

Solaris 10 Container Leitfaden

Funktionsweisen, Anwendungsfälle, Best Practices, Kochbücher

Ulrich Gräf, Ambassador Operating Systems
Detlef Drewanz, Ambassador Operating Systems

Sun Microsystems GmbH
Stand 07.11.2006

Inhaltsverzeichnis

1. Einführung	5
2. Allgemein	6
2.1. Solaris Zonen / Solaris Container	6
2.1.1. Überblick.....	6
2.1.2. Zonen und die Installation von Software.....	8
2.1.3. Zonen und Security.....	8
2.1.4. Zonen und Ressource Management.....	8
2.1.4.1. CPU Ressourcen.....	9
2.1.4.2. Memory Ressource Kontrolle.....	9
2.1.4.3. Netzwerk Ressourcen Kontrolle (IPQoS = IP Quality of Services).....	10
2.1.5. User Interfaces für Zonen.....	10
2.1.6. Zonen und Hochverfügbarkeit.....	10
2.2. Virtualisierungstechnologien im Vergleich	11
2.2.1. Stufe 0: Konsolidierung in einem Rechner.....	12
2.2.2. Stufe 1: Domains/physikalische Partitionen.....	13
2.2.3. Stufe 2: Logische Partitionen.....	14
2.2.4. Stufe 3: Container (Solaris Zonen) in einem OS.....	15
2.2.5. Zusammenfassung zu den Virtualisierungstechnologien.....	16
3. Anwendungsfälle	17
3.1. Grid Computing mit Isolation.....	17
3.2. Kleine Webserver.....	18
3.3. Multi-Netzwerk Konsolidierung.....	19
3.4. Multi-Netzwerk Monitoring.....	20
3.5. Multi-Netzwerk Backup.....	21
3.6. Konsolidierung Development/Test/Integration/Produktion.....	22
3.7. Konsolidierung von Test-Systemen.....	23
3.8. Schulungssysteme.....	24
3.9. Server Konsolidierung.....	25
3.10. Security Kapselung.....	26
3.11. Developer Testsysteme.....	27
3.12. Hosting für verschiedene Firmen auf einem Rechner.....	28
4. Best Practices	29
4.1. Konzepte	29
4.1.1. Arten der Softwareinstallation in Solaris.....	29
4.1.2. Softwareinstallation in einer Zone.....	29
4.1.3. Softwareinstallation in der globalen Zone für lokale Zonen.....	30
4.1.4. Sparse-root Zonen.....	31
4.1.5. Whole-root Zonen.....	31
4.1.6. Vergleich Sparse-root Zone und Whole-root Zone.....	32
4.1.7. Storage-Konzepte.....	32
4.1.7.1. Storage für das Root-Filesystem der lokalen Zonen.....	32
4.1.7.2. Storage für Daten.....	32
4.1.7.3. Storage für Programme/Applikationen.....	33
4.1.7.4. Rootplatten Layout.....	33
4.1.7.5. ZFS in einer Zone.....	33
4.1.8. Netzwerk-Konzepte.....	34
4.1.8.1. Vorbereitung Netzwerk für lokale Zonen.....	34
4.1.8.2. Verwaltung der Netzwerk-Adressen.....	34
4.1.8.3. IP-Stack und Routing zwischen Zonen.....	34
4.1.8.4. Zonen und Limitierungen im Netzwerk.....	35
4.1.9. Separate Name Services in Zonen.....	35
4.1.9.1. hosts-Datenbank.....	35
4.1.9.2. User-Datenbank (passwd, shadow, user_attr).....	35
4.1.9.3. Services.....	35
4.1.9.4. Projekte.....	35
4.2. Paradigmen	36
4.2.1. Delegation von Admin Rechten an die Applikations-Abteilung.....	36
4.2.2. Applikationen nur in lokalen Zonen.....	36
4.2.3. Eine Applikation pro Zone.....	37
4.2.4. Zonen im Cluster.....	37
4.3. Konfiguration / Administration	39
4.3.1. Manuelle Konfiguration von Zonen mit zonecfg.....	39
4.3.2. Manuelle Installation von Zonen mit zoneadm.....	39
4.3.3. Manuelle De-Installation von Zonen mit zoneadm.....	39

4.3.4.	Manuelles Entfernen einer installierten Zone mit zonecfg.....	39
4.3.5.	Duplizieren einer installierten Zone.....	39
4.3.6.	Standardisiertes Erzeugen von Zonen.....	40
4.3.7.	Automatische Konfiguration von Zonen per Script.....	40
4.3.8.	Automatisiertes Provisionieren der Services.....	40
4.4.	Lifecycle Management.....	41
4.4.1.	Patchen eines Systems mit lokalen Zonen.....	41
4.4.2.	Neuinstallation und Service Provisionieren statt Patchen.....	41
4.4.3.	Patch-Update durch Flash und Live Upgrade.....	41
4.4.4.	Backup und Recovery von Zonen.....	42
4.4.5.	Migration (Umzug) einer Zone mit zoneadm detach/attach.....	42
4.5.	Management und Monitoring.....	43
4.5.1.	Überwachung der Zonen-Auslastung.....	43
4.5.2.	Extended Accounting mit Zonen.....	43
4.5.3.	Auditing der Operationen in der Zone.....	43
4.5.4.	DTrace von Prozessen in einer Zone.....	43
5.	Kochbücher.....	44
5.1.	Installation und Konfiguration.....	44
5.1.1.	Konfigurationsdateien.....	44
5.1.2.	Relevante OS-Befehle.....	45
5.1.3.	Rootplattenlayout.....	47
5.1.4.	Konfiguration einer Sparse-root Zone: Erforderliche Aktionen.....	48
5.1.5.	Konfiguration einer Whole-root Zone: Erforderliche Aktionen.....	49
5.1.6.	Konfiguration einer Zone: Optionale Aktionen.....	50
5.1.7.	Storage in einer Zone.....	50
5.1.8.	ZFS in einer Zone.....	51
5.1.9.	Konfiguration einer Zone durch Kommando-Datei oder Template.....	51
5.1.10.	Installation einer Zone.....	51
5.1.11.	Uninstall einer Zone.....	51
5.1.12.	Automatische Konfiguration von Zonen durch sysidcfg.....	52
5.1.13.	Automatische Schnell-Installation von Zonen.....	52
5.1.14.	Hardening von Zonen.....	53
5.2.	Netzwerk.....	54
5.2.1.	Netzwerk und Routing.....	54
5.2.1.1.	Globale und lokale Zone mit gemeinsamem Netzwerk.....	54
5.2.1.2.	Zonen in getrennten Netzwerksegmenten.....	55
5.2.1.3.	Zonen in getrennten Netzwerken.....	56
5.2.1.4.	Zonen mit Verbindung in unabhängige Kunden-Netzwerke.....	57
5.2.1.5.	Verbindung von Zonen über externe Router.....	59
5.2.2.	IPQoS.....	60
5.3.	Lifecycle Management.....	61
5.3.1.	Boot einer Zone.....	61
5.3.2.	Softwareinstallation per mount.....	61
5.3.3.	Software Installation mit Provisioning System.....	62
5.3.4.	Migration einer Zone (mit OpenSolaris attach/detach).....	62
5.3.5.	Herunterfahren einer Zone.....	62
5.4.	Management und Monitoring.....	63
5.4.1.	Accounting über eine Zone.....	63
5.4.2.	Audit einer Zone.....	63
5.5.	Ressource Management.....	63
5.5.1.	Begrenzung von /tmp-Größe in einer Zone.....	63
5.5.2.	Ressource Pools mit Prozessor Sets.....	64
5.5.3.	Fair Share Scheduler.....	64
5.5.4.	Statisches CPU Ressource Management zwischen Zonen.....	64
5.5.5.	Dynamisches CPU Ressource Management zwischen Zonen.....	65
5.5.6.	Statisches CPU Ressource Management in einer Zone.....	65
5.5.7.	Dynamisches CPU Ressource Management in einer Zone.....	65
5.5.8.	Physikalischen Hauptspeicherverbrauch eines Projektes begrenzen.....	65
6.	Literatur.....	66

Disclaimer

Die Sun Microsystems GmbH übernimmt keine Gewähr für die Vollständigkeit und die Fehlerfreiheit der in diesem Dokument enthaltenen Informationen und Beispiele.

Versionierung

<i>Version</i>	<i>Inhalt</i>	<i>Wer</i>
1.3	07.11.2006 Zeichnungen 1 - 6 als Image	Detlef Drewanz
1.2	06.11.2006 Verallgemeinerung Virtualisierung Weitere Netzwerkbeispiele	Detlef Drewanz Ulrich Gräf
1.1	27.10.2006 Versionierungstabelle umgestellt (das Neueste nach oben) Literaturangabe ergänzt 5.1.14 Hardening von Zonen ergänzt	Detlef Drewanz
1.0	24.10.2006 Feedback eingearbeitet, Korrekturen Bilder: 2.2, Anwendungsfälle und Netzwerk 2.1.6, 4.2.4 hinzugefügt (Zonen im Cluster) Komplette Überarbeitung Kapitel 2.2, 2.1.3	Detlef Drewanz Thorsten Früauf/Detlef Uhlerr Ulrich Gräf
Draft 2.2	1. Netzbeispiel, Korrekturen 31. 07. 2006	Ulrich Gräf
Draft 2.0	2. Draft veröffentlicht – 28.07.2006	Ulrich Gräf, Detlef Drewanz
Draft 1.0	1. Draft (intern) - 28.06.2006	Ulrich Gräf, Detlef Drewanz

1. Einführung

[dd/ug] Mit der Verfügbarkeit von Solaris 10 am 31. Januar 2005 ist durch Sun Microsystems ein Betriebssystem mit bahnbrechenden Neuerungen fertiggestellt worden. Eine dieser Neuerungen sind die Solaris 10 Container, die unter anderem zur Konsolidierung und Virtualisierung von OS-Umgebungen, für die Isolation von Anwendungen und zum Ressource Management eingesetzt werden können.

Die Nutzung dieser neuen Möglichkeiten erfordert Know-How, Entscheidungshilfen und Beispiele, die wir in diesem Leitfaden zusammengefasst haben. Er richtet sich an Manager, RZ-Leiter, IT-Fachgruppen und IT-Administratoren.

Das Dokument gliedert sich in die Kapitel Einführung, Allgemeiner Teil, Anwendungsfälle, Best Practices, Kochbücher (erprobte Beispiele) und eine Literaturliste. Nach der kurzen Einführung folgt der allgemeine Teil mit der Darstellung heutiger RZ-typischer Anforderungen in Bezug auf Virtualisierung und Konsolidierung, sowie der Darstellung und dem Vergleich der Container Technologie. Im Anschluss daran werden in verschiedenen Anwendungsfällen die Einsatzmöglichkeiten für Container diskutiert. Deren konzeptionelle Umsetzung wird anhand von Best Practices dargestellt. Im Kapitel Kochbücher sind an konkreten Beispielen die Kommandos zur Umsetzung der Best Practices dargestellt. Alle Kochbücher wurden durch die Autoren selbst erprobt und verifiziert.

Das Dokument selber ist als Referenz-Dokument gedacht. Es ist zwar möglich, das Dokument von vorne nach hinten zu lesen, dies ist aber nicht zwingend notwendig. Ein RZ-Leiter wird vielleicht den Überblick über die Container-Technologie erwerben wollen oder sich die Anwendungsfälle ansehen. Ein IT Architekt sieht sich die Best Practices an, um Lösungen zu bauen. Währenddessen probiert ein System-Administrator die Kommandos in den Kochbüchern aus, um Praxis zu gewinnen. Daher bietet das Dokument für jeden etwas und darüber hinaus Referenzen um in den anderen Bereich hinüber zu schauen.

Vielen Dank an alle diejenigen, die durch Bemerkungen, Beispiele und Zusätze zu diesem Dokument beigetragen haben. Ein besondere Dank geht an die Kollegen (alphabetisch): Constantin Gonzalez, Thorsten Früauf, Franz Haberhauer, Matthias Pfützner, Roland Rambau, Franz Stadler, Heiko Stein, Detlef Uhlerr und Holger Weihe.

Für Rückmeldungen und Anregungen stehen die Autoren gerne zur Verfügung.

Langen und Berlin im November 2006

Ulrich Gräf (Ulrich.Graef@sun.com), Detlef Drewanz (Detlef.Drewanz@sun.com)

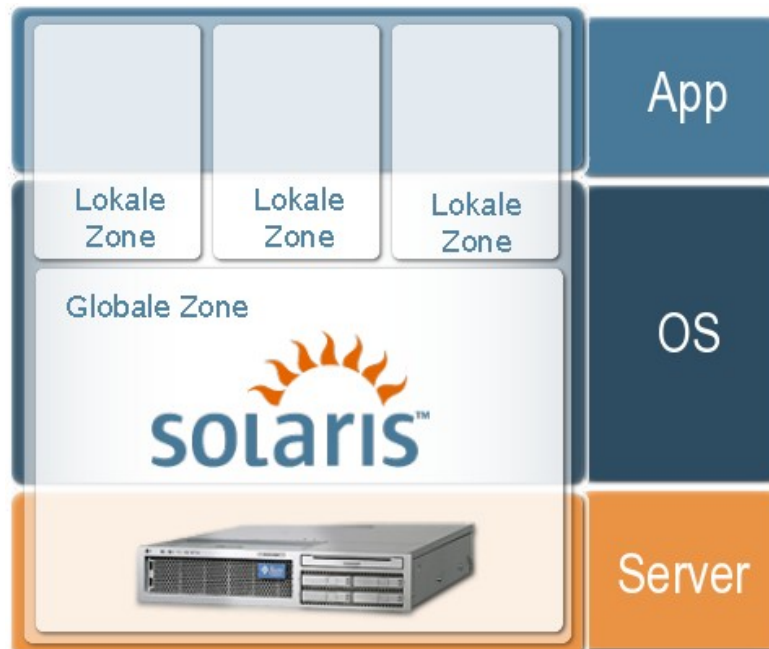
2. Allgemein

2.1. Solaris Zonen / Solaris Container

2.1.1. Überblick

[ug] Solaris 10 Zonen sind eine Virtualisierung auf der Schnittstelle zwischen Betriebssystem und Applikation.

- Es gibt eine globale Zone, die dem bisherigen Solaris-Betriebssystem entspricht.
- Zusätzlich können sogenannte lokale Zonen definiert werden, die auch nicht-global (non-global) Zone genannt werden.
- Alle shared Objekte (Programme, Libraries, der Kernel) werden nur einmal geladen; daher ist der Overhead bezüglich Hauptspeicher sehr gering.
- Bezüglich Filesystem ist eine lokale Zone von der globalen Zone getrennt. Das heißt, sie benutzt ein Unterverzeichnis der globalen Zone als root-Verzeichnis (wie bei chroot-Umgebungen).
- Eine Zone hat eine oder mehrere eigene Netzwerk-Adresse.
- Physikalische Devices sind in den lokalen Zonen nicht sichtbar, können jedoch definiert werden (ist ggf. sinnvoll im Zusammenhang mit Platten oder Volumes).
- Lokale Zonen haben eigene OS-Einstellungen wie z.B. für den Nameservice.
- Die lokalen Zonen sind bezüglich Prozessen untereinander und gegen die globale Zone abgetrennt, das heißt, eine lokale Zone kann nicht die Prozesse einer anderen Zone sehen.
- Die Abtrennung erstreckt sich auch auf Shared Memory und IP Adressen.
- Der Zugriff auf eine andere lokale Zone auf dem gleichen Rechner ist daher nur über Netzwerk möglich.
- Die globale Zone kann jedoch zur Steuerung und Überwachung (Accounting) alle Prozesse der lokalen Zonen sehen.



Zeichnung 1: [dd] Schematische Darstellung von Zonen

[ug] Eine lokale Zone sieht somit ein exklusiv nutzbares Solaris-Environment, welches von den anderen Zonen getrennt ist. Gleichzeitig werden viele Ressourcen gemeinsam genutzt, wodurch der Overhead gering ist.

Solaris Container sind die Kombination von Solaris Zonen mit dem Solaris Resource Management. Das Solaris Resource Management wurde bereits in Solaris 9 eingeführt und erlaubt die Kontrolle von CPU-, Memory- und Netzwerk-Ressourcen.

Weitere Details sind in der folgenden Tabelle zusammengefaßt:

<i>Shared Kernel:</i>	Der Kernel wird von der globalen Zone und den lokalen Zonen gemeinsam genutzt. Der durch das OS initiierte Overhead fällt nur einmal an. Die Kosten für eine lokale Zone sind daher, gemessen in Hauptspeicher-Bedarf, CPU-Leistung und Plattenplatz gering.
<i>Shared Objekte:</i>	In Unix werden alle Objekte wie Programme, Dateien und shared Libraries nur einmal als shared Memory Segment geladen, wodurch insgesamt die Performance verbessert wird. Dies erstreckt sich bei Solaris 10 auch über die Zonen; d.h. egal wie häufig zB ein Programm oder eine shared Library in Zonen benutzt wird: sie belegt im Hauptspeicher nur einmal Platz. (Dieses Sharing ist bei virtuellen Maschinen nicht möglich.)
<i>Filesystem:</i>	Der sichtbare Teil des Filesystems der lokalen Zone kann auf einen Teilbaum oder mehrere Teilbäume der globalen Zone eingeschränkt werden. Die Dateien in der lokalen Zone können verzeichnisweise shared mit der globalen Zone oder in Kopie konfiguriert werden.
<i>Patches:</i>	Pakete (Solaris Packages), die in der lokalen Zone als Kopie installiert sind, können dort auch separat gepatcht werden.
<i>Netzwerk:</i>	Zonen haben eigene virtuelle IP-Adresse (ggf. mehrere) auf einem oder mehreren Interfaces. Der Netzwerkverkehr zwischen Zonen läuft über den shared Netzwerk-Stack und bleibt somit im Rechner.
<i>Prozess:</i>	Jede lokale Zone sieht nur die eigenen Prozesse. Dies ist ein Teilbaum des gesamten Prozessbaumes. Die globale Zone sieht alle Prozesse der lokalen Zonen.
<i>Separation:</i>	Der Zugriff auf die Ressourcen der globalen Zone oder anderer lokalen Zonen, soweit nicht extra so konfiguriert (devices, storage), wird verhindert. Auftretende Softwarefehler werden durch Fehlerisolation auf die jeweilige lokale Zone begrenzt.
<i>Dedicated Devices:</i>	In der Defaultkonfiguration einer lokalen Zone sind keine physikalischen Devices enthalten. Es ist jedoch möglich, Devices (z.B. Platten, Volumes, DVD-Laufwerke, ...) ein oder mehreren lokalen Zonen zuzuordnen. So lassen sich auch zusätzliche Treiber nutzen.
<i>Shared Storage:</i>	Zusätzlich können weitere Teile des Dateibaumes (Filesysteme oder Directories) von der globalen Zone aus einer oder mehreren lokalen Zonen zugeordnet werden.
<i>Physikalische Devices:</i>	Die Administration von allen Dingen, die mit physikalischen Devices zu tun haben, wird von der globalen Zone aus erledigt. Die lokalen Zonen haben auf die Zuteilung dieser Devices keinen Zugriff.
<i>Root Delegation:</i>	Eine lokale Zone hat einen eigenen root-Account. Daher kann die Administration von Applikationen und Services in einer lokalen Zone vollständig an andere Personen delegiert werden (auch der root-Anteil), ohne dass die Betriebssicherheit in der globalen Zone oder der anderen lokalen Zonen beeinflusst wird. Root der globalen Zone hat generellen Zugang zu allen lokalen Zonen.
<i>Name Space:</i>	Lokale Zonen verfügen über eine unabhängige Namensumgebung mit Hostnamen, Netzwerkdiensten, Usern, Rollen und Prozessumgebungen. Eine Zone kann zur Nutzung von lokalen Dateien konfiguriert sein, eine andere auf dem gleichen Rechner kann z.B. LDAP oder NIS nutzen.
<i>System Einstellungen:</i>	Die Einstellungen in <code>/etc/system</code> gelten für alle Zonen. Jedoch sind die wichtigsten Einstellungen früherer Solaris Versionen (shared memory, semaphore und message queues) ab Solaris 10 online in jeder Zone möglich (sogar noch feiner mit sogenannten Projekten).

Tabelle 1: [ug] Eigenschaften von Solaris 10 Zonen

2.1.2. Zonen und die Installation von Software

[dd] Die geplante Funktionsweise der lokalen Zonen bestimmt die Art wie Software in Zonen installiert wird. Die meiste Software wird durch Solaris oder Softwarehersteller im pkg-Format bereitgestellt. Wird diese Software mit *pkgadd* in der globalen Zone installiert, steht diese Software automatisch auch allen anderen Zonen zur Verfügung. Dadurch wird die Installation und Pflege von Software erheblich vereinfacht, da auch wenn viele Zonen installiert sind, die Softwarepflege zentral aus der globalen Zone heraus vorgenommen werden kann.

Software kann auch exklusiv für eine lokale oder die globale Zone installiert werden, um z.B. in einer Zone unabhängige Änderungen an der Software vornehmen zu können. Dieses kann durch die Installation mit speziellen Optionen von *pkgadd* erreicht werden oder dadurch, daß Software nicht als pkg-Software installiert wird.

In jedem Falle werden jedoch der Kernel und die Treiber von Solaris zwischen allen Zonen geteilt, können aber nur in der globalen Zone direkt installiert, modifiziert und angesprochen werden.

2.1.3. Zonen und Security

[dd] Durch die Bereitstellung eigener root-Verzeichnisse für jede Zone, können durch die lokalen Nameservices in den Zonen eigene Festlegungen zu den Security-Einstellungen getroffen werden (Role Based Access Control, passwd-Datenbank).

Lokale Zonen verfügen über weniger mögliche Prozeß-Privilegien als die globale Zone, wodurch die Ausführung einiger Kommandos innerhalb einer Zone nicht möglich ist. Zu den Einschränkungen gehören u.a.:

- System-Ressourcen
 - Konfiguration von Swap und Prozessorsets
 - Veränderung des Prozeß Scheduler und Shared Memory
 - Anlegen von Device Files
 - Load/Unload von Kernel Modulen
 - lock und unlock von Hauptspeicher
 - real time Anwendungen
- Netzwerk
 - Konfiguration von Netzwerkinterfaces
 - Tracen des Netzwerkverkehrs (snoop)
 - Zugriff auf raw-sockets
- Dtrace

Diese Zugriffe sind nur in der globalen Zone erlaubt.

Solaris 10 wird zur Zeit wie bereits vorige Versionen nach Common Criteria EAL4+ zertifiziert. Die Zertifizierung wird durch die Canadian CCS durchgeführt. Canadian CCS ist Mitglied der Gruppe der Zertifizierungsstellen von westlichen Staaten, bei der auch das BSI Mitglied ist. Diese Zertifizierung wird automatisch von der BSI anerkannt. Bestandteil der Zertifizierung sind Einbruchssicherheit, Separation und neu in Solaris 10 die Abschottung der Zonen. Der Abschluss der Zertifizierung wird Ende 2006 erwartet. Details dazu sind nachlesbar unter:

<http://www.cse-cst.gc.ca/services/common-criteria/ongoing-evals-e.html>

2.1.4. Zonen und Ressource Management

[ug] In Solaris 9 wurde Ressource Management auf Basis von Projekten, Tasks und Ressource Pools eingeführt. In Solaris 10 erfährt dieses Feature eine Erweiterung mit der Einführung von Zonen. Kontrollierbar sind die folgenden Ressourcen:

- CPU Anteile (Prozessor-Sets und Fair-Share-Scheduler)
- Real Memory-Verbrauch (*rcapd*)
(Zonen-spezifische Einstellungen in einem späteren Solaris Update)
- Kontrolle des Netzwerk-Verkehrs (IPQoS = IP Quality of Services)

2.1.4.1. CPU Ressourcen

[ug] Mit Zonen sind 3 Stufen von Ressource Management möglich:

- Partitionierung der CPUs in Prozessorsets, die Ressource Pools zugeordnet werden können. Lokale Zonen können dann den Ressource-Pools zugeordnet werden und laufen dann ausschließlich in der definierten CPU Menge.
- Einsatz des Fair-Share-Scheduler (FSS) zwischen lokalen Zonen in einem Ressource Pool. Dies erlaubt die feingranulare Zuteilung von CPU-Ressourcen in einem definierten Verhältnis.
- Einsatz des FSS in einer lokalen Zone. Damit kann die der Zone zugeteilte CPU-Zeit nochmals im definierten Verhältnis auf Projekte zugeordnet werden.

Prozessor-Sets in einem Ressource-Pool

Einer lokalen Zone läßt sich genau wie einem Projekt ein Default Resource Pool zuordnen, in dem alle Prozesse der Zone ablaufen (`zonecfg: set pool=`). Man kann CPUs einem Ressource Pool zuordnen. Dann laufen die Prozesse der Zonen nur auf den dem Ressource-Pool zugeordneten CPUs ab.

Einem Ressource-Pool können auch mehrere Zonen (oder auch Projekte) zugeordnet sein, die sich dann die CPU-Ressourcen des Ressource Pools teilen.

Fair-Share-Scheduler in einem Ressource-Pool

Mit dem Fair Share Scheduler (FSS) läßt sich die Zuordnung von CPU-Ressourcen innerhalb eines Ressource Pools steuern. Dazu kann man jeder Zone bzw jedem Projekt einen Anteil (`share`) zuordnen. Die Einstellungen der Zonen und der Projekte in einem Ressource Pool werden dazu herangezogen, die Aufteilung der CPU-Ressourcen vorzunehmen.

