

Alternative zum ASCC auf GPL-Basis

-

UNIX-Stammtisch Dresden 01.11.2006

Zustandsbeschreibung

- Anlass: Die Linux Internet Präsentation der Stadt Dresden ist unsicher bei Hardwareproblemen
- Ziel:
 - ausfallsicheres System durch Überwachung der Hardware und Software
 - standardisierte Software für alle Linux-Verfahren
 - Senkung des Administrations Aufwands für die Pflege
 - zentrale Steuerung aller Linux Systeme
- Lösung – Ausschreibung mit vier möglichen Lösungen als Ergebnis:
 - (1) Wir stellen Ihnen unsere Hardware zur Verfügung mit FC-HW. Unsere Stunde kostet:
 - ...
 - (2) VMWARE-Lösung mit FC-HW
 - (3) FC-HW Lösung, bei sich
 - die SW selbst überwacht
 - auf jedem Server jeder Dienst mit läuft (auch bei DB)
 - (4) ASCC
- Entscheidung: ASCC
 - NFS als File Server – läuft immer und ist nicht zu teuer
 - zentrale Überwachung und Steuerung möglich; zentrale Systempflege
 - Load Balancing

Struktur

- Netz Filer (FAS 250)
 - NFS (nur am internen, privaten Netz)
 - Betriebssystem Images der Applikation Nodes
 - sonstige Daten Ressourcen
 - Sicherung über FC (nach Austausch FAS250 durch FAS270)
- 2 Control Nodes
 - eigenes Betriebssystem
 - gegenseitiges Überwachen
 - Steuerung des Gesamtsystems
 - internes und externes Netz
 - steuern das Netz Boot der Applikation Nodes
- Applikation Nodes
 - internes und externes Netz
 - Netz Boot
 - Entwicklungsumgebung (manueller Modus) und Applikations Umgebung (online Modus)
- Struktur des ASCC
 - File-Server
 - Applikations-Knoten ohne eigens System
 - Steuer-Server (gegenseitige Überwachung)

Realitäten

- Eine-Person-Bedienoberfläche für ein Multi-User-System
 - nur Test einer Konfiguration auf einem Node zu einem Zeitpunkt möglich
 - Oberfläche reagiert stark Zeitversetzt auf Benutzerinteraktionen
- offensichtliche Schwächen und Startprobleme
 - keine doppelte Anbindung an das Hausnetz (wurde beseitigt)
 - Netz-Filer nur einfach eingebunden (wurde beseitigt)
 - Ausfall des Gesamtsystems noch möglich:
 - Ausfall des Rahmens, in dem die Knoten untergebracht sind
 - Ausfall des Netz Filers
Daran wird gearbeitet; es sollte ein zweites ASCC daneben stehen können, auf das die Daten gespiegelt werden und das im Notfall die Arbeit übernimmt.
 - Die Sicherung des Netz Filers ist noch nicht realisiert! (Lösung ist auch mit TSM über FC möglich!; internes Problem)
- Clon-Technologie
 - Erzeugen eines Images dauert 6 bis 24 Minuten (auch mit NDMP-Copy)
 - Verlust der Protokolle
- Jede Änderung einer Applikation macht ein 'Take-Image' + 'Deploy Image' für alle Nodes nötig (weil die Konfiguration im Image gespeichert wird).
- es mangelt an Bedienkomfort: Kopieren von Teilen der Konfiguration oder Löschen (Beispielsweise NFS Ressource) geht nicht!

