

Ein *günstiger* Hochverfügbarkeits-Webserver (Teil 1)

[Frank Rennemann](#), [Rolf Gauss](#)

Am Beispiel der "alten" Portal-Hardware werden wir in diesem Artikel eine Möglichkeit, relativ kostengünstig zu einer befriedigenden HA-Lösung (High Availability = Hochverfügbarkeit) zu kommen, skizzieren.

Vorbetrachtungen

Eine sehr gute Einführung zum Thema Hochverfügbarkeit befindet sich schon auf dem Portal, [Hochverfügbare Systeme unter Linux \[1\]](#). Ausgehend davon, wie wichtig die Verfügbarkeit eines Serverdienstes ist, sollte man sich für einen vernünftigen Kompromiss zwischen Kosten und Nutzen entscheiden. Natürlich gibt es Situationen, wo die Kosten für Hardware und Betrieb gegenüber den möglichen Schäden bei Nichtverfügbarkeit keine oder nur eine untergeordnete Rolle spielen. Sollten Sie eine derartige Installation vornehmen wollen, wird die hier skizzierte Möglichkeit sicher keine brauchbare Lösung darstellen.

Bei einem Webserver ist zwar eine besonders hohe Aussenwirkung zu erwarten, andererseits wird aber kein potentieller Besucher Schaden an Leib, Leben oder Geldbeutel nehmen, wenn dieser Dienst aufgrund eines Strom-, Netzwerk- oder Hardwareausfalls mal für eine Stunde oder im Extremfall sogar für ein ganzes Wochenende nicht zur Verfügung steht. Auf der anderen Seite macht es natürlich auch keinen guten Eindruck auf potentielle Besucher, wenn das "Online-Schaufenster" einer Firma längere Zeit sozusagen zugeklebt ist.

Bis jetzt ist uns allerdings noch kein Fall bekannt geworden, wo ein Firmen-Manager zu seinem Webmaster gesagt hätte "Setz uns einen Firmen-Webserver auf, Geld spielt keine Rolle". Im Gegenteil, oft sind es unterdimensionierte Maschinen im Dauerstress, die mit veralteter, längst abgeschriebener Hardware gut besuchte Websites ins Netz bringen. Fällt so ein System ohne Netz und doppelten Boden dann aus, ist es oft schon ein Problem, geeignete Ersatzteile zu beschaffen. Meist bedeutet es für den Systemadministrator eine schlaflose, durchgearbeitete Nacht, um eine schnell herbeigezauberte, noch ältere Maschine als Ersatz zu konfigurieren. Und aus eigener Erfahrung wissen wir, was es heisst, Entscheidungsträger von der Notwendigkeit der Anschaffung neuer Hardware zu überzeugen ("Wieso, der alte Rechner, den Sie als Ersatz eingesetzt haben, funktioniert doch").

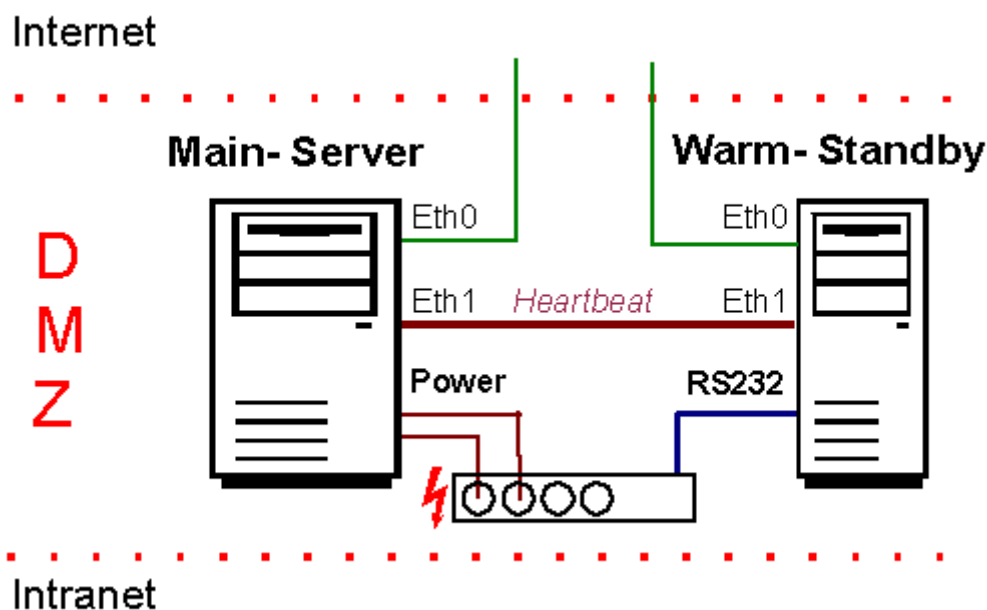
Die von uns im Folgenden beschriebene Lösung hat den Vorteil, dass die evtl. schon vorhandene Hardware als Warm-Standby weiter verwendet werden kann, wenn es der Argumentation gegenüber dem Management dient. Es ist auch nicht nötig, beim Neuaufsetzen eines Servers symmetrisch vorzugehen, d.h. der Standby-Server muss nicht über die gleiche teure Hardware verfügen, die den Hauptrechner möglicherweise etwas kostspieliger macht.

