktiv-Systeme
  - zentrales Tool ist LDOM-Manager
- Service-Domain
  - Domain, die virtuelle I/O-Services für Guest-Domains bereitstellt (NIC, HDD, SAN)
  - werden virtuelle Services zur Verfügung gestellt, dann ist die Control-Domain auch eine Service-Domain
  - mehrere Service-Domains bilden Redundanzen (PCI-Erreichbarkeit)
- Guest-Domain
  - Domain, die als eigene OS-Instanz als Produktiv- oder Entwicklungs-System dient
- I/O-Domain
  - Domain, die direkten Zugriff auf PCI-Bus hat (ohne CPU & Memory)
  - jede Service-Domain ist auch eine I/O-Domain
- Root-Domain
  - Guest-Domain mit direktem Zugriff auf kompletten PCI-Bus incl. CPU & Memory (Direct I/O, Root Complex)
  - spezielle Form einer I/O-Domain

Begrifflichkeiten 2

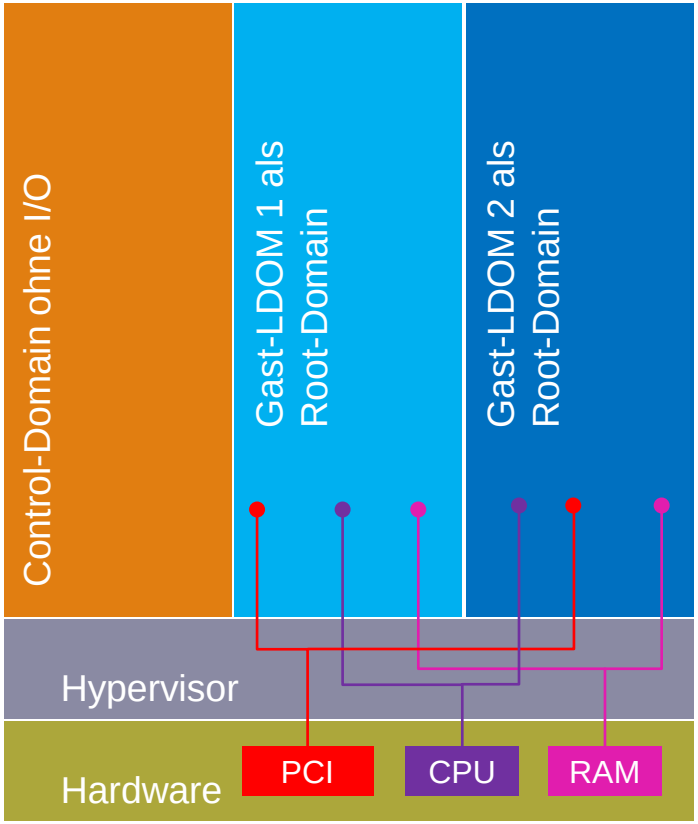
- vsw
  - virtual switch service
  - Service zur Netzwerk-Virtualisierung
  - nur in Service-Domains
- vds
  - virtual disk server
  - Service zur Disk-Virtualisierung
  - nur in Service-Domains
- vcc
  - virtual console concentrator service
  - stellt virtuelle Port-Ranges für serielle Konsolen zur Verfügung und bindet diese an vntsd
  - nur in Control- bzw. Service-Domains
  - Achtung, eine Guest-Domain kann nur an einen vcc gebunden werden (keine Redundanz)!
- vntsd
  - virtual network terminal server daemon
  - nur in Control- bzw. Service-Domains (abhängig von vcc)
  - Terminal-Server des Oracle Solaris
  - Standard-Staus „disabled“(!)