Das geht so:

- Ist die Auslastung des Prozessore-Sets unter 100%, dann wird keine Steuerung vorgenommen, da ja noch freie CPU Leistung zur Verfügung steht.
- Liegt die Auslastung bei 100%, dann wird der Fair-Share-Scheduler aktiv, indem die Priorität der beteiligten Prozesse so verändert wird, dass die zugeteilte CPU-Leistung einer Zone oder eines Projektes dem definierten Anteil entspricht.
- Der definierte Anteil berechnet sich aus dem Share-Wert einer aktiven Zone (oder Projekt) dividiert durch die Summe der Shares aller aktiven Zonen/Projekte.

Die Zuordnung ist auch dynamisch im laufenden Betrieb änderbar.

CPU-Ressource Kontrolle in einer Zone

In einer lokalen Zone ist es nochmals möglich, Projekte und Ressource-Pools zu definieren und die Zuordnung von CPU-Ressourcen mittels FSS auf die in der Zone laufenden Projekte zu kontrollieren (siehe vorigen Absatz).

2.1.4.2. Memory Ressource Kontrolle

[ug] In Solaris 10 (auch in einem Update von Solaris 9) ist der Hauptspeicher-Verbrauch auf der Ebene von Zonen, Projekten und Prozessen kontrollierbar. Dies wird mit dem sogenannten Ressource-Capping-Daemon (`rcapd`) realisiert. Für die entsprechenden Objekte wird ein Limit für den Realspeicherverbrauch definiert. Steigt der Verbrauch eines der Projekte über das definierte Limit an, dann veranlasst der `rcapd` das Auslagern von wenig genutzten Hauptspeicher-Seiten von Prozessen. Als Meßgröße wird die Summe des Platzverbrauches der Prozesse herangezogen. Damit wird der definierte Hauptspeicher-Bedarf eingehalten. Die Leistung der Prozesse in dem entsprechenden Objekt sinkt, da sie gegebenenfalls Seiten wieder einlagern müssen, wenn die Speicherbereiche wieder benutzt werden.

Als Vereinfachung wird in einem Update von Solaris 10 ein Memory Limit für jede Zone einstellbar sein.

Memory Sets ist ein Entwicklungsprojekt für einen späteren Update von Solaris 10. Hier werden sogenannte Memory-Sets definiert. Die Prozesse einer Zone (Projekt, Resourcepool) können den Speicher dann nur aus dieser Menge Speicher beziehen.

2.1.4.3. Netzwerk Ressourcen Kontrolle (IPQoS = IP Quality of Services)

[ug] In Solaris 10 (auch Solaris 9) ist es möglich, den Netzwerk-Verkehr zu klassifizieren und die Datenrate der Klassen zu kontrollieren.

Ein Beispiel wäre die Bevorzugung des Netzwerk-Verkehrs eines Webservers vor dem Netzwerk Backup. Wenn der Netzwerk-Backup läuft, soll der Dienst am Kunden darunter nicht leiden.

Die Konfiguration erfolgt mit Regeln in einer Datei, die mit dem Kommando `ipqosconf` aktiviert wird.

Die Regeln bestehen aus einem Teil, mit dem der Netzwerkverkehr klassifiziert wird und aus Aktionen, mit denen die Datenrate / Burstrate gesteuert werden kann. Die Klassifizierung kann unter anderem nach den folgenden Parametern erfolgen:

- Adresse des Senders
- Adresse des Empfängers
- Port-Nummer des Empfängers
- Datenverkehr Typ (UDP,TCP)
- Userid des lokalen Prozesses
- Projekt des lokalen Prozesses (`/etc/project`)
- TOS Feld des IP-Verkehrs (Veränderung Priorität in einer laufenden Verbindung)

Näheres siehe Kochbuch zu IPQoS.

2.1.5. User Interfaces für Zonen

[dd] Für die Arbeit mit Zonen und Containern stehen verschiedene Werkzeuge zur Verfügung. Solaris liefert selbst eine Reihe von Command Line Interface (CLI)-Tools wie `zoneadm`, `zonectfg` und `zlogin` mit denen auf der Kommandozeile oder in Scripten die Konfiguration und Installation von Zonen vorgenommen werden kann.

Als Grafisches User Interface (GUI) steht für Solaris/SPARC der Solaris Container Manager (http://www.sun.com/software/products/container_mgr/) zur Verfügung. Dabei handelt es sich um ein separates Produkt, das zusammen mit dem Sun Management Center (SunMC) betrieben wird. Der Container Manager gestattet durch sein Nutzerinterface eine einfache und schnelle Neuerzeugung oder Umkonfiguration von Zonen und eine effektive Benutzung des Ressource Managements.

Webmin stellt mit dem Zonenmodule ein Browser User Interface (BUI) zur Installation und dem Management von Zonen zur Verfügung. Da das Module noch nicht vollständig in Solaris integriert ist, kann es unter <http://www.webmin.com/standard.html> als `zones.wbm.gz` heruntergeladen werden.

2.1.6. Zonen und Hochverfügbarkeit

[tf/du]Ein einzelnes System hat bei allen vorhandenen RAS-Fähigkeiten nur die Verfügbarkeit eines Rechners. Ist diese Verfügbarkeit nicht ausreichend, so kann man mit Sun Cluster 3.1 08/05 und dem HA Solaris Container Agenten sogenannte Failover Zonen implementieren und so Zonen zwischen Cluster Knoten schwenken. Damit erhöht sich die Verfügbarkeit des Gesamtsystems erheblich. Zusätzlich wird hier aus einem Container ein flexibler Container. Das heißt, es ist völlig unerheblich auf welchem der am Cluster beteiligten Rechner der Container abläuft. Ein Verschieben des Containers ist durch administrative Aktionen oder bei Ausfall eines Rechners automatisch möglich.

Mit der einer zukünftigen Sun Cluster Version ist geplant auch Services zwischen Zonen schwenken zu können, wobei Zonen dann virtuelle Cluster Nodes darstellen.

2.2. Virtualisierungstechnologien im Vergleich

Die herkömmlichen RZ Technologien umfassen

- Applikationen auf separaten Rechnern
Hierzu gehören auch mehrstufige Architekturen mit Firewall, Loadbalancing, Web- und Application-Server und Datenbanken.
- Applikationen auf einem Rechnerverbund
Dies sind die MPP-Systeme oder Grid-Verbund-Rechner die meist nur im High-Performance Computing Bereich (Technical Computing) benutzt werden.
- Viele Applikationen auf einem großen Rechner

Die Separierung von Applikationen auf Rechnern vereinfacht die Installation der Applikationen, erhöht jedoch die Maintenance Kosten, da die Betriebssysteme mehrfach installiert und separat gepflegt werden müssen. Weiterhin sind die Rechner in der Regel schlecht ausgelastet (< 30 %).

MPP-Systeme oder Grid-Verbunde machen nur in Bereichen Sinn, in denen die Applikationen bereits angepaßt sind. Die Anpassung ist ein größerer Schritt und nur teilweise für heute verfügbare Standard-Applikationen durchgeführt. In Zukunft kann diese Technologie daher mit der Anpassung der Applikationen interessanter werden.

Viele Applikationen in einem Rechner ist die Betriebsart, mit der heute Mainframes und größere Unix Systeme betrieben werden. Die Vorteile liegen in der besseren Auslastung der Systeme (mehrere Applikationen) und geringeren Zahl von zu wartenden Betriebssystem-Installationen. Daher ist gerade diese Variante für die Konsolidierung im Rechenzentrum interessant.

Die Herausforderungen bestehen darin, für die Applikationen eine Umgebung zu schaffen, in der die Applikationen zwar unabhängig ablaufen können (Separation), aber doch Teile miteinander teilen (Sharing) um Kosten zu sparen. Die im einzelnen interessanten Bereiche sind:

- Separation. Wie weit sind die Environments der Applikationen getrennt?
- Applikation. Wie fügt sich die Applikation in das Environment ein?
- Auswirkungen auf die SW-Maintenance
- Auswirkungen auf die HW-Maintenance
- Delegation: Können Administrations-Tätigkeiten an das Environment delegiert werden?
- Skalierung der Environments?
- Overhead der Virtualisierungstechnologie?
- Kann man unterschiedliche OS-Versionen in den Environments benutzen?

Dazu wurden verschiedene Techniken zur Virtualisierung entwickelt, die im folgenden vorgestellt werden.

Vergleiche dazu auch: <http://en.wikipedia.org/wiki/Virtualization>

2.2.1. Stufe 0: Konsolidierung in einem Rechner

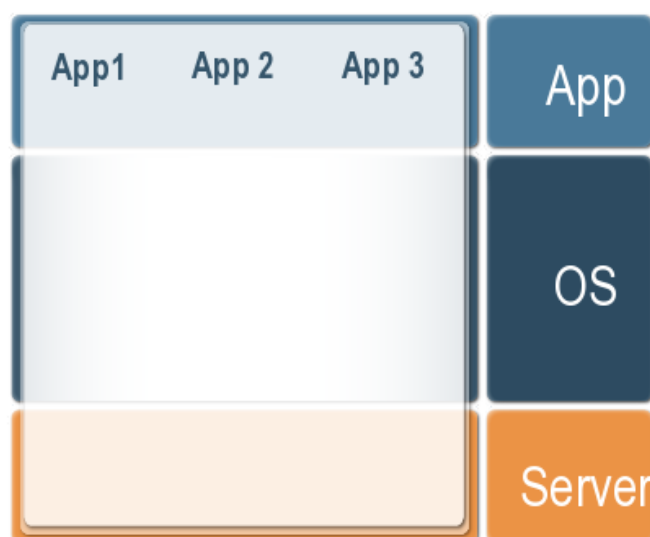
[ug] Die Applikationen werden auf einem Rechner installiert und unter unterschiedlichen Userids benutzt. Das ist die Konsolidierung, die von Haus aus in den modernen Betriebssystemen möglich ist.

Vorteile:

- Applikation: Alle Applikationen sind lauffähig, sofern sie nur im Basis-Betriebssystem lauffähig sind und keine eigenen OS-Treiber nutzen. Einschränkungen gibt es allerdings dann, wenn unterschiedliche Versionen der Applikationen mit definiertem Install-Verzeichnis gebraucht werden oder zwei Instanzen die gleiche Userid brauchen (z.B. Oracle Instanzen) bzw. die Konfigurations-Dateien an den selben Stellen im Filesystem liegen.
- Skalierbarkeit: Die Leistung einer Anwendung läßt sich online ändern.
- OS-Maintenance: Nur für ein OS muß die OS-Installation, Patches und Umsetzung Firmen-interner Standards erfolgen. Das heißt mit dem Administrations-Aufwand für eine Maschine kann man viele Applikationen betreiben.
- Overhead: Der Overhead ist gering, da nur die Applikationsprozesse pro Applikation laufen müssen.

Nachteile:

- HW-Maintenance: Bei Ausfall einer gemeinsam genutzten Komponente sind ggf. viele oder alle Applikationen betroffen.
- OS-Maintenance: Die Administration wird kompliziert, sobald Applikationen auf unterschiedliche Versionen einer Software basieren (z.B. Versionen für Oracle, Weblogic, Java, ...). Ohne genaue Dokumentation und Change-Management wird ein solches System schwer beherrschbar. Ein Fehler in der Dokumentation wird nicht sofort bemerkt, kann sich jedoch beim Upgrade (HW oder OS) später fatal auswirken.
- Separation: Die Applikationen können sich über gemeinsam genutzte Hardware und das OS gegenseitig beeinflussen. Der Einfluss kann bei Solaris mit Resource Management und Netzwerk-Bandbreitenkontrolle reduziert werden.
- Delegation: Die für die Applikation / den Service zuständige Abteilung braucht für einen Teil der Ablaufsteuerung Root-Rechte oder muss mit dem Rechner-Betrieb bezüglich Änderungen kommunizieren. Das kann daher die Sicherheit beeinträchtigen oder aufwändiger werden/länger dauern.
- OS-Versionen: Unterschiedliche Betriebssysteme/Versionen sind nicht möglich.



Zeichnung 2: [dd] Konsolidierung in einem Rechner

Diese Konsolidierung wird von vielen modernen Betriebssystemen ermöglicht die es erlauben, mehrere Software-Pakete zu installieren, ggf. auch die gleiche Software in unterschiedlichen Versionsständen.

2.2.2. Stufe 1: Domains/physikalische Partitionen

[ug] Ein Rechner kann dabei durch Konfiguration in Teil-Rechner (Domain, Partition) zerlegt werden. Die Domains sind fast vollständig physikalisch getrennt, da die elektrischen Verbindungen abgeschaltet werden.

Gemeinsam benutzte Teile sind entweder sehr ausfallsicher (Gehäuse) oder redundant aufgebaut (Service-Prozessor, Netzteile).

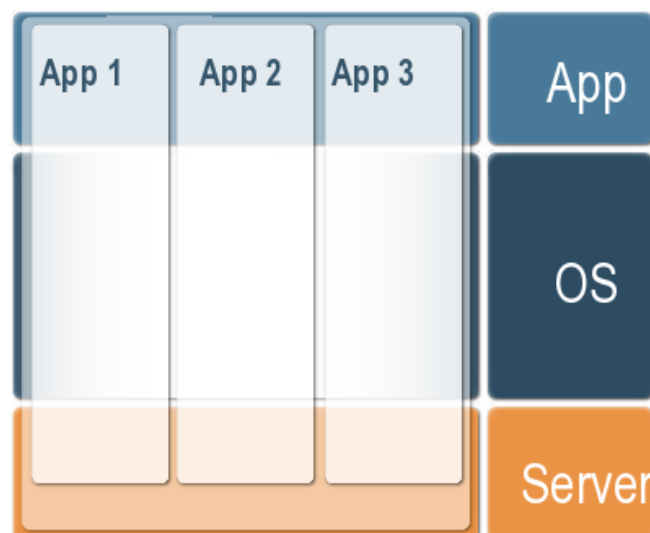
Vorteile:

- Separation: Die Applikationen sind gut voneinander separiert, ein gegenseitiger Einfluss über das OS oder ausgefallene gemeinsame Hardware ist nicht möglich.
- Applikation: Alle Applikationen sind lauffähig, sofern sie nur im Basis-Betriebssystem lauffähig sind.
- Skalierbarkeit: Die Leistung einer Virtualisierungsinstanz (hier eine Domain) läßt sich bei einigen Implementierungen im laufenden Betrieb ändern (Dynamic Reconfiguration), indem HW-Ressourcen zwischen Domains verschoben werden.
- HW-Maintenance: Bei Ausfall einer Komponente kann bei entsprechender Auslegung der Domain die Applikation trotzdem betrieben werden. Reparaturen sind durch Dynamic Reconfiguration im laufenden Betrieb möglich (bei redundanter Auslegung). Nur um Total-Ausfälle abzufangen, muß noch ein Cluster eingerichtet werden (Stromversorgung, Brand des Gehäuses, Ausfall des RZ, Software-Fehler).
- OS-Versionen: Unterschiedliche Betriebssysteme/Versionen sind möglich.

Nachteile:

- OS-Maintenance: Jede Maschine muß separat administriert werden. OS-Installation, Patches und die Umsetzung Firmen-interner Standards müssen für jede Maschine separat erfolgen.
- Delegation: Die für die Applikation / den Service zuständige Abteilung braucht für einen Teil der Ablaufsteuerung Root-Rechte oder muss mit dem Rechner-Betrieb bezüglich Änderungen kommunizieren. Das kann daher die Sicherheit beeinträchtigen oder aufwändig werden/länger dauern.
- Overhead: Jede Maschine hat den Overhead eines eigenen Betriebssystems.

Sun bietet Domains in den Rechnern der Midframe Serie ab SunFire 4800 – SunFire 6900 und bei den High-End Systemen SunFire 12k, 15k, 20k und 25k an.



Zeichnung 3: [dd] Domains/Physikalische Domains

Diese Virtualisierungstechnik wird von einigen Herstellern zur Verfügung gestellt (Sun Dynamic System Domains, Fujitsu-Siemens Partitions, HP nPars). HW-Support und (wenig) OS-Support sind notwendig.

2.2.3. Stufe 2: Logische Partitionen

[ug] Auf der Hardware eines Rechners läuft ein minimales Betriebssystem, der Hypervisor, der die Schnittstelle zwischen Hardware und OS des Rechners virtualisiert. Auf den entstehenden sogenannten virtuellen Maschinen lässt sich jeweils ein eigenes Betriebssystem (Gast-Betriebssystem) installieren.

Bei einigen Implementierungen läuft der Hypervisor auch als normales Applikationsprogramm, was jedoch erhöhten Overhead bedeutet.

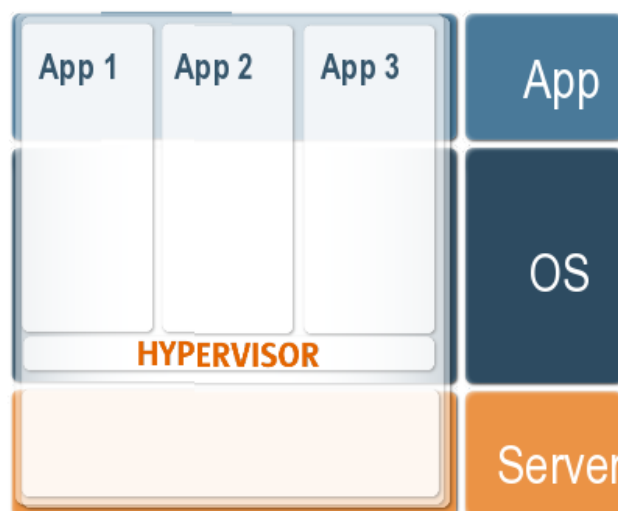
In der Regel werden durch Emulation aus den realen Devices virtuelle Devices erzeugt; reale und virtuelle Devices werden den logischen Partitionen durch Konfiguration zugeteilt.

Vorteile:

- Applikation: Alle Applikationen des Gast-Betriebssystems sind lauffähig.
- Skalierbarkeit: Die Leistung einer logischen Partition lässt sich teilweise im laufenden Betrieb ändern, wenn das OS und der Hypervisor es zulassen.
- Separation: Die Applikationen sind voneinander separiert, ein direkter gegenseitiger Einfluss über das OS ist nicht möglich.
- OS-Versionen: Unterschiedliche Betriebssysteme/Versionen sind möglich.

Nachteile:

- HW-Maintenance: Bei Ausfall einer gemeinsam genutzten Komponente sind ggf. viele oder alle logischen Partitionen betroffen. Es wird jedoch versucht, durch vorbeugende Analyse Anzeichen eines zukünftigen Ausfalls zu erkennen, um vorab die Fehler auszugrenzen.
- Separation: Die Applikationen können sich über gemeinsam genutzte Hardware gegenseitig beeinflussen. Ein Beispiel ist hier das virtuelle Netzwerk, da der Hypervisor einen Switch emulieren muß. Auch die virtuellen Platten, die zusammen auf einer realen Platte liegen und sich dort gegenseitig den Schreib-/Lese-Kopf „wegziehen“, sind ein Beispiel für dieses Verhalten. Um das zu vermeiden, kann man reale Netzwerk-Interfaces und/oder dedizierte Platten verwenden, was den Aufwand zur Nutzung von logischen Partitionen allerdings erhöht.
- OS-Maintenance: Jede Partition muss separat administriert werden. OS-Installation, Patches und die Umsetzung Firmen-interner Standards müssen für jede Partition separat erfolgen.
- Delegation: Wenn die für die Applikation / den Service zuständige Abteilung Root-Rechte benötigt, dann sind alle Aspekte des Betriebssystems in der logischen Partition administrierbar. Das kann die Sicherheit beeinträchtigen.
- Overhead: Jede logische Partition hat den Overhead eines eigenen Betriebssystems, insbesondere wird der Hauptspeicherbedarf der einzelnen Systeme beibehalten..



Zeichnung 4: [dd] Logische Partitionen

Zu logischer Partitionierung gehören das IBM VM Betriebssystem, IBM LPar auf z/OS und AIX, HP vpars, sowie VMware und XEN.

Sun bietet für die SPARC Architektur (sun4v) sogenannte Logical Domains (ab Solaris 10 11/06) und wird zukünftig im x64-Bereich XEN anbieten.

2.2.4. Stufe 3: Container (Solaris Zonen) in einem OS

[ug] In einer Betriebssystem-Installation werden voneinander unabhängige Ausführungsumgebungen geschaffen. Der Kernel wird mandantenfähig: er ist nur einmal da, erscheint aber jeder Zone als wäre er exklusiv zugeordnet.

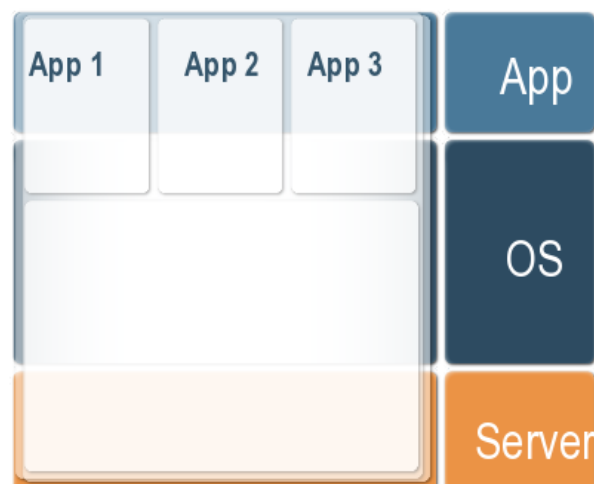
Die Separation wird implementiert, indem der Zugriff auf Ressourcen eingeschränkt wird, wie z.B. die Sichtbarkeit der Prozesse (modifiziertes procfs), Nutzbarkeit der Devices (modifiziertes devfs), die Sichtbarkeit des Dateibaumes (wie bei chroot).

Vorteile:

- Applikation: Alle Applikationen sind lauffähig, sofern sie keine eigenen Treiber benutzen oder sonst systemnahe Features nutzen. Eigene Treiber sind jedoch über Installation in der globalen Zone nutzbar.
- Skalierbarkeit: Die Leistung eines Containers ist änderbar (mit Ressource Management).
- Separation: Die Applikationen sind voneinander separiert, ein direkter gegenseitiger Einfluss über das OS ist nicht möglich.
- OS-Maintenance: Nur an zentraler Stelle (in der globalen Zone) muß die OS-Installation, Patches und Umsetzung Firmen-interner Standards erfolgen.
- Delegation: Die für die Applikation / den Service zuständige Abteilung braucht für einen Teil der Ablaufsteuerung Root-Rechte. Sie kann hier die Root-Rechte in der Zone erhalten, ohne die anderen lokalen Zonen oder die globale Zone beeinträchtigen zu können. Die Zuweisung von Ressourcen ist allein der globalen Zone vorbehalten.
- Overhead: Alle Prozesse lokaler Zonen sind aus Sicht der globalen Zone nur normale Applikationsprozesse. Der OS-Overhead (Memory Management, Scheduling, Kernel) und der Speicherbedarf für shared Objekte (Dateien, Programme, Libraries) entsteht nur einmal. Jede Zone hat zusätzlich nur eine kleine Zahl von Systemprozessen. Daher sind hunderte von Zonen auf einem 1-Prozessor-System möglich.

Nachteile:

- HW-Maintenance: Bei Ausfall einer gemeinsam genutzten Komponente sind ggf. viele oder alle Zonen betroffen. Mit FMA (Fault Management Architecture) erkennt Solaris 10 Anzeichen eines zukünftigen Ausfalls und kann die betroffenen Komponenten (CPU, Memory, Bus-Systeme) online deaktivieren bzw. ersetzen. Durch Einsatz von Cluster Software (Sun Cluster) kann die Verfügbarkeit der Applikation in der Zone verbessert werden (Cluster Agent für Solaris Zonen).
- Separation: Die Applikationen können sich über gemeinsam genutzte Hardware gegenseitig beeinflussen. Der Einfluss kann bei Solaris mit Ressource Management und Netzwerk-Bandbreitenkontrolle minimiert werden.
- OS-Versionen: Unterschiedliche Betriebssysteme/Versionen sind nicht möglich.



Zeichnung 5: [dd] Container (Solaris Zonen) in einem OS

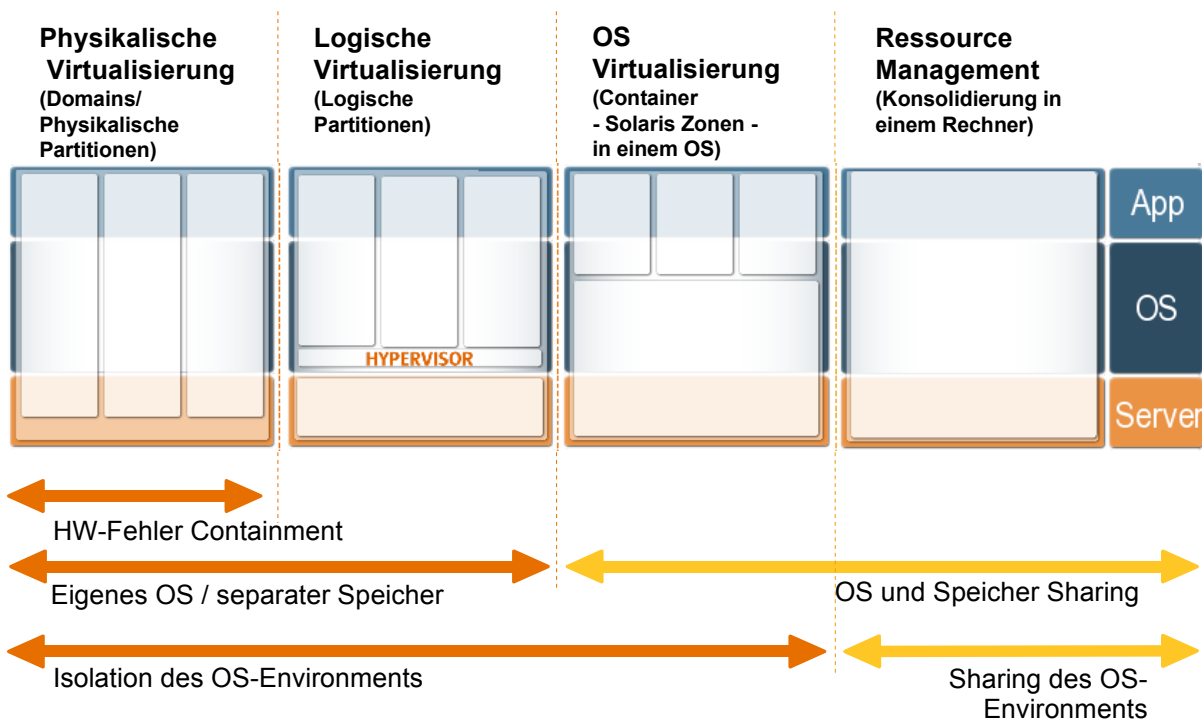
Implementierungen sind im freien BSD Betriebssystem die Jails, in Solaris die Zonen, im Linux das vserver Projekt. HW Voraussetzungen sind nicht erforderlich.

