Einsatz von Fujitsu Siemens Primecluster als HV-Lösung (DE)

Ulrich Hansmair

ulrich.hansmair@siemens.com

Dieses Dokument beschreibt die Hochverfügbarkeitslösung Primecluster von Fujitsu Siemens Computers (FSC) auf Linux/i386 mit einer Einführung in das technische Konzept sowie einer Fall Studie aus der Praxis.

1. FSC Primecluster

1.1. Produktbeschreibung

Die hier beschriebene Software RMS (Reliant Monitoring System) in der Version 3.1E30 ist Ende 2000 für Linux von Fujitsu Siemens Computers (FSC) verfügbar geworden und stellt meines Wissens damit eines der ersten kommerziellen Produkte in diesem Bereich dar. Es handelt sich um die Linux-Portierung einer seit Jahren weiterentwickelten Software, die erst für das Siemens eigene UNIX (Reliant Unix), dann auf Solaris weiterentwickelt wurde. Die aktuelle Version ist bereits bei der Nummer 4.0XXX angelangt und wird nun unter dem Namen Primecluster vertrieben. Vergleichbare Produkte im propietären Umfeld sind HP ServiceGuard und SUN HA-Cluster.

1.2. Hochverfügbarkeitsstrategie

FSC Primecluster verfolgt den klassischen Ansatz auf mehreren physikalischen Servern (Cluster Knoten) einen oder mehrere sog. logischen Host zu erzeugen, der zwischen den Cluster Knoten umgeschalten werden kann. Im Gegensatz zur Solarisversion ist die Anzahl der Cluster Knoten bei Linux auf zwei beschränkt. Am Markt werden jedoch vorwiegend Zwei-Knoten-Server nachgefragt, meist um eine bestimmte, wichtige Applikation hochverfügbar zu machen. Diese Applikation läuft dann innerhalb dieses logischen Hosts.

Wer sich unter dem Begriff "logischer Host" die komplexe Simulation einer Linux-Umgebung vorstellt (vergleichbar etwa VMWARE) liegt völlig falsch. Der logische Host besteht meist nur aus einer

"virtuellen" IP-Adresse, die per IP-Aliasing auf die physikalische Netzwerkschnittstelle des Cluster Knoten gebunden wird, sowie aus zwischen den Cluster Knoten schaltbare Platten (FibreChannel-/SCSI-Dual-Hosted oder NFS). Die HV-Applikaton läuft komplett in der lokalen Umgebung des Cluster Knotens! (Dieses Design erfordert somit symmetrische Clusterknoten, was in der Praxis oft zu Problemen führt, siehe unten.) Der Schaltvorgang des logischen Hostes (durch Administrations-Kommando oder im Fehlerfall automatisch durch die Cluster-Software) besteht somit aus dem Beenden der Applikation, der Dekonfiguration der virtuellen IP-Adresse, des Umounts auf die schaltbaren Platten sowie das Ganze in umgekehrter Reihenfolge auf dem Ziel Cluster Knoten. D.h. Der Schaltvorgang führt zu einer Downtime der Applikation von ca. 2 Minuten (z.B. Web-Service) bis zu 30 Minuten (z.B. Datenbank) oder länger. Dies stellt auch den wesentlichen unterschied von Hochverfügbarkeit und Fehler Toleranz (Applikation läuft ohne Unterbrechung weiter) dar. Der Unterschied ist vielen Leuten, insbesondere oft dem Kunden(!), unklar.

Die Überwachung des Clusters erfolgt mit sog. Detektoren. Dies sind Programme als C-Binary oder Shell-Skripte, die von dem "Cluster Base Monitor" zyklisch über den Zustand des Clusters befragt werden. Dazu werden die zu überwachenden Resourcen in einer logischen Baum Struktur gegliedert. Jede Resource dieses Baumes kann parametriesiert werden und in Abhängigkeit zu anderen Resourcen gesetzt werden. Damit wird ein geordnetes Offline-/Online-Prozessing der Cluster Applikation möglich. Es geht dabei um Fragen wie z.B.