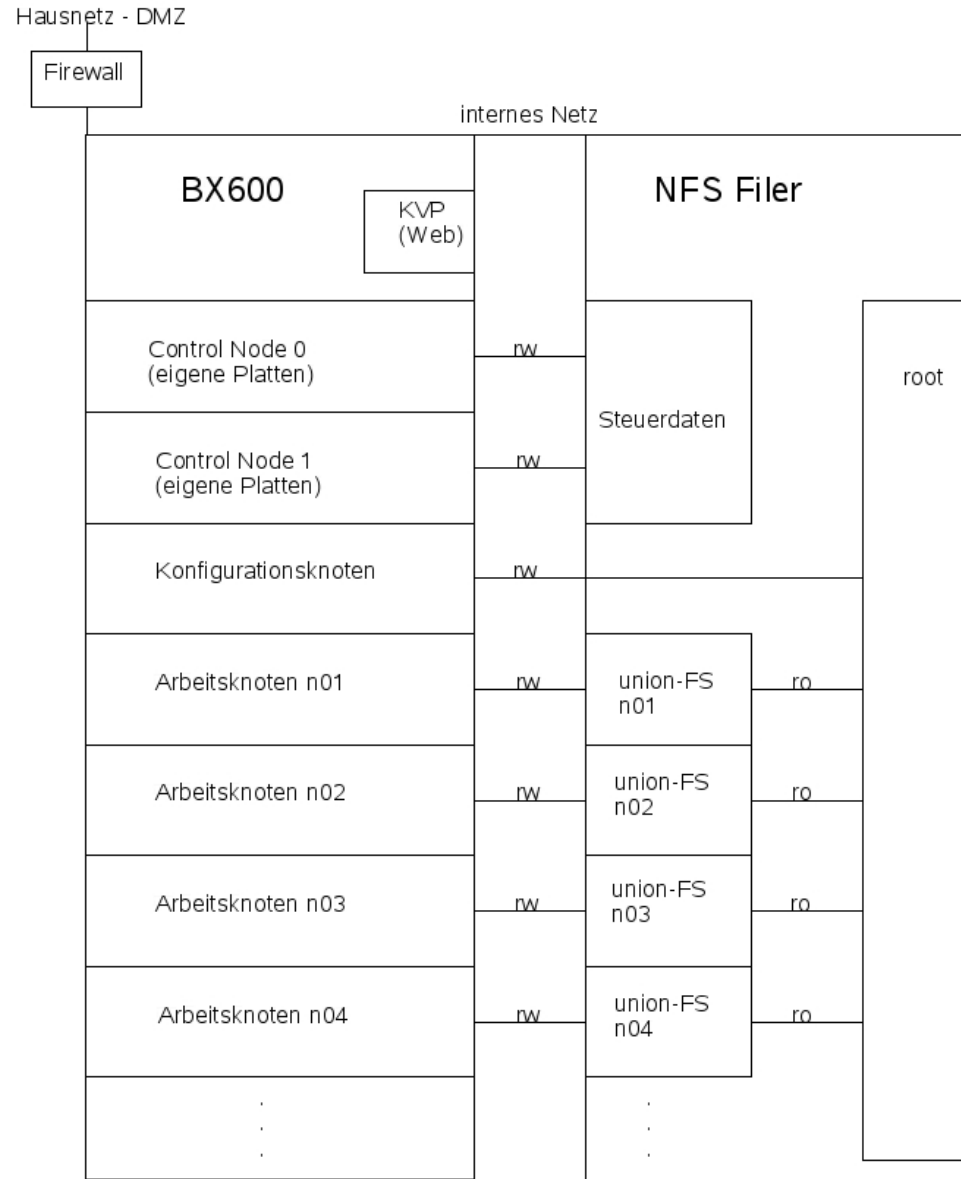
Wie würde ich das ASCC neu bauen?