2.2.5. Zusammenfassung zu den Virtualisierungstechnologien

[ug] Die diskutierten Virtualisierungstechnologien lassen sich in der folgenden Tabelle zusammenfassen.

	Stufe 0: Konsolidierung in einem Rechner	Stufe 1: Domains/ Physikalische Partitionen	Stufe 2: Logische Partitionen	Stufe 3: Container (Solaris Zonen) in einem OS
Separation	--	++	+ (Software) -- (Hardware)	+ (Software) -- (Hardware)
Applikation	--	++	++	++
SW-Maintenance	++	--	--	++
HW-Maintenance	./.	++	./.	./.
Delegation	--	--	--	++
Skalierbarkeit	+	++	++	++
Overhead	++	--	--	++
OS-Versionen	eine	mehrere	mehrere	eine

Tabelle 2: [ug] Zusammenfassung zu den Virtualisierungstechnologien



Zeichnung 6: [dd] Vergleich von Virtualisierungstechnologien

3. Anwendungsfälle

Das folgende Kapitel diskutiert verschiedene Anwendungsfälle für Container und bewertet diese.

3.1. Grid Computing mit Isolation

Anforderung

[ug] Im Unternehmen besteht Bedarf eine gewisse Menge Rechenarbeit im Hintergrund unter Nutzung der freien Zyklen der bestehenden Rechner durchzuführen.

Dafür ist Grid Software wie Sun N1 Grid Engine geeignet. Jedoch soll den Prozessen die dann im Grid laufen, die Sicht auf die anderen Prozesse des Systems versperrt werden.

Lösung

[ug] Auf den Rechnern mit freier Kapazität wird Sun Gridware in jeweils einer Zone installiert:

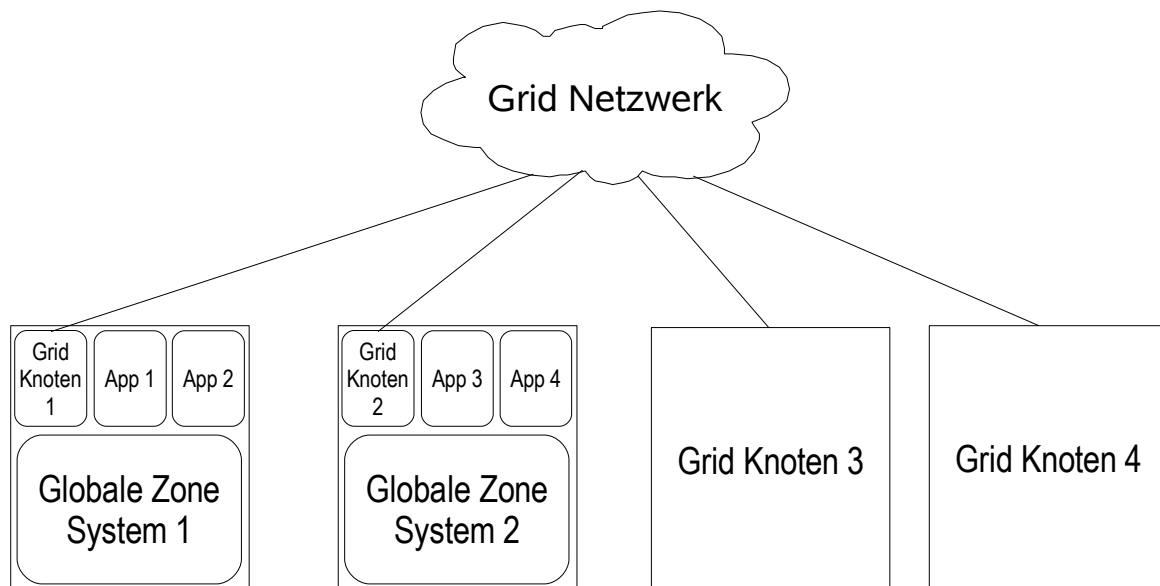
- Sparse-root Zones werden benutzt (Kochbuch).
- Software und Daten werden per NFS in der globalen Zone installiert.
- Sun Grid wird in der lokalen Zone installiert.
- Automatische Erzeugung der Zone (Kochbuch)
- Software-Installation per lofi von der globalen Zone (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Die Rechner können zu den definierten Zeiten die freien Kapazitäten nutzen.
- Die Daten der anderen Applikationen bleiben vor Einsicht geschützt. Dadurch wird die Bereitschaft der Applikations-Zuständigen besser, die Rechner dafür zur Verfügung zu stellen.
- Die Applikations-Abteilung kann die freien Kapazitäten den Grid-Nutzern „verkaufen“.

Über alles werden Rechner-Kapazitäten eingespart.



Zeichnung 7: [dd] Anwendungsfall Grid Computing mit Isolation

3.2. Kleine Webserver

Anforderung

[ug] Eine der folgenden Situationen besteht:

- Ein ISP möchte ohne große Kosten die Möglichkeit haben, Webserver automatisch aufzusetzen. Der ISP will auf Basis dieser Technologie ein günstiges Angebot für Webserver mit Root-Zugang erstellen.
- Ein Rechenzentrum einer größeren Firma hat Requests aus vielen Abteilungen für interne Webserver zum Informationsaustausch zwischen den Abteilungen. Das Interesse der Kunden ist, möglichst einfach ohne viele Regularien und Absprachen, Web-Content abzulegen (wenig Aufwand). Daher strebt die Fachabteilung einen Webserver an, auf dem sonst niemand sonst arbeitet. Das Interesse des Rechenzentrums ist, die Administrationskosten niedrig zu halten. Daher soll es möglichst kein eigener Rechner sein, zumal auch der Datenverkehr wahrscheinlich klein sein wird und keinen eigenen Rechner rechtfertigt.

Lösung

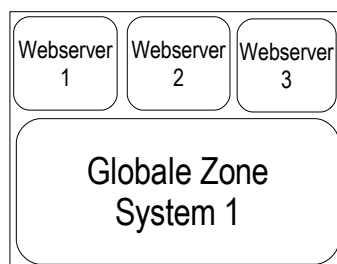
[ug] Die Webserver werden durch automatisch installierte Zonen realisiert. Im einzelnen werden die folgenden Details verwendet:

- Sparse-root Zones, das heißt die Zonen erben möglichst alles von der globalen Zone (Kochbuch).
- Das Softwareverzeichnis des Webserver, z.B. `/opt/webserver` wird nicht vererbt (`inherit-pkg-dir`), um verschiedene Web-Server Versionen zu ermöglichen.
- Automatische Erzeugung einer Zone per Script (Kochbuch)
- Automatische Systemkonfiguration in der Zone mit `sysidcfg` (Kochbuch)
- Automatische IP-Adressverwaltung (Kochbuch)
- Option: Automatische Schnell-Installation einer Zone (Kochbuch)
- Option: Applikations-Administrator mit root-Zugang (Kochbuch)
- Option: Software-Installation per mount (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften.

- Die Betriebsabteilung hat geringe Aufwände bei der Erzeugung der Zonen.
- Da die erwartete Last sehr klein ist, ist der Konsolidierungseffekt sehr groß, da nur die aktiven Prozesse in der Zone Ressourcen brauchen.
- Die Nutzer der automatisch erzeugten Web-Server in den Zonen haben die Freiheit, diverse verschiedene Versionen zu nutzen, ohne sich umgewöhnen zu müssen. Das heißt, ihre Kosten sind gering und es entsteht kaum Schulungsbedarf.
- Die Webserver können mit den Standardports und den voreingestellten IP-Adressen der Container arbeiten - es sind keine besonderen Konfigurationen nötig, wie sonst, wenn mehrere Webserver auf einem System betrieben werden sollen.



Zeichnung 8: [dd] Anwendungsfall kleine Webserver

3.3. Multi-Netzwerk Konsolidierung

Anforderung

[dd] Die folgende Situation besteht:

- Ein Kunde verfügt über mehrere unterschiedliche Netzwerke, die entweder in mehreren Stufen durch Firewalls oder durch Router getrennt sind. In den einzelnen Netzwerken laufen Anwendungen. Der Kunde möchte sehr einfach Anwendungen aus verschiedenen Netzwerken oder Security-Bereichen gemeinsam auf einem physikalischen System benutzen, da die Anwendungen nicht jeweils ein komplettes System auslasten.

Lösung

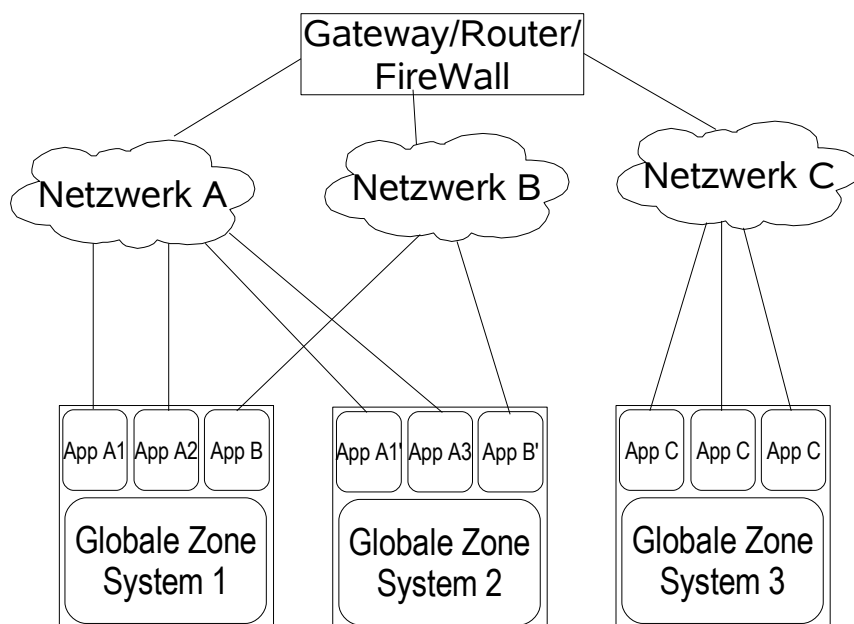
[dd] Die einzelnen Anwendungen werden jeweils in eine Zone installiert. Zonen werden nach bestimmten Kriterien (Redundanz, ähnliche Anwendung, Lastverhalten, ...) zusammen auf physikalische Server gruppiert. Das Routing zwischen den Zonen ist ausgeschaltet, um die Netzwerke zu trennen. Im einzelnen werden die folgenden Details verwendet:

- Erzeugung von Zonen (Kochbuch)
- Zonen als Laufzeitumgebung für je eine Anwendung. (Kochbuch).
- Das Routing der Globalen Zone auf den Interfaces ist ausgeschaltet, damit Zonen sich nicht gegenseitig erreichen können. D.h. die Zonen erreichen jeweils nur Adressen in ihrem Netzwerk (Kochbuch)

Bewertung

[dd] Dieser Anwendungsfall hat die folgenden Eigenschaften.

- Die Netzwerkstruktur vereinfacht sich durch die Einsparung von Routen und Routern.
- Die Anzahl benötigter Systeme verringert sich.
- Anwendungen können nach neuen Gesichtspunkten geordnet werden, z.B. alle Web-Server auf einen physikalischen Server bzw. für Webserver werden z.B. T2000 benutzt, für Proxy Server werden T1000 benutzt, für alle Datenbanken UltraSPARC IV+ Systeme, etc.
- Die globale Zone kann als zentrale Administrationsinstanz aller Zonen eines Systems benutzt werden. Über die globalen Zonen kann ein separates Administrationsnetz gelegt werden.
- Die Administration der Anwendungen liegt innerhalb der Zone bei den Anwendungsadministratoren. Werden gleiche Anwendungen auf Systemen zusammengruppiert, kann ein Anwendungsadministrator aus der globalen Zone heraus alle Anwendungen in den Zonen einfacher administrieren oder durch die Nutzung von Sparse-root Zones die Administration vereinfachen.



Zeichnung 9: [dd] Anwendungsfall: Multi-Netzwerk Konsolidierung

3.4. Multi-Netzwerk Monitoring

Anforderung

[dd] Die folgende Situation besteht:

- Ein Kunde verfügt über mehrere unterschiedliche Netzwerke, die entweder in mehreren Stufen durch Firewalls oder durch Router getrennt sind. In den einzelnen Netzwerken sind verschiedene Rechner installiert. Der Kunde möchte die Administration vereinfachen und von einer zentralen Stelle aus direkt in alle Netzwerke „hineinschauen“ können und von einer zentralen Stelle aus administrieren können.

Lösung

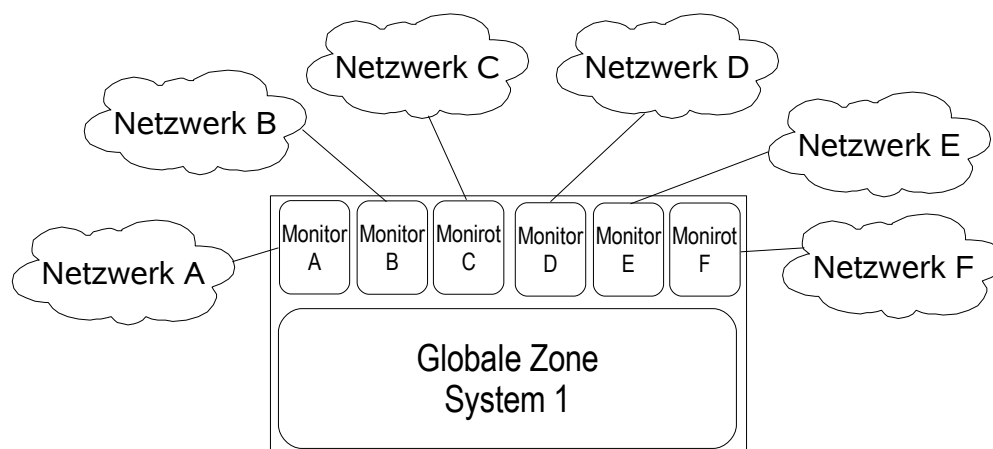
[dd] Es wird ein zentraler Monitoring- und Administrator-Server installiert. Auf diesem Server werden mehrere Zonen erzeugt, die in jedes Netzwerk eine Netzwerkverbindung haben. Aus den Zonen erfolgt das Monitoring oder die Administration der Rechner der einzelnen Netzwerke. Im Einzelnen werden die folgenden Details verwendet:

- Sparse-root Zones, das heißt die Zonen erben möglichst alles von der globalen Zone (Kochbuch).
- Alle Zonen verwenden die selben Monitoring- und Administrationstools.
- Monitoring-Daten werden in zwischen Zonen gemeinsam genutzte Filesysteme abgelegt. (Kochbuch)
- Die Daten können aus einer lokalen Zone oder aus der globalen Zone heraus zentral ausgewertet werden.
- Von einer zentralen Stelle aus (die globale Zone) können zentrale Konfigurationsdateien über die Zonen an alle Systeme in den Netzwerken direkt verteilt werden. Umständliche Wege über Router oder Firewalls entfallen.
- Das Routing zwischen den Zonen muß abgeschaltet sein. (Kochbuch)

Bewertung

[dd] Dieser Anwendungsfall hat die folgenden Eigenschaften.

- Die Betriebsabteilung hat geringe Aufwände bei der Erzeugung der Zonen.
- Der Administrationsaufwand sinkt für Systeme in den Netzwerken, da keine mehrfachen Logins über Router oder Firewalls hinweg ausgeführt werden müssen.
- Es kann ein Single-Point-of-Administration geschaffen werden.
- Entlastung der Router und Firewalls von Netzlast und zusätzlichen Konfigurationen.
- Nutzung einheitlicher Monitoring Werkzeuge.
- Nutzung einheitlicher Konfigurationen wird vereinfacht.



Zeichnung 10: [dd] Anwendungsfall Multi-Netzwerk Monitoring

3.5. Multi-Netzwerk Backup

Anforderung

[dd] Die folgende Situation besteht:

- Ein Kunde verfügt über mehrere unterschiedliche Netzwerke, die entweder in mehreren Stufen durch Firewalls oder durch Router getrennt sind. In den einzelnen Netzwerken sind verschiedene Rechner installiert. Der Kunde möchte das Backup vereinfachen und von einer Stelle aus das direkte Backup der Systeme in den einzelnen Netzwerken durchführen.

Lösung

[dd] Es wird ein Server mit Backup-Serversoftware installiert. Auf diesem Server werden mehrere Zonen erzeugt, die in jedes Netzwerk eine Netzwerkverbindung haben. In den Zonen werden Backup-Clients gestartet, die mit dem Backup-Server in der globalen Zone über das interne Netzwerk kommunizieren oder als Backup-Server direkt auf ein verfügbares Backup-Gerät schreiben. Im Einzelnen werden die folgenden Details verwendet:

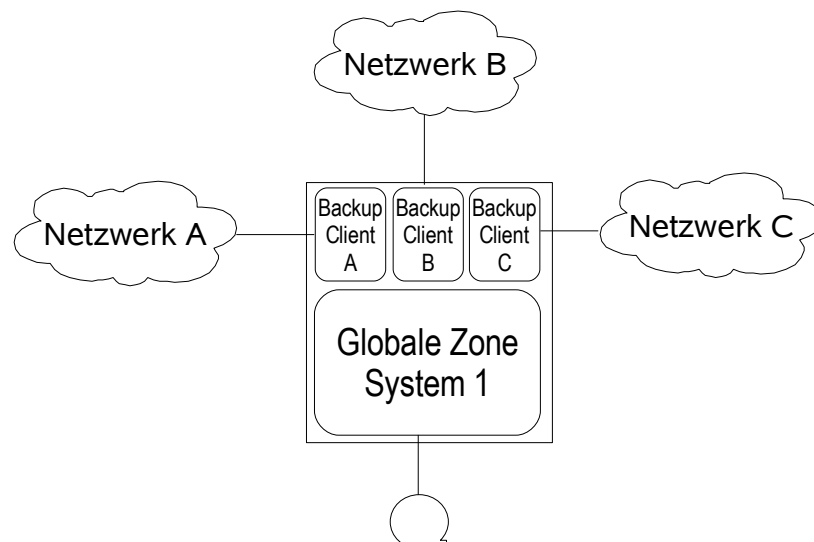
- Sparse-root Zones, das heißt die Zonen erben möglichst alles von der globalen Zone (Kochbuch).
- Die Backup-Client oder -Server Software ist für jede Zone separat installiert.
- Bereitstellung eines Gerätes aus der globalen Zone an eine lokale Zone. (Kochbuch)
- Netzwerksetup zur Verbindung von der globalen mit der lokalen Zone (Kochbuch)

Bewertung

[dd] Dieser Anwendungsfall hat die folgenden Eigenschaften.

- Die Betriebsabteilung hat geringe Aufwände bei der Erzeugung der Zonen.
- Backup und Restore kann zentral von einer Stelle aus organisiert und durchgeführt werden.
- Durch direktes Backup können höhere Backup-Geschwindigkeiten erreicht werden. Router oder Firewalls werden nicht durch Backup-Daten belastet.
- U.u. Einsparung von Lizenzkosten für Backup-Software.
- Sharing von Backup-Hardware zwischen Abteilungen und Netzwerken.

Bisher hat sich leider gezeigt, daß die Softwarehersteller Backup-Server Software noch nicht für die Benutzung in Zonen freigegeben haben. So ist also dieser Anwendungsfall nur bedingt einsetzbar. Backup-Clients sind vereinzelt bereits zur Benutzung in Zonen freigegeben.



Zeichnung 11: [dd] Anwendungsfall Multi-Netzwerk Backup

3.6. Konsolidierung Development/Test/Integration/Produktion

Anforderung

[ug] Zu einer Applikation, die in Produktion ist, existieren in der Regel noch weitere Systeme, die die gleiche Applikation tragen:

- Development Systeme
- Test-Systeme
- Integrations-Systeme ggf. mit Simulation der Applikationsumgebung
- Disaster-Recovery-Systeme

Lösung

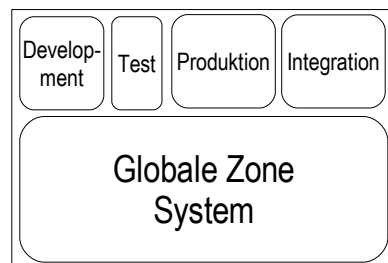
[ug] Die verschiedenen Systeme können auf einem Rechner in Zonen realisiert werden. Die Details:

- Sparse-root Zones, das heißt die Zonen erben möglichst alles von der globalen Zone (Kochbuch).
- Option: Ressource Pools mit eigenem Prozessor-Set für die Produktion (Kochbuch)
- Option: Applikations-Administrator mit root-Zugang (Kochbuch)
- Option: Software-Installation per mount (Kochbuch)
- Option: OS Release Upgrade per Flash-Image (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Es sind weniger Rechner insgesamt zu betreiben. Daher können Einsparungen bei Platz, Stromverbrauch und Klimatisierung erzielt werden.
- Die Anwendungen können auf genau dem Environment getestet werden, auf dem sie auch später ablaufen sollen.
- Ein Umstieg auf eine neue Version in der Produktion ist einfach durch Umleiten der Last auf die Zone mit der neuen Version möglich. Eine Installation nach Test ist nicht erforderlich.



Zeichnung 12: [dd] Anwendungsfall Konsolidierung Development/Test/Integration/Produktion

3.7. Konsolidierung von Test-Systemen

Anforderung

[ug] Für Tests von Software und Applikationen existieren im RZ-Environment viele Test-Systeme die immer nur bei Tests benutzt werden. Größtenteils werden sie nur für qualitative Tests benutzt, wobei die Systeme nicht wie bei Lasttests belastet werden. Ein Neu-Installieren der Testsysteme je nach benötigtem Test kommt allerdings nicht in Frage, weil man ohne Wartezeit auf das Environment gehen können will, genau so wie es in der Produktion läuft.

Die Testsysteme sind daher sehr schlecht ausgelastet.

Lösung

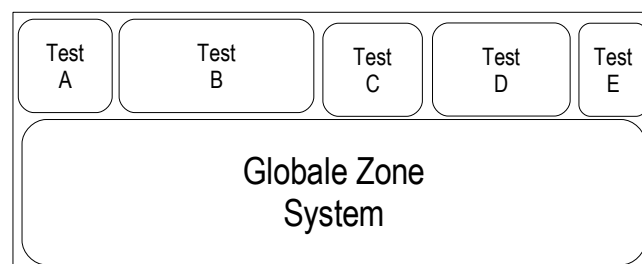
[ug] Die Testsysteme werden durch Zonen auf einem größeren Server realisiert. Die Details:

- Sparse-root Zones / Whole-root Zones, je nach Bedarf (Kochbuch).
- Filesystem Entscheidungen analog zum Produktionssystem
- Option: Ressource Management mit Prozessor Sets für parallel ablaufende Tests (Kochbuch)
- Option: Automatische Erzeugung der Zone (Kochbuch)
- Option: Software-Installation per mount (Kochbuch)
- Option: Move der Zone zwischen Rechnern (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Die Betriebsabteilung benötigt viel weniger Rechner die als Testsysteme dienen. Es müssen viel weniger Installationen durchgeführt werden.
- Teure Systeme für Lasttests müssen nur einmal und nicht für jede Applikation angeschafft werden. Lasttests auf den mittels Zonen geharten Maschinen müssen jedoch abgesprochen werden (Betrieb).
- Die Anwender haben die Testinstallation immer im Zugriff, jedenfalls qualitativ; gegebenenfalls nicht mit der vollen Performance.



Zeichnung 13: Anwendungsfall Konsolidierung von Testsystemen

3.8. Schulungssysteme

Anforderung

[ug] In Schulungs-Abteilungen müssen die Rechner häufig wieder neu aufgesetzt werden, die den Schulungsteilnehmern (auch Schüler/Studenten) zur Verfügung stehen.

