Der Artikel zeigt anhand eines Beispiels aus der Praxis, wie das Netzwerk in einer virtualisierten Solaris-Umgebung aus Oracle VM Server SPARC in Kombination mit Solaris-Zonen konfiguriert wird. Dabei kommen die Technologien Solaris, LDom, VLAN, Link Aggregation 802.3ad / Trunking, Cisco Trunking ISL/802.1Q, Solaris IPMP und exclusive IP von Solaris-Zonen zum Einsatz.

Eine schwere Netzwerk-Aufgabe mit der Solaris-Virtualisierungslösung Oracle VM Server SPARC

Roman Gächter, Trivadis AG

Vom Namen her kann man die Virtualisierungs-Lösung „Oracle VM Server SPARC“ leicht mit dem Produkt „Oracle VM Server X86“ verwechseln. Es handelt sich dabei um zwei verschiedene Technologien, das letzte Wort macht den Unterschied aus. „Oracle VM Server X86“ ist das Produkt für die Intel-X86-Hardware und baut auf der XEN-Technologie auf. In diesem Artikel wird VM Server SPARC thematisiert, das auch unter dem Namen „Logical Domains“ (LDoms) bekannt und nur auf SPARC-Hardware verfügbar ist.

Ausgangslage

Oft ist man vor vollendete Tatsachen gestellt und muss aus den bestehenden Möglichkeiten das Beste herausholen. Im vorliegenden Fall war die Hardware bereits gekauft und konfiguriert. Im Rahmen eines In-Sourcing-Projekts wurde eine Banken-Applikation auf neuer SPARC-Hardware in den firmeneigenen Rechenzentren aufgesetzt. Um die Hardware-Kosten niedrig zu halten, entschied man sich für Oracle VM Server SPARC. Es war notwendig, mehrere Umgebungen (Produktion, Entwicklung und Abnahme) aufzubauen. Zudem musste die Hardware redundant über zwei Rechenzentren verteilt bereitgestellt werden. So wurden zwei SPARC-T4-2-Boxen gekauft und verteilt auf zwei autonome Rechenzentren installiert.

Pro Umgebung wurde jeweils eine „Guest LDom“ aufgesetzt. Die einzelnen Systeme der Applikation wurden als Solaris-Zonen in der „Guest LDom“ konfiguriert. Die gesamte Installation der Zonen liegt auf dem SAN. Im Desaster-Fall können die Zonen in das andere Rechenzentrum verschoben werden (siehe Abbildung 1).

Knacknuss Netzwerk

Es standen insgesamt acht 1-GB-Ethernet-Ports zur Verfügung. Die Frage war nun: „Wie lege ich das Netzwerk aus, damit eine Primary Domain, drei Guest Domains und fünfzehn Zonen verteilt auf drei Umgebungen redundant und mit optimalem Durchsatz angeschlossen werden können? Es musste ein Netzwerkkonzept ausgearbeitet werden, das die folgenden Vorgaben erfüllt:

- Verwendung der bestehenden T4-2-Hardware, die mit zwei „Quad

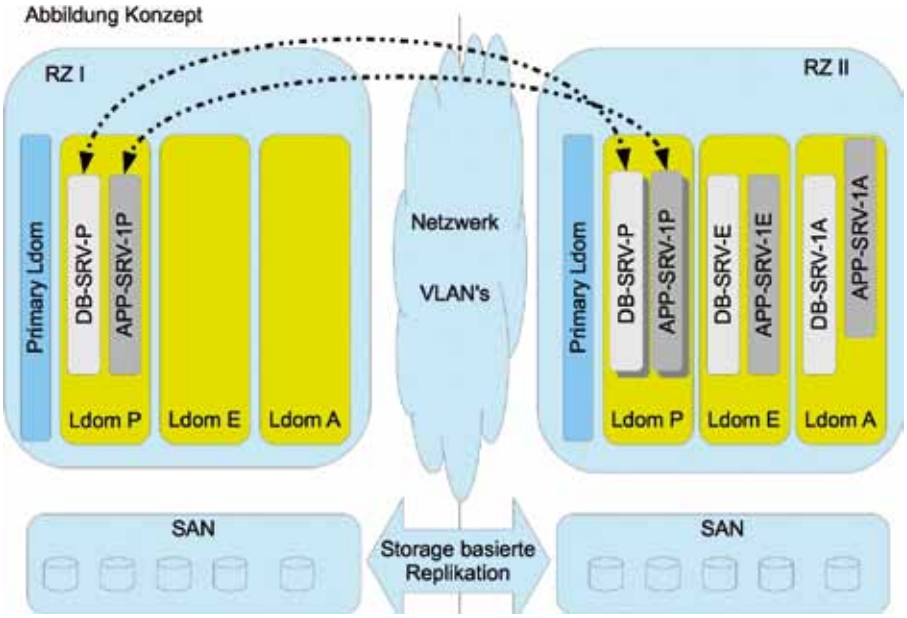

Abbildung 1: Das Konzept

Port“ Network Interface Controllern (NICs), also acht 1-GBPorts, bestückt ist.
- Neben der Primary Domain müssen noch drei weitere Domains betrieben werden.
- Pro Umgebung sind fünf Solaris-Zonen installiert. Die einzelnen Zonen innerhalb der Umgebungen sind durch Firewalls zu separieren, da für mehrere Mandanten ausgelegt
- Die SPARC-Hardware ist direkt an Cisco Core Switches „Trunking Ports“ angeschlossen. Die virtuellen Switches und Vnets der LDoms müssen mit VLANs konfiguriert werden.
- Die Netzwerk-Performance ist für die produktive Umgebung kritisch.
- Das Netzwerk-Konzept muss auf Ausfallsicherheit ausgelegt sein.

Die beiden von Solaris unterstützten Technologien, um Netzwerk-Interfaces zu bündeln, sind „IP Multipath“ (IPMP) und „Link Aggregation“. Beide Technologien bieten Features an, die sich zum Teil überschneiden, jedoch auf unterschiedlichen Netzwerk-Schichten des OSI-Modells implementiert sind: „Link Aggregation“ in der MAC-Schicht, IPMP in der IP-Schicht. Es war schnell klar, dass in dieser virtualisierten Umgebung eine Gruppierung von NICs notwendig war. „Link Aggregation 802.3ad“ bietet folgende Vorteile:

- Implementiert auf dem MAC-Layer
- Erhöhte Bandbreite durch Bündelung mehrere NICs
- Automatisches „Failover/Failback“ von Links des Aggregats
- Load Balancing, Verteilung des „Inbound und Outbound Traffic“ gemäß der gewählten Policy
- Redundanz ist möglich

Das Zusammenspiel dieser Technologie mit Cisco Trunking wurde anhand eines Proof of Concept (POC) überprüft. Bei der Variante 1 (siehe Abbildung 2) hat man ein Aggregat über je einen Port des e1000g- und igb-NIC gebildet und an zwei Switches angeschlossen. In LDom wurden ein virtueller Switch und zwei virtuelle Netzwerke konfiguriert und diese den Solaris-Zonen exclusiv zur Verfügung gestellt. Die Zonen mussten zwingend in unterschiedlichen VLANs betrieben werden (Trennung der Systeme beziehungsweise der logischen Netzwerke).

Die Idee dieser Konfiguration war: Es gibt eine redundante Netzwerk-Infrastruktur. Ob nun ein Switch, ein NIC oder ein einzelner NIC Port ausfällt – das Netz bleibt mit reduzierter Bandbreite verfügbar.

Leider hat diese Konfiguration in einem Fehlerfall (Ausfall eines Core Switch) nicht funktioniert. Durch den simulierten Ausfall eines der Switches hat jeweils eine der Zonen die Netzverbindung verloren. Erst nach dem Gratuitous-ARP-Paket der Zone, das bei Solaris in einem Intervall von fünf Minuten gesendet wird, wurde der Link-Status „down des Vnets“ erkannt.

Das Gratuitous-ARP-Package hat unter anderem den Effekt, ARP Caches im Netzwerk zu aktualisieren. Die Protokolle „Link Aggregation 802.3ad“ und Cisco Trunking sind in dieser Konfiguration nicht kompatibel. Die notwendigen VLAN- und Link-Informationen wurden zwischen den verschiedenen Protokollen nicht korrekt ausgetauscht. Da die notwendigen Konfigurationsänderungen im Netzwerk-Bereich nicht vorgenommen werden konnten, musste man eine andere Lösung suchen.