Virtualisierter oder Direct I/O

Klassische Control-/Service-Domain mit Gästen

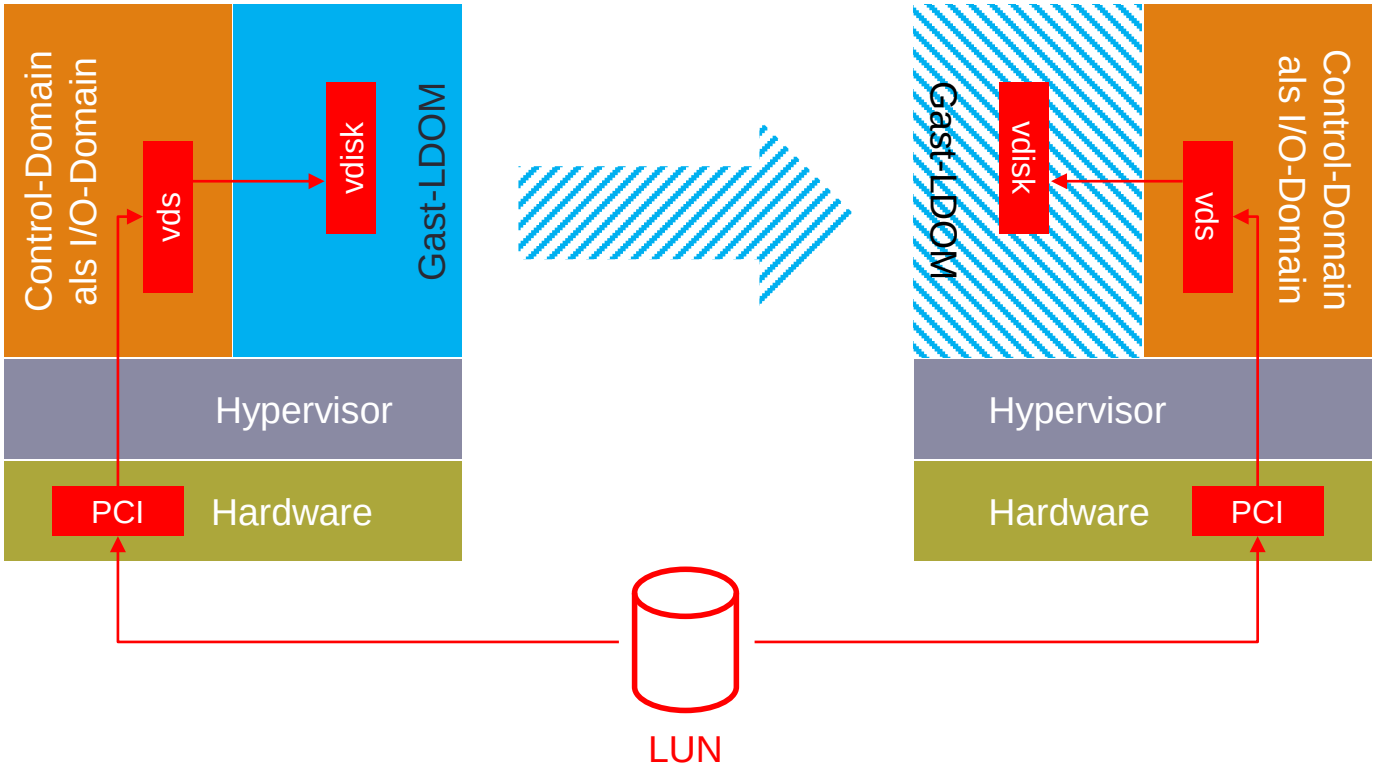


Control-Domain mit zwei I/O-Domains als Gäste