Lösung

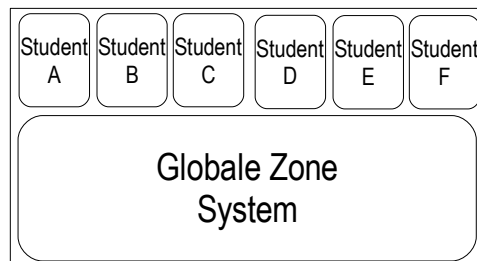
[ug] Die Schulungssysteme werden durch automatisch installierte Zonen realisiert:

- Sparse-root Zones, das heißt die Zonen erben möglichst alles von der globalen Zone (Kochbuch).
- Automatische Erzeugung einer Zone per Script (Kochbuch)
- Automatische Systemkonfiguration in der Zone mit `sysidcfg` (Kochbuch)
- Automatische IP-Adressenverwaltung (Kochbuch)
- Option: `/opt` ist nicht vererbt (`inherit-pkg-dir`), für Schulung von Installation (Kochbuch)
- Option: Automatische Schnell-Installation einer Zone (Kochbuch)
- Option: Applikations-Administrator mit root-Zugang (Kochbuch)
- Option: Installation gemeinsamer Software per mount (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Der Betrieb der Schulung ist extrem einfach, da die Zonen für die Schulung einfach für die nächste Schulung neu erzeugt werden können.
- Die Schulungsteilnehmer sind in der Zone selbst root, womit sie alle wesentlichen Administrationsfunktionen ohne Gefahr für das Gesamtsystem ausüben können. Die Neuinstallation des Rechners entfällt somit.
- Der Schulungsleiter hat von der globalen Zone aus Einblick in die Arbeit der Schulungsteilnehmer.
- Die Kosten für das Schulungssystem sinken daher drastisch.



Zeichnung 14: Anwendungsfall Schulungssysteme

3.9. Server Konsolidierung

Anforderung

[ug] Im Rechenzentrum werden mehrere Applikationen betrieben, deren Auslastung zu gering ist (oft weit unter 50%). Die Rechner selber benötigen in der Regel Strom, Kühlung und Platz.

Lösung

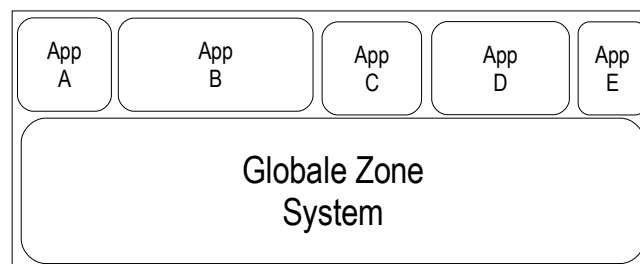
[ug] Mehrere Anwendungen werden in Zonen auf einem Rechner konsolidiert. Die Details:

- Dies kann in der Regel mit Sparse-root Zones erfolgen. (Kochbuch).
- Genau eine Applikation pro Zone installieren. (Best Practices)
- Option: Software-Installation durch shared Directory (Kochbuch)
- Option: Automatische Software Installation (Kochbuch)
- Option: Ressource Management mit Ressource Pools (Kochbuch)
- Option: Ressource Management mit Fair Share Scheduler (Kochbuch)
- Option: Netzwerk Ressource Management mit IPQoS (Kochbuch)
- Option: Hochverfügbarkeit durch Cluster
- Option: Software-Installation per Provisioning (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Die Betriebsabteilung hat geringere Kosten, da weniger Platz, Energie und Kühlung benötigt werden.
- Die Anwendungsabteilung hat eine Kapselung der Applikation und genau definierte Schnittstellen. Die Verfügbarkeit der Applikation steigt.



Zeichnung 15: Anwendungsfall Server Konsolidierung

3.10. Security Kapselung

Anforderung

[ug] Im Rechenzentrum laufen Applikationen auf verschiedenen Rechnern, weil

- Bestimmte Abteilungen sicher sein wollen, dass die Daten und Prozesse nicht von anderen Abteilungen gesehen werden.
- Services für verschiedene Kunden sollen konsolidiert werden. Die Kunden wünschen Vertraulichkeit der Daten und der Prozesse (die ggf. Rückschluss auf Geschäftsprozesse erlauben).

Lösung

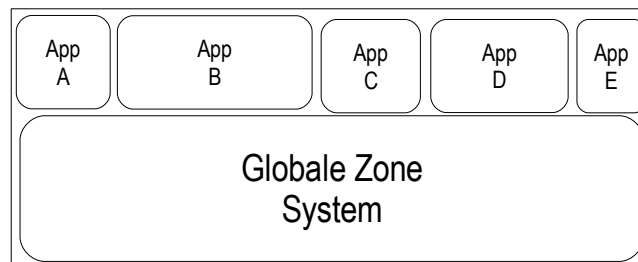
[ug] Die Applikationen der Kunden werden in verschiedenen Zonen installiert:

- Applikationen werden nur in lokalen Zonen installiert (Best Practices)
- Die Filesysteme/Devices mit den Daten der jeweiligen Kunden werden nur in den entsprechenden Zonen zur Verfügung gestellt. Damit ist die Vertraulichkeit der Daten auch ohne strenge Disziplin bei den Dateizugriffsrechten gewährleistet.
- Option: Software-Installation in der Zone in ein nur dort vorhandenes lokales Filesystem oder ein nicht gehartetes /opt .

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Die Betriebsabteilung hat eine bessere Auslastung der Systeme.
- Die Anwender / Kunden behalten die Vertraulichkeit.
- Die Basis-Kosten sind geringer, wodurch der Service bei höherer Marge günstiger angeboten werden kann.



Zeichnung 16: Anwendungsfall Security Kapselung

3.11. Developer Testsysteme

Anforderung

[ug] Entwickler brauchen zum Ausprobieren ihrer Applikation Testsysteme. Häufig ist auch das Zusammenspiel von mehreren Rechnern zu testen. Die Ressourcen für die Installation der Testsysteme sind in der Regel limitiert. Da die Developer die meiste Zeit entwickeln haben die Testsysteme eine geringe Auslastung.

- Eine geshartes Benutzen der Testsysteme ist möglich, jedoch erhöht es die Dauer bis der Test durchgeführt ist, da die Systeme neu installiert bzw. wiederhergestellt werden müssen.
- Es ist nicht auszuschließen, dass es Zeitkonflikte gibt, wenn die Developer gleichzeitig auf die Systeme zugreifen wollen.
- Backup/Restore bzw. Installation sind in der Regel in der Betriebsabteilung lokalisiert. Die herkömmliche Nutzung der Testsysteme erzeugt hier einigen Aufwand.

Lösung

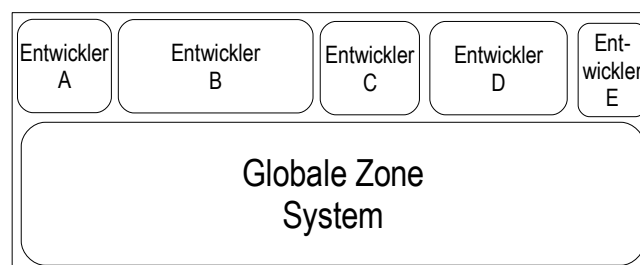
[ug] Die Testsysteme der Developer werden mittels mittelgroßer Rechner und Zonen realisiert.. Die Details:

- Sparse-root Zones / Whole-root Zones je nach Bedarf. (Kochbuch).
- Daten liegen lokal (Kochbuch)
- Automatische Erzeugung der Zone (Kochbuch)
- Automatische Systemkonfiguration in der Zone mit sysidcfg (Kochbuch)
- Automatische IP-Adressenverwaltung (Kochbuch)
- Der Developer wird Applikations-Administrator mit root-Zugang (Kochbuch)
- Zu testende Software wird lokal installiert.
- Option: Software-Installation (teilweise) per mount (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Die Betriebsabteilung hat viel weniger Testsysteme zu betreiben, jedes neue Testsystem braucht daher nur Plattenplatz und keinen eigenen Rechner mehr.
- Das Erstellen der Testsysteme kann voll automatisiert werden, der Aufwand für die Betriebsabteilung ist daher viel geringer.
- Die Developer können mehrere Testsysteme gleichzeitig bestehen lassen; nur die gerade gebrauchten müssen hochgefahren werden. Vergleich von Funktionalitäten in den Versionen ist ohne viel Aufwand möglich.
- Um einen Test durchzuführen ist keine Wartezeit (f. Restore) oder Koordination (mit anderen Tests auf der Maschine) notwendig.



Zeichnung 17: Anwendungsfall Developer Testsysteme

3.12. Hosting für verschiedene Firmen auf einem Rechner

Anforderung

[ug] Ein Application Service Provider betreibt Systeme für verschiedene Firmen. Die Systeme sind nicht ausgelastet.

Lösung

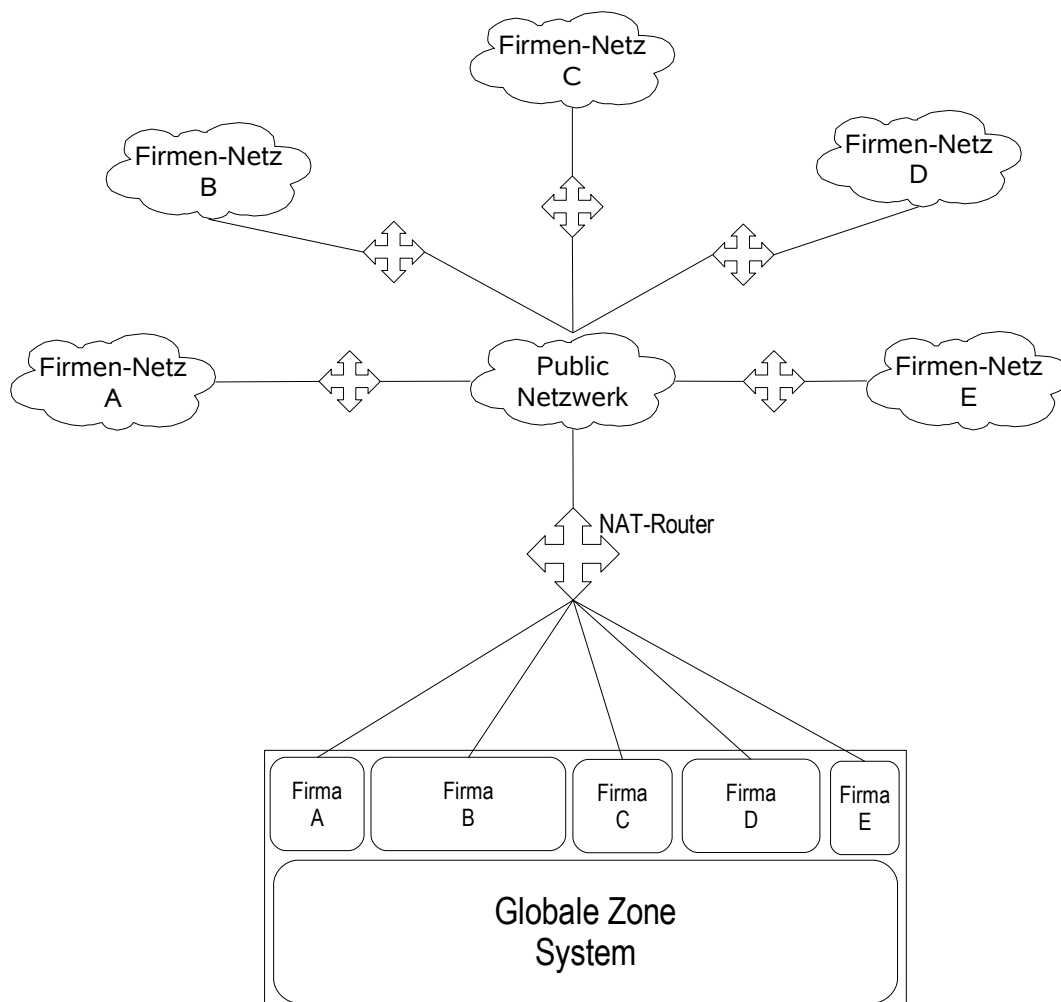
[ug] Die Applikationen werden in Zonen auf einem Rechner konsolidiert. Die Details:

- Getrennte Filesysteme für jede Zone (Kochbuch).
- Option: Integration der Server in die Firmennetze per NAT (Kochbuch)
- Option: Software-Installation per mount (Kochbuch)

Bewertung

[ug] Dieser Anwendungsfall hat die folgenden Eigenschaften:

- Der Application Service Provider spart RZ-Platz, Energie und Kühlung. Er kann daher seine Dienstleistung günstiger anbieten.
- Der Kunde des ASP kann die Dienstleistung günstiger einkaufen.



Zeichnung 18: Anwendungsfall Hosting für verschiedene Firmen auf einem Rechner

4. Best Practices

Das folgende Kapitel beschreibt Konzeptionen zur Umsetzung von Architekturen mit Containern.

4.1. Konzepte

4.1.1. Arten der Softwareinstallation in Solaris

[dd]

- Softwareinstallation als Package mit *pkgadd*
 - Software wird im pkg-Format verteilt und mit *pkgadd* installiert (*P-Software*)
 - pkg-Informationen sind in der pkg-Datenbank
 - Paketverwaltung wie Update, Patch, Remove und Zoneninstallation nutzen pkg-Datenbank
- Softwareinstallation aus einem Archiv
 - Software wird per Installationskript installiert, z.B. mit *cp* oder *tar* (*A-Software*)
 - keine Informationen über die installierte Software in der pkg-Datenbank
 - Löschen und Updates der Software durch die Software selbst

Zur Vereinfachung der Betrachtungsweise soll im Folgenden die Software in *P-Software* und *A-Software* unterteilt werden.

4.1.2. Softwareinstallation in einer Zone

[dd] Software kann so für Zonen installiert werden, daß sie nur in dieser einen Zone verfügbar ist

- Installation von Software in einer lokalen Zone
 - Software wird in einer lokalen Zone installiert und kann nur dort genutzt werden
 - Die Verantwortung für die Pflege dieser Software liegt bei dem Administrator der Zone. Dafür besteht hier der Vorteil, lokale Änderungen an der Software selbst vornehmen zu können.
 - Für die Softwareinstallation ist zusätzlicher Plattenplatz in der Zone vorzusehen. Die Installation von *P-Software* oder *A-Software* kann in der üblichen Form vorgenommen werden.
 - Für Anwender stellt sich die Frage, ob ihre Software in einer lokalen Zone läuft und von dem Softwarehersteller dafür auch zertifiziert ist. Anwendungen können nach http://developers.sun.com/solaris/articles/zone_app_qualif.html zur Nutzung in lokalen Zonen qualifiziert werden.
- Installation von Software in der globalen Zone
 - Software wird in der globalen Zone installiert und soll nur dort nutzbar sein
 - *P-Software* muß mit dem Kommando *pkgadd -G <package>* installiert werden, damit nur die Einträge in die pkg-Datenbank der globalen Zone erfolgen. Diese Software wird ebenfalls in einer besonderen pkg-Datenbank (*/var/sadm/install/gz-only-packages*) eingetragen und bei einer Installation einer Zone nicht mitverwendet. *P-Software* exklusive für die globale Zone können zum Beispiel der Kernel und die Treiber sein.
 - *A-Software* nimmt keine Ergänzungen an der pkg-Datenbank vor und muß in ein Verzeichnis installiert werden, das ausschließlich in der globalen Zone sichtbar ist.

4.1.3. Softwareinstallation in der globalen Zone für lokale Zonen

[dd] Software kann durch die globale Zone für lokale Zonen bereitgestellt werden. Diese Software wird als *P-Software* oder *A-Software* installiert und gepflegt. Die Verantwortung für die Installation und Pflege der Software liegt bei dem Administrator der globalen Zone.

- Bereitstellung von *A-Software* durch die globale Zone:
 - Software wird lokal in der globalen Zone durch *tar*, *cp* oder entsprechende Installationsrichtlinien in Verzeichnissen installiert
 - pkg-Datenbank wird nicht verändert
 - Software wird durch Bereitstellung des Installationsverzeichnisses an die entsprechende lokale Zone bereitgestellt (*zonecfg*-Parameter *add fs*)
 - Um die Software in einer lokalen Zone benutzen zu können, müssen ggf. in der lokalen Zone noch besondere Installationsprozeduren ausgeführt werden. Dazu zählen z.B. das Anpassen von config-Files oder das Erzeugen von log-Verzeichnissen.
 - Mit dieser Variante kann durch die Bereitstellung des Installationsverzeichnisses für jede Zone separat entschieden werden, ob bestimmte Software nutzbar sein soll oder nicht
- Bereitstellung von *P-Software* durch die globale Zone
 - *P-Software* wird mit *pkgadd* in der globalen Zone installiert und wird in der pkg-Datenbank der globalen Zone eingetragen.
 - Diese Software wird allen existierenden und noch zu installierenden lokalen Zonen zur Verfügung gestellt. Dieser Vorgang kann durch die Angabe des *pkginfo(4)*-Parameters *SUNW_PKG_ALLZONES* erzwungen werden, und schließt die Benutzung von *pkgadd -G* aus.
 - Die Installation findet durch den Administrator der globalen Zone statt. Im Ergebnis ist die so installierte Software in allen Zonen identisch.
 - Solaris 10 hat eine besondere Form der Softwarebereitstellung von *P-Software* aus der globalen Zone an die lokalen Zonen eingeführt. Durch den *zonecfg*-Parameter *inherit-pkg-dir* werden Verzeichnisse (und Unterverzeichnisse) der globalen Zone an eine lokale Zone readonly zur Verfügung gestellt. Technisch wird dieses durch einen readonly Loopback-Mount eines Verzeichnisses in die lokale Zone hinein realisiert. Mit diesem Verfahren werden Dateien und pkg-Informationen für lokale Zonen verfügbar, obwohl die eigentliche Installation der Software in der globalen Zone vorgenommen wurde. Dabei wird bei der Installation entschieden, ob eine zu installierende Datei kopiert werden muß. Befindet sich eine zu installierende Datei in einem *inherit-pkg-dir*, so wird die Datei nicht kopiert, sondern in der pkg-Datenbank nur die Verfügbarkeit der Datei für die Zone eingetragen.
 - Zonen, die nahezu die gesamte Software von der globalen Zone "erben", benötigen sehr wenig Plattenplatz und werden auch *Sparse-root Zonen* genannt.
 - Zonen die keine Software der globalen Zone "erben" und alle Packages (außer den Kernel selbst) lokal installieren, brauchen nahezu soviel Platz wie eine vollständige OS Installation und werden *Whole-root Zonen* genannt.
 - Es gibt Mischformen zwischen Sparse-root und Whole-root Konfigurationen.

4.1.4. Sparse-root Zonen

[dd] Sparse-root Zonen werden Zonen genannt, an die die folgenden Verzeichnisse standardmäßig von der globalen Zone aus als *inherit-pkg-dir* "vererbt" werden:

- */lib*
- */platform*
- */sbin*
- */usr*

- */opt* wird im Standard nicht als *inherit-pkg-dir* angelegt. Soll Software in der globalen Zone installiert werden und in den lokalen Zonen shared mitbenutzt werden, dann kann man */opt* auch als *inherit-pkg-dir* eintragen.
- Der Plattenplatz wird für diese Verzeichnisse nur einmal in der globalen Zone auf der Festplatte belegt und mehrfach in den Sparse-root Zonen benutzt. So wird durch eine Sparse-root Zone nur ein minimaler Plattenplatz benötigt, der je nach Architektur und Umfang der Softwareinstallation ca. 70 MB (x86/x64) oder 100MB (SPARC) beträgt.
- Die OS- und Softwareverwaltung wird einfacher, da Softwarepakete nur einmal zentral in der globalen Zone installiert werden und in allen lokalen Zonen readonly und unter Eintragung in der pkg-Datenbank der Zone bereitgestellt werden.
- Die Softwarepflege (Update, Patching) von vererbten Packages erfolgt zentral aus der globalen Zone heraus für alle installierten Zonen.
- Durch die Vererbung werden Sparse-root Zonen erheblich schneller installiert und schneller gepatched.
- Programme und Bibliotheken werden genau wie shared Memory nur einmal in den Hauptspeicher geladen. Da Sparse-root Zonen die gleichen Programme und Bibliotheken mit anderen Zonen teilen, wird der Platz im Hauptspeicher nur einmal belegt, unabhängig davon, wie häufig es aufgerufen worden ist und wie viele Zonen es benutzen. Das führt bei häufiger Benutzung der gleichen Software in unterschiedlichen Zonen zu einer Einsparung in benötigtem Hauptspeicher.
- Sparse-root Zonen sind nicht dafür geeignet, unterschiedliche Softwarestände in den Zonen zu betreiben, wenn sich diese Software in einem *inherit-pkg-dir* befindet. Hierfür sind entweder Whole-root Zonen Konfigurationen notwendig oder die Software darf nicht in einem *inherit-pkg-dir* installiert werden.
- Einige Installationsroutinen von Software benötigen Schreibrechte in einem *inherit-pkg-dir*. Um solche Software in Zonen installieren und nutzen zu können, muß die Whole-root Zonekonfiguration oder eine Mischform gewählt werden.

4.1.5. Whole-root Zonen

[dd] Whole-root Zonen werden Zonen genannt, die eine eigene Solaris Installation in der Zone enthalten. D.h. bei der Erzeugung der Zone, werden alle Packages der globalen Zone in die lokale Zone kopiert. Ausgenommen hiervon sind Packages, die mit *pkgadd -G* in der globalen Zone installiert wurden. Diese stehen ausschließlich der globalen Zone zur Verfügung (z.B. der Kernel und die Treiber).

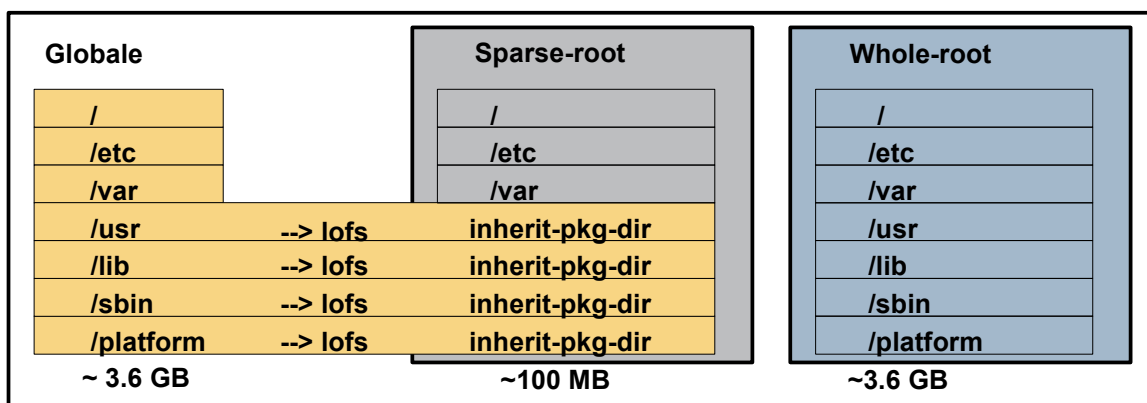
- Der benötigte Platz für eine Whole-root Zone umfasst ca. den Bedarf einer kompletten Solaris Installation, je nach Architektur und Umfang der Softwareinstallation ca. 2500 MB (x86/x64) oder 3600 MB (SPARC).
- Die Erzeugung und das Patchen einer Whole-root Zone erfordert mehr Zeit.
- Whole-root Zonen teilen sich bis auf den Kernel keine Programme und Bibliotheken untereinander oder mit der globalen Zone. Deshalb werden sie nochmals in den Hauptspeicher geladen, unabhängig davon, ob eine andere Zone eine eigene Kopie des gleichen Programms bereits in den Hauptspeicher geladen hat. Das führt zu einem erhöhten Bedarf an benötigten Hauptspeicher-Ressourcen durch Whole-root Zonen.
- Whole-root Zonen haben eine nahezu vollständige Unabhängigkeit von der OS-Installation der globalen Zone, da fast alle Packages (außer Kernel und Treiber) in der Kopie vorliegen und separat modifiziert und gepatched werden können. Dies schließt auch den Kernel Jumbo Patch aus, der die Kernel Pakete in der lokalen Zone nicht findet.

4.1.6. Vergleich Sparse-root Zone und Whole-root Zone

[dd] Aus den vorangegangenen Betrachtungen ergibt sich eine Vergleichstabelle zu Sparse-root Zonen und Whole-root Zonen. In den überwiegenden Fällen werden Sparse-root Zonen eingesetzt.

	Sparse-root Zone	Whole-root Zone
Benötigter Plattenplatz	70 - 100 MB	2600 - 3600 MB
Schnelle Erzeugung von Zonen	+	-
Schnelles Patchen pro Zone	+	-
Zentrale Softwareinstallation und Pflege	+	+
Effektive Hauptspeichernutzung	+	-
Unabhängige Installation von OS-Teilen in der Zone	-	+
Schreibmöglichkeit in OS-Teilen	-	+

Tabelle 3: [dd] Vergleich Sparse-root Zone und Whole-root Zone



Zeichnung 19: [dd] Schematischer Vergleich Sparse-root Zone und Whole-root Zone

4.1.7. Storage-Konzepte

4.1.7.1. Storage für das Root-Filesystem der lokalen Zonen

[ug] Normalerweise reicht es aus, ein Filesystem für mehrere Zonen zu sharen.

Bestehen erhöhte Anforderungen an die Minimierung der gegenseitigen Beeinflussung ist zu empfehlen, jeder Zone ein eigenes Filesystem zu geben. Mit dem neuen ZFS Filesystem von Solaris 10 ist das mit geringem Aufwand möglich (ab Solaris 10 6/06). Zur Zeit wird wegen möglicher Probleme eines Upgrades von Solaris 10 auf ein späteres Solaris 10 Release nicht empfohlen, ZFS als zoneroot zu verwenden.

Für ein System auf dem eine hohe Zahl an Zonen installiert werden soll ist zu empfehlen, daß ein Storage-Subsystem mit Cache für die Root-Filesysteme eingesetzt wird. Jede OS-Instanz und damit auch jede Zone schreiben in unregelmäßigen Abständen Messages, Log-Einträge, utmp-Einträge usw. auf das Root-Filesystem. Die Applikationen nutzen ggf. ebenfalls das Root-Filesystem für Logs. Bei vielen Instanzen kann dies zu einem Engpass auf dem Filesystem führen (bei ZFS weniger kritisch).