Bei der Variante 2 (siehe Abbildung 3) kamen zwei Aggregate zum Einsatz. Diese sind nicht mehr Switch-übergreifend. In LDom wurden nun zwei virtuelle Switches und vier virtuelle Netzwerke konfiguriert. Das neue Element ist hier IP Multipath (IPMP) von Solaris. Die Vnets der Zonen sind auf zwei Aggregate beziehungsweise zwei Cisco-Switches verteilt. Die Idee dieser Konfiguration auch hier ist: Wir haben eine redundante Netzwerk-Infrastruktur. Ob nun ein Switch, ein NIC oder ein einzelner NIC Port ausfällt – das Netz bleibt mit reduzierter Bandbreite verfügbar.

Mit dieser Konfiguration wurde dank IPMP eine funktionierende Redundanz erreicht. Auch der Ausfall eines Switch wird korrekt erkannt und IPMP verwendet nur noch das Vnet mit Link-Status „up“. Wichtig ist hier, dass in der LDom-Konfiguration für die Vnets das Property „link_state“ auf „physical“ gesetzt ist, sonst klappt es nicht. Abbildung 4 zeigt schematisch die Übersicht des Netzwerks mit allen Aggregaten und LDom sowie symbolisch jeweils zwei Zonen pro LDom.

Die Aggregate „aggr 1“ und „aggr 3“ sind am ersten Core-Switch, „aggr 2“ und „aggr 4“ am zweiten angeschlossen. Die Vnets sind über IPMP immer so aufgesetzt, dass sie sich über zwei verschiedene Aggregate und Core-Switches erstrecken. Die produktive Umgebung hat die Bandbreite von 4 x 1 Gbps zur Verfügung. Die Umgebungen „Entwicklung“ und „Abnahme“ teilen

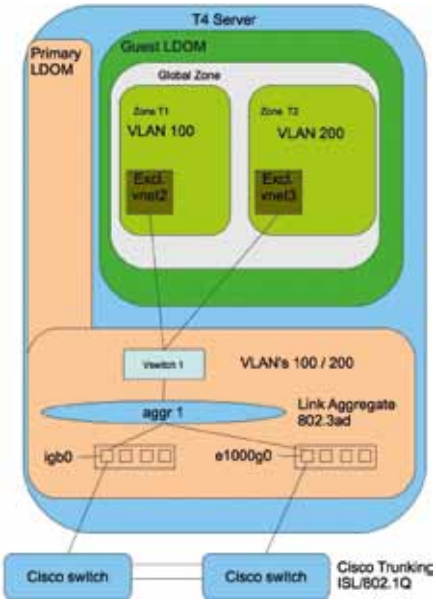

Abbildung 2: POC Variante 1

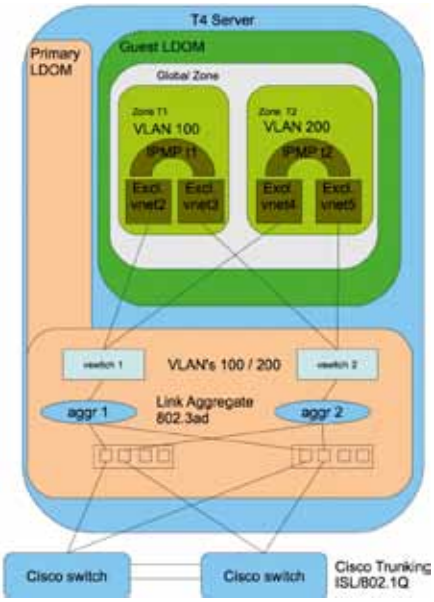

Abbildung 3: POC Variante 2

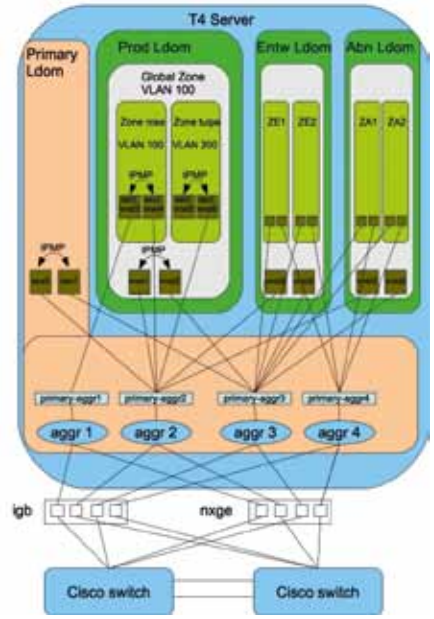

Abbildung 4: Übersicht über das Netzwerk

sich den Rest. In dieser Konfiguration müssen die virtuellen Switches im Hypervisor in der Lage sein, mehrere VLANs zu managen.