- Die Lösung ist nicht akzeptable! Darum wird eine eigene Lösung angestrebt!
- File Server:
 - Linux – Shares auf LVM aufbauen
 - doppelt halten mit 'Heartbeat' und 'DRDB' (Siehe Linux-Magazin 2004/07)
 - sollte neben NFS auch tftp und pxelinux beinhalten
 - Bereitstellen der System Platten für alle Knoten: Knoppix Prinzip mit Union-FS (Siehe Linux-Magazin 2006/03)
- Steuer-Server:
 - ohne System Platten wie die restlichen Knoten (außer /tmp und swap)
 - Steuerung: Apache + Perl
 - Bedienoberfläche: einfacher Browser (ohne Java + Java Script)
 - doppelt halten: siehe File Server
- Applikation Nodes:
 - ohne eigene Platten (außer /tmp und swap)
 - Load Balancing: 'pound' oder 'LVS' einsetzbar
- Realitäten:
 - es gibt bereits einen NetApp Filer, der zu nutzen ist!
 - NetApp war nicht zu PXE-BOOT zu überreden, ebensowenig zum DHCP für PXE-Boot

hacc-

- Neuaufbau mit frei Software (außer NetApp) unter Beibehaltung der Grundstruktur:
 - Filer (zweiter Filer geplant - Ausfallsicherheit): NFS Services für:
 - Daten zum Boot der Arbeitsknoten (pxelinux,cfg, tftp-Kernel u. Initrd-Verweis)
 - Nagios Steuerdaten und Steuerscripte für die Arbeitsknoten
 - Root Partitionen
 - Union-FS Partitionen
 - Home Partition für die Arbeitsknoten und Daten-Partitionen für die Arbeitsknoten
 - Steuerknoten
 - eigenes Betriebssystem (Linux)
 - zwei Steuerknoten überwachen sich gegenseitig mit Heartbeat
 - Dienste für Boot: dhcp, tftp (atftpd) (PXE Boot Daten liegen auf NetApp Filer)
 - Überwachung und Gewährleistung der Ausfallsicherheit: Nagios
 - rinetd-Weiterleitung für Sicherung des NetApp Filers (keine Hausnetzanbindung)
 - Mail-Relais mit Postfix (Alarm des NetApp)
 - Arbeitsknoten
 - werden per PXE gebootet und per NFS und Union-FS mit Daten versorgt
 - sind austauschbar
 - Konfigurationsknoten (Einspielen von Patches in das Gesamtsystem)

hacc- Struktur



Management Blade

Blade Server System Information - Mozilla Firefox <@hacc-cn0>

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://192.168.100.20/

SUSE LINUX Entertainment News Internet Search Reference Maps and Directions Shopping People and Companies

FUJITSU COMPUTERS SIEMENS BX600

PRIMERGY

BX600

System Property
Management Blade
Switch Blade
Server Blade
Server Blade-1
Server Blade-4
Server Blade-5
Server Blade-6
Server Blade-7
Recovery
ASR
Auto Configuration
Power Control
Boot Option
Blade Info
Server Blade-8
Server Blade-9
Server Blade-10

Power Control

Power Status Power On

Power Switch power on Apply

Power On/Off

	On Time	Off Time
Sunday	disabled	disabled
Monday	disabled	disabled
Tuesday	disabled	disabled
Wednesday	disabled	disabled
Thursday	disabled	disabled
Friday	disabled	disabled
Saturday	disabled	disabled

hour : minute or disabled

Apply

Everyday

Power Setting

When power off fail, force power off after 5 minutes.

Apply

Controller time: 11/02/2006 08:16:41
© 2000-2004 Fujitsu Siemens Computers

http://192.168.100.20/US/7/power_control.htm

Netzanbindung - Sicherheit

- Das System arbeitet mit einem eigenen, abgeschotteten, inneren Netz (eth1)
 - Der NetApp Filer hat nur dort Zugriff
 - Die Ressourcen des Filers werden nur im inneren Netz angeboten
 - Booten der Arbeitsknoten -> innere IP Adresse
 - Zugriff der Arbeitsknoten auf das Betriebssystem und auf das vorgeblendete Union-FS
 - Zugriff auf die Daten
 - Überwachung
- Über das externe Netz wird auf die Dienste und auf die Steuerrechner zugegriffen (bond0, bestehend aus eth0 und eth2):
 - IP Bonding (doppelt angebunden) -> Äußere IP Adresse
 - keine Routing zum inneren Netz
 - Überwachung der Zugriffe durch einen externen Firewall
- Ausbau:
 - Aufbau eines zweiten NetApp Filers (Aufbau NetApp Cluster)
 - Aufstellung einer zweiten BX600 und Verteilung der Steuerknoten
- Sicherheit:
 - Einzelsicherung der Steuerknoten über TSM
 - Sicherung NetApp Filer über TSM und FC