4.1.7.2. Storage für Daten

[ug] Die Planung für den Storage für Daten (Files, Datenbanken) erfolgt in gleicher Weise wie für dedizierte Rechner. Die Storage Devices werden an den Rechner, der die Zonen enthält, angeschlossen. Die Filesysteme werden in der globalen Zone direkt gemountet und an die Zonen per loopback-mount übergeben (`zonecfg add fs`). Dann können die Filesysteme auch von anderen Zonen zum Datenaustausch genutzt werden, sofern konfiguriert.

Raw-Devices z.B. für Datenbanken können ebenfalls dediziert einer lokalen Zone übergeben werden. (`zonecfg add device`) Das Übergeben von Raw-Devices an mehrere Zonen macht

nur im Ausnahmefall Sinn.

4.1.7.3. Storage für Programme/Applikationen

[ug] Für die von den Applikationen genutzten Programme kann man ebenfalls diese Vorgehensweise (s.o.) wählen.

Mit Zonen kann man alle Software in einem Verzeichnis der globalen Zone installieren und dieses Verzeichnis den lokalen Zonen zuordnen (`zonecfg add fs`). Dies ist vergleichbar dazu, daß Software auf einem NFS-Server im RZ liegt; hier wird ein lokales Filesystem verwendet, das der Zone zur Verfügung gestellt ist.

Damit ist nach einmaliger Installation die Software für alle Zonen nutzbar. Die Eigenständigkeit des Rechners bleibt unangetastet. Diese Art der Softwareverteilung ist nur für A-Software (non-pkg-software) sinnvoll. P-Software (pkg-software) wird durch Installationsmechanismen der Zone in die Zonen verteilt bzw. vererbt.

4.1.7.4. Rootplatten Layout

[dd] Je nach Verfügbarkeitsanforderungen werden die root-Platten innerhalb eines Systems über interne Platten gespiegelt oder über unterschiedliche Controller und externe Storage Devices.

Die gesamte OS-Installation wird heute bereits in vielen Fällen in `/` installiert, d.h. ohne weitere Unterteilung in `/usr` oder `/opt`. Lediglich `/var` sollte in ein separates Filesystem installiert werden um ein Überlaufen von `/` durch große log-Files zu vermeiden. Weiterhin sollte zusätzlicher Platz für ein späteres geplantes Live Upgrade des Systems vorgesehen werden.

Wenn die Anzahl benötigter Slices pro Platte erschöpft ist, sollte `/var/crash` in `/var` mit hineinverlagert werden oder die Benutzung von Softpartitionen in Erwägung gezogen werden.

Oft wird `/usr/local` als schreibbares Verzeichnis in einer lokalen Zone benötigt. Wenn aber z.B. eine Sparse-root Zone benutzt wird und `/usr` ein `inherit-pkg-dir` ist, kann in `/usr/local` in einer lokalen Zone nicht geschrieben werden. Empfehlenswert ist in solchen Fällen, in der globalen Zone einen Softlink z.B. `/usr/local -> ../opt/local` anzulegen. Da `/opt` normalerweise für jede Zone separat ist, besteht hier für jede Zone Schreibrecht. Wichtig ist hier, relative links zu benutzen, damit auch ein Zugriff aus der globalen Zone auf z.B. `/zones/zone1/root/usr/local` im richtigen Verzeichnis ankommt.

Die Software der Companion DVD installiert sich nach `/opt/sfw`. Wenn diese zentral installiert werden soll, wäre `/opt/sfw` zusätzlich als `inherit-pkg-dir` festzulegen.

4.1.7.5. ZFS in einer Zone

[ug] Mit Solaris 10 6/06 ist erstmals das neue Filesystem ZFS verfügbar. Da ZFS nicht auf einem special device basiert, das gemountet wird, ist ein Mechanismus im `zonecfg`-Kommando implementiert, mit dem man ein ZFS Filesystem an eine Zone weitergeben kann. Dies geht mit dem `zonecfg`-Unterkommando `add dataset`. Das ZFS Filesystem ist dann in der Zone sichtbar und kann dort verwendet und (limitiert) administriert werden.

Bei ZFS können Attribute eines Filesystems gesetzt werden, die sofort aktiv werden. Hier sind die Attribute `quota` (maximal belegbarer Platz) und `reservation` (vorreservierter Platz für das Filesystem) interessant. Wenn ein Filesystem mit `add dataset` einer Zone übergeben ist, dann kann der Administrator der globalen Zone die Attribute `quota` und `reservation` setzen. Der Administrator der lokalen Zone kann diese Parameter nicht verändern, um keinen Zugriff auf Ressourcen anderer Zonen zu nehmen.

ZFS erlaubt es, Filesysteme hierarchisch zu unterteilen. Dies gilt auch für die einer Zone weitergegebenen ZFS Filesysteme, die vom Administrator der Zone weiter unterteilt werden können. Sind die Attribute `quota` und `reservation` für das übergebene Filesystem gesetzt, gilt die Limitierung auch in der Zone weiter. Somit sind die in der Zone nutzbaren Ressourcen beschränkt.

4.1.8. Netzwerk-Konzepte

4.1.8.1. Vorbereitung Netzwerk für lokale Zonen

[ug] Eine Netzwerk-Adresse ist bei der Konfiguration einer Zone nicht zwingend vorgeschrieben. Dienste in einer Zone sind von außerhalb nur über das Netzwerk zu erreichen, daher ist in der Regel mindestens eine Netzwerk-Adresse pro Zone notwendig (konfigurierbar mit `zonecfg`).

Die Netzwerk-Adresse einer Zone wird als virtuelles Interface auf ein beliebiges physikalisches Interface gelegt. Dazu muß entweder die globale Zone ebenfalls eine Adresse auf dem Interface besitzen oder das Interface muß mit `ifconfig <interface> plumb` aktiviert sein.

Eine Zone kann mehrere Netzwerk-Adressen auch auf unterschiedlichen Interfaces erhalten. Routen dafür lassen sich aber nur in der globalen Zone eintragen, da Stand heute nur ein Netzwerk-Stack gemeinsam für alle Zonen genutzt wird.

Mit dem Projekt *Crossbow* ist unter anderem geplant, optional einer Zone eine eigene IP-Stack Instanz geben zu können.

4.1.8.2. Verwaltung der Netzwerk-Adressen

[ug] DHCP ist für die Adressen von Zonen nicht möglich, da DHCP auf der HW-Adresse des Netzwerk-Interfaces basiert und Zonen eine virtuelle Adresse auf einem shared Netzwerk-Interface nutzen. Sie haben daher die gleiche MAC-Adresse wie das Interface in der globalen Zone. Die Verwaltung der Netzwerk-Adressen für Zonen muß daher auf andere Weise erfolgen.

Es sind prinzipiell die folgenden Arten der Verwaltung möglich:

- Manuelles Führen einer Liste.
Bei der Zonen-Konfiguration muss daher die IP-Adresse in der Liste markiert werden.
- Vor-Definieren der IP-Adressen mit dem Zonen-Namen im Name-Service.
Bei der Zonen-Konfiguration kann ein Script daher die IP-Adresse der Zone automatisch ermitteln, wenn der IP-Name aus dem Zonen-Namen berechnet werden kann (Kochbuch).
- Wenn auf einem System viele Zonen eingerichtet werden sollen, dann ist es empfehlenswert einen ganzen Bereich vorab für das System zu allokatieren, bei dem die Netzwerk-Adresse gleich dem vorgesehenen Zonnennamen ist. Damit ist eine eindeutige Zuordnung gewährleistet.
- Andere Integration in die IP-Namensvergabe des Ziel-Environments.
Hier ist eine Integration in die Prozesse der IP-Vergabe des Unternehmens gemeint.

4.1.8.3. IP-Stack und Routing zwischen Zonen

[dd] Jede Zone verfügt über mindestens eine eigene IP-Adresse und eigene TCP- und UDP-Portnummern. Anwendungen, die in Zonen benutzt werden, binden sich auf die in der Zone sichtbaren IP-Adressen und nutzen diese auch als Absenderadressen.

Wenn Zonen sich durch entsprechende Adressvergabe in unterschiedlichen logischen Subnetzen befinden und eine Kommunikation der Zone mit anderen Netzen notwendig ist, muß für jede Zone eine Defaultroute existieren. Diese wird von der globalen Zonen aus gesetzt, da die Routing Tabelle sich im TCP/IP-Stack befindet, der zwischen allen Zonen shared wird.

Ist so eine Defaultroute für eine Zone gesetzt, findet eine Inter-Zonen-Kommunikation (lokale Zone zu lokale Zone) direkt über den gemeinsam genutzten TCP/IP-Stack statt und verläßt nicht das physikalische Interface. Wenn diese Inter-Zonen-Kommunikation unterbunden werden soll, müssen sogenannte reject-routes eingesetzt werden, die jegliche Kommunikation zwischen zwei IP-Adressen im IP-Stack unterbinden.

Ist eine gezielte Kommunikation zwischen zwei lokalen Zonen notwendig, die aber z.B. über einen externen Router, Load Balancer oder Firewall geführt werden soll, müssen NAT-fähige Router eingesetzt werden. Entsprechende Setups werden im Abschnitt Kochbücher diskutiert.

Die Netzwerk-Trennung der lokalen Zonen findet auf logischer IP-Ebene statt, trotzdem teilen sich alle Zonen und alle genutzten Netzwerk Interfaces einen TCP/IP-Stack. Für eine weitergehende Netzwerk-Trennung der Zonen ist eine Separierung des TCP/IP-Stacks notwendig. In diesem Falle konfiguriert jede Zone ihre Netzwerkinterfaces in einer separaten IP-Stack Instanz. Das Projekt Crossbow unter <http://www.opensolaris.org/os/project/crossbow/> soll in einem späteren Solaris Update diese Funktionalität zur Verfügung stellen.

4.1.8.4. Zonen und Limitierungen im Netzwerk

[dd] Zonen haben im Zusammenhang mit Netzwerkkonfigurationen verschiedene Limitierungen, die im allgemeinen darauf zurückzuführen sind, daß die Zonen einen gemeinsamen TCP/IP-Stack benutzen.

Netzwerkfunktionalitäten, die nur in der globalen Zone nutzbar sind

- NFS-Server
- DHCP-Server
- DHCP-Client
- RAW-Netzwerkverkehr
- Snoop
- NCA

Netzwerkfunktionalitäten, die in der globalen Zone konfiguriert werden müssen, aber in der lokalen Zone zur Verfügung stehen:

- Default Router für lokale Zonen
- IPQoS
- IKE für IPsec
- IPMP
- IPFilter

4.1.9. Separate Name Services in Zonen

[ug] Name Services umfassen unter anderem die hosts-Datenbank, die Userids (passwd, shadow) und werden mit der Datei `/etc/nsswitch.conf` konfiguriert, die in der lokalen Zone unabhängig von der globalen Zone existiert. Die Name Services sind daher in lokalen Zonen unabhängig von den globalen Zonen definiert. Die wichtigsten Aspekte dazu werden in diesem Abschnitt behandelt.

Wenn man der Empfehlung von anderer Stelle in dem Dokument folgt, dass man in der globalen Zone keine Applikationen laufen läßt, dann muß auch die globale Zone nicht in NIS oder LDAP eingefügt werden. Dies limitiert weiter den Zugang von außen und vermindert die Abhängigkeit der globalen Zone von anderen Rechner (Nameservice Server).

4.1.9.1. hosts-Datenbank

[ug] Die Rechner, die mittels Namen ansprechbar sein sollen müssen hier eingetragen werden. Es findet keine automatische Kopie der `/etc/hosts` von der globalen Zone statt, wenn die Zone installiert wird (ganz im Sinne, dass in der lokalen Zone eine eigene OS Umgebung existiert). Die bessere Alternative ist natürlich, dass man einen Name-Service wie NIS, NIS+, DNS oder LDAP verwendet. Bei einer automatischen Installation kann das über eine `sysidcfg`-Datei eingestellt werden.

4.1.9.2. User-Datenbank (passwd, shadow, user_attr)

[ug] Die User-Einstellungen in den lokalen Zonen können wie bei einem separaten Rechner durch einen Nameservice ergänzt werden. Wichtig ist, dass ja in der Zone andere Userid-Nummern gelten können. Eine Kopie der Dateien von der globalen Zone wird nicht empfohlen, ein Nameservice wie NIS oder LDAP ist besser geeignet.

4.1.9.3. Services

[ug] Die `/etc/services` oder der entsprechende Name-Service muß ebenfalls an die in der Zone laufenden Applikationen angepasst werden.

4.1.9.4. Projekte

[ug] Um in einer lokalen Zone Ressource-Management mit Fair-Share-Scheduler oder Extended Accounting lokal zu betreiben, muß die entsprechende Name-Service Datenbank in `/etc/project` oder der entsprechende Name-Service in der Zone angepasst werden.

4.2. Paradigmen

Paradigmen sind Design Regeln zum Aufbau von Zonen. Je nach Anwendung muß entschieden werden, welche davon zur Anwendung kommen sollen.

4.2.1. Delegation von Admin Rechten an die Applikations-Abteilung

[ug] Die Administration bezüglich einer Applikation kann an die für die Applikation zuständige Abteilung delegiert werden. Durch die Zonen-Isolation kann der Root-Administrator nur Ressourcen beeinflussen, die der Zone zugeordnet sind. Dies gilt auch für andere definierte privilegierte User in der Zone (siehe Process Privileges, *ppriv*).

- Dediziert ist das Netzwerk (Stand 10/2006) nur in der globalen Zone konfigurierbar.
- Filesysteme können von der globalen Zone festgelegt sein (*zonecfg add fs*)
- Die Administration der Filesysteme kann dem Administrator der lokalen Zone übergeben werden (*zonecfg add device*).

4.2.2. Applikationen nur in lokalen Zonen

[ug] Wenn lokale Zonen für Applikationen genutzt werden empfiehlt es sich, die globale Zone nicht auch für Applikationen zu nutzen.

- Nur dann ist es möglich, daß die Administration der Hardware des Rechners rein bei den Plattform-Administratoren der globalen Zone bleibt.
- Plattform-Administratoren sind die Administratoren, die die globale Zone administrieren. Sie haben den Zugriff auf die Hardware (Gehäuse, Netzkabel, Festplatten) und führen die Solaris Installation in der globalen Zone durch. Das Patching des Kernels und der Reboot des Gesamtsystems obliegt ebenfalls dem Plattform-Administrator.
- Es ist nicht notwendig, dem Applikations-Admin Zugang zu der globalen Zone zu geben.
- Sofern der Applikations-Admin root-Zugang braucht, kann er das Root-Passwort der lokalen Zone seiner Applikation erhalten. Er muss dann allerdings in Absprache mit dem Betrieb die Verantwortung für die Verfügbarkeit der Applikation übernehmen.
- Requests für Storage/Network erfolgen über die Plattform-Administration, die nach Freigabe durch die Storage- oder Netzwerk-Administration (sofern separat) die Ressourcen der entsprechenden lokalen Zone zuordnen kann.

In der globalen Zone werden nur noch systemnahe Applikationen installiert, die zum Management, der Überwachung oder zu Backup-/Restore-Zwecken notwendig sind. Zur Erhöhung der Verfügbarkeit kann auch Cluster-Software eingesetzt werden.

Vorteile:

- Die Zuständigkeiten für System, Applikation und Storage können klar getrennt werden.
- Der root-User Zugriff auf dem Basis-System ist auf die System-Administration eingeschränkt. Die Stabilität ist dadurch besser.

Nachteile:

- Einige Applikationen sind noch nicht explizit für die Nutzung in Zonen freigegeben. In der Regel funktionieren die Applikationen in den Zonen, jedoch sind sie nur noch nicht vom Hersteller der Applikation zertifiziert worden.

4.2.3. Eine Applikation pro Zone

[ug] Ein weiteres Paradigma ist, immer nur eine Applikation pro Zone zu installieren. Der Overhead einer Zone ist sehr gering; im Prinzip werden nur die Applikationsprozesse vom Rest des Systems separiert, indem sie mit der ID der Zone markiert werden. Das wird durch die Zonen-Technologie realisiert.

Die Gewinne durch diese Entscheidung sind sehr hoch:

- Der Administrator der Applikation kann das Passwort für Root in der lokalen Zone erhalten, ohne daß er den Betrieb des gesamten Rechners bei einem Fehler gefährden kann.
- Gerade bei Konsolidierung von vielen Applikationen wird die Anzahl der Benutzer mit Root-Rechten reduziert.
- Wenn man die Applikation automatisch installieren möchte ist dies viel einfacher, weil man ja sicher weiß, das keine andere Applikation da ist und das System verändert hat.
- Abhängigkeiten/Störungen zwischen Applikationen im Filesystem und/oder Konfigurationsdateien entfallen vollständig, wodurch der Betrieb sicherer wird.

4.2.4. Zonen im Cluster

[tf/du] Container können im Cluster als Blackbox Container oder in ausgeformten Ressource Topologien konfiguriert werden. In einem Blackbox Container werden die betriebenen Applikationen nur im Container konfiguriert. Der Cluster kennt sie nicht, sie werden innerhalb des Containers nur von Solaris kontrolliert. Dies führt zu besonders einfachen Cluster Konfigurationen.

Zusätzlich kann man in der Cluster Topologie Applikationen im Container unter Cluster Kontrolle bringen. Hierfür können Standard Agenten oder die Shellskript- oder SMF-Komponente des Sun Cluster Container Agenten verwendet werden. Mischformen zwischen den beiden Konzepten sind jederzeit zulässig. Diese Cluster können in aktiv-passiv Konfiguration oder in aktiv-aktiv Konfiguration betrieben werden.

Aktiv-passiv bedeutet, ein Knoten bedient die konfigurierten Container und der zweite Knoten wartet auf den Ausfall des ersten.

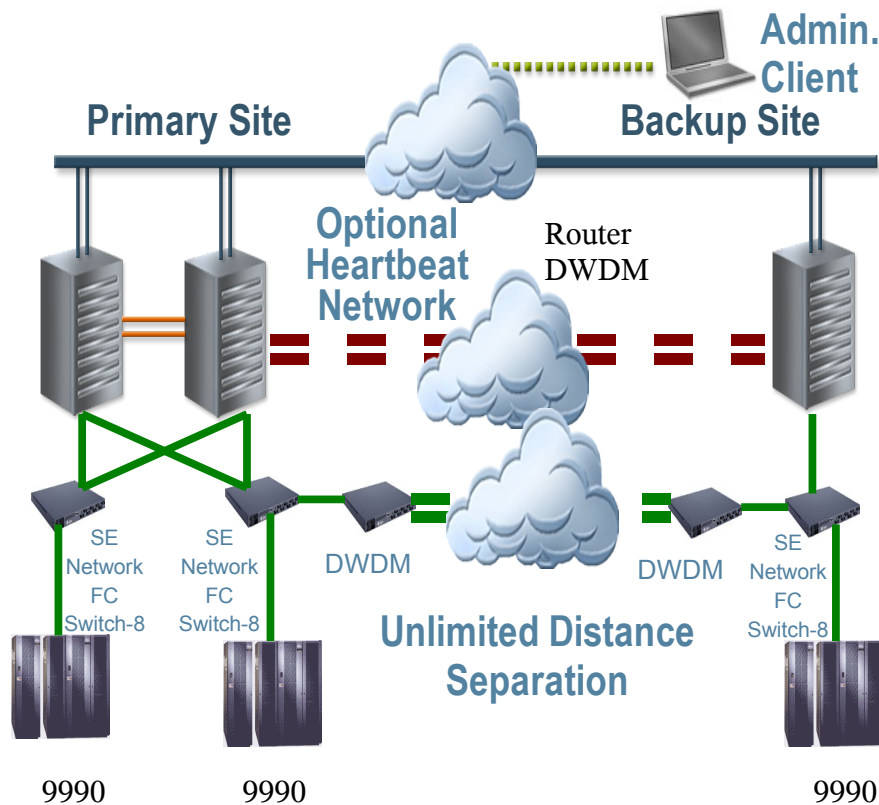
Aktiv-aktiv bedeutet, daß jeder Knoten Container bedient. Reicht die Leistung jedes Knoten nicht für alle Container aus, so ist bei Ausfall eines Knotens eine geringere Performance zu akzeptieren.

Ist die Anforderung Hochverfügbarkeit in einem Rechenzentrum oder Hochverfügbarkeit zwischen zwei Rechenzentren, so genügt Sun Cluster, um Container geplant und im Notfall zwischen den Rechnern zu verschieben. Die Maximale Entfernung zwischen zwei Cluster Knoten ist bei Sun Cluster jedoch zur Zeit auf 400 km begrenzt.

Genügt diese Entfernung nicht oder genügt die Latenz der involvierten Spiegelungstechnik nicht den Performance Anforderungen, so ist hier Sun Cluster Geographic Edition einsetzbar.

Sun Cluster Geographic Edition setzt laufende Cluster in mehreren Rechenzentren voraus. Der verwandte Storage wird über geeignete Techniken vom Rechenzentrum A in das Rechenzentrum B repliziert. Zur Zeit wird Sun StorEdge Availability Suite (AVS) oder Hitachi Truecopy unterstützt, EMC SRDF ist geplant. Sun Cluster Geographic Edition ermöglicht es, die Container vom Rechenzentrum A in das Rechenzentrum B zu verschieben. Fällt ein Rechenzentrum aus, so schlägt Sun Cluster Geographic Edition einen Rechenzentrum-Schwenk vor. Nach Bestätigung durch einen Administrator wird dieser dann auch durchgeführt. Mit den hier beschriebenen Software Produkten lassen sich die Anforderungen von Visualisierung und Flexibilisierung von Containern bis hin zu Disaster Recovery Konzepten voll abdecken.

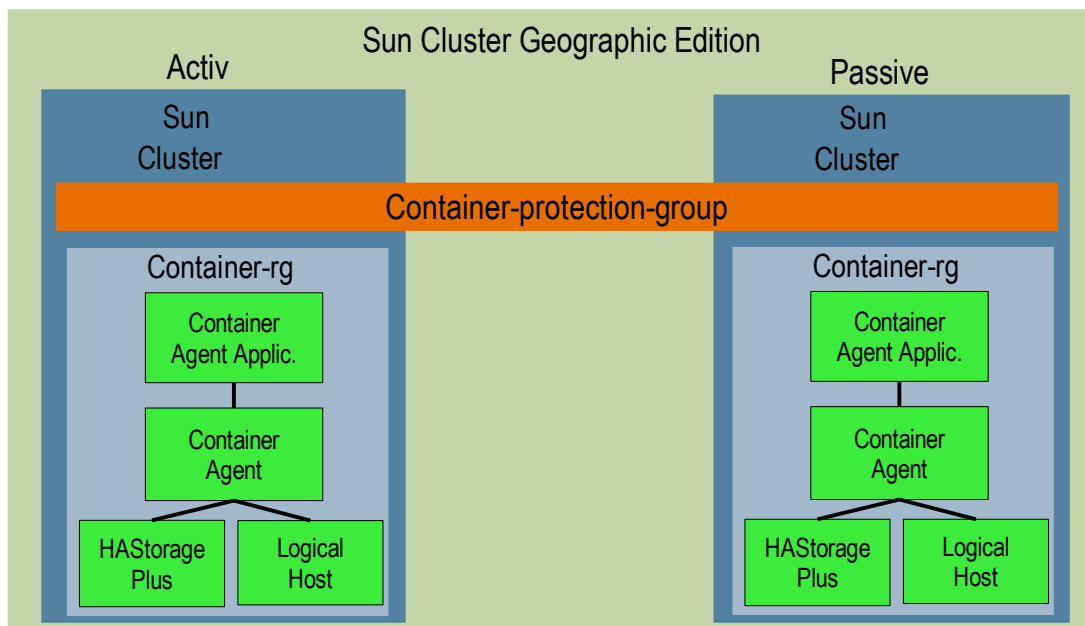
Beispiel:



Zeichnung 20: [tf/du] Typische Konfiguration: Sun Cluster Geographic Edition

Die hier dargestellte Hardware Architektur besteht aus einem zwei Knoten Cluster aus SF6900 Servern im primären Rechenzentrum und einem Ein-Knoten Cluster in dem für Disaster Recovery vorgesehenen Ausweichrechenzentrum. Die verwendeten Storage Systeme sind Sun StorageTek 9990 Systeme und die Datenreplikation erfolgt mittels Truecopy. Die Steuerung erfolgt über Sun Cluster Geographic Edition.

Sollte keine Disaster Recovery gewünscht werden, so kann man jedes Rechenzentrum auch für sich betreiben.



Zeichnung 21: [tf/du] Ressourcegruppen und Protectiongroup Topologie

In der hier dargestellten Ressourcegruppen Topologie verwaltet der aktive Cluster in Hamburg einen Container. Der passive Cluster in München wird für Disaster Recovery Zwecke benutzt. Die Cluster bilden eine Partnerschaft und verwalten die Resourcegruppe in einer sogenannte Protectiongroup.

4.3. Konfiguration / Administration

4.3.1. Manuelle Konfiguration von Zonen mit `zonecfg`

[ug] Zur Konfiguration einer Zone gibt es das Kommando `zonecfg`, siehe Beispiel im Kochbuch.

Die Konfiguration beschreibt lediglich das Verzeichnis (`zonecfg: zonepath`) in dem die Dateien für die Zone abgelegt werden sollen und wie System-Verzeichnisse von der globalen Zone übernommen werden sollen (Kopieren/Loopback-Mount). Als Option kommen noch Netzwerk-Adressen, User-Verzeichnisse, Devices... hinzu.

Jede einmal konfigurierte Zone lässt sich auch als Template für weitere Zonen verwenden, wodurch das Erstellen von Zonen gleichartiger Konfiguration erheblich vereinfacht wird.

