LDoms Deep Dive – IO Best Practices für Oracle VM Server für SPARC

Stefan Hinker
Oracle Deutschland B.V. & Co. KG
Düsseldorf

Schlüsselworte
Virtualisierung, Netzwerk, IO, Oracle VM Server for SPARC

Einleitung

Virtualisierung von IO hat viele Vorteile. Ressourcen werden besser ausgenutzt, die Verwaltung wird flexibler und nicht zuletzt erlaubt die Live Migration eine vollständige Entkopplung eines Systems von der physischen Hardware. Der Nachteil ist jedoch, dass Virtualisierung in der Regel zu höheren Latenzen und dadurch zu teilweise deutlichen Leistungseinbußen führt, unabhängig davon, welche Virtualisierungslösung betrachtet wird. Inzwischen ist dieses Problem erkannt und es gibt verschiedene Ansätze, wie bessere IO-Leistung erzielt werden kann. In diesem Beitrag werden erprobte Methoden zur Leistungsoptimierung des virtuellen IO mit Oracle VM Server for SPARC dargestellt. Des weiteren wird beschrieben, wie durch Verwendung von redundantem virtuellem IO eine hohe Verfügbarkeit der Gastsysteme bei gleichzeitig einfacher Wartbarkeit der Infrastruktur sichergestellt werden kann.

Grundlage: Virtuelles IO bei Oracle VM Server for SPARC

Wie in Abbildung 1 dargestellt, wird virtuelles IO bei Oracle VM Server for SPARC (auch LDoms) als ein Dienst der Service-Domain zur Verfügung gestellt. Diese ist in Besitz der physischen IO Geräte und betreibt hierfür Virtual Disk Servcies (vds) und Virtual Switches (vswitch, vsw). Die virtuellen Geräte der Gastsysteme werden mittels Logical Domain Channels hieran angeschlossen. Die LDCs sind dabei schnelle punkt-zu-punkt Verbindungen, die die Hypervisor-Hardware zur Verfügung stellt. Trotz einer sehr effizienten Implementierung ist offensichtlich, daß der zusätzliche Weg durch den Hypervisor und die Service-Domain zu zusätzlicher Latenz im IO führen muss.

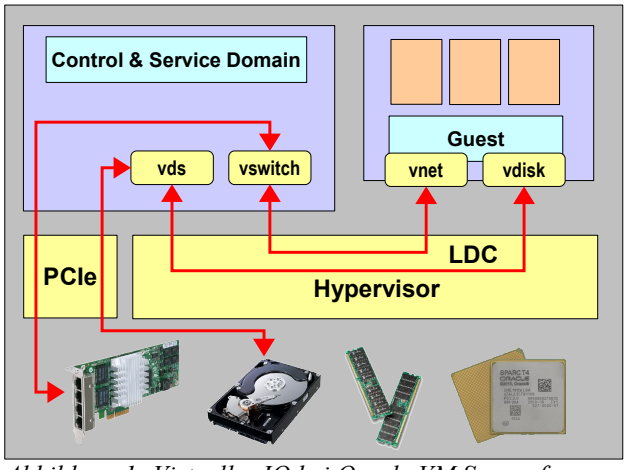

Abbildung 1: Virtuelles IO bei Oracle VM Server for SPARC

Disk-IO Virtualisierung mit hoher Leistung

Methoden zur Leistungsoptimierung bei physischem Platten-IO sind lange bekannt. Insbesondere die Verwendung von mehreren LUNs mittels RAID-0 (oder Raid 1-0) ist eine bewährte Methode, um mehr als die Leistung einer einzelnen Platte zu erzielen. Während hierbei die Latenz der einzelnen Operationen gleich bleibt, kann durch parallelen Zugriff auf alle Platten der Gesamtdurchsatz vervielfacht werden – oft linear mit der Anzahl der Platten. Dies hat nicht nur damit zu tun, dass mehrere physische Festplatten gleichzeitig verwendet werden, sondern auch damit, dass die Übertragungswege parallelisiert werden, so dass ein Queueing in mehreren Stufen stattfinden kann. Diesen Effekt kann man bei der Konfiguration von virtuellem Plattenspeicher bei LDoms ebenfalls ausnutzen. Indem man statt nur einer großen virtuellen LUN bspw. 4 oder 8 entsprechend kleinere LUNs an den Gast weiter reicht, wird das IO nun 4-fach bzw. 8-fach parallel abgewickelt. D.h. man erhält statt einer SCSI-Queue in der Service-Domain 8, statt einem LDC 8, und nicht zuletzt erneut statt einer SCSI-Queue im Gastsystem ebenfalls 8. Auf diese Weise kann der erreichbare Durchsatz erheblich gesteigert werden. Bei internen Tests wurden so Werte von über 1GB/s erreicht.