Konfigurations-Beispiele

Die folgenden Beispiele zeigen, wie die oben beschriebene Konfiguration erstellt werden kann. Sie gelten für Oracle VM Server SPARC 3.0 und Solaris 10:

- *Konfiguration Link Aggregation 802.3ad*
  Es wurden verschiedene „load balancing policies" getestet und sich am Schluss für „L2" entschieden. Damit wird das „outbound device" gemäß der MAC-Adressen in den Paketen selektiert. Das „link aggregation control protocol" (LACP) wurde nicht konfiguriert („off mode"), was dem Solaris-Default entspricht. Listing 1 zeigt, wie auf der „Primary Domain" ein solches Aggregat mit dem „dladm"-Command erzeugt werden kann.
- *Netzwerk-Konfiguration im Hypervisor*
  Das Beispiel zeigt wie zwei virtuelle Switches auf den vorher erstellten Aggregaten, um die VLANs 100, 200, 300 und 400 zu managen. Mit der „vid"-Property wird bestimmt, dass der Switch das VLAN-Tagging für die entsprechenden VLANs durchführen soll (siehe Listing 2). Es ist von Vorteil, die Mac-Adressen selber eindeutig zu vergeben. Der „Logical Domain Manager" ist zwar in der Lage, diese automatisch zu vergeben, erkennt jedoch nicht die Adressen weiterer LDoms auf anderer Hardware. Anschließend erstellt man für die eben erzeugten Switches Vnets, die dann für die IPMP-Konfiguration verwendet werden können. Die „pvid"-Property bestimmt, zu welchem VLAN das Vnet gehören soll. Wichtig ist das Property „linkprop=phys-state". Es wird gebraucht, um den Link-Status der physischen Netzwerk-Devices an die virtuellen durchzureichen. IPMP kann nur dann funktionieren, wenn ein physischer Link-Fehler an die virtuellen Devices weitergegeben wird (siehe Listing 3).
- *Link based IPMP*
  Dieses Beispiel zeigt, wie IPMP mit „link-based failure detection" und „active active mode" für die Interfaces „vnet3" und „vnet4" konfiguriert werden kann. Der verwendete Hostname ist „tulpe", der Name der IPMP-Gruppe „pz1". Für das erstes NIC-File „/etc/hostname. vnet3" nimmt man „tulpe netmask + broadcast + group pz1 up" und für das zweite NIC-File „/etc/hostname. vnet4" heißt es „group pz1 up".

```
dladm create-aggr -d nxge0 -d igb0 1
dladm modify-aggr -P L2 1
```

Listing 1

```
ldm add-vsw mac-addr=00:14:4f:fc:00:00 vid=100,200,300,400 net-
dev=aggr1 primary-aggr1 primary
ldm add-vsw mac-addr=00:14:4f:fc:00:01 vid=100,200,300,400 net-
dev=aggr2 primary-aggr2 primary
```

Listing 2

```
ldm add-vnet mac-addr=00:14:4f:fc:00:03 linkprop=phys-state pvid=100 excl_rose_1 primary-aggr1 prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:04 linkprop=phys-state pvid=100 excl_rose_2 primary-aggr2  prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:05 linkprop=phys-state pvid=200 excl_tulpe_1 primary-aggr1 prodldom
ldm add-vnet mac-addr=00:14:4f:fc:00:06 linkprop=phys-state pvid=200 excl_tulpe_2 primary-aggr2 prodldom
```

Listing 3

Fazit

In einer virtuellen Umgebung kommt man kaum darum herum, Netzwerk-NICs zu bündeln. In diesem Beispiel ist aufgezeigt, wie die Technologien „Link Aggregation“ und IPMP kombiniert werden können. In Zusammenarbeit mit den Kollegen vom Netzwerk-Team wurde mit der oben beschriebenen Konfiguration eine gute Lösung gefunden. Die Systeme sind redundant und mit optimaler Performance am Netzwerk angeschlossen. Auch im Fall von Wartungen im Netzwerk-Bereich können die Systeme ohne Unterbrechung weiterbetrieben werden. Weil „Load Balancing“ implementiert ist, zeigten die Messungen eine optimale Verteilung der Netzwerk-Last über die Netzwerk-Ports. Auch der Durchsatz entsprach den Erwartungen.