- Für welche Resourcen macht ein "Autorecovery" Sinn, bevor der Cluster einen Schaltvorgang ausführt?
- Wie hoch soll der Timeout sein, bevor eine Resource als nicht Verfügbar angesehen wird?
- In welcher Reihenfolge müssen die Resourcen beim Online-Prozessing aktiviert werden? (z.B. macht es keinen Sinn die Applikation zu starten, wenn die Plattenresourcen noch nicht aktiviert sind)

Eine wichtiges Thema stellt die Cluster Knoten Eliminierung dar. Bei grösseren Defekten (z.B. CPU) ist es einem Cluster Knoten u.U. nicht mehr möglich, ein geordnetes Offline Prozessing für den Schaltvorgang durchzuführen. Dies verhindert natürlich ein erfolgreiches Schalten der HV-Applikation (z.B. virutelle IP-Adresse ist noch am alten Knoten konfiguriert). Für diesen Fall muss ein Cluster Knoten den anderen "eliminieren" können. Bei proprietären UNIXen (z.B. SUN) wird der entsprechende Knoten dann an der Konsole automatisch angehalten (Kernel Debugger, OBP bei SUN). Die PC Architektur bietet nicht diese Möglichkeit. Deshalb greift man auf einen "Network Power Switch" (NPS) zurück, mit dem sich die Cluster Knoten gegenseitig den Strom abschalten können. Hier beginnt auch die (m.E. sehr theoretische) Diskussion, was passiert, wenn die Cluster Knoten den Kontakt untereinander (über "private interconnect") verlieren ("split brain syndrom").

1.3. Cluster Konfiguration

Die Cluster Konfiguration ist ein komplexes Aufgabengebiet. Die Qualität der Konfiguration ist massgeblich für die tatsächliche Verfügbarkeit der Applikation. Um Support vom Hersteller zu bekommen, ist es üblich, die Cluster Konfiguration als Dienstleistung eben vom Hersteller selbst einzukaufen (z.B. SUN HA-Cluster) bzw. zumindest abnehmen zu lassen. Hier ist bei FSC Primecluster

mit den "Wizard Tools" eine leistungsfähige Unterstützung zur Konfiguration gängiger Lösungen integriert. Die Konfiguration von hochverfügbarem Oracle, Informix, SAP R/3, Apache und einiger anderer Applikationen wird wesentlich vereinfacht und auf hohen Qualitätsniveau gesichert.

Die "Wizard Tools" bestehen aus bereits konfigurierten Modulen, die mit einem Konfigurationseditor zu einem vollständigen Resourcebaum, d.h. einer vollständigen Konfiguration, kombiniert werden können. Der Auswahl Dialog der Module kann so aussehen (es sind nicht alle Wizard Pakete installiert):

```
Creation: Application type selection menu:
 1) HELP             8) R3any           15) generic         22) r3combi-inf8vd
 2) QUIT             9) R3ci            16) ipalias         23) r3combi-intf
 3) RETURN          10) Rawdisk         17) nfs             24) r3combi-ora7fs
 4) Cmdline         11) allround        18) nfs-client      25) r3combi-ora7vd
 5) Ipalias         12) basic-if        19) nfs-server      26) r3nfs
 6) Mpoint          13) filesys         20) r3combi-adabas  27) workload
 7) Nfs             14) foreign-code    21) r3combi-any
Application Type:
```

Durch die Module generic, Cmdline, allround sind dem skriptwilligen Administrator keine Grenzen gesetzt.

1.4. Software-Komponenten

FSC Primecluster wird im rpm-Format ausgeliefert.

```
Obligatorisch:


SMAWRrms-3.1E30-15_SuSE7.0.i386.rpm
Reliant Monitor Software, d.h. Daemonen, Kommandos, Manpages

SMAWskel-1.00-2.i386.rpm
Skeleton

SMAWRhv-ba-32A20-22_i386.rpm
SMAWRhv-do-32A20-22_i386.rpm
SMAWRhv-ge-32A20-22_i386.rpm
SMAWRhv-li-32A20-22_i386.rpm
SMAWRhv-to-32A20-22_i386.rpm
SMAWRhv-r3-32A20-22.i386.rpm
......
Wizard Tools Standard ergänzt um Modul für SAP R/3 (weiter Module für
Oracle, Apache usw. sind möglich)


Optional:
```

```
SMAWcf-1.2A00-23_SuSE7.0.i386.rpm
Reliant Cluster, Cluster Foundation (RC-CF) provides fundamental cluster
services like consistent cluster membership, high speed IPC and reliable
transport across the interconnect, which are then used by the higher-level
service layers from which clustering applications are developed.

SMAWcip-1.2A00-08_SuSE7.0.i386.rpm
Reliant Cluster IP (CIP).

SMAWdtcp-2.1A00-33_SuSE7.0.i386.rpm
Reliant Cluster Scalable Internet Services

SMAWjre-1.1.8v1-2.i386.rpm
Java Runtime Environment.

SMAWrcadm-1.2A00-18.i386.rpm
Java GUI for CF, SIS, and RMS
```

