

```
field = bodyText;
wrap = <div class="Yh_text"></div>
```

Listing 5

Zuletzt wird nun das Menü neu definiert, da die ursprüngliche horizontale Version nicht gut mit dem meist zur Verfügung stehenden Joystick eines Handys bedient werden kann. Bei der Festlegung der Site-Struktur ist zu bedenken, dass ein Menü nicht zu viele Elemente enthalten sollte, da es auf kleinen Bildschirmen sonst schnell unübersichtlich werden kann.

TYPO3GRUPP

```
page: 10 >
page: 100 = MENU
page: 100.wrap = <div class="Yh_menu"></div>
page: 100.1 = MENU
page: 100.1.MO 1
altWrap = <div style="font-weight: bold;"></div>
```

Listing 6

Nach dem „lendi“ folgt noch der Inhalt als einfache Liste der Content-Elemente aus der „Normal“-Spalte.

TYPO3GRUPP

```
# der Inhalt
page: 20 < Styles.content.get
```

Listing 7

Durch diese einfache Condition, die die von WURFL gepflegte Eigenschaft „is_wireless_content“ prüft, wurde eine mobile Variante der Website erzeugt. Für eine reale Website sind sicher noch weitere Content-Anpassungen notwendig. Das Prinzip bleibt aber immer dasselbe: Man überschreibt die relevanten TypoScript-Einstellungen für die jeweiligen per WURFL ausgelassenen Geräteigenschaften. Für eine Produktions-Site sollte man die Templates modularisieren. In der TYPO3-Dokumentation ist dies beschrieben. Leider kann man mit dem TYPO3-Conditions keine komplexen Booleschen Ausdrücke aufbauen. Der Autor arbeitet daher an einer Weiterentwicklung von wurfl2. Darüber hinaus ist eine Umstellung von der Standard-WURFL-PHP-Bibliothek auf TerawURFL [7] geplant. TerawURFL speichert die Gerätebeschreibungen von WURFL in einer MySQL-Datenbank und ist damit deutlich schneller. Da TYPO3 ohnehin eine Datenbank voraussetzt, liegt dieser Schritt nahe.

Links und Literatur

Softlink 1461

- [1] TYPO3 Conditions: http://typo3.org/documentation/document-library/references/doc_core_tseef/4.0.0/view/4/1/
- [2] WURFL 2. Extension für TYPO3: <http://typo3.org/extensions/repository/view/wurfl/2.0.8.1/>
- [3] Template-Doku: [http://typo3.org/documentation/document-library/tutorials/doc_tut_template_dercurrent/library/tutorials/doc_tut_template_dercurrent/](http://typo3.org/documentation/document-library/core-documentation/doc_core_templates/current/library/tutorials/doc_tut_template_dercurrent/library/tutorials/doc_tut_template_dercurrent/)
- [4] Typo3 Templates: http://typo3.org/documentation/document-library/tutorials/doc_tut_tfb1/gercurrent/
- [5] TypoScript-Doku: http://typo3.org/documentation/document-library/references/doc_core_tseef/4.0.0/view/
- [6] TypoScript-Doku: http://typo3.org/documentation/document-library/references/doc_core_tseef/4.0.0/view/
- [7] TerawURFL: <http://wurfl.sourceforge.net/utilites/>

Luc de Louw

Hochverfügbare Server-Infrastrukturen

Verfügbarkeit rund um die Uhr durch den Einsatz von zwei Servern und DRBD

Die beste Content-Management-Software und das beste Webdesign sind nutzlos, wenn der Webserver nicht erreichbar ist. Durch die zunehmende Globalisierung sind Unternehmen vermehrt auf Onlineapplikationen angewiesen, die 24 Stunden am Tag verfügbar sind. Dank der hier vorgestellten reinen Software-Lösung ist diese hohe Verfügbarkeit selbst für kleine und mittelständische Unternehmen erschwinglich.