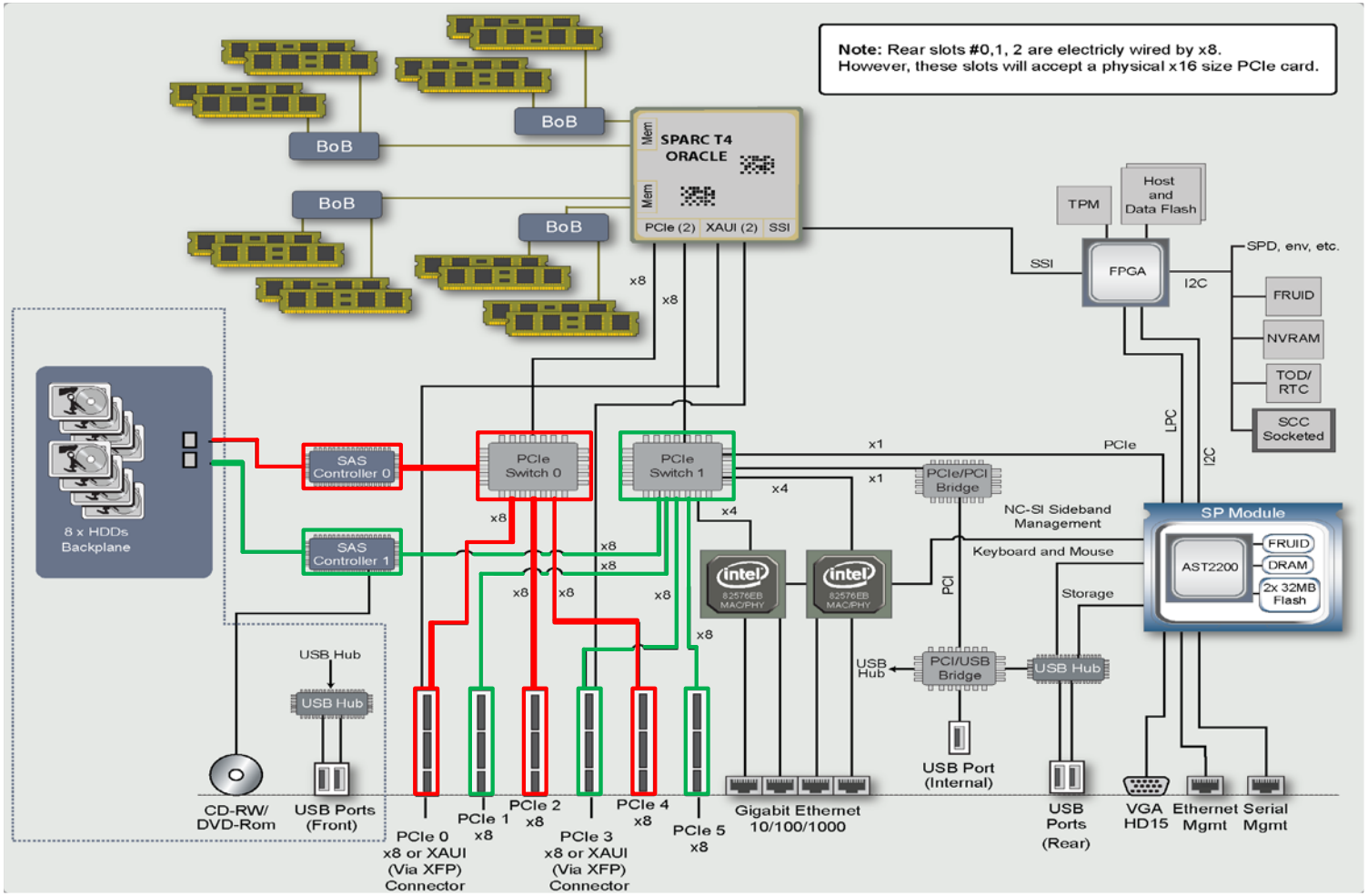


Live Migration

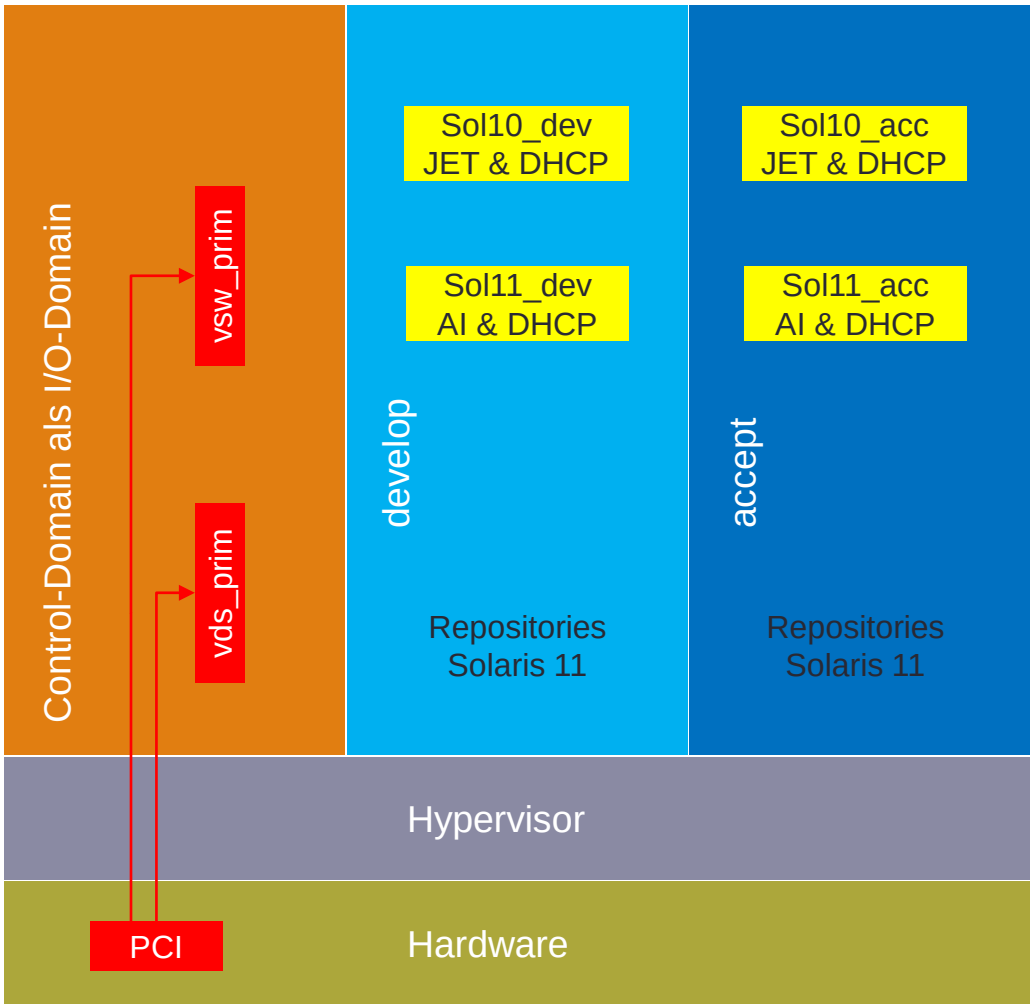
- Voraussetzungen sind virtueller I/O, gesharte SAN-LUN‘s und identische Namen für „vds“ und „vdisk“