2. Fall Studie

2.1. Systemumgebung

Die Cluster Konfiguration der Fall Studie lässt sich mit der Statusabfrage des laufenden Cluster (Befehl: "hvdisp -a") darstellen:

```
Local System:  CLUSTERKNOTEN_A_RMS1
Configuration: cluster1.us

Resource            Type            HostName            State
------------------------------------------------
-
CLUSTERKNOTEN_A_RMS1       SysNode                             Online
CLUSTERKNOTEN_B_RMS1       SysNode                             Online
R3_MAIN             userApplication                     Online
Machine001_R3_MAIN  andOp           CLUSTERKNOTEN_A_RMS1
Machine000_R3_MAIN  andOp           CLUSTERKNOTEN_B_RMS1       Online
CI_R3_ZENTRALINSTANZ gResource                           Online
Interface000_VIRTUELLE_IP_PUBLIC_LAN gResource
Online
Interface000_VIRTUELLE_IP_BATCH_LAN gResource
Online
NfsMountPoint000_NFS_SAPWORK gResource                     Online
NfsMountPoint000_NFS_SAPMNT gResource                     Online
```

```
NfsMountPoint000_NFS_SAPIO gResource                              Online
NfsMountPoint000_NFS_ORACLE gResource                              Online
```

Die ersten beiden Zeilen zeigen den Cluster Knoten an auf dem man sich befindet sowie die aktuell laufende Cluster Konfiguration. Dann (Zeile 7 und 8) folgen die Cluster Knoten, die beide "online" sind. Nun kommt der eigentliche Resource Baum. Die Wurzel ist die userApplicaton "R3_MAIN" (vom Typ: generic), die nur auf einem der beiden Cluster Knoten online sein kann (hier: CLUSTERKNOTEN_A_RMS1). Die Resource "CI_R3_ZENTRALINSTANZ" (vom Typ: R3ci) steht logisch darunter. Die Blätter werden durch die letzten acht Zeilen dargestellt. Es sind zwei virtuelle IP-Adressen (vom Typ: Ipalias) sowie vier schaltbare Platten (vom Typ: Nfs).

2.2. Praktische Erfahrungen

2.2.1. Applikations Administration SAP R/3

Das Projekt der Fall Studie hat einen administrativen Aufwand, der Arbeitsteilung unabdingbar macht. Insbesondere wird zwischen Applikationsmanagement und Betriebssystem Administration Unterschieden. Die klare Rollenaufteilung zwischen "root" und Applikations-Benutzer (hier: "sidadm") wird hier durchbrochen, weil die Cluster Software die Applikationssoftware (hier: SAP R/3) steuert und überwacht. Falls der "sidadm" die Applikation stoppen will, um z.B. einen Fehler zu beheben, muss er dies der Cluster Software mitteilen, damit diese nicht über eine Autorecovery Funktion die Applikation sofort wieder zu starten versucht. Diese Besonderheit muss auch von Administratoren beachtet werden, die z.B. für die Sicherung zuständig sind.

2.2.2. Betriebssystem Administration Linux

Die Administration der Cluster Software ist Aufgabe von "root". Sie fügt zusätzlich Komplexität ins das Linux System, die die Administration oder eine etwaige Fehlersuche nicht gerade erleichtern. Häufige Fehlerursache ist die Asymmetrie der Cluster Knoten. Ein Eintrag in /etc/hosts auf einem Cluster Knoten ist natürlich im Schaltfall wertlos, wenn er auf dem zweiten Cluster Knoten nicht vorhanden ist.

2.2.3. IP-Aliasing

Die IP-Adresse des logischen Hosts wird mit IP-Aliasing auf eine physikalische Schnittstelle gebunden. Dadurch kann der logische Host Daten unter eigener IP-Adresse empfangen. Will der logische Host allerdings Daten versenden, erscheint als "Source IP Adress" im IP-Paket beim Empfänger die physikalische Schnittstelle des Cluster Knotens, auf dem der logische Host läuft. Dies kollidiert mit Sicherheits Kontroll Mechanismen wie z.B. TCP-Wrapper oder "HOST Authentication" (Sicherheits Einstellung bei BS2000/OS-390 Filetransfer mit OpenFT).