Ebenfalls analog zur physischen Welt ist ggf. eine Trennung von latenz-sensitivem Verkehr vom restlichen IO empfehlenswert. Ein gutes Beispiel hierfür sind die Redo-Logs der Oracle Datenbank. Hier empfiehlt sich die Verwendung von mindestens einer separaten LUN, was gleichzeitig einen dedizierten LDC und eine eigen SCSI-Queue zur Folge hat. So wird der hauptsächlich sequentiell schreibende Verkehr des Logwriters nicht durch anderen lesenden oder schreibenden Verkehr gestört.

Virtuelle Netzwerke

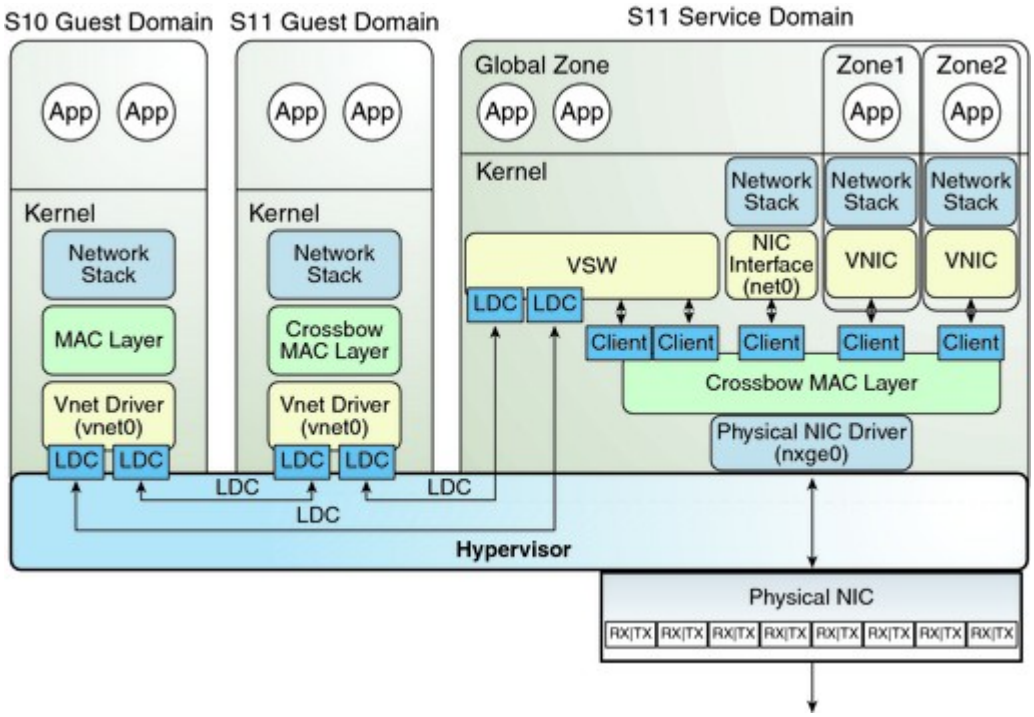

Abbildung 2: Netzwerk Stack mit LDoms und Solaris 11 (Abb. aus dem LDom Admin Guide)

Virtuelle Netzwerke in LDoms werden grundsätzlich mit Hilfe von Virtual Switches realisiert. Der virtuelle Switch funktioniert dabei genau wie ein Ethernet-Switch in der wirklich Welt und vermittelt Pakete auf Ethernet-Ebene vom Sender zum Empfänger. Die Verbindung zur Außenwelt wird dabei über ein an den vSwitch angeschlossenes physisches Interface hergestellt, die Verbindung zu LDom

Gästen über LDCs. Sind mehrere Gäste an einen Switch angeschlossen, werden auch die einzelnen möglichen Gast zu Gast Verbindungen jeweils durch einen LDC realisiert, um eine höchstmögliche Leistung zu erreichen.

Reduzierte Latenz bei virtuellem Netzwerk

Anders als bei Platten-IO ist es nicht möglich, höhere Durchsätze durch das Bündeln mehrerer LDCs für ein einzelnes Netzwerk-Interface zu erreichen. Die Latenz läßt sich jedoch spürbar reduzieren. Seit der Version 2.1 gibt es die Option „`extended-mapin-space`“, die eine Eigenschaft des jeweiligen Gastes ist. Wird sie sowohl für die Service-Domain als auch für einen Gast auf „`on`“ gestellt, verwendet die Service-Domain einen hierfür reservierten Speicherbereich des Hypervisors um die Kommunikation zu beschleunigen. Pro Gast benötigt der Hypervisor 4MB zusätzliches RAM, bei vielen Gästen sollte man daher ggf. prüfen, ob wirklich alle Gäste die so reduzierte Latenz benötigen.