Boot

- dhcpd.conf:
DHCPD_INTERFACE = "eth1";
... (PXE Optionen)
filename "pxelinux.0";
next-server 192.168.100.101;
shared-network "Auto DHCP" {
 subnet 192.168.100.0 netmask 255.255.255.0 {
 group {# Server managed by NETboot
 host C0A86406 {
 hardware ethernet 00:C0:9F:99:EB:E7;
 fixed-address 192.168.100.6;
 }
 }
 }
...
▪ /etc/sysconfig/atftpd: ATFTPD_DIRECTORY="/data/vol_root/tftpdir" (NFS Resource)
hacc-cn0:/data/vol_root/tftpdir # ls
04-initrd (-> ../04/boot/initrd)
04-vmlinuz (-> ../04/boot/vmlinuz)
messages pxelinux.cfg pxelinux.0

PXE

- Je zu startenden Knoten gibt es in pxelinux.cfg ein File; z.B. C0A86407:
default 04_SLES10
label 04_SLES10
KERNEL 04-vmlinuz
APPEND initrd=04-initrd ip=192.168.100.7:192.168.100.100::255.255.255.0::hacc-
n01P:eth1:off root=192.168.100.100:/vol/vol_root/04 acpi=off
- Der Knoten wird dadurch versorgt mit:
 - einem Root File-System: liegt in 192.168.100.100:/vol/vol_root/04
 - unter /vol/vol_root werden die verschiedenen Versionen des Betriebssystems aufbewahrt.
 - Mit jedem Patch wird eine neue Version erstellt. Hier wird der Stand 04 benutzt.
 - einer inneren IP Adresse; hier; 192.168.100.7
 - Netzwerkanbindung zum inneren Netz
 - die letzte Stelle (7) ist ein symbolischer Link auf das Stück Union-FS, das für diesen Arbeitsknoten vor das Root-System gekettet wird

Union-FS

- hacc-cn0:/data/vol_union # ls -l
7 -> n01
8 -> n02
9 -> n03
n01 (vor geschaltete Union-FS Systeme)
n02
n03
n04
n05
XXX (Platzhalter für defekte Knoten)
- Union-FS: Knoppix:
 - Verketteten mehrere Filesysteme so hintereinander, dass sie als ein Filesystem sichtbar sind
 - lesen in allen verketteten Systemen, schreiben im obersten
 - besondere Vorkehrungen für das Löschen
 - Voraussetzung: installieren der zum Linux Kernel passenden Treiber: runter laden, übersetzen (nur der speziellen Treiber), installieren im aktuellen Kernel, initrd Modulliste um 'unionfs' ergänzen und initrd neu erstellen mit 'mkinitrd -D eth1'

Arbeitsknoten

- speichert seine Daten in seinem individuellen Union-FS, das beim Boot vorgelagert wurde
 - Start der Kundendienste, die angeboten werden
 - besitzt eine primäre IP Adresse (und je Dienst eine Aliasadresse)
 - enthält alle Konfigurationsdateien des Servers
 - enthält alle Logs
 - erscheint wie ein normales Linux-System
- Initialisierung dieses Union-FS
 - Kopieren eines vorhandenen Union-FS und entfernen aller Dateien bis auf ein Grundgerüst
 - Anlegen eines Links für die innere IP Adresse, mit der die Initialisierung vorgenommen werden soll
 - Modifizieren einiger Dateien:
 - /etc/HOSTNAME
 - /etc/fstab
 - /etc/sysconfig/network/ifcfg-bond0 (neue Adressen eintragen)
 - Starten und weitere Konfigurationen, wie bei einem normalen Linux
- Jeder Knoten besitzt ein lokales Swap und ein lokales /tmp!