Zwei Primergys und eine Steckdose

Das SuSE-Portal, wie es vor knapp einem Jahr, Anfang Februar 2001, an den Start ging, bestand im Wesentlichen aus zwei Siemens Primergy Servern auf Intel-Basis, einer E200 mit zwei 933 MHz Pentium III Prozessoren und einer Primergy 170 mit einem 866 MHz Pentium III. Beide Rechner verfügten über jeweils 512 Mb Arbeitsspeicher und über ca. 20 GB Speicherplatz. Die E200 hatte ein

RAID-Array aus vier 9 GB Festplatten, drei über RAID-Level 5 mit insgesamt 18 GB nutzbarem Festplattenplatz verknüpft, die vierte als Hot Spare, d.h. beim Ausfall einer der anderen Platten wurde sie sofort als Ersatz verwendet. Der zweite Server hatte aus Kostengründen nur eine 20 GB IDE-Festplatte. Beide Systeme standen in der SuSE-DMZ (DMZ = DeMilitarisierte Zone, zwischen zwei Firewalls), verfügten über jeweils zwei Netzwerkkarten und wurden über rsync-Scripte von einem Rechner im Intranet symmetrisch auf dem gleichen Datenbestand gehalten.

Der Clou der Anlage war eine per serielllem Kabel an den Warm-Standby-Server angeschlossene schaltbare Steckerleiste mit vier Steckdosen[2]. Über diese Steckerleiste wurde der Hauptserver über seine beiden Netzteile mit zwei getrennten Netzkabeln mit Strom versorgt (siehe Abbildung). Per Script konnte der Zweitserver beide Steckdosen, die für die Stromversorgung des Hauptservers zuständig waren, über die serielle Schnittstelle abschalten. So war sichergestellt, dass bei einem versehentlichen Signal an eine der Steckdosen der Hauptserver nicht unbeabsichtigt abgeschaltet wurde. Nur wenn beide Steckdosen abgeschaltet würden, würde der Hauptserver den Dienst einstellen und die Übernahme durch den Zweitserver gelingen.



Der Heartbeat zwischen den beiden Servern wurde über ein wget-Shellscript von Warm-Standby-Server realisiert, das in regelmässigen Abständen versuchte, per wget die robots.txt des Hauptservers zu erreichen. Damit ließ sich nicht nur die Funktion des Servers überprüfen, sondern auch, ob evtl. nur der Apache-Webserver seinen Dienst eingestellt hat. Blieb der Heartbeat aus, kappte ein Script auf dem Zweitserver die Stromversorgungen des Hauptservers und konfigurierte die Netzwerkkarte, die ins Internet zeigte, um, damit sie nun die gleiche IP-Adresse hatte wie vorher der nun von Netz genommene Hauptserver. Diese Umschaltung passierte binnen ca. 5 Minuten, ein zwar merkbares, aber noch nicht unangenehmes Intervall, das sich bei verkürzten Heartbeat-Zeiten auch noch hätte verringern lassen.

Beim ersten Stromausfall hat sich das System sofort bewährt, denn bedingt durch das RAID-Array mit ext2-Filesystem benötigte der Hauptserver inklusive fscheck immer eine ganze Weile, bis er nach einem harten Shutdown wieder dienstbereit war. Auch in diesem Fall übernahm der Zweitserver, weil er dank Reiser-Filesystem extrem schnell dienstbereit war. Da er auf seine Heartbeat-Anfragen noch keine Antwort bekam, nahm er nach kürzester Zeit den Hauptserver wieder vom Netz. Der ersten Übernahmescript-Version fehlte noch eine eMail-Schnittstelle, so dass es drei Tage dauerte (über ein Wochenende), bis die Übernahme überhaupt bemerkt wurde. Diese Funktion wurde danach natürlich nachgebessert, um die Admins darauf aufmerksam zu machen, dass der Hauptserver kontrolliert gebootet und wieder ans Netz gebracht werden muss. Gleichzeitig hatte der Ernstfall aber auch die Tauglichkeit der Installation bewiesen.

Alles in allem hat diese Lösung ca. 9.000 Euro für den Hauptserver, ca 3000 Euro für den Zweitserver und ca 200 Euro für die Schaltsteckerleiste gekostet. Die komplette Konfiguration wurde testweise auf

zwei normalen Workstations entwickelt. Welche Serverhardware man im Ernstfall einsetzt, hängt nur von Einsatzzweck und Budget ab. Die Lösung erwies sich als extrem wartungsarm, einmal installiert lief die ganze Anlage, unterbrochen von zwei kurzen Stromausfällen störungsfrei, die letzten fünf Monate sogar ohne irgendeinen Admin-Eingriff.

Fazit

Diese kostengünstige Variante einer HA-Lösung erreicht zwar nicht die Ausfallsicherheitswerte anderer Anlagen, bietet aber einen akzeptablen Kompromiss für nicht "mission-critical" Anwendungen. Darüber hinaus hat sie den Vorteil, dass auch ältere Hardware eingebunden und somit "recycled" werden kann, was sich auch als Argument gegenüber Entscheidungsträgern immer gut macht. Ein weiterer Vorteil liegt z.B. in der Verfügbarkeitssicherheit bei Betriebssystem-Updates. Soll ein neuer Kernel eingespielt werden, macht man das Update zuerst auf dem Warm-Standby. Läuft der danach wieder einwandfrei, löst man eine Dienstübernahme aus (Ausfallzeit unter 30 Sekunden), führt das Update auch auf dem Hauptsystem aus und kann auch hier in aller Ruhe die Änderungen auf Funktionsfähigkeit testen, bevor man den Hauptserver wieder den Dienst übernehmen lässt.

Im zweiten Teil des Artikels werden wir den genauen Ablauf und die Scripte, die die Übernahmen regeln, näher beleuchten.



Linux auf dem Server 20.01.2002