Netzwerk-Konfiguration für sehr viele Gäste

Die Anzahl der LDCs, mittels derer die virtuellen Netzwerke innerhalb des Hypervisors aufgespannt werden ist beschränkt, die genaue Anzahl hängt vom jeweiligen System ab. Sind besonders viele Gäste oder besonders komplexe Netzwerke zu konfigurieren, kann es durch die vollständige Vermaschung aller Netzwerk-Interfaces zu einem Engpass bei der Anzahl der LDCs kommen. In solchen Fällen gibt es die Möglichkeit, bei einzelnen vSwitches auf diese Vollvermaschung zu verzichten. Wird bei einem vSwitch die Option „`inter-vnet-link`“ auf „`off`“ gesetzt, entfallen alle Gast-zu-Gast LDC-Verbindungen, es wird pro Gast dann nur ein LDC zur Verbindung an diesen vSwitch benötigt. Auf diese Weise werden eine große Zahl von LDCs für andere Zwecke frei. Die Gäste können selbstverständlich weiter miteinander kommunizieren, allerdings mit etwas höherer Latenz.

SR-IOV: Virtuelle Hardware

Neben der „klassischen“ Virtualisierung von IO ist es seit einiger Zeit auch möglich, Virtualisierung direkt als Funktion der Hardware zu nutzen. Die PCI SIG „SR-IOV“ (Single Root IO Virtualization) hat mit dem gleichnamigen Standard die Grundlage dafür geschaffen. Hierbei bietet die Netzwerk-Karte selbst die Möglichkeit, eine gewissen Anzahl von sogenannten Virtual Functions (VF) zu konfigurieren. Diese Virtual Functions bieten die gleichen Möglichkeiten wie die physische Karte und finden sich ähnlich wie die Physical Function – die klassische Karte – im Devicebaum des Systems. Von dort können Sie dann, genau wie bspw. ein einzelner PCIe Slot oder ein ganzer Root Complex, einem Gast zugewiesen werden. Dieser kann die Virtual Function dann nutzen wie eine vollwertige Netzwerk-Karte, ohne den sonst mit virtuellem IO verbunden Overhead. Derzeit gibt es Karten mit SR-IOV nur für Ethernet, nicht aber bspw. für Fibre Channel oder Infiniband. Während technisch gesehen nichts dagegen spricht, SR-IOV auch für HBAs zu nutzen, ist es derzeit ausschließlich für Netzwerk-Karten verfügbar. Sollte es in Zukunft auch HBAs geben, die SR-IOV unterstützen, ist eine entsprechende Erweiterung des LDom Subsystems denkbar.

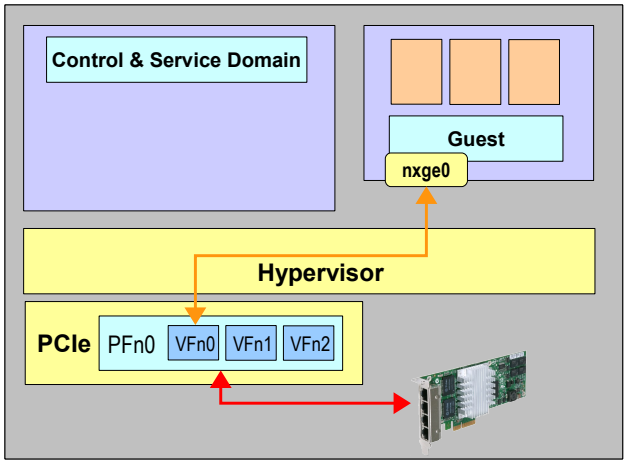

Abbildung 3: Zuweisung von SR-IOV Virtual Functions

Der Vorteil dieser Art der Virtualisierung ist offensichtlich. Nachteilig ist neben der begrenzten Anzahl der Virtual Functions pro Karte vor allem die Einschränkung in Bezug auf Live Migration. Da die Virtual Functions eine Funktion der Hardware sind, ist die Live Migration eine Gastes mit solchen Devices nicht möglich, selbst wenn auf dem Zielsystem eine äquivalente Konfiguration vorhanden wäre. Die Schwierigkeit hierbei liegt u.A. in den diversen Registern und sonstigen Puffern, die die Karte selbst vorhält und die bei einer Migration nicht ohne weiteres auf das Zielsystem übertragbar sind.

Zu beachten ist bei der Verwendung von VFs die Verwaltung der zugehörigen MAC-Adressen. Die Netzwerk-Karte kann eine gewissen Anzahl von MAC-Adressen verwalten und den VF zuweisen. Diese können dann bspw. für virtuelle Netzwerk-Ports von Solaris Zonen genutzt werden, die innerhalb eines Gastes konfiguriert werden. Netzwerk-Verkehr von Zonen, die keine solche MAC-Adresse verwenden, wird jedoch nicht transportiert werden.