Es lohnt sich, im Vorfeld genug Zeit zu investieren und ein gutes Netzwerk-Design auszuarbeiten. Wichtig sind auch ausführliche Tests der möglichen Varianten.

Literatur und Links

- https://blogs.oracle.com/droux/entry/link_aggregation_vs_ip_multipathing
- http://docs.oracle.com/cd/E19253-01/816-4554/
- http://docs.oracle.com/cd/E37707_01/html/E29665/preface.html
- http://www.ieee802.org/3/hssg/public/apr07/frazier_01_0407.pdf

1. http://standards.ieee.org/findstds/standard/802.1Q-2011.html

Roman Gächter
Roman.Gaechter@trivadis.com

Ein knappes Jahr ist seit der Veröffentlichung von Oracle VM 3.1 für x86 vergangen. Dies soll Anlass für ein Review und den Überblick über wesentliche Features der aktuellen Version sein.

OVM 3 (x86) – was sich getan hat

Dirk Läderach, Robotron Datenbank-Software GmbH

Während die ersten Versionen (3.0.1 bis 3.1) bei vielen Anwendern nach Tests oder Upgrades für Ablehnung beziehungsweise Schmunzeln bis Verärgerung sorgte, kann mittlerweile von vielen Seiten eine stete, schrittweise positive Resonanz im Umgang mit der Virtualisierung-Lösung beobachtet werden. Im letzten Jahr hat die DOAG eine Liste der aus Sicht ihrer Mitglieder aktuellen Probleme veröffentlicht und auch Oracle diesbezüglich um Stellung gebeten. Viele dieser Themen sorgten auch bei unseren Kunden für eine schleppende Akzeptanz der Lösung. In der Zwischenzeit ist die Version 3.2.2.520 verfügbar und viele der Mängel sind behoben oder es existiert zumindest ein zufriedenstellender Workaround.

Ein paar positive Beispiele

Upgrade-Probleme gehören seit der Version 3.1.1. nahezu der Vergangenheit an und mittlerweile gibt es auch ein funktionierendes Rollback. Bei fünfzehn vom Autor selbst durchgeführten Migrationen auf 3.2.1 und 3.2.2 hat eine einzige nicht funktioniert; diese wurde sauber wieder zurückgerollt und es erfolgte ein „redeploy“ der Anwendung.

Einbinden von ISO-Dateien funktioniert zwar weiterhin nur über den Umweg des OVM-Hosts, aber die Import-Limitierung (http, ftp) ist inzwischen einem funktionierenden Repository-Import gewichen, der über eine Aktualisierung des Repository gestartet wird. Diese Funktionalität ist übrigens auch in der Lage, per „scp“ in das Repository kopierte Objekte wie virtuelle Disks, Templates, Assemblies bis hin zu den Konfigurationsdateien der VMs zu importieren (wenn die Syntax stimmt) und über die Oberfläche bereitzustellen.

Das oft genutzte Feature der Hard-Partitionierung mittels CPU-Pinning erforderte anfangs noch manuelles Editieren der VM-Konfigurationsdatei („vm.cfg“). Ebenfalls genutzt werden konnten die OVM-Utils. Im aktuellen Release besteht nun mit den CPU-Pools eine sehr gelungene, automatisierte Variante, das CPU-Pinning für die jeweilige Lizenzierung und/oder die Performance-Ansprüche umzusetzen. Für die Oracle Database Appliance ist diese Methode Pflicht.

Auch die Integration in den Enterprise Manager Cloud Control, der Voraussetzung für eine rollenbasierte Nutzerverwaltung ist, und das OPS-Center

```
[root@ovm01 /]# dmidecode|grep UUID
UUID: 20BDCCAA-D378-4C3E-B968-74AB09200A4E

/etc/ovs-agent/agent.ini:
…
[server]
fakeuuid=20BDCCAA-D378-4C3E-B968-74AB09200A4E
…
```

Listing 1