Von einer Konfiguration von Zonen durch das Editieren der XML-Dateien unter `/etc/zones` wird abgeraten, da dieses Verfahren sehr fehleranfällig ist und durch Solaris nicht unterstützt wird. Ab Solaris 10 1/06 sind Veränderungen in den XML-Dateien wegen Prüfsummen in `/etc/zones/index` kaum mehr möglich.

4.3.2. Manuelle Installation von Zonen mit `zoneadm`

[ug] Nachdem eine Zone konfiguriert ist, besteht der nächste Schritt in der Installation mit `zoneadm -z <zone> install` (siehe Kochbuch). Dabei werden alle Pakete (Solaris pkg), die nicht zum Kernel des Solaris gehören, in der Zone installiert, wobei die Dateien, die per Loopback-Mount von der globalen Zone zur Verfügung stehen, nicht kopiert werden. Hierbei werden auch die postinstall-Scripte der Pakete für die Zone ausgeführt. Die Zone hat am Ende eine eigene Paket-Datenbank (`/var/sadm/install/*`). Als Quelle für die zu installierenden Pakete der lokalen Zone werden die Pakete der globalen Zone genutzt.

Danach ist die Zone bootbar.

4.3.3. Manuelle De-Installation von Zonen mit `zoneadm`

[ug] Wird eine Zone nicht mehr gebraucht, dann kann man die Zone mit `zoneadm -z <zone> uninstall` deinstallieren. Dies entfernt alle Dateien im Zonepath, auch die, die nicht bei der Installation sondern später entstanden sind.

Die Konfiguration der Zone bleibt bestehen.

4.3.4. Manuelles Entfernen einer installierten Zone mit `zonecfg`

[ug] Um eine Zone zu entfernen, sollte zunächst eine De-Installation durchgeführt werden. Dann kann mit dem Kommando `zoneadm -z zone destroy` die Konfiguration der Zone entfernt werden.

4.3.5. Duplizieren einer installierten Zone

[ug] Zonen können schnell dupliziert werden, wenn die `inherit-pkg-dir` Konfigurationen identisch sind. Dann kann man eine Zone als Template erzeugen und für jede Zone durch Kopieren des Inhalts schnell Duplizieren.

In einem Update von Solaris 10 wird diese Funktion mit zusätzlichen Prüfungen als Unterkommando `clone` ins Kommando `zoneadm` mit eingebunden.

4.3.6. Standardisiertes Erzeugen von Zonen

[ug] Zonen können folgendermaßen erzeugt werden:

- mit dem Kommando `zonecfg` und dann mit `zoneadm` zum Installieren der Zone
- mit dem N1 Grid Console – Container Manager
- mit Webmin, einem freien Management-Werkzeug, das bei Solaris 10/OpenSolaris beige-packt ist (Solaris Zones Modul: neuester Stand `zones.wbm.gz` downloadbar bei <http://www.webmin.com>)
- per scripts, die `zonecfg` und `zoneadm` aufrufen

Der Container Manager und Webmin eignet sich für die Admins, die selten Zonen einrichten.

Mit `zonecfg` kann man Zonen gemäß eigenen Standards erzeugen, indem man Zonen als Templates einrichtet. Eine Zone kann dann mit dem `zonecfg` Unterkommando `create -t <template>` als Template eingesetzt werden.

In der Regel werden einige Richtlinien lokal festgelegt, zum Beispiel:

- Welche Filesysteme von der globalen Zone geerbt werden sollen (`inherit-pkg-dir`).
- Filesysteme mit Software, die jede Zone nutzen können soll.
- shared Verzeichnisse von der globalen Zone.
- Ob die Zone bei Reboot des Rechners automatisch gestartet werden soll.
- Der Ressource Pool und andere Ressource Einstellungen für die Zone.

Diese Standard-Einstellungen lassen sich als Templates ablegen und wieder benutzen. Man muss jedoch immer noch die IP-Adresse und das Zone-Directory mit `zonecfg` konfigurieren, sowie beim ersten Booten der Zone den Rechnernamen (`/etc/nodename`), die Zeitzone und den Name-Service konfigurieren.

Denkbar ist hier auch die Nutzung des N1 Service Provisioning System (N1SPS) mit dem eine automatisierte Erzeugung von Zonen möglich ist.

4.3.7. Automatische Konfiguration von Zonen per Script

[dd] Zonen können automatisch per Script erzeugt und konfiguriert werden. Die Erzeugung von Zonen folgt vorher festgelegten Policies. Dazu sind die folgenden Schritte im Script auszuführen:

- Das `zonecfg`-Kommando wird von einem Script aufgerufen, das die IP-Adresse, Zone-Directory, Filesysteme, Ressource-Pool nach Policies setzt (Kochbuch), zum Beispiel nach:
 - IP-Adresse ist die Adresse des IP-Namens von `z-<zone>`
- Das Zone-Directory wird für diese Klasse von Zonen z.B. auf `/export/zone2/<zone>` gesetzt.
- Die Einstellungen, die die Zone direkt nach dem Booten anfordert, lassen sich mit einer Datei namens `sysidcfg` vornehmen, die vor dem Booten der Zone in das `/etc`-Verzeichnis der Zone gestellt wird (`<zonepath>/root/etc`). Mehr Details siehe Kochbuch.

4.3.8. Automatisiertes Provisionieren der Services

[dd] Mit der automatischen Konfiguration von Zonen per Script und der Festlegung genau einen Service pro Zone einzusetzen, kann der Prozeß der Service-Provisionierung ebenfalls automatisiert werden. Dazu kann z.B. das Service Provisioning System (N1SPS)

http://www.sun.com/software/products/service_provisioning/index.xml oder ein eigenes Script verwendet werden. Wenn die Provisionierung des Services gut automatisiert ist, läßt sich so ein Service sehr schnell auf vielen Zonen einsetzen. Die folgenden Schritte sind notwendig:

- Installation der Software
- Erzeugung oder Kopieren der Applikationsdaten
- Modifikation der Zone um die Erfordernisse der Software abzudecken
 - User-Accounts, Umgebungs-Einstellungen, log-Files
- Start des Services organisieren
 - smf-service oder rc*-Script installieren

Zur Entfernung bzw. Bewegung eines Services in eine andere Zone sind die entsprechende Schritte zum Export des Services und der Daten des Services vorzusehen.

4.4. Lifecycle Management

4.4.1. Patchen eines Systems mit lokalen Zonen

[dd] Um ein System mit installierten lokalen Zonen zu patchen, werden die Patches mit *patchadd* in der globalen Zone eingespielt. Hierbei wird die Patchprozedur wie üblich für die globale Zone ausgeführt und anschließend der Reihe nach für jede lokale Zone. Dazu muß die Zone laufen.

- Erfordert der Patch die Installation im Single-User Modus, wird die Zone in den Single-User Modus gebracht, der Patch installiert und die Zone wieder hochgefahren.
- Ist die Zone installiert, aber angehalten, wird sie für die Installation des Patches kurzzeitig in den Single-User Modus hochgefahren.

Dieses Verfahren betrifft alle installierten Zonen (Zustand *installed*) und kann bei vielen einzuspielenden Patches und vielen vorhandenen Zonen sehr viel Zeit in Anspruch nehmen (mehrere Stunden). Um hier etwas Zeit zu sparen empfiehlt es sich, die globale Zone und alle installierten lokalen Zonen in den single-user Mode zu booten und dann gemeinsam zu patchen.

Bemerkungen:

- Bis zur Behebung von **Bug 6188748** müssen die zu installierenden Patches zunächst auf ein lokales Filesystem kopiert werden, bevor *patchadd* aufgerufen werden kann.
- Sollen Patches nur in eine bestimmte Zone installiert werden, z.B. bei Whole-root Zonen zum Testen von Patches, können Patches mit *patchadd -G* in eine Zone installiert werden.
- Generell wird empfohlen, in den Zonen gleiche Patchstände des Betriebssystems zu betreiben. Lediglich für die benutzte Anwendungssoftware sollte werden, in Zonen unterschiedliche Patch-Ständen oder Versionen zu benutzen.

4.4.2. Neuinstallation und Service Provisionieren statt Patchen

[dd] Das Patchen von Zonen führt durch das Booten der Zonen in den Single-User Mode zum Ausfall des bereitgestellten Services. Dieser Service-Ausfall kann je nach Anzahl der Zonen und Anzahl zu installierender Patches unterschiedlich lang sein. Um diesen Zeitraum zu verkürzen und eine Vorhersagbarkeit der Service-Ausfallzeit zu erreichen, kann der Weg der Neuinstallation anstatt individuelles Patchen gewählt werden.

Zonen können in fast beliebiger Zahl schnell erzeugt werden. Die Zonen-Installation ist erheblich schneller als die Neu-Installation eines Rechner oder das individuelle Patchen von Zonen.

Daher hat man einen Geschwindigkeitsvorteil, wenn man anstatt zu patchen Zonen auf einem Zielsystem neu erzeugt. Diese haben so automatisch den aktuellen Patchstand des Zielsystems. Danach werden die Anwendungsdaten und ggf. die Anwendung von der existierenden Zone in die neue Zone bewegt. Für diesen Zeitraum ist ein Service-Ausfall einzukalkulieren, der aber im Vergleich zur Standard-Patch-Methode vorhersagbar und kurz ist.

Voraussetzung für dieses Verfahren ist, daß die Applikation und die Daten zwischen Zonen bewegt werden können.

4.4.3. Patch-Update durch Flash und Live Upgrade

[ug] Die Idee hier ist, die Ausfallzeiten beim Installieren von Patches zu reduzieren, indem ein Ersatzrechner mit dem neuen Patchstand parallel zum Betrieb vorbereitet wird:

- Wenn ein neuer OS-Patchstand benötigt wird, dann wird ein Rechner ähnlicher Größe mit dem benötigten Patchstand vorbereitet. Die Zonen des Produktionssystems werden auf dieses System dupliziert (Kopie des Baumes + Anpassung */etc/zones/index*, bzw.; mit *zoneadm attach* (zur Zeit nur in OpenSolaris)).
- Auf diesem System werden nun die Patches eingespielt und die funktionalen Tests durchgeführt.
- Danach kann dieses System in einem Flash-Archiv gesichert werden und auf einem alternativen Boot-Environment des Produktionssystems mittels Live Upgrade installiert werden.
- Letzte Änderungen in den Zonen können übernommen werden bevor man das alternative Bootenvironment testweise hochfährt. Dazu kann man das in Solaris 10 vorhandene Tool *bart* verwenden, das Veränderungen in Filesystemen anzeigen kann.

Alternativ kann man das Testsystem als neues Produktionssystem in Betrieb nehmen.

Eine ähnliche Vorgehensweise ist in der Industrie unter dem Namen rotierendes Integrations- und Testsystem üblich.

4.4.4. Backup und Recovery von Zonen

[dd] Zonen können einfach mit Backuptools aus der globalen Zone gesichert werden. Die Sicherung kann die Anwendungsdaten der Zone mit enthalten oder diese separat sichern. Für die Sicherung von Zonen sind die folgenden Dinge notwendig:

- Sicherung der Zonekonfiguration mit
`zonecfg -z <zone1> export -f <zone1>.zonecfg` oder direkt der Datei
`/etc/zones/<zone>.xml`
- u.U. Sicherung der zugehörigen Zeile in `/etc/zones/index`
- Sichern des Verzeichnisbaumes z.B. `/zones/zone1` mit Backup Tools

Beim Recovery sind die Schritte in umgekehrter Richtung auszuführen.

- `/zones/zone1` erzeugen
- `chmod 700 /zones/zone1` (Standard für Zonen-Verzeichnisse)
- u.U. Unter mounts herstellen, wenn z.B. `/zones/zone1/root/var` ein eigenes Filesystem sein soll
- Backup einspielen
- ggf. `<zone1>.zonecfg` anpassen (neuer Pfad, andere Mountpunkte)
- mit `zonecfg -z <zone1> -f <zone1>.zonecfg` die Zonenkonfiguration erstellen
- im `/etc/zones/index` den Status der Zone von `configured` in `installed` ändern

Beim Sichern des Zoneverzeichnisses muß beachtet werden, daß sparse-root Zonen `inherit-pkg-dir` enthalten. Verzeichnisse wie z.B. `/usr`, `/sbin` pro Zone mitzusichern, wäre sicherlich in den meisten Fällen nicht sinnvoll. D.h. in der Backup Konfiguration sollte hinterlegt werden, daß `inherit-pkg-dir` nicht mitzusichern sind.

Die Sun StorEdge Enterprise Backup Software (EBS) kann z.B. über `.nsr`-Dateien in Verzeichnissen derart gesteuert werden, daß solche Verzeichnisse nicht mitgesichert werden.

Veritas Netbackup (NBU) verfügt über eine zentrale Konfiguration. D.h. in diesem Falle sind entsprechende Konfigurationen zentral mit globalen Rules einzustellen. Bei einem späteren geplanten Umzug von Zonen und einem Backup/Restore aus der globalen Zone, muß diese Konfiguration entsprechend systemübergreifend ausgelegt werden.

Durch den Einsatz eines Backup-Clients in der globalen Zone für mehrere lokale Zonen, ist die Einsparung von Lizenzgebühren möglich. Zur Sicherung von besonderen Anwendungsdaten (z.B. Datenbanken) muß u.U. der Backup-Client oder das Backup-Modul direkt in der lokalen Zone laufen (z.B. Sicherung von Oracle mit RMAN).

4.4.5. Migration (Umzug) einer Zone mit `zoneadm detach/attach`

[dd] Seit OpenSolaris Build 36 ist die Funktionalität der Migration von installierten lokalen Zonen von einem System zu einem anderen System enthalten. Das Feature der Zonen Migration wird in einem zukünftigen Release von Solaris 10 enthalten sein.

Zur Migration einer Zone wird die Zone gestoppt, mit seinen Daten migriert und auf dem Zielsystem neu gestartet. Auf beiden Systemen müssen die installierten Packages und Patches identisch sein.

4.5. Management und Monitoring

4.5.1. Überwachung der Zonen-Auslastung

[ug] Mit dem Kommando *prstat* (seit Solaris 8) sieht man die Prozesse mit der höchsten Auslastung ähnlich dem von auch anderen Plattformen bekannten Kommando *top*. Mit Solaris 10 hat das Kommando *prstat* eine Option *-z* erhalten, mit der man summarisch die Auslastung für jede Zone (auch die globale Zone) mit anzeigen kann. Dadurch ist der Status der Zonen leicht überwachbar.

4.5.2. Extended Accounting mit Zonen

[ug] Mit Solaris 9 wurde das Extended Accounting eingeführt. Im Gegensatz zu dem traditionellen Unix Accounting (das es nach wie vor gibt), kann man bei Extended Accounting die Datenfelder, die protokolliert werden sollen definieren. Mit Solaris 10 steht als weiteres optionales Datenfeld auch der Name der Zone zur Verfügung.

Daher kann man in der globalen Zone das Extended Accounting so konfigurieren, daß mit jedem Accounting Datensatz auch der Zonen-Name mitgeschrieben wird. Die Daten (z.B. verbrauchte CPU-Zeit) lassen sich daher nach Zonen getrennt summieren und einer Kapazitätsplanung oder Abrechnung zuführen.

4.5.3. Auditing der Operationen in der Zone

[dd] Audit kann zur Überwachung von Systemaktivitäten eingesetzt werden. Das System-Audit erfolgt in der globalen Zone. Audit kann auch zur Überwachung der Aktivitäten einer lokalen Zone konfiguriert werden oder durch den Administrator einer Zone zur Überwachung der Nutzerprozesse einer Zone. Audit in lokalen Zonen kann zwar die Aktivitäten vom Kernel nicht überwachen, aber User-Aktivitäten innerhalb der Zonen.

4.5.4. DTrace von Prozessen in einer Zone

[dd] DTrace ist Stand heute (Solaris 10 6/06) nur in der globalen Zone nutzbar.

DTrace Scripte können um die Variable *zonename* erweitert werden, um z.B. nur Systemcalls einer Zone zu tracen.

```
global# dtrace -n 'syscall:::/zonename=="sparse"/
{@[probefunc]=count()}'
```

oder den I/O von Zonen zu messen.

```
global# dtrace -n io:::start'@[zonename] = count()'
```

Ab OpenSolaris Build 37 und in einer folgenden Release von Solaris 10, wird DTrace auch innerhalb einer lokalen Zone nutzbar sein. Das wird möglich, wenn die Privilegien *dtrace_proc* und *dtrace_user* einer Zone zugewiesen werden können. So wird innerhalb einer Zone das Tracen von Prozessen der Zone selbst möglich. Ausgeschlossen bleibt das Tracen des Kernels und von Prozessen der globalen Zone.

5. Kochbücher

Das Kapitel Kochbücher stellt die Realisierung der konzeptionellen Best Practices an konkreten Beispielen dar.

5.1. Installation und Konfiguration

5.1.1. Konfigurationsdateien

[dd]

Datei: `/etc/zones/index`

listet alle konfigurierten Zonen eines Systems auf, inkl. Zonenstatus und Rootverzeichnis.

Beispiel:

```
# DO NOT EDIT: this file is automatically generated by zoneadm(1M)
# and zonecfg(1M). Any manual changes will be lost.
#
global:installed:/
sparse:installed:/zones/sparse:ae343d81-3439-cd4d-ff52-f110feeff8d2
whole:configured:/zones/whole
```

Datei: `/etc/zones/<zonenname>.xml`

Enthält die Konfiguration einer Zone <zonenname>

Beispiel: `/etc/zones/sparse.xml`

kann auch als Template benutzt werden (`zoneadm: create -t <zone>`)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE zone PUBLIC "-//Sun Microsystems Inc//DTD Zones//EN"
"file:///usr/share/lib/xml/dtd/zonecfg.dtd.1">
<!--
    DO NOT EDIT THIS FILE. Use zonecfg(1M) instead.
-->
<zone name="sparse" zonepath="/zones/sparse" autoboot="false"
pool="sparse">
  <inherited-pkg-dir directory="/lib"/>
  <inherited-pkg-dir directory="/platform"/>
  <inherited-pkg-dir directory="/sbin"/>
  <inherited-pkg-dir directory="/usr"/>
  <inherited-pkg-dir directory="/opt"/>
  <network address="192.168.1.20" physical="bge0"/>
</zone>
```

Datei: `/etc/zones/SUNWblank.xml`

legt die Standardvorgabe fest, wie blank-Zones (Kommando `zoneadm create -b`) erzeugt werden

```
<!DOCTYPE zone PUBLIC "-//Sun Microsystems Inc//DTD Zones//EN"
"file:///usr/share/lib/xml/dtd/zonecfg.dtd.1">
<zone name="blank" zonepath="" autoboot="false">
</zone>
```

Datei: `/etc/zones/SUNWdefault.xml`

legt die Standardvorgabe fest, wie sparse-root Zonen (default) erzeugt werden

```
<!DOCTYPE zone PUBLIC "-//Sun Microsystems Inc//DTD Zones//EN"
"file:///usr/share/lib/xml/dtd/zonecfg.dtd.1">
<zone name="default" zonename="" autoboot="false">
  <inherited-pkg-dir directory="/lib"/>
  <inherited-pkg-dir directory="/platform"/>
  <inherited-pkg-dir directory="/sbin"/>
  <inherited-pkg-dir directory="/usr"/>
</zone>
```

Datei: `/var/sadm/install/gz-only-packages`

listet alle Packages, die in der globalen Zone mit `pkgadd -G` installiert wurden und nicht für die Erzeugung von lokalen Zonen benutzt werden.

5.1.2. Relevante OS-Befehle

[dd] Die Einführungen von Zonen in Solaris 10 führte zu einer Reihe von neuen und modifizierten Kommandos:

neue Befehle	Beschreibung
<code>df(1M)</code>	-Z zeigt Mounts der Zonen an
<code>ifconfig(1M)</code>	zone <zonename> weist eine virtuelle IP-Adresse einer Zone zu -zone gibt eine virtuelle IP-Adresse der globalen Zone zurück -Z führt das Kommando für alle Interfaces in der Zone aus
<code>ipcrm(1)</code>	-z <zonename> löscht eine Message Queue, ein Semaphore Set oder eine Shared Memory ID einer Zone
<code>ipcs(1)</code>	-z <zonename> zeigt Interprozess Kommunikations-Parameter einer Zone an
<code>patchadd(1M)</code>	-G installiert einen Patch nur in die aktuelle Zone
<code>pgrep(1)</code>	-z <zoneid> wendet pgrep auf Prozesse in Zone zoneid an
<code>pkill(1)</code>	-z <zoneid> wendet pkill auf Prozesse in Zone zoneid an
<code>pkgadd(1M)</code>	-G installiert ein Package lediglich in die aktuelle Zone
<code>poolbind</code>	-i zoneid <zonename> bindet alle Prozesse der Zone <id> oder <zonename> an den Ressource Pool
<code>prctl</code>	-i zoneid <zonename> Anzeigen und Modifizieren von Ressource Controls der Prozesse einer Zone
<code>priocntl(1)</code>	-i zoneid <zoneidlist> zeigt die Scheduling Parameter der Prozesse einer Zonen zoneid
<code>prstat</code>	-z <zonename> zeigt die Prozess-Statistiken der Prozesse von <zonename> an -Z zeigt Informationen zusammen über Prozesse und Zonen an

neue Befehle	Beschreibung
<i>ps</i>	-o pid,...,zone zone existiert als Formatfeld für konfigurierbare Ausgaben -z <zonename> zeigt nur Prozesse von <zonename> an -Z Anzeige des Zonenamens zu dem der Prozess gehört
<i>renice(1)</i>	-i zoneid <zoneidlist> modifiziert die Priorität laufender Prozesse in <zoneidlist>
<i>zoneadm(1M)</i>	Administration von Zonen (Installation/Uninstallation, Boot/Reboot/Halt, List)
<i>zonecfg(1M)</i>	Konfiguration der Zone
<i>zlogin(1)</i>	Zonenlogin von der globalen Zone aus -C Verbindung mit der Console der Zone
<i>zonename(1)</i>	Ausgabe des Namens der aktuellen Zone

Tabelle 4: [dd] Relevante OS-Befehle

5.1.3. Rootplattenlayout

[dd] Die nachfolgende Tabelle stellt beispielhaft ein Rootplattenlayout eines Systems mit einer lokalen Zone dar. Die Annahmen über die Größe der Filesysteme stellen Erfahrungswerte und Beispiele dar und können je nach Umfang der Softwareinstallation und lokalen Erfordernissen variieren.

Pfad	Größe des Filesystems (Beispiel)	Bemerkung
/	8 GB	/opt für die globale Zone kann hier mit enthalten sein
swap	2 GB	je nach Hauptspeichergröße, Anforderung und Erfahrung In das Swap-Device ist auch das Dump-Device konfiguriert.
/var	6 GB	je nach Anforderung und Erfahrung
/var/crash	4 GB	kann u.U. auch innerhalb von /var liegen Größe je nach Hauptspeichergröße und Erfahrung
ABE (/)	8 GB	Alternatives Boot-Environment (ABE) für /
ABE (/var)	6 GB	ABE für /var /var/crash und swap können zwischen beiden BE shared werden
metadb	100MB	SVM Metadatenbank, wenn der Solaris Volume Manager eingesetzt wird
/zones	2 GB	Root-Filesystem aller Zonen, wenn mehrere Zonen sich ein Filesystem teilen
/zones/zone1	1 GB	Root-Filesystem exklusiv für Zone1
/zones/root	1 GB	Root-Filesystem für mehrere Zonen, wenn mehrere Zonen sich ein Filesystem teilen und /var der lokalen Zone separat sein soll. Weitere Aufteilung z.B. in /zones/root/zone2, /zones/root/zone3,etc.
/zones/var	1GB	Var-Filesystem für mehrere Zonen, wenn mehrere Zonen sich ein Filesystem teilen sollen, aber /var separiert wird und getrennt überwacht wird. Weitere Aufteilung z.B. in /zones/var/zone2, /zones/var/zone3, etc.
/anwendung1	1 GB	enthält die Anwendung, die innerhalb der Zone laufen soll /anwendung1 wird z.B. in der Zone an /opt/anwendung1 gemountet, ist vom rootfs der Zone separiert und kann also unabhängig von der Zone verschoben werden.

Tabelle 5: [dd] Rootplatten-Layout

5.1.4. Konfiguration einer Sparse-root Zone: Erforderliche Aktionen

[dd] Aus Sparse-root Zonen können nur mit Neuinstallation Whole-root Zonen erzeugt werden. Also ist vor Beginn der Konfiguration auszuwählen, welche Art der Zone erzeugt werden soll.

Entscheidungshilfen dazu sind bereits in diesem Dokument diskutiert worden. Die folgenden Angaben sind für die Konfiguration erforderlich:

- Festlegung eines Zonennamens
- Festlegung des root-Verzeichnisses der Zone
- Zuordnung eines Interfaces
- Zuordnung einer IP-Adresse
- Hier wird `/opt` für die lokale Zone kopiert, so daß in der Zone lokal Software nach `/opt` installiert werden kann.