Schaltplan T4-1

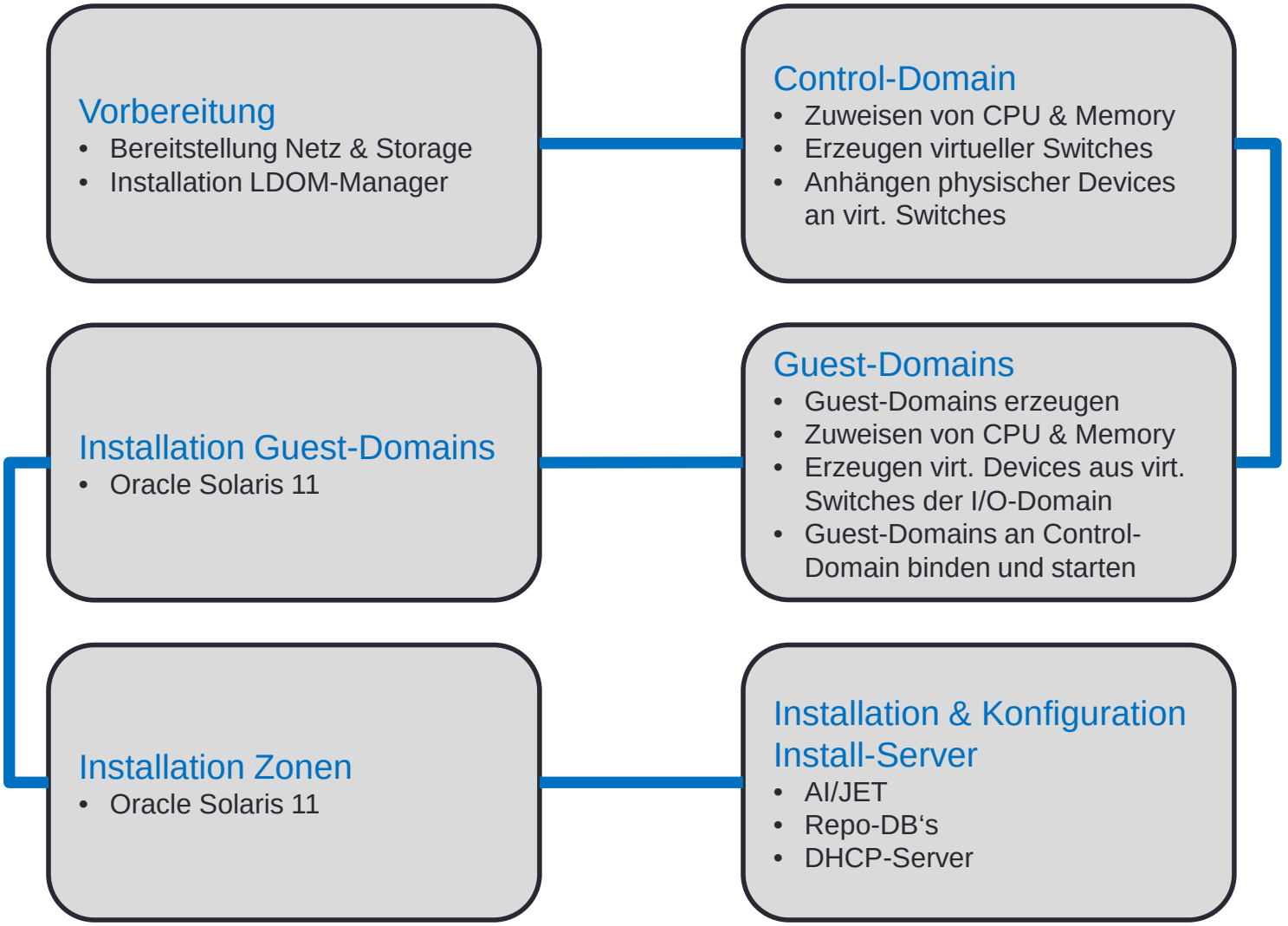


Testaufbau detailliert



- System soll als Entwicklungs- und Abnahme-Server für Solaris-Installationen dienen
- eine Service-Domain als Control- und I/O-Domain
- zwei Guest-Domains (develop & accept)
- jeweils zwei Zonen (eine für Solaris 10-Clients, eine für Oracle Solaris 11-Clients)
- Repositories liegen in den globalen Zonen, um die lokalen Zonen zu installieren
- in allen Domains und Zonen wird Solaris 11 installiert

Konfigurationsablauf



LDOM-Manager

LDOM-Manager ist zentrales Konfigurations-Tool und somit Grundvoraussetzung

```
root@esther:~# pkg list ldomsmanager
NAME (PUBLISHER)                                  VERSION                    IFO
system/ldoms/ldomsmanager (solaris)               3.0.0.4-0.175.1.9.0.4.0    i--
```

```
root@esther:~# pkg info system/ldoms/ldomsmanager
          Name: system/ldoms/ldomsmanager
       Summary: Logical Domains Manager
   Description: LDoms Manager - Virtualization for SPARC T-Series
      Category: System/Virtualization
         State: Installed
     Publisher: solaris
       Version: 3.0.0.4
 Build Release: 5.11
        Branch: 0.175.1.9.0.4.0
Packaging Date: June 28, 2013 09:40:36 PM
          Size: 4.06 MB
          FMRI: pkg://solaris/system/ldoms/ldomsmanager@3.0.0.4,5.11-