Nagios

- Nagios ist ein Überwachungssystem für das Netzwerk und für Systemkomponenten
- Steuerung: Konfigurationsfiles, Scripte und einer Web-Oberfläche
- Die Standard-Skripte der RPM Pakete von SuSE wurden stark verändert, da dieses Produkt so nicht einsetzbar war (evtl. hätte es sich gelohnt, die neueste Version herunter zu laden und einzusetzen)
- Je IP Adresse wurde ein Host definiert; alle externen IP Adressen eines Hosts wurden als 'parent' verbunden
- Für die Services wurden selbst programmierte Prüfroutinen zugefügt.
- Für die kritischen Services wurden event-Handler erstellt, die in Abstufung
 - den Dienst neu starten
 - den Server neu starten
 - für ssh-Probleme wurde ein Event-Handler erstellt, der ein Reset für den Knoten durchführt (über das Blade Control Center physisches RESET)
- für die Hosts wurde ein Event-Handler erstellt, der bei kritischen Hosts eingebunden werden kann und der in Abstufungen:
 - 15 Minuten warten
 - immer noch nicht erreichbar:
 - Auswahl eines neuen Knotens
 - Kennzeichnen des bisherigen Knotens als defekt

Konfigurationsknoten

- Beim Ersteinrichten wurden einige Anpassungen vorgenommen
 - `/etc/sysconfig/network/config`: 'FORCE_PERSISTENT_NAMES=no'
 - `/etc/init.d/network`: im Stop-Teil: Herunterfahren von eth1 durch Modifikation des Scripts verhindern (ifconfig bond0, eth0 und eth2 down und 'exit 0' wurde am Anfang des Stop-Teils eingebaut)
 - Einbau eines Scripts in den Bootprozeß, der das für die inneren IP Adresse ausgewählte Union-FS vorkettet (oder keines vorkettet beim Konfigurationsknoten)
 - `/etc/init.d/boot` wurde dafür entsprechend modifiziert (scriptaufruf eingebaut) (Siehe Quellen)
 - ein Script, von 48 Zeilen, welches dieses Einketten entsprechend vornimmt, wurde erstellt (Nur wenn ein Union-FS Filesystem für die IP Adresse existiert wird es vorgekettet)
- Über den Konfigurationsknoten werden Änderungen im Root Filesystem vorgenommen
 - Erstellen eines neuer Root Systems
 - Einspielen der Patches
 - Beim Einspielen von Kernel Updates (über neues Root System):
 - Neu Installieren Union-FS Treiber
 - Installieren der besonderen initrd mit unionfs Treiber: `mkinitrd -D eth1`

Script: /etc/init.d/union.sh

- Einbinden in /etc/init.d/boot:

```

union.sh (/etc/init.d) - GVIM <@hacc-n00>
Datei Editieren Werkzeuge Syntax Puffer Ansicht Hilfe
#!/bin/bash
# union.sh
# Vorketten des union-fs Filer bereichs vor Root

# Boot Interface
IF='eth1'
# unionfs Resource Path
UNION='192.168.100.100:/vol/vol_union'

# IPT = <letzte Stelle der IP Adresse des Hostes auf Interface IP>
IPT=$(ifconfig $IF | head -2 | tail -1 | \
    awk '{ print $2 }' | awk -F. '{ print $4 }')

echo Resource: $UNION/$IPT/

set -x

# Anlegen der Ressourcen zum Mounten
mkdir -p /ram
mount -n -t tmpfs tmpfs /ram -o rw
mkdir /ram/changes /ram/union

# Mounten des eigenen Bereichs des Servers
if (mount -n -t nfs $UNION/$IPT/ /ram/changes -o rw,nolock)
then
    # alte mtab-Files entfernen
    rm /ram/changes/etc/.wh.mtab*

    # neu mounten Root
    mount -n -o remount,ro,nolock /

    # zusammenfuehren der Ressourcen
    mount -n -t unionfs -o dirs=/ram/changes=rw:/=ro unionfs /ram/union

    # einfüegen der Systembereiche
    mkdir -p /ram/union/changes
    cat /proc/mounts
    mount -n --move /proc /ram/union/proc
    mount -n --move /sys /ram/union/sys
    mount -n --move /dev /ram/union/dev
    mount
    pivot_root /ram/union /ram/union
    mount
    cat /proc/mounts
    echo Ende unionfs einbinden
    # sleep 60
fi
14,0-1 Alles