```
global# zonecfg -z zone1
zone1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/zones/zone1
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=bge0
zonecfg:zone1:net> set address=192.168.1.1/24
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> info
zonepath: /zones/zone1
autoboot: false
pool:
inherit-pkg-dir:
    dir: /lib
inherit-pkg-dir:
    dir: /platform
inherit-pkg-dir:
    dir: /sbin
inherit-pkg-dir:
    dir: /usr
net:
    address: 192.168.1.1/24
    physical: bge0
zonecfg:zone1> exit
global#
```

5.1.5. Konfiguration einer Whole-root Zone: Erforderliche Aktionen

[dd] Whole-root Zonen enthalten keine `inherit-pkg-dir` und werden mit `zonecfg create` aus der Vorgabedatei `/etc/zone/SUNWdefault.xml` erzeugt. Im Anschluß daran werden mit `remove inherit-pkg-dir` alle `inherit-pkg-dir` entfernt. Von der Benutzung von `zonecfg create -b` wird abgeraten, da so eine Zone mit der Vorgabe aus `/etc/zones/SUNWblank.xml` erzeugt wird, die nicht unbedingt einer whole-root Zone entsprechen muß. Die folgenden Angaben sind für die Konfiguration einer whole-root Zone erforderlich:

- Festlegung eines Zonennamens
- Festlegung des root-Verzeichnisses der Zone
- Zuordnung eines Interfaces
- Zuordnung einer IP-Adresse

```
global# zonecfg -z whole
whole: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:whole> create
zonecfg:whole> set zonename=whole
zonecfg:whole> set zonepath=/zones/whole
zonecfg:whole> remove inherit-pkg-dir dir=/sbin
zonecfg:whole> remove inherit-pkg-dir dir=/usr
zonecfg:whole> remove inherit-pkg-dir dir=/platform
zonecfg:whole> remove inherit-pkg-dir dir=/lib
zonecfg:whole> add net
zonecfg:whole:net> set physical=bge0
zonecfg:whole:net> set address=192.168.1.1/24
zonecfg:whole:net> end
zonecfg:whole> verify
zonecfg:whole> commit
zonecfg:whole> info
zonename: whole
zonepath: /zones/whole
autoboot: false
pool:
net:
    address: 192.168.1.1/24
    physical: bge0
zonecfg:whole> exit
global#
```

5.1.6. Konfiguration einer Zone: Optionale Aktionen

[dd] Nach der Konfiguration der Zone oder auch zu einem späteren Zeitpunkt können bestimmte Änderungen an der Konfiguration vorgenommen werden.

- weitere `inherit-pkg-dir` hinzufügen (das muß vor der Installation einer Zone erfolgen)
- automatischen Boot der Zone bei Systemboot erlauben

```
global# mkdir /opt/sfw
global# zonecfg -z zone1
zonecfg:zone1> add inherit-pkg-dir
zonecfg:zone1:inherit-pkg-dir> set dir=/opt/sfw
zonecfg:zone1:inherit-pkg-dir> end
zonecfg:zone1> set autoboot=true
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> info
zonepath: /zones/zone1
autoboot: true
pool:
inherit-pkg-dir:
    dir: /lib
inherit-pkg-dir:
    dir: /platform
inherit-pkg-dir:
    dir: /sbin
inherit-pkg-dir:
    dir: /usr
inherit-pkg-dir:
    dir: /opt/sfw
net:
    address: 192.168.1.1/24
    physical: bge0
zonecfg:zone1> exit
global#
```

5.1.7. Storage in einer Zone

[dd] Der Zugriff auf Storage von lokalen Zonen aus, kann auf verschiedene Arten erfolgen. Die Zugriffsarten werden durch die folgenden Eigenschaften beschrieben:

- Device
 - Filesystem in der lokalen Zone nutzen
 - Raw-Device in der lokalen Zone nutzen
- Mount
 - durch die globale Zone unabhängig vom Boot der Zone
 - durch die globale Zone beim Booten einer lokalen Zone
 - durch die lokale Zone
- Zugriff
 - read/write
 - readonly
- Sharing von Storage
 - exklusiv für eine lokale Zone
 - shared durch mehrere lokale Zonen oder mit der globalen Zone

Die Konfiguration des Zugriffs auf Storage erfolgt in der globalen Zone oder den lokalen Zonen durch `mount` und Eintrag in `/etc/vfstab` oder durch Konfiguration der lokalen Zone mit `zonecfg add fs` oder `add device`.

5.1.8. ZFS in einer Zone

[ug] Ein ZFS Filesystem kann man einer Zone übergeben, so daß der Administrator der Zone das Filesystem weiterverwenden kann.

- Mit `add dataset` des `zonecfg`-Kommando kann das ZFS der Zone übergeben werden.
- In der Zone kann der Administrator weitere ZFS Filesysteme mit `zfs create` von dem Filesystem ableiten.
- Das in der globalen Zone gesetzte Attribut `quota` kann in der Zone nicht verändert oder überschritten werden.
- Der Administrator der Zone kann jedoch den Mountpoint innerhalb der Zone festlegen.

5.1.9. Konfiguration einer Zone durch Kommando-Datei oder Template

[dd] Gleichartige Zonen können durch die Nutzung von Kommando-Dateien für `zonecfg` oder durch die Nutzung von Templates konfiguriert werden. So ist eine schnelle und fehlerfreie Konfiguration vieler Zonen möglich.

- Nutzung von `zonecfg`-Kommando-Datei
Eine Kommando-Datei wird erstellt, die für die automatische Konfiguration der Zone benutzt wird.
 1. `zonecfg` Kommando-Datei erstellen
 - mit `zonecfg -z <zone> export -f <datei>` einer vorhandenen Zone
 - alternativ: mit einem Texteditor erstellen, ein Kommando pro Zeile
 2. Kommando-Datei zur Konfiguration einer neuen Zone benutzen
 - `zonecfg -z <zone> -f <zone>.zonecfg`
- Nutzung von Templates
Eine neue Zone wird durch die Nutzung einer bereits konfigurierten Zone konfiguriert. Danach werden notwendige Konfigurationsänderungen vorgenommen.
 - `zonecfg -z <zone> create -t <template-zone>`

5.1.10. Installation einer Zone

[dd] Die Installationszeit einer Zone variiert je nachdem, ob eine Sparse-root oder Whole-root Zone installiert werden soll. Weiterhin bestimmt die Menge der in der globalen Zone installierten Software, die Art und Weise der benutzten Plattentechnologie (SATA, IDE, SCSI, FC) und ob ein Schreibcache in dem benutzten Plattensubsystem vorhanden sind, die Zeit zur Installation einer Zone. Diese Zeit kann für eine Sparse-root Zone zwischen 3 und 20 Minuten variieren.

```
global# zoneadm -z zone1 install
Preparing to install zone <zone1>.
Creating list of files to copy from the global zone.
Copying <2373> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <985> packages on the zone.
Initialized <985> packages on zone.
Zone <zone1> is initialized.
Installation of these packages generated warnings: <SUNWxwfnt
SUNWxwcft SUNWolrte SUNWpprou SUNWjxmft SUNWilof SUNWjxcft SUNWxwoft>
The file </zones/zone1/root/var/sadm/system/logs/install_log> contains
a log of the zone installation.
global#
```

5.1.11. Uninstall einer Zone

[dd] Installierte Zonen werden durch `zoneadm -z <zone> uninstall` deinstalliert. Dabei werden die folgenden Aktionen ausgeführt:

- Löschen der Daten in dem `zonepath`-Unterverzeichnis
- Umwandlung des Zustands der Zone in `/etc/zones/index` in `configured`

5.1.12. Automatische Konfiguration von Zonen durch `sysidcfg`

[ug] Bei der Installation eines OS ist es notwendig, dass das Betriebssystem konfiguriert wird. Die Konfiguration kann auf verschiedene Arten dem OS mitgeteilt werden (Siehe JumpStart Dokumentation). Eine der Möglichkeiten bei Zonen ist, dass man eine Datei namens `sysidcfg` erzeugt, die die entsprechenden Parameter enthält. Eine OS-Instanz, die zum ersten Male hochfährt, benutzt die `sysidcfg`-Datei, sofern sie unter `/etc` steht.

Mit Zonen funktioniert das also folgendermaßen:

1. Zone konfigurieren (`zonecfg`)
2. Zone installieren (`zoneadm`)
3. `sysidcfg` Datei erzeugen und ins `/etc` der Zone kopieren (das ist in der globalen Zone unter `<zonepath>/root/etc`)
4. Erzeugen einer Datei `.NFS4inst_state.domain` in `<zonepath>/root/etc`, damit beim ersten Boot die NFSv4 Domain nicht abgefragt wird (noch mit Solaris 10 6/06).
5. Zone booten

5.1.13. Automatische Schnell-Installation von Zonen

[ug] Die Installation von Zonen dauert abhängig von der Geschwindigkeit des Filesystems (Platten) und der zu kopierenden Software der Zone wenige bis einige Minuten.

Dies kann erheblich verkürzt werden, wenn viele Zonen erzeugt werden sollen, die bezüglich der Konfiguration identisch sind (wesentlich sind hier die `inherit-pkg-dir` Einstellungen).

Vorbereitung:

1. `zonecfg` einer Template Zone mit der gewünschten Konfiguration, die später nicht benutzt werden darf (zum Beispiel mit dem Namen `template1`). Hierbei muß `zonepath` gesetzt werden, die Konfiguration der Netzwerk-Adresse sollte fehlen, weil die Zone ja nur als Template dient und nicht in Betrieb gehen soll.
2. Installieren der Zone `template1` mit `zoneadm`.
3. Sichern des Inhaltes der Zone `template1` in einer tar-Datei `template1.tar`. (Anmerkung: `cp -r` ist nicht geeignet um Zonen zu kopieren!)

Erzeugen von duplizierten Zonen:

1. Konfiguration mit `zonecfg` unter Nutzung der Template Funktion (`create -t template1`). (Die `inherit-pkg-dir` Einstellungen dürfen nicht verändert werden!)
2. Ändern von `zonepath` mit `zonecfg` (kann per script geschehen)
3. Eintragen von Netzwerkadresse(n) mit `zonecfg`.
4. Auspacken von `template1.tar` unterhalb `zonepath` der neuen Zone.
5. Editieren von `/etc/zones/index` um die neue Zone auf `installed` zu setzen.
6. Konfigurieren der neuen Zone von außen mittels Kopieren einer passenden `sysidcfg`-Datei in das `<zonepath>/root/etc` der neuen Zone.
7. Erzeugen einer Datei `.NFS4inst_state.domain` in `<zonepath>/root/etc`, damit beim ersten Boot die NFSv4 Domain nicht abgefragt wird (Stand Solaris 10 6/06).
8. Booten der neuen Zone zum Betrieb.

Die neue Zone ist von einer herkömmlich installierten Zone nicht zu unterscheiden.

Wenn Patches eingespielt werden sollen, dann werden die installierten Zonen in den single-user Modus hochgefahren. Da die Template Zone nicht direkt startbar ist, ist zum Einspielen der Patches die folgende Vorgehensweise notwendig:

1. Konfigurieren der Zone `template1` (manuell oder mit `sysidcfg`-Datei)
2. Einspielen der Patches (am besten sind alle konfigurierten Zonen hochgefahren)
3. De-Konfigurieren der Zone `template1` mit dem Kommando `zlogin template1 sys-unconfig`
4. Neu Erzeugen der Zonen-Inhaltsdatei `template1.tar`.

5.1.14. Hardening von Zonen

Zum Hardening von Solaris wird grundsätzlich das Solaris Security Toolkit empfohlen. Die kompletten Abläufe und Mechanismen sind hier abgelegt:

http://www.sun.com/products-n-solutions/hardware/docs/Software/enterprise_computing/systems_management/ssl/index.html

Innerhalb des Toolkits werden die Besonderheiten beschrieben, die zur Härtung von Sparse root oder Whole root Zonen notwendig sind. Details dazu sind hier zu finden:

<http://www.sun.com/products-n-solutions/hardware/docs/html/819-1503-10/introduction.html#pgfId-1001177>

5.2. Netzwerk

5.2.1. Netzwerk und Routing

[dd/ug] Die folgende Abschnitte beschreiben 5 Szenarien im Umfeld Zonen, Netzwerke und Routing. Folgende Einschränkungen existieren:

- In den angeschlossenen Netzen darf die gleiche IP-Adresse nicht doppelt vergeben sein. Kann dies auf Grund von organisatorischen Gegebenheiten nicht verhindert werden, muß eine Abtrennung mit NAT-Routern (Szenario 3) benutzt werden.
- Die vorgenommene Netzwerkseparierung wird im Solaris auf logischer TCP/IP-Ebene realisiert. Für viele Einsatzfälle ist das ausreichend. Ist eine Trennungen auf physikalischer Netzwerkebene notwendig, muß diese durch Domains realisiert werden.
- Eine Verbesserung der Netzwerkseparierung innerhalb von Solaris ist mit dem Projekt Crossbow angestrebt (<http://www.opensolaris.org/os/project/crossbow/>)

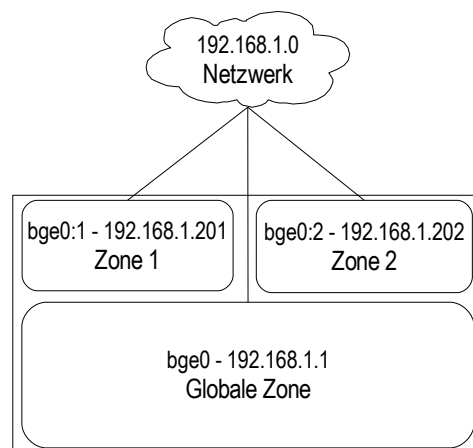
5.2.1.1. Globale und lokale Zone mit gemeinsamem Netzwerk

[dd/ug] Zwei lokale Zonen zone1 und zone2 befinden sich im gleichen Netzwerksegment wie die globale Zone.

- Jede lokale Zone kann auf dem gleichen Netzwerk-Interface wie die globale Zone eingerichtet werden.
- Das Routing, das für die globale Zone einrichtet ist, gilt auch für die lokalen Zonen. Alle Zonen (globale und lokale) können miteinander kommunizieren.

Realisierung:

- Die Zonen werden mit dem Netzwerk-Interface der globalen Zone eingerichtet; ist dies `bge0`, dann wird mit `zonecfg: add net` die Einstellung `set physical=bge0` vorgenommen.
- Jede lokale Zone muß eine Adresse aus dem Netzwerk der globalen Zone erhalten.



Zeichnung 22: [dd] Globale und lokale Zone mit gemeinsamem Netzwerk

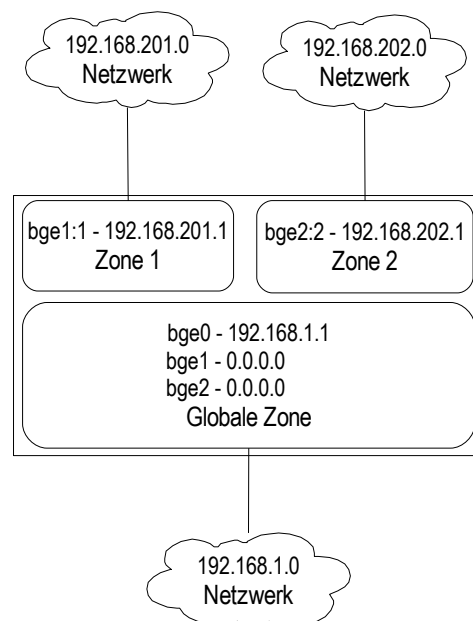
5.2.1.2. Zonen in getrennten Netzwerksegmenten

[dd/ug] Zwei lokale Zonen zone1 und zone2 befinden sich in logisch getrennten Netzwerksegmenten und stellen für diese Netzwerksegmente Dienste zur Verfügung.

- Jede lokale Zone soll ein eigenes physikalisches Interface in dem Netzwerksegment haben.
- An dem Netzwerksegment ist kein weiteres Netzwerk angeschlossen.
- Routing wird nicht eingesetzt.
- Eine Kommunikation zwischen den lokalen Zonen soll nicht stattfinden.
- Eine Kommunikation zwischen der globalen Zone und den lokalen Zonen ist nicht vorgesehen.

Realisierung:

- Das für die lokale Zone vorgesehene Netzwerk-Interface (z.B. *bge1*) darf in der globalen Zone nicht anderweitig benutzt sein.
- Zur Vorbereitung für lokale Zonen muß das Interface geplumbt werden (aber nicht aktiviert):
`ifconfig bge1 plumb down`
Damit erhält das Interface die Adresse 0.0.0.0, die aber nicht aktiv ist.
- Routing-Einträge werden nicht vorgenommen.
- Option: Wenn eine Kommunikation zwischen der globalen und der lokalen Zone möglich sein soll, ist in der globalen Zone das Interface mit einer Adresse zu konfigurieren, die sich im Netzwerk der lokalen Zone befindet.



Zeichnung 23: [dd] Zonen in getrennten Netzwerksegmenten

5.2.1.3. Zonen in getrennten Netzwerken

[dd/ug] Zwei lokale Zonen zone1 und zone2 befinden sich in logisch getrennten Netzwerken und stellen Dienste für andere Netze zur Verfügung.

- Jede lokale Zone soll ein eigenes physikalisches Interface in dem Netzwerk haben.
- An dem Netzwerksegment sind weitere Netzwerke angeschlossen.
- Routing wird eingesetzt.
- Eine Kommunikation zwischen den lokalen Zonen soll nicht stattfinden.
- Eine Kommunikation zwischen der globalen Zone und den lokalen Zonen ist nicht vorgesehen.

Realisierung:

- Das für die lokale Zone vorgesehene Netzwerk-Interface (z.B. *bge1*) darf in der globalen Zone nicht anderweitig benutzt sein.
- Zur Vorbereitung für lokale Zonen muß das Interface geplumbt werden (aber nicht aktiviert):

```
ifconfig bge1 plumb down
```

 Damit erhält das Interface die Adresse 0.0.0.0, die aber nicht aktiv ist.
- Die Netzwerk-Konfiguration der Zonen wird hochgefahren, in dem man die Zonen auf den Zustand *ready* setzt.

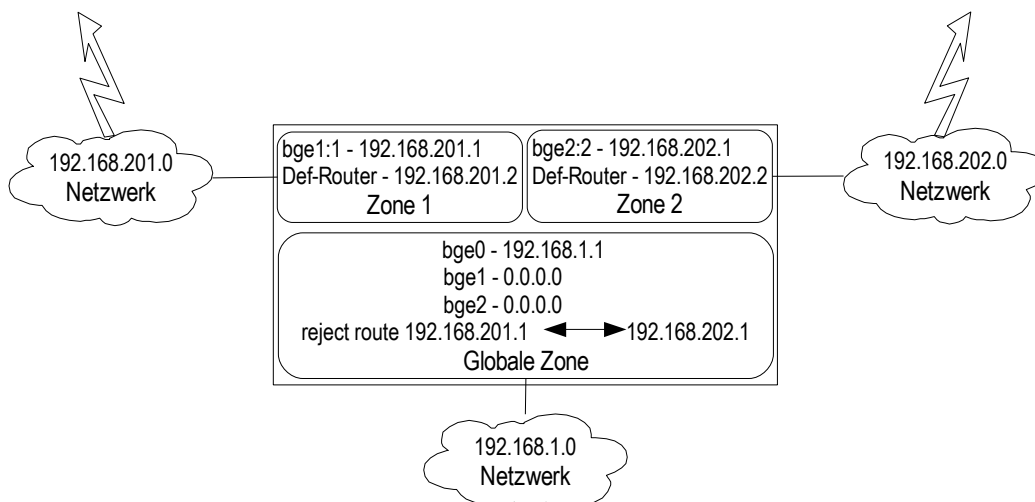
```
zoneadm -z zone1 ready
zoneadm -z zone2 ready
```

 Die in der Konfiguration der Zonen angegebenen Adressen (*zone1: 192.168.201.1* und *zone2: 192.168.202.1*) sind nun aktiv.
- Für die Kommunikation der lokalen Zonen mit anderen Netzwerken wird in der globalen Zone für die lokalen Zonen jeweils eine Default Route gesetzt.

```
route add default 192.168.201.2
route add default 192.168.202.2
```
- Damit keine Kommunikation zwischen den lokalen Zonen durch den shared TCP/IP-Stack stattfindet, müssen in der globalen Zone reject-routes gesetzt werden, die eine Kommunikation zwischen zwei IP-Adressen unterbinden.

```
route add 192.168.201.1 192.168.202.1 -interface -reject
route add 192.168.202.1 192.168.201.1 -interface -reject
```
- Die Zonen können jetzt zum Betrieb gebootet werden:

```
zoneadm -z zone1 boot
zoneadm -z zone2 boot
```
- Option: Wenn eine Kommunikation zwischen der globalen und der lokalen Zone möglich sein soll, ist in der globalen Zone ein Interface zu konfigurieren, das sich in dem logischen Netzwerk der lokalen Zone befindet.



Zeichnung 24: [dd] Zonen in getrennten Netzwerken

5.2.1.4. Zonen mit Verbindung in unabhängige Kunden-Netzwerke

[dd/ug] Zwei lokale Zonen zone1 und zone2 befinden sich in logisch getrennten Netzwerken und stellen Dienste für unterschiedliche Kunden in eigenen Netzwerken zur Verfügung.

- Jede lokale Zone soll ein eigenes physikalisches Interface in dem Netzwerk haben.
- An dem Netzwerksegment sind weitere Netzwerke des Kunden angeschlossen.
- Eine Koordination der Adressvergabe in den Netzwerken findet nicht statt; eine Adresse könnte mehrfach vergeben sein (jeweils 1x pro Kunden-Netzwerk). Das ist mit der heute üblichen Verwendung der privaten IP-Adressen einigermaßen wahrscheinlich.
- Die Zonen zone1 und zone2 sollen aus anderen Netzwerken erreichbar sein.
- Die Zonen zone1 und zone2 können keine Verbindungen in andere Netzwerke initiieren.
- Eine Kommunikation zwischen den lokalen Zonen soll nicht stattfinden.
- Eine Kommunikation zwischen der globalen Zone und den lokalen Zonen ist nicht vorgesehen.

Realisierung:

- Das für die lokale Zone vorgesehene Netzwerk-Interface (z.B. *bge1*) darf in der globalen Zone nicht anderweitig benutzt sein.
- Zur Vorbereitung für lokale Zonen muß das Interface geplumbt werden (aber nicht aktiviert):

```
ifconfig bge1 plumb down
```

 Damit erhält das Interface die Adresse 0.0.0.0, die aber nicht aktiv ist.
- Die Netzwerk-Konfiguration der Zonen wird hochgefahren, in dem man die Zonen auf den Zustand *ready* setzt.

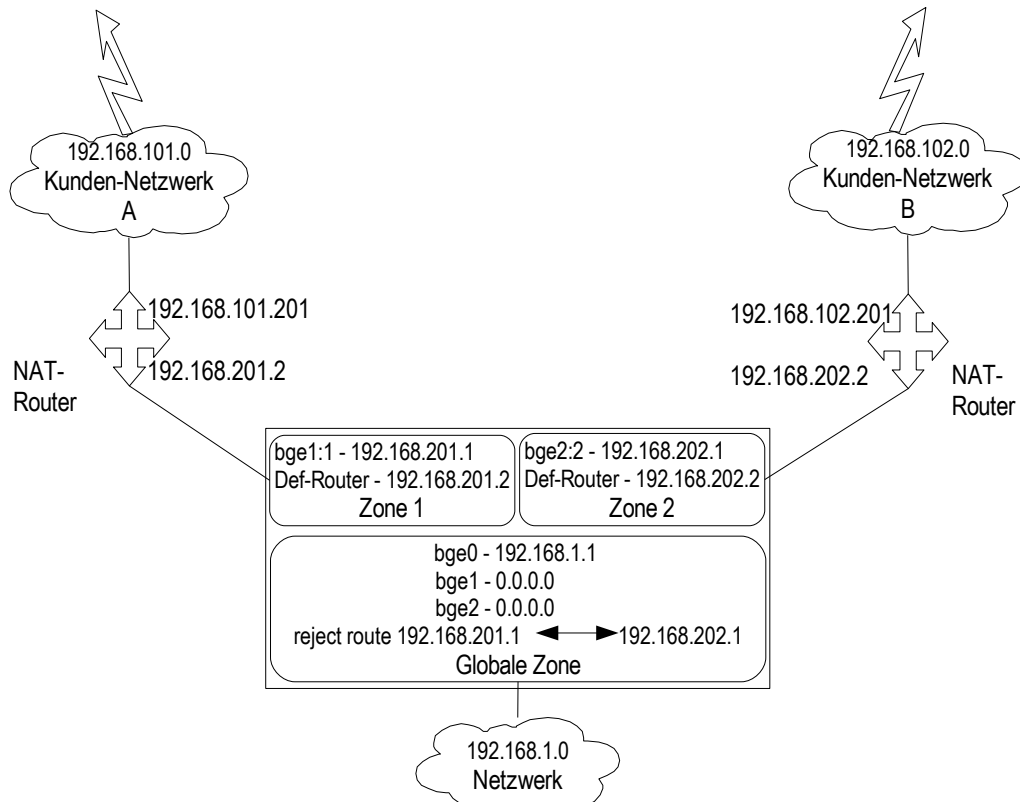
```
zoneadm -z zone1 ready
zoneadm -z zone2 ready
```

 Die in der Konfiguration der Zonen angegebenen Adressen (*zone1: 192.168.201.1* und *zone2: 192.168.202.1*) sind nun aktiv.
- Für die Kommunikation der lokalen Zonen mit anderen Netzwerken wird in der globalen Zone für die lokalen Zonen jeweils eine Default Route gesetzt.

```
route add default 192.168.201.2
route add default 192.168.202.2
```
- Damit keine Kommunikation zwischen den lokalen Zonen durch den shared TCP/IP-Stack stattfindet, müssen in der globalen Zone reject-routes gesetzt werden, die eine Kommunikation zwischen zwei IP-Adressen unterbinden.