Um dieses Thema zu bewältigen, müssen für das virtuelle Interface expliziete Routen gesetzt werden

```
root@CLUSTERKNOTEN_A:/root>route add 1.2.3.4 eth0:1
```

oder bei Bedarf ein verändertes Default-Routing nach dem Muster

```
root@CLUSTERKNOTEN_A:/root>route add DEFAULT-GW eth0:1
root@CLUSTERKNOTEN_A:/root>route add default gw DEFAULT-GW
```

Man muss also zwischen "lokalen Routen" (nur den Cluster Knoten betreffende, z.B. /etc/route.conf bei SuSE) und "virtuelle Routen" (die von der Cluster Software gesetzt werden) unterscheiden. Die Konfigurationsdatei bei FSC Primecluster sieht z.B. wie folgend aus (incl. Default-Routing):

```
#uname-n IfName Interface(s) Netmask    Routes
#uname-n  : the UNIX name of the system as reported by uname -n
#IfName   : the name of the address as defined within the
#           hosts file and the wizard configuration
#Interface: the name(s) of the LAN controllers, for instance
#           zx0, or et0,et1 if there are redundant controllers
#Netmask  : the netmask in hex-notation, e.g. 0xffffffc0
#Routes   : the arguments specified are passed to the route command
#           directly: "-net 192.168.20.0 netmask 255.255.255.0 gw "
#           will be passed as "route add -net ..." or "route del -net ..."
#           respectively. If there is more than one route command to
trigger,
#           separate the individual commands by a comma. The actual
interface
#           name is represented by the string . It is replaced by the
#           actual name of the interface upon invocation of the command.
#Note, dot notation is not allowed; do not forget to specify the networks in
#      the networks file and the names in the hosts file; do not use any
#      domain name service; redundancies are processed from left to right;
#      use different netmasks to do load balancing among several controllers
#      within the same subnet.

#uname-n IfName Interface(s) Netmask    Routes
#robo    O22msg eth0,eth1    0xffffff00 default dev  , -net 192.168.20.0
netmask 255.255.255.0 dev
CLUSTERKNOTEN_A CLUSTERKNOTEN_A_VIRT eth1 0xffffff80 default gw 192.168.1.1 eth1:1 , -
net
192.18.48.0 netmask 255.255.240.0 gw 192.168.1.61 eth1:1 , -net 141.29.64.0
netmask 255.255.224.0 gw 192.168.1.61 eth1:1 , -net 141.29.96.0 netmask
255.255.240.0 gw 192.168.1.61 eth1:1 , -net 192.18.90.0 netmask
255.255.255.0 gw 192.168.1.61 eth1:1 , -net 192.18.120.0 netmask
```

```
255.255.248.0 gw 192.168.1.61 eth1:1 , -net 192.18.220.0 netmask
255.255.252.0 gw 192.168.1.61 eth1:1 , -net 192.18.224.0 netmask
255.255.252.0 gw 192.168.1.61 eth1:1 , -net 192.18.232.0 netmask
255.255.248.0 gw 192.168.1.61 eth1:1 , -net 192.25.0.0 netmask 255.255.0.0
gw 192.168.1.61 eth1:1
CLUSTERKNOTEN_B CLUSTERKNOTEN_A_VIRT eth1 0xffffff80 default gw 192.168.1.1 eth1:1 , -
net
192.18.48.0 netmask 255.255.240.0 gw 192.168.1.61 eth1:1 , -net 141.29.64.0
netmask 255.255.224.0 gw 192.168.1.61 eth1:1 , -net 141.29.96.0 netmask
255.255.240.0 gw 192.168.1.61 eth1:1 , -net 192.18.90.0 netmask
255.255.255.0 gw 192.168.1.61 eth1:1 , -net 192.18.120.0 netmask
255.255.248.0 gw 192.168.1.61 eth1:1 , -net 192.18.220.0 netmask
255.255.252.0 gw 192.168.1.61 eth1:1 , -net 192.18.224.0 netmask
255.255.252.0 gw 192.168.1.61 eth1:1 , -net 192.18.232.0 netmask
255.255.248.0 gw 192.168.1.61 eth1:1 , -net 192.25.0.0 netmask 255.255.0.0
gw 192.168.1.61 eth1:1
```

Ein Konflikt ergibt sich, wenn zwei logische Hosts "virtuelles Default Routing" benötigen. Dies ist leider nicht möglich.

2.2.4. ps-Kommando (Folie)

Das ps-Kommando spielt bei der Überwachung der Applikationen und Subapplikationen eine wichtige Rolle. Viele Detektoren auf Shell-Ebene "grep’en" in der Prozesstabelle nach den gewünschten Prozessen. Hier wird ein grundsätzlicher Unterschied von Linux und UNIX unter Umständen zum Problem.