0.175.1.9.0.4.0:20130628T214036Z
```

Nach Installation wird automatisch Control-Domain erzeugt - immer „primary“, nicht änderbar

```
root@esther:~# ldm list
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  NORM  UPTIME
primary          active     -n-c--  UART    64    261632M  1.0%  1.0%   17d 1h 42m
```

Control-Domain konfigurieren

CPU‘s zuweisen

```
root@esther:~# ldm set-vcpu 4 primary
```

Speicher zuweisen

```
root@esther:~# ldm set-memory 8g primary
```

virtuelle Disk-Switch „vds-prim“ erzeugen

```
root@esther:~# ldm add-vds vds-prim primary
```

LUN‘s an virtuelle Disk-Switch binden (komplette Disk → Slice 2)

```
root@esther:~# ldm add-vdsdev /dev/dsk/c0t484954414348492037373031303836343030343430d0s2 \
c0t484954414348492037373031303836343030343430d0@vds-prim
root@esther:~# ldm add-vdsdev /dev/dsk/c0t484954414348492037373031303836343030343433d0s2 \
c0t484954414348492037373031303836343030343433d0@vds-prim
```

virtuelle Netzwerk-Switch „vsw-prim“ erzeugen und Netz-Interface an diese binden

```
root@esther:~# ldm add-vsw net-dev=net12 mac-addr=2:21:28:57:47:17 vsw-prim primary
```

virtuelle Console „vcc-prim“ erstellen und Port-Range festlegen

```
root@esther:~# ldm add-vcc port-range=5000-5100 vcc-prim primary
```