INFO:ZUM TUTORIAL	Experten
Schwierigkeitsgrad	4 Stunden
Voraussetzungen/Zeitaufwand	Zwei Rechner oder VMware Workstation mit installiertem Linux und einer gleich großen unformatierten Partition.
Hinweise/Voraussetzungen	

Das Ziel dieses Tutorials ist ein lauffähiger, hochverfügbarer LAMP (Linux, Apache, MySQL, PHP) Stack auf zwei Servern. Für den Test reichen zwei ältere Rechner oder zwei VMware-Instanzen. VMware Workstation 5.5 bietet außerdem „Teaming“ an, was für Experimente mit diesem Tutorial bestens geeignet ist. Das Linux-System sollte Apache, MySQL und PHP bereits vorinstalliert haben. Wie diese Komponenten installiert und konfiguriert werden, wird im Handbuch der jeweiligen Distribution erklärt. Die zum Einsatz kommende Distribution sollte einen Kernel der Serie 2.6 enthalten.

Was ist ein HA-Cluster?

Jede Art von Cluster ist eine Zusammenarbeit von zwei oder mehr Rechnern in einem Verbund. In diesem Tutorial wird das Aufsetzen eines HA (High Availability, hochverfügbaren) Clusters mit zwei Rechnern beschrieben.

Wieso hochverfügbare Server eingesetzt werden

Durch redundante vorhandene Server übernimmt im Falle eines Hard- oder Softwaredefekts automatisch ein überlebender Server die Dienste. Dadurch wird Zeit gewonnen, um einen defekten Server reparieren zu können. Bei geplanten Wartungsarbeiten, wie zum Beispiel Hardware-Erweiterungen oder der Installation eines neuen Linux-Kernels, können die Dienste manuell durch ein kontrolliertes „Switchover“ auf den zweiten Server verlegt werden.

Kleine und mittlere Unternehmen profitieren von der erhöhten Verfügbarkeit, da für Kunden die Website immer zur Verfügung steht. Außerdem kann auf Personal im Bereitschaftsdienst verzichtet werden, da beispielsweise ein Serverausfall mitten in der Nacht automatisch überbrückt wird.

Großbetriebe, welche schon vorher hardwarebasierte HA-Systeme einsetzen, profitieren bei reinen Softwarelösungen durch erhebliche Einsparungen bei den Investitions- und Wartungskosten.

Was bei der Wahl der Hardware zu beachten ist

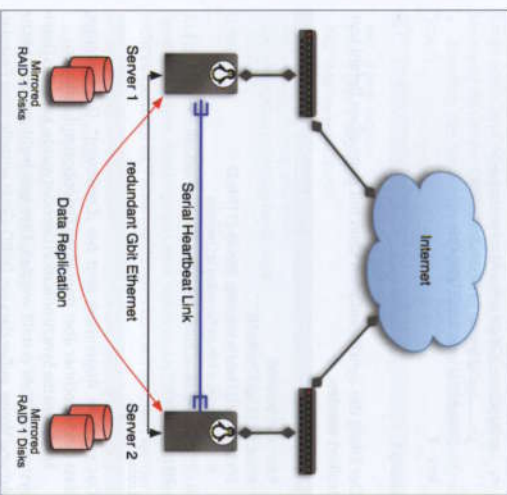
Trotz redundanter Serverinfrastruktur sollten die einzelnen Geräte über jeweils redundante Festplatten, Netzteile und Kühlsysteme verfügen. Dadurch können die Verfügbarkeit und Zuverlässigkeit weiter gesteigert werden.

Die zum Einsatz kommende Software läuft praktisch auf allen Systemen, für die aktuelle Linux-Distributionen erhältlich sind.

Auf produktiven Systemen ist der Einsatz von Gigabit Ethernet notwendig, um bei der Echtzeitreplikation keine größeren Leistungseinbußen zu erleiden.

Systemaufbau und Funktionsweise

Der Aufbau des Systems besteht aus zwei Servern, die zusätzlich zur üblichen Netzwerkanbindung über eine dedizierte Gigabit Ethernet-Leitung (Crossover) und ein Nullmodem-Kabel miteinander verbunden sind.



Schematische Darstellung eines HA-Clusters

Die dedizierte Gigabit-Ethernet-Verbindung wird für die Echtzeitreplikation der DRBD-Geräte benötigt, während das Nullmodemkabel der zusätzlichen Sicherheit für die Serverüberwachung dient.

DRBD

DRBD [1] ist ein Linux-Kernelmodul, das eine zusätzliche Schicht zwischen einem Blockgerät und dem Dateisystem implementiert. Für das Dateisystem spielt es keine Rolle, ob das darunterliegende Gerät eine Festplatte oder ein DRBD-Gerät ist. Dank dieser Tatsache ist DRBD für alle Anwendungen transparent.