Der Linux Kernel ist so designed, dass alle Statusabfragen von z.B. System Befehlen über das /proc-Dateisystem abstrahiert sind. Will beispielsweise der ps-Befehl Auskunft über das aufrufenden Befehl eines Prozesses haben, so muss "ps" den Filedeskriptor der Datei /proc/PID/cmdline öffnen, den Inhalt auslesen und schliesslich den Deskriptor wieder schliessen. Proprietäre UNIXe haben zwar auch ein /proc-Dateisystem, die System Befehle wie ps lesen jedoch direkt im Kernel die entsprechenden Tabellen aus. Das Linux-/proc dagegen kann als API für System Befehle betrachtet werden.

Der Vorteil des Linux Designs liegt auf der Hand: der Kernel kann leicht ausgetauscht werden. Für einen Kerneltausch ist es nicht erforderlich, System Befehle wie z.B. ps neu zu kompilieren. Es entstehen jedoch zwei gravierende Nachteile.

Bedingt durch die Datei System Operationen im /proc Dateisystem ist der Linux "ps" wesentlich langsamer/aufwändiger als z.B. bei SUN Solaris. Praktische Vergleiche deuten mindestens auf den Faktor 1000 hin. Bei ausgiebigem Gebrauch, wie im Cluster Umfeld, erzeugt die Überwachung bereits eine erhebliche Grundlast auf dem Cluster Knoten. Der Cluster Knoten wird unperformant.

Das zweite Problem liegt wesentlich versteckter. Ich habe bisher nur ein der ps-Manpage eine Andeutung darüber gefunden:

```
hansmair@ABCDEFGH:/home/hansmair>man ps
.....
 Programs  swapped  out to disk will be shown without command line arguments, and un-
less the c option is given, in brack
       ets.
.....
```

Ein Prozess der ausge'swap't wurde, enthält anscheinend in /proc/PID nur noch sehr verkürzte Informationen. Dies führt dazu, dass der ps Befehl eine stark verkürzte Ausgabe der obigen cmdline bringt. Aus "/usr/sbin/apmd -w 10 -P /usr/sbin/apmd_proxy" wird z.B. "[apmd]". In Speicher Hochlast Situationen führt dies dazu, das der Cluster schaltet, weil ein Detektor lügt. Eine Anpassung des regulären Ausdrucks, nach dem ge'grep't wird, ist meist nicht möglich. Es würde mich sehr interessieren, aus welchen Gründen die Kernelprogrammierer dies so lösen!

2.2.5. Dynamische Geräte Erkennung

Ein generelles Problem im Serverbereich ist die Dynamische Erkennung von Geräten bei Linux. Vor allem bei Festplatten und Netzwerk Schnittstellen sind betroffen. Angenommen in einem Server sind folgende Festplatten eingebaut:

```
/dev/sda
/dev/sdb
/dev/sdc
/dev/sde
/dev/sdf
/dev/sdg
```

Falls /dev/sda wegen Hardware Versagen ausfällt, wird aus /dev/sdb nun /dev/sda usw. Alle in /etc/fstab beschriebenen Zuordnungen sind nun verschoben und nach einem Reboot (bei dem typischerweise Festplatten versagen) geht ganz oder teilweise schief.

Der Einbau einer Netzwerkkarte in einen Server sollte immer unmittelbar mit der Konfiguration zusammen passieren. Wird eine Netzwerkkarte in einen Slot vor den bestehenden Netzwerkkarten eingebaut, wird aus "eth0" nun "eth1", usw. und der Server kommt nach dem Einbau nicht mehr ans LAN.

2.3. Fazit

Nach fast zwei Jahre Erfahrung kann insgesamt ein sehr positives Urteil über die vorgestellte HV-Lösung gefällt werden. Der Hersteller bietet ein sehr ausgereiftes Produkt an, das ruhig und kontinuierlich weiterentwickelt wird. Die frühe Vorstellung des Produktes speziell für Linux ergab einen Vorteil am

Markt. Leider ist trotzdem kein hoher Bekanntheitsgrad, auch in Fachkreisen, festzustellen. Dabei beweist das Produkt FSC Primecluster, dass Linux auch im "Profi" Bereich durchaus mit anderen UNIXen mithalten kann. Vielleicht kann diese Dokument etwas dazu beitragen, HV-Lösungen auch unter Linux mehr in Betracht zu ziehen.

A. Web-Links

Anbei zwei Links, die als Ausgangspunkt für eine Recherche im World Wide Web dienen können:

Link mit vielen Infos aus der opensource/Linux Welt rund um das Thema HV: *http://linux-ha.org* (http://linux-ha.org/)

Link zu FSC Primecluster: *http://www.fujitsu-siemens.com/rl/products/software/clustertechnology.html*