Konfiguration als „DOAG-2013“ abspeichern, anschl. System-Reboot

```
root@esther:~#  ldm add-config DOAG-2013; init 6
```

Prüfen der Konfiguration - 1/2

gesicherte Konfigurationen

```
root@esther:~# ldm list-config
factory-default
DOAG-2013 [current]
```

LDOM-Übersicht (Kurzausgabe)

```
root@esther:~# ldm list
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  NORM  UPTIME
primary          active     -n-cv-  UART    4     8G       0.9%  0.9%  10m
```

Bedeutungen der Spalte „FLAGS“:

-	Placeholder
c	Control domain
d	Delayed reconfiguration (Konfiguration wurde geändert)
e	Error
n	Normal
s	Column 1 - starting or stopping Column 6 - source domain
t	Column 2 - transition Column 6 - target domain
v	Virtual I/O service domain (Control- oder Service-Domain)

Prüfen der Konfiguration - 2/2

I/O-Konfiguration (vds, vsw und vcc)

```
root@esther:~# ldm list -o disk,network,console
NAME
primary

MAC
    00:10:e0:2a:d8:b6

VCC
    NAME             PORT-RANGE
    vcc-prim         5000-5100

VSW
    NAME             MAC               NET-DEV   ID   DEVICE     LINKPROP   DEFAULT-VLAN-ID PVID
    vsw-prim         02:21:28:57:47:17 net12     0    switch@0              1               1

VDS
    NAME             VOLUME         OPTIONS          MPGROUP        DEVICE
    vds-prim         c0t484954414348492037373031303836343030343430d0
/dev/dsk/c0t484954414348492037373031303836343030343430d0s2
                     c0t484954414348492037373031303836343030343433d0
/dev/dsk/c0t484954414348492037373031303836343030343433d0s2
```

Guest-Domain manuell erstellen

CPU‘s, Memory und I/O

```
root@esther:~# ldm add-domain develop
root@esther:~# ldm add-vcpu 16 develop
root@esther:~# ldm add-memory 64g develop
root@esther:~# ldm add-vdisk disk01_develop c0t4849544143484920373730313038363430303430d0@vds-prim develop
root@esther:~# ldm add-vnet vnet_develop vsw-prim develop
```

„auto-boot?=false“ für OPB der Guest-Domain setzen

```
root@esther:~# ldm set-var auto-boot\?=false boot-device=disk01_develop develop
```

Guest-Domain an die Service-Domain binden (hiermit werden die konfigurierten Ressourcen persistent gebunden)

```
root@esther:~# ldm bind develop
```

Guest-Domain starten

```
root@esther:~# ldm start develop
LDom develop started
```

LDOM-Übersicht

```
root@esther:~# ldm list
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  NORM  UPTIME
primary          active     -n-cv-  UART    4     8G       0.2%  0.2%  2d 21h 14m
develop          active     -t----  5000    16    64G      6.2%  6.2%  27s
```

Guest-Domain mit XML-File erstellen

XML-File aus einer vorhandenen LDOM erzeugen (hier develop)

```
root@esther:~# ldm list-constraints -x develop > /tmp/ovm/develop.xml
```

XML-File bearbeiten (develop → accept, Platten tauschen, uuid entfernen → wird automatisch neu erzeugt)

```
root@esther:~# cp /tmp/ovm/develop.xml /tmp/ovm/accept.xml
root@esther:~# gsed -i 's/develop/accept/' /tmp/ovm/accept.xml
root@esther:~# gsed -i \
's/c0t48495441434849203737303130383634303034 30d0/c0t48495441434849203737303130383634303034 33d0/' \
/tmp/ovm/accept.xml
root@esther:~# gsed -i '/uuid/d' /tmp/ovm/accept.xml
```

Guest-Domain „accept“ erzeugen

```
root@esther:~# ldm add-domain -i /tmp/ovm/accept.xml
```

Guest-Domain an die Service-Domain binden & starten

```
root@esther:~# ldm bind accept; ldm start accept
LDom accept started
```