Bei jedem Schreibzugriff auf die Festplatte repliziert DRBD die geschriebenen Blöcke auf den zweiten Server. Der so genannte „write-commit“ erfolgt, sobald auf beiden Servern die Datenblöcke erfolgreich auf die Festplatte geschrieben wurden. Durch die synchrone Datenspiegelung wird eine Datenkonsistenz auf Blockgeräteebene sichergestellt.

Heartbeat

Heartbeat [2] ist die Software, die für die gegenseitige Überwachung der beiden Server zum Einsatz kommt. In regelmäßigen Intervallen kommunizieren die beiden Server miteinander, sie

„pingen“ sich an. Kommt keine Antwort innerhalb des konfigurierten Zeitrahmens, nimmt der passive Server an, dass der aktive nicht mehr verfügbar ist. In diesem Szenario werden automatisch die entsprechenden „Failover“-Maßnahmen ergriffen. Diese Maßnahmen sind in der Konfigurationsdatei von Heartbeat festgelegt. Üblicherweise behalten sie das Aktivieren der lokalen DRBD-Blockgeräte, das Einhängen der Dateisysteme und den Start der verschiedenen Dienste.

Annahmen

In dem Konfigurationsbeispiel werden folgende IP-Adressen, Rechnernamen und Dateisysteme symbolisch verwendet:

- „links“ ist der Rechnername des primären Rechners
- „rechts“ ist der Rechnername des sekundären Rechners
- Als „öffentliches“ Netzwerk wird 10.0.0.0 angenommen
- Als „privates“ Netzwerk wird 192.168.0.0 angenommen
- Das DRBD-Blockgerät wird in /data eingehängt
- Das DRBD-Blockgerät ist /dev/drbd0
- Die Ressource heißt r0

Entsprechend der vorhandenen Netzwerkkonfigurationen im Rechenzentrum muss das öffentliche Netzwerk angepasst werden. Die angenehmen Rechnernamen sind frei wählbar und sollten bei der Installation des Basisbetriebssystems festgelegt werden, können aber mit etwas Aufwand auch nachträglich geändert werden. Wichtig ist, dass sowohl DRBD als auch Heartbeat mit den Rechnernamen konfiguriert werden müssen.

Installation und Konfiguration von DRBD

Als erstes müssen auf beiden Rechnern die entsprechenden Partitionen angelegt werden. Diese können mit dem Befehl fdisk oder einem anderen distributionsabhängigen Werkzeug erstellt werden.

Zu beachten ist, dass DRBD für die Speicherung der Metadaten 128MByte von der Partition für sich beansprucht.

Als nächstes wird DRBD installiert und konfiguriert. Die meisten Linux-Distributionen verfügen über vorkompilierte DRBD-Pakete. Bis zum Erscheinen von DRBD8 sollten nur die aktuellsten Versionen aus dem 0.7.x-Zweig verwendet werden. Sind bei der eingesetzten Distribution keine entsprechenden Pakete verfügbar, kann der Quellcode [3] heruntergeladen werden. Um den Quellcode übersetzen zu können, müssen der Kernelquellcode, das Programm „make“ sowie der GCC Compiler installiert sein.

SHELL

```
links--# tar xzf drbd-0.7.x.tar.gz
links--# cd drbd-0.7.x
links--# make && make install
```

Listing 1

Die entsprechenden Pakete haben bei verschiedenen Distributionen unterschiedliche Namen. Wie diese konkret heißen, kann im Handbuch der entsprechenden Distribution nachgeschlagen werden. Die GCC-Version, die verwendet werden soll, muss mit der Version identisch sein, mit der der eingesetzte Kernel übersetzt wurde.

Kompilieren und Installieren von DRBD

Wenn die Installation nach /etc/drbd.conf geschriebene, wird eine Beispielkonfiguration nach /etc/drbd.conf geschrieben. Diese muss den entsprechenden Systemen angepasst werden. Wie in den Voraussetzungen beschrieben, wird je eine gleich große, unformatierte Partition auf beiden Servern benötigt. Diese Partition und die Netzwerkkonfiguration müssen angepasst werden. Die zu ver-