Redundantes Virtuelles IO

Bei physischen Systemen wird die Verfügbarkeit von IO in der Regel durch redundante Pfade erhöht. So schützt man sich mit MPxIO oder RAID 1 gegen den Ausfall eines HBAs, Kabels oder einer Platte, mit LACP oder IPMP gegen den Ausfall einer Netzwerk-Karte oder -Kabels. Diese Techniken können unverändert auch in der Service-Domain verwendet werden. Allerdings muß auch die Service-Domain selbst von Zeit zu Zeit gewartet werden, was häufig einen Reboot einschließt. Dieser bedeutet für die Gastsysteme einen vorübergehenden Stillstand im IO-System. Um diesen zu vermeiden, können redundante Service-Domains konfiguriert werden.

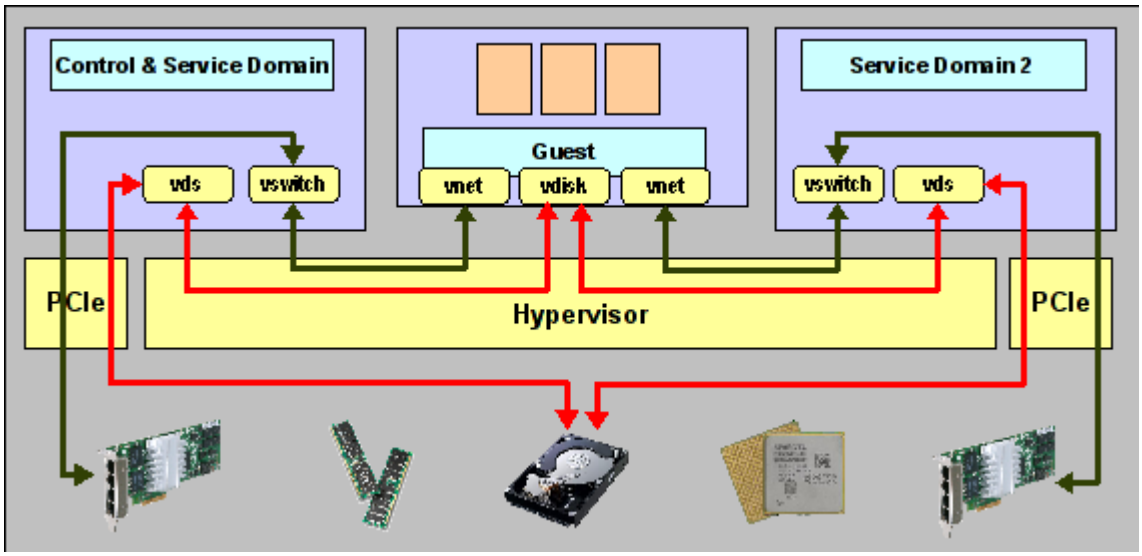

Abbildung 4: Redundante IO-Domains

Wie in der Abbildung dargestellt, werden dem Gastsystem hierbei einfach zwei virtuelle Netzwerk-Interfaces zugewiesen. Über diese kann der Gast dann mittels IPMP redundante Kommunikation konfigurieren. Der Linkstatus des physischen Netzwerks kann je nach Bedarf weitergereicht oder versteckt werden. Für virtuelle Platten wird, ähnlich wie bei MPxIO, ein multi-pathing Treiber im Gast verwendet. Die beiden Virtual Disk Services in den beiden Service-Domains verweisen hierbei beide auf den selben Backend Speicher. Fällt nun eine der beiden Service-Domains aus, werden die Datendienste von der verbleibenden Domain übernommen. Wichtig hierbei ist, dass die Service-Domains selbst nicht von der jeweils anderen Domain abhängen, also mit eigenen PCIe Ressourcen bootfähig sind.

Auf diese Weise ist eine Wartung der beiden Service-Domains möglich, ohne dass die Gäste hiervon betroffen wären. Auch gegen einen ungeplanten Ausfall einer Service-Domain ist man so geschützt. Mit Live Migration könnte die Wartung der Service-Domain ebenfalls von den Gästen entkoppelt werden. Allerdings ist Live Migration immer nur für geplante Wartungsarbeiten einsetzbar, nicht aber ein Schutz gegen ungeplante Ausfälle aller Art.

Kontaktadresse:
Stefan Hinker
Oracle Deutschland B.V. & Co. KG
Hamborner Str. 51
D-40472 Düsseldorf

	Stefan Hinker
Telefon:	+49 211 7483-9848
E-Mail	stefan.hinker@oracle.com
Internet:	https://blogs.oracle.com/cmt