```
route add 192.168.201.1 192.168.202.1 -interface -reject
route add 192.168.202.1 192.168.201.1 -interface -reject
```
- Die Zonen können jetzt zum Betrieb gebootet werden:

```
zoneadm -z zone1 boot
zoneadm -z zone2 boot
```
- Der Default-Router ist ein NAT-Router, der die IP-Adresse der lokalen Zone zum Kunden hin verbirgt. Auf der Seite des Kunden ist er mit einer IP-Adresse aus dessen Netzwerk konfiguriert. Somit können keine Konflikte bei Adressen mehr auftreten.
- Option: Wenn eine Kommunikation zwischen der globalen und der lokalen Zone möglich sein soll, ist in der globalen Zone ein Interface zu konfigurieren, das sich in dem logischen Netzwerk der lokalen Zone befindet.



Zeichnung 25: [dd] Zonen mit Verbindung in unabhängige Kunden-Netzwerke

5.2.1.5. Verbindung von Zonen über externe Router

[dd/ug] Ein Webserver in der zone1 wird aus dem Internet kontaktiert und braucht zur Erfüllung der Aufträge den Applicationsserver in der zone2.

- Die zone1 soll über ein separates Netzwerk mit dem Internet verbunden sein.
- Die Verbindung von zone1 nach zone2 soll über einen externen Load-Balancing Router erfolgen.
Wegen der Übersichtlichkeit sind hier keine weiteren Instanzen für Web- und Applicationsserver enthalten.
- Eine Kommunikation direkt zwischen den lokalen Zonen soll nicht möglich sein, wohl aber über den externen Router.
- Eine Kommunikation zwischen der globalen Zone und den lokalen Zonen ist nicht vorgesehen.

Realisierung:

- Die für die lokalen Zonen vorgesehenen Netzwerk-Interfaces (*bge1*, *bge2* und *bge3*) dürfen in der globalen Zone nicht anderweitig benutzt sein.
- Zur Vorbereitung für die lokalen Zonen müssen die Interfaces geplumbt werden (aber nicht aktiviert); damit erhalten die Interfaces die Adresse 0.0.0.0:


```
ifconfig bge1 plumb down
ifconfig bge2 plumb down
ifconfig bge3 plumb down
```
- Die Netzwerk-Konfiguration der Zonen wird hochgefahren, in dem man die Zonen auf den Zustand *ready* setzt.


```
zoneadm -z zone1 ready
zoneadm -z zone2 ready
```

 Die in der Konfiguration der Zonen angegebenen Adressen (*zone1*: *192.168.201.1*, *192.168.200.1* und *zone2*: *192.168.202.1*) sind nun aktiv.
- Für die Kommunikation der Zone *zone1* mit dem Internet wird in der globalen Zone eine Default Route gesetzt. Weiterhin braucht man eine Route zur scheinbaren Adresse der *zone2* hinter dem NAT-Router.

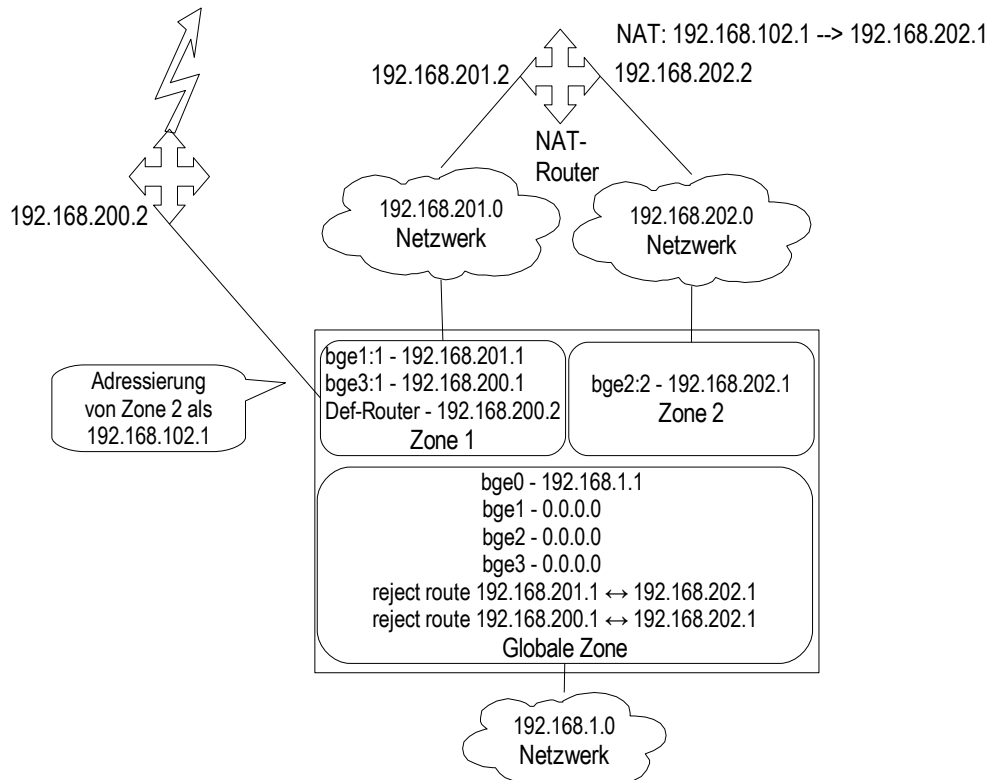

```
route add default 192.168.200.2
route add 192.168.102.0 192.168.201.2
```
- Damit keine Kommunikation zwischen den lokalen Zonen durch den shared TCP/IP-Stack stattfindet, müssen in der globalen Zone reject-Routes gesetzt werden, die eine Kommunikation zwischen den IP-Adressen der beiden Zonen unterbinden.


```
route add 192.168.201.1 192.168.202.1 -interface -reject
route add 192.168.202.1 192.168.201.1 -interface -reject
route add 192.168.200.1 192.168.202.1 -interface -reject
route add 192.168.202.1 192.168.200.1 -interface -reject
```
- Die Zonen können jetzt zum Betrieb gebootet werden:


```
zoneadm -z zone1 boot
zoneadm -z zone2 boot
```
- Die reject-Route führt zu einer vollständigen Unterbindung der Kommunikation zwischen *zone1* und *zone2*, die aber nach obigen Vorgaben in diesem Szenario gefordert ist. Daher muß der konfigurierte Default Router NAT unterstützen. Er muß die Adresse *192.168.102.1* in die Adresse *192.168.202.1* umsetzen. Die Kommunikation über den NAT-Router umgeht so die reject-Routen.
- Option: Wenn eine Kommunikation zwischen der globalen und der lokalen Zone möglich sein soll, ist in der globalen Zone ein Interface zu konfigurieren, das sich in dem logischen Netzwerk der lokalen Zone befindet.

Der Ablauf ist nun folgendermaßen:

- Ein HTTP-Request wird von außen an die Zone zone1 gestellt.
- Teile des Requests kann sie selbst bearbeiten, ein anderer Teil muss vom Applicationserver kommen, der über die Adresse 192.168.102.1 adressiert wird.
- Diese Adresse wird über den NAT Router geroutet, der die Adresse auf der anderen Seite in die Adresse 192.168.202.1 umsetzt.
- Das ist die Adresse der Zone zone2 die den Applicationserver trägt, der die fehlenden Teile des Requests bearbeitet und über die bestehende Verbindung zurückliefert.



Zeichnung 26: [dd] Verbindung von Zonen über externe Router

5.2.2. IPQoS

[dd] Der IP-Verkehr einer Zone zu einer IP-Adresse kann mit IPQoS begrenzt werden. Mit dem Kommando `ipqosconf(1M)` wird in der globalen Zone die Datei `/etc/inet/ipqosinit.conf` erstellt, die die Konfiguration für IPQoS enthält. Diese Konfiguration wird benutzt, um bestimmten Zonen nur eine bestimmte Netzwerkbandbreite zur Verfügung zu stellen.

5.3. Lifecycle Management

5.3.1. Boot einer Zone

[dd] `zoneadm -z <zone> boot` bootet eine Zone, mountet die Filesysteme, initialisiert die Netzwerkinterfaces, setzt die Ressource Controls und startet den Service Manager der Zone. Beim ersten Boot einer Zone werden, wie bei einer Solaris Neuinstallation, alle smf-Service Manifeste importiert und das initiale smf-Repository erstellt.

```
global# zlogin -C zone1
[Connected to zone 'zone1' console]

[NOTICE: Zone booting up]

SunOS Release 5.10 Version Generic_118855-15 64-bit
Copyright 1983-2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: zone1

zone1 console login:
```

Weiterhin wird das Verzeichnis `<zonepath>/dev` erzeugt, das alle Devices enthält, auf die die Zone zugreifen kann.

```
global# ls /zones/zone1/dev
arp          dtrace      msglog      rdisk       syscon      ticlts      vt00
conslog     dtremote    null        rmt         sysevent    ticots
zconsole
console     fb0         poll        sad          sysmsg      ticotsord   zero
cpu         fd          pool        stderr      systty      tty         zfs
crypto      kstat      ptmx        stdin       tcp         udp
cryptoadm   log        pts         stdout      tcp6        udp6
dsk         logindmux   random      swap        term        urandom
```

5.3.2. Softwareinstallation per mount

[ug] Um eine Applikation in der Zone laufen zu lassen, wird außer in trivialen Fällen (Apache, Perl, ...) noch zusätzlich installierte Software (nicht aus Sun-pkg Format) gebraucht.

Eine Methode diese Software zu verteilen ist in der globalen Zone ein geschartes Software-Repository zu pflegen und dieses ganz oder teilweise in den Zonen sichtbar zu machen:

- Global gibt es das Verzeichnis `/software`
- Die Software ist in dem Verzeichnis `/software/produkt/version/` installiert, z.B.:
 - `/software/oracle/9i_05/`
 - `/software/oracle/9i_06/`
 - `/software/oracle/10g_01/`
 - `/software/siebel/3.14/`
 - `/software/tomcat/3.7/`
- Mit `zonecfg: add fs` werden die notwendigen Verzeichnisse in den jeweiligen Zonen readonly sichtbar gemacht (mit `lofs`).

Damit ist gewährleistet, daß auch die Nutzung von Software nachvollzogen werden kann, um gegebenenfalls die Lizenzkosten zu kontrollieren.

Einfacher ist es, das gesamte Verzeichnis `/software` in jede Zone einzubringen.

Zur Überwachung der Nutzung der Software müssen dann andere Mittel benutzt werden (`audit` und `dtrace`).

5.3.3. Software Installation mit Provisioning System

[ug] Die N1 SPS Software kann Software auch in Zonen provisionieren. Die Voraussetzungen sind:

- Ein schreibbares Verzeichnis, in dem die Software installiert werden kann. Dies kann `/opt` sein, das aber nicht als `inherit-pkg-dir` konfiguriert sein darf.
- Wenn die Software unter `/usr` oder in einem anderen System-Verzeichnis installiert werden muß, kann das entweder mit einer whole-root Zone oder mit einem in den `/usr`-Baum gemounteten schreibbaren Verzeichnis (z.B. `/usr/local`) implementiert werden.
- Die Software muß in der Zone lauffähig sein.

5.3.4. Migration einer Zone (mit OpenSolaris attach/detach)

[dd] Zonen können ab Solaris 10 11/06 zwischen physikalischen Systemen bewegt werden. Dafür das Feature `zoneadm detach/attach` enthalten. Für die Migration von Zonen sind die folgenden Bedingungen zu beachten:

- Zonen müssen vor der Migration angehalten werden
- der Patch- und Package-Stand zwischen beiden Systemen muß gleich sein

Eine lokale Zone kann wie folgt zwischen Systemen migriert werden:

- Detach der Zone mit `zoneadm -z <zone> detach`
- Bewegen des `zonepath`-Verzeichnis zum Zielsystem, z.B. im SAN oder mit `tar/cp`
- Anlegen der Zonenkonfiguration `zonecfg -z <zone> create -a <zonepath>`
- Attach der Zone `zoneadm -z <zone> attach`

Eine installierte lokale Zone wird vor der Migration auf einem System detached. Dieser Prozeß erzeugt alle Informationen die benötigt werden, um diese Zone an einem anderen System zu attachen. Diese Informationen werden in der Datei `<zonepath>/SUNWdetached.xml` gespeichert. Der Zustand der Zone wird von „`installed`“ in den Zustand „`configured`“ umgewandelt. So wird diese Zone nachfolgend bei Package Installationen, Patches, Boot, etc. nicht mehr betrachtet.

Vor einem Attach wird die Zonenkonfiguration aus der `SUNWdetached.xml`-Datei erzeugt. Durch das `attach` wird auf dem Zielsystem geprüft, ob die lokale Zone auf das Zielsystem passt. Dazu wird überprüft ob die installierten Packages und Patches in der migrierten Zone und der globalen Zone übereinstimmen.

5.3.5. Herunterfahren einer Zone

[dd] Zonen können aus der lokalen Zone selbst oder aus der globalen Zone heraus heruntergefahren werden. Je nachdem welche Variante benutzt wird, werden dabei laufende Services noch beendet oder einfach gestoppt.

- `zoneadm -z zone halt` wird in der globalen Zone aufgerufen, hält eine Zone an und stoppt alle Prozesse der Zone sofort.
- `zlogin <zone> init 5` wird in der globalen Zone aufgerufen, wechselt in die Zone und fährt die Zone mit dem Service Manager herunter und beendet alle Prozesse ordnungsgemäß.
- `init 5` wird in der Zone selbst aufgerufen, fährt die Zone mit dem Service Manager herunter und beendet alle Prozesse ordnungsgemäß.

5.4. Management und Monitoring

5.4.1. Accounting über eine Zone

[ug] Mit Extended Accounting (Kommando *acctadm*) kann man prozessweise Accounting einschalten. Im vordefinierten Ressource-Umfang namens *extended* wird zusätzlich auch noch der Name der Zone mitgeschrieben. Damit ist es möglich, die Accounting Daten den jeweiligen Zonen zuzuordnen und den Verbrauch der Zonen summarisch abzurechnen, ohne daß aufwendig die Kommandos den einzelnen Applikationen zugeordnet werden müssen, wie es mit traditionellem Unix-Accounting notwendig ist.

Mit Solaris 10 wird eine Library (*libexacct(3LIB)*) und ein Beispielprogramm (*/usr/demo/libexacct/*) mitgeliefert, mit dem die Daten einfach ausgewertet werden können.

5.4.2. Audit einer Zone

[dd] Audit kann im Zusammenhang mit lokalen Zonen auf zwei unterschiedlichen Wegen genutzt werden:

- Audit wird in der globalen Zone konfiguriert. Durch Setzen der *zonename* Policy in */etc/security/audit_startup* trägt audit in jeden audit-record den Zonennamen mit ein. Mit *auditreduce -z <zonename>* werden die entsprechenden audit-records extrahiert und können mit *praudit* ausgewertet werden. Die Konfiguration und Sammlung der audit-Daten erfolgt komplett aus der globalen Zone heraus.
- Audit wird in der globalen Zone konfiguriert. Zusätzlich wird die *perzone* Policy in */etc/security/audit_startup* gesetzt. Dadurch startet jede Zone einen eigenen *auditd* und hält pro Zone eigene Konfigurationen und log-Files. Die Kontrolle über die audit-Konfiguration liegt hier beim Administrator der lokalen Zone.

Je nachdem ob die Kontrolle über die Audit-Konfiguration und der vollständige Zugriff auf die Audit-Daten erforderlich ist, wird die Entscheidung für eine der beiden Konfigurationsmöglichkeiten ausfallen.

5.5. Ressource Management

5.5.1. Begrenzung von /tmp-Größe in einer Zone

[dd] */tmp* wird in vielen Fällen als *tmpfs* in swap benutzt. Das führt dazu, daß der swap-Bereich durch */tmp* in jeder Zone von allen Zonen shared benutzt wird. Die */tmp*-Bereiche sind zwar nur für jede Zone selbst sichtbar, aber es werden globale Ressourcen benutzt. Zur Begrenzung des Platz-Verbrauches sollte */tmp* in der */etc/vfstab* der Zonen mit der *size*-Option gemountet werden.

<i>swap</i>	-	<i>/tmp</i>	<i>tmpfs</i>	-	<i>yes</i>	<i>size=250m</i>
-------------	---	-------------	--------------	---	------------	------------------

Diese Begrenzung kann durch den root-Administrator der Zone verändert werden.

5.5.2. Ressource Pools mit Prozessor Sets

[ug] In Solaris (ab Solaris 9) lassen sich die CPUs auf Resource Pools aufteilen. Solaris Zonen kann man diesen Resource-Pools zuordnen und damit läßt sich auf einfache Weise eine Entkopplung der CPU-Ressourcen durchführen:

- Mit den Kommandos `poolcfg` und `pooladm` wird das Ressource-Management eingeschaltet und der zusätzliche Resource-Pool erzeugt.
- Mit dem Kommando `poolcfg` wird der Prozessor-Set (`pset`) erzeugt und dem Resource-Pool zugeordnet.
- Mit dem Kommando `zonecfg` kann man das Attribut `pool` der Zone ändern und dort den neuen Resource-Pool eintragen. Dies ist dann die Einstellung, die nach Reboot des Systems ohne weitere Maßnahmen gilt.
- Mit dem Kommando `poolbind` kann die Zuordnung der Zonen zu Resource-Pools dynamisch verändert werden. Dies hat aber keine Auswirkung auf die mit `zonecfg` vorgenommene Default-Einstellung.

Die Prozesse einer Zone, die einem Resource Pool zugeordnet sind, laufen dann nur auf den Prozessoren ab, die zum Prozessorset des Resource-Pools gehören.

Sind mehrere Zonen einem Resource-Pool zugeordnet, dann kann das Verhältnis der verbrauchten CPU-Zeit mit dem Fair-Share Scheduler eingestellt werden.

5.5.3. Fair Share Scheduler

[ug] Das Verhältnis des CPU-Verbrauchs zwischen Zonen oder Projekten läßt sich einstellen.

Dazu wird der sogenannte Fair Share Scheduler benutzt. Dabei werden CPU-shares zugeteilt:

- Für Zonen geht das mit `add rctl` und dem Attribut `zone.cpu-shares` im `zonecfg`-Kommando.
- Bei Projekten wird dies in der project-Database (`/etc/project`, oder NIS oder LDAP) und dem Attribute `project.cpu-shares` eingestellt (globale und/oder lokale Zone).
- Die Zonen/Projekte müssen dem gleichen Resource-Pool zugeordnet werden.
- In dem Resource-Pool muß der Fair Share Scheduler eingeschaltet werden.
- In der globalen Zone kann man den Resource-Pool auch auf eine Untermenge der Prozessoren im System limitieren.

Das Betriebssystem sorgt dann für Fairness, wenn die CPUs des Resource Pool ausgelastet werden. Dazu werden die zugeteilten Shares der Projekte und Zonen mit aktiven Prozessen dazu benutzt, den Sollwert für den CPU-Verbrauch zu berechnen.

Ist der Anteil der Verbrauchten CPU-Zeit größer als der durch die `cpu-shares` definierte Teil, steuert der Fair Share Scheduler durch Verringerung der Priorität dagegen.

Weiterhin:

- Im laufenden Betrieb (dynamisch) können die Einstellungen für die `cpu-shares` mit dem Kommando `prctl` umgestellt werden.
- Die Zugehörigkeit zu Resource-Pools kann mit dem Kommando `poolbind` jederzeit umgestellt werden.

5.5.4. Statisches CPU Ressource Management zwischen Zonen

[ug] Statisches Ressource-Management zwischen Zonen bestimmt, wie im Normalfall nach einem Bootvorgang die Ressourcen zwischen Zonen verteilt werden sollen:

- Zum Erstellen eines Resourcepools werden die Kommandos `poolcfg` und `pooladm` benutzt.
- Die Zugehörigkeit einer Zone zu einem Resourcepool kann bei `zonecfg` mit dem Attribut `pool` eingestellt werden.
- Die Einstellungen für Fairness zwischen den Zonen werden mit `add rctl` im `zonecfg` eingestellt. Empfohlen wird die Einstellung von `zone.cpu-shares` (Anteil CPU) und `zone.max-lwps` (maximale Anzahl Threads).

5.5.5. Dynamisches CPU Ressource Management zwischen Zonen

[ug] Hiermit sind die Kommandos gemeint, mit denen man im laufenden Betrieb die Ressource-Kontrolle umstellen kann, um auf eine temporäre Lastsituation zu reagieren.

Dazu werden die Kommandos *prctl* und *poolbind* verwendet. Gegebenenfalls können mit *poolcfg* und *pooladm* neue Ressourcepools temporär erstellt werden.

5.5.6. Statisches CPU Ressource Management in einer Zone

[ug] Das Verhältnis des CPU-Verbrauchs von Applikationen in einer Zone läßt sich ebenfalls einstellen.

Dazu werden innerhalb der Zone Ressourcepools definiert und der Fair Share Scheduler eingesetzt. Eine Zuordnung von CPUs zu den Ressource-Pools ist in der Zone nicht möglich.

- Die Zuordnung von Applikationen zu Projekten erfolgt mit Konfiguration in den Dateien */etc/project* und der */etc/user_attr*.
- In der */etc/project* ist definierbar, in welchem Ressource-Pools ein Projekt laufen soll.
- Zum Erstellen eines Ressource-Pools in der Zone werden die Kommandos *poolcfg* und *pooladm* benutzt, Prozessoren können allerdings nicht zugewiesen werden.
- In dem Ressource Pool muß der Fair Share Scheduler (FSS) eingeschaltet werden.
-

5.5.7. Dynamisches CPU Ressource Management in einer Zone

[ug] Hiermit sind die Kommandos gemeint, mit denen man im laufenden Betrieb in der Zone die Ressource-Kontrolle umstellen kann, um auf eine temporäre Lastsituation zu reagieren.

Dazu werden die Kommandos *prctl* und *poolbind* verwendet. Gegebenenfalls können mit *poolcfg* und *pooladm* neue Ressourcepools temporär erstellt werden.

5.5.8. Physikalischen Hauptspeicherverbrauch eines Projektes begrenzen

[dd] Zur Begrenzung des physikalischen Hauptspeicherverbrauches eines Projekte kann der Ressource Capping Daemon *rcapd(1M)* verwendet werden. Wenn die resident set size (RSS) eines Projektes die Capping-Vorgabe (*rcap.max-rss* in Bytes) überschreitet, reduziert der *rcapd* die RSS und lagert benutzte Speicherseiten auf das Paging-Device aus. *rcapd* wird durch *rcapadm(1M)* konfiguriert und mit *rcapstat(1)* überwacht.

Der *rcapd* kann in Zonen im Zusammenhang mit Projekten eingesetzt werden und den physikalischen Hauptspeicherverbrauch eines Projektes innerhalb einer Zone begrenzen.

6. Literatur

- [1] Jeff Victor, "Solaris Containers Technology Architecture Guide", Sun Blueprint, Mai 2006, <http://www.sun.com/blueprints/0506/819-6186.html>
- [2] Jeff Victor, "Zones and Containers FAQ", OpenSolaris FAQ, <http://opensolaris.org/os/community/zones/faq/>
- [3] Sun Microsystems Inc., "Solaris Containers Learning Center", http://www.sun.com/software/solaris/containers_learning_center.jsp
- [4] Joost Pronk van Hoogeveen, "Working with Solaris Containers and the Solaris Service Manager", Sun Blueprint, Mai 2006, <http://www.sun.com/blueprints/0506/819-4328.html>
- [5] Menno Lageman, "Solaris Containers --What They Are and How to Use Them", Sun Blueprint, Mai 2005, <http://www.sun.com/blueprints/0505/819-2679.pdf>
- [6] Sun Microsystems Inc., "System Administration Guide: Solaris Containers-Resource Management and Solaris Zones", Solaris 10 Manual, 2006, <http://docs.sun.com/app/docs/doc/817-1592>
- [7] OpenSolaris Project: „Crossbow: Network Virtualization and Resource Control“ <http://opensolaris.org/os/project/crossbow/>
- [8] Marcelo Leal's Solaris Screencasts zu Solaris Containers - Resource Management and Zones http://www.posix.brte.com.br/blog/index.php?page_id=30
- [9] Websphere Application Server im Container
http://www.sun.com/software/whitepapers/solaris10/websphere6_sol10.pdf
und http://blogs.sun.com/roller/page/sunabl?entry=websphere_deployment_on_solaris_10
- [10] Statement von IBM, dass Websphere MQ nur in globalen Zonen und whole root lokalen Zonen unterstützt wird
<http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg21233258>
- [11] Solaris Containern bei der Apache Software Foundation
<http://www.tbray.org/ongoing/When/200x/2006/03/06/Apache-Server>
- [12] Oracle im Container
<http://www.sun.com/blueprints/0505/819-2679.pdf> pp. 22-33
und Oracle Metalink <https://metalink.oracle.com> (Note: 317257.1)
- [13] Sun Cluster Data Service for Solaris Containers Guide <http://docs.sun.com/app/docs/doc/819-2664>
- [14] Webserver/ftpd und Backup/Restore in einer Zone
<http://www.sun.com/blueprints/0506/819-6186.pdf> pg. 8 und pp. 10-13
- [15] Qualification Best Practices for Application Support in Non-Global Zones
http://developers.sun.com/solaris/articles/zone_app_qualif.html
- [16] "Bringing Your Application Into the Zone".
http://developers.sun.com/solaris/articles/application_in_zone.html
- [17] Sreekanth Setty, "Deploying Sun Java Enterprise System on the Sun Fire T2000 Server using Solaris Containers", Sun Blueprint, August 2006, <http://www.sun.com/blueprints/0806/819-7663.pdf>
- [18] <http://en.wikipedia.org/wiki/Virtualization>
- [19] Zonenspezifische Einstellungen in späteren Solaris Updates:
<http://www.opensolaris.org/os/community/arc/caselog/2006/496/spec-txt>
- [20] Grafische Überwachung des Zonen-Auslastung
<http://www.asyd.net/home/projects/zonestats>