LDOM-Übersicht

```
root@esther:~# ldm list
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  NORM  UPTIME
primary          active     -n-cv-  UART    4     8G       0.2%  0.2%  3d 1h 39m
accept           active     -t----  5001    16    64G      6.2%  6.2%  24s
develop          active     -t----  5000    16    64G      6.2%  6.2%  3h 14m
```

log in

„virtual network terminal server daemon“ (vntsd) in Control-/Service-Domain starten

```
root@esther:~# svcs -a |grep vntsd
disabled       Nov_01   svc:/ldoms/vntsd:default

root@esther:~# svcadm enable vntsd
```

Verbindung mit Console herstellen (hier develop)

```
root@esther:~# telnet localhost 5000
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "develop" in group "develop" ....
Press ~? for control options ..

~ ?
{0} ok ~?
Supported escape sequences:
~.  - terminate connection
~B  - send a BREAK to the remote system
~C  - open a command line
~R  - Request rekey (SSH protocol 2 only)
~^Z - suspend ssh
~#  - list forwarded connections
~&  - background ssh (when waiting for connections to terminate)
~?  - this message
~~  - send the escape character by typing it twice
```

MAC-Adresse für DHCP

MAC aus „Banner“ ist nicht die Richtige!

```
{0} ok banner

SPARC T4-1, No Keyboard
Copyright (c) 1998, 2012, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.34.2.a, 65536 MB memory available, Serial #83395546.
Ethernet address 0:14:4f:f8:83:da, Host ID: 84f883da.
```

Anzeige Gerätenamen-Aliases

```
{0} ok devalias
disk01_develop           /virtual-devices@100/channel-devices@200/disk@0
vnet_develop             /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
```

Eigenschaften Netzwerk-Device (tatsächliche MAC)

```
{0} ok cd vnet_develop
{0} ok .properties
local-mac-address        00 14 4f fb ee 3f
max-frame-size           00004000
address-bits             00000030
reg                      00000000
compatible               SUNW,sun4v-network
device_type              network
name                     network
```

Fertigstellung

- OS-Installationen der LDOM‘s
- Erstellen der Repo-DB‘s pro globaler Zone (Guest-Domain)
- Konfiguration und Installation der Zonen
  - Sol10_dev, Sol11_dev, Sol10_acc, Sol11_acc
- Installation Automated Installer (AI) pro Sol11-Zone
  - Sol11_dev & Sol11_acc
- Installation & Konfiguration JET in den Sol10-Zonen
  - Sol10_dev & Sol10_acc
- Installation und Konfiguration der DHCP-Server
  - Sol10_dev, Sol11_dev, Sol10_acc, Sol11_acc

Bedingungen Live Migration

- identischer Firmeware-Stand auf beiden Systemen
- LDOM-Manager muss in identischer Version installiert sein
- LUN‘s müssen von beiden Systemen erreichbar sein (shared storage)
- beide Systeme müssen über Netzwerk kommunizieren könen
- I/O muss auf beiden Systemen virtualisiert sein (Service-Domains)
- vsw, vcc und vds müssen auf beiden Systemen identischen Bezeichner haben

Test Live Migration

Migrations-Test im dry-run

```
root@esther:~# ldm migrate-domain -n develop magnus
Target Password:
```

Guest-Domain develop von Server „esther“ auf Server „magnus“ migrieren

```
root@esther:~# ldm migrate-domain develop magnus
Target Password:
```

Migrations-Übersicht auf „magnus“

```
root@magnus:~# ldm list -o status develop
NAME
develop

STATUS
OPERATION    PROGRESS        SOURCE
migration    23%             esther
```

LDOM-Übersicht auf „magnus“

```
oot@magnus:~# ldm list
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  NORM  UPTIME
primary          active     -n-cv-  UART    4     8G       0.2%  0.2%  3d 1h 39m
develop          active     -t----  5000    16    64G      6.2%  6.2%  1d 9h 14m
```

Fazit

- günstige Variante für Entwicklungs- und Abnahme-Umgebungen
- Ressourcen für diese Zwecke völlig ausreichend
- für produktive Systeme eher Hardware mit mehreren Bussen
  - dann Direct I/O
    - Verbesserung des Durchsatzes, da der zusätzliche Layer der virtualisierten Services entfällt
  - oder zusätzliche Service-Domain(s)
    - Nutzung von IPMP und MPxIO zur Erhöhung der Verfügbarkeit & Ausfallsicherheit
    - Live Migration

Mobility Networks Logistics

Fragen?