```

```

boot (/etc/init.d) - GVIM <@hacc-n00>
Datei Editieren Werkzeuge Syntax Puffer Ansicht Hilfe
*) DO_CONFIRM= ;;
esac
unset answer
echo
fi
export DO_CONFIRM

#
# unionfs einfüegen
#
echo -n "Starten Dateisysteme"
/bin/sh /etc/init.d/union.sh

#
# Start blogd, requires /proc and /dev/pts.
# Export the real device in variable REDIRECT.
#
test -x /sbin/blogd -a -n "$REDIRECT" && /sbin/blogd $REDIRECT
if test -z "$REDIRECT" ; then
    if (echo -n > /dev/tty) 2>/dev/null ; then
        REDIRECT=/dev/tty
    else

```

147,28 44%

Ergebnis

- Alle Arbeitsknoten sind wie normale Linux-Rechner administrierbar
 - Keine Pfad-Verbiegungen
 - Keine zusätzlichen Modifikationen im System durch symbolische Links oder spezielle Mount-Bind-Operationen
 - SW-Patches dürfen nur auf dem Konfigurationsknoten eingespielt werden!
- Problem: neues Patch-Regime von Novell für SLES10
 - Jeder Knoten soll sich bei Novell melden (schlimmer als Microsoft)
 - dann holt er individuell die Patches und installiert sie
 - das kann so nicht gemacht werden, weil:
 - teilweise Softwarepakete geändert werden müssen nach der Installation, damit sie laufen (Nagios, Tomcat5)
 - die Server keine aktive Internetverbindung haben sollen
 - Software nicht auf den Arbeitsknoten sondern nur auf dem Konfigurationsknoten gespeichert werden soll
 - Lösung:
 - SLES10: Zur Zeit werden die Patches manuell runter geladen und irgendwann eingespielt
 - Langfristige Lösung: andere Distribution!

Weiterentwicklungsmöglichkeiten

- Nagios weiter einbinden:
 - Auslastung registrieren und grafisch darstellen
 - Standard Layout ändern
- Web Oberfläche für die Konfiguration des Systems
 - Installation neuer Blades
 - Einbinden neuer Applikationen in die Überwachung
 - Erstellen einer einheitlichen Administrationsoberfläche (Web), für:
 - das Web Kontrol Center (KVP)
 - die Switches
 - Nagios
 - die zu entwickelnde Konfigurationsoberfläche
- Einbindung weiterer BX600 Systeme in ein System der Steuerung unter Beachtung eines Total Ausfalle (Strom Ausfall) eines BX600
- Aktives Betriebssystem Versions-Management, das die gezielte Einbindung mehrerer Betriebssystemversionen erlaubt

Quellen

- [1] <http://www.am-utils.org/project-unionfs.html>
- [2] <http://www.gi-hill.de/fachtagung2005/linux-diskless.pdf>
- [3] Linux-magazin 2006/03 Wurzelimport
- [4] http://www.kolbitsch.org/research/technical/Unattended_Linux.pdf
- [5] <ftp://unionfs-mirror.linux-live.org/unionfs/>
- [6] http://www.novell.com/de-de/documentation/suse10/pdfdoc/suse10_ref/suse10_ref.pdf