

## Ein Asterisk für den Katastrophenfall Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

von Daniel Meinhold, Dominik Zöller



Die Open Source Telefonanlage Asterisk wird zunehmend auch für kleine und mittlere Unternehmen interessant und steht dort in direkter Konkurrenz zu kommerziellen Lösungen von bekannten Herstellern wie z.B. Alcatel-Lucent, Cisco oder Siemens.

Dabei werden im Unternehmenseinsatz hohe Anforderungen an die Erreichbarkeit der Telefonie-Infrastruktur gestellt. Die Be-

nutzer erwarten eine nahezu 100%-ige Verfügbarkeit. Ein Ausfall kann je nach Unternehmen gravierende Konsequenzen nach sich ziehen. Hersteller wie Siemens können beim Thema Hochverfügbarkeit (engl.: High Availability, abgekürzt HA) auf jahrzehntelange Entwicklungsarbeit zurückgreifen. Dabei bezeichnet Hochverfügbarkeit die Fähigkeit eines Systems, bei Ausfall einer Komponente einen uneingeschränkten Betrieb zu gewährleisten. Im

Gegensatz zu Open Source Lösungen hat die Entwicklung von Hochverfügbarkeits-Lösungen bei den kommerziellen Herstellern einen hohen, wenn nicht sogar entscheidenden Stellenwert. Dies gilt jedoch nicht ohne Einschränkungen: Zum Beispiel hat Cisco seinen Communications Manager 6.0 nicht für den Einsatz in Feuerwehreinrichtungen und ähnlich kritischen Einsatzorten freigegeben.

## Schwerpunktthema

## Ein Asterisk für den Katastrophenfall

### Hochverfügbare Telefonie auf Basis einer Open Source Lösung?



Dipl.-Inform. Daniel Meinhold ist Consultant bei der ComConsult Beratung und Planung GmbH. Bereits während seines Studiums hat er sich intensiv mit Linux, Virtualisierung und hochverfügbaren Server-Lösungen befasst. Bei ComConsult hat er sich daher sehr schnell auf Hochverfügbarkeitslösungen im Telekommunikationsumfeld spezialisiert. Im Competence Center ist er vor allem für die Konzeption von Telekommunikationslösungen sowie für die Durchführung von Tests und Messungen verantwortlich.



Dominik Zöllner ist seit 2006 Berater bei der ComConsult Beratung und Planung. Während seines Studiums konzentrierte er sich auf die Themengebiete der Kommunikationsnetze und der Betriebssysteme. Bei ComConsult ist er vorwiegend mit der Evaluierung, Planung und Ausschreibung professioneller Unified Communications, Kollaborations- und Video-Konferenz-Systeme befasst.

#### Open Source: Vielfalt vs. Qual der Wahl

Solche Ausnahmen bestätigen aber die Regel: Bei allen namhaften, kommerziellen Herstellern findet man auch entsprechende Hochverfügbarkeits-Lösungen vor. Demgegenüber hat man bei Open Source Software eine fast unüberschaubare Vielzahl an Projekten, deren Vor- und Nachteile auf den ersten Blick nicht zu erkennen sind. Hat man sich für ein Projekt entschieden, muss man entweder über das Fachwissen verfügen, dieses zu realisieren oder einen Systemintegrator finden, der diese Aufgabe sowie den Support übernimmt. Dank Open Source hat man jedoch auch die Möglichkeit, Änderungen und Erweiterungen vorzunehmen. Dies ist mit proprietären Lösungen nicht möglich.

#### Hochverfügbarkeit mit Asterisk: Problematisch

In der relativ kurzen Entwicklungszeit von Asterisk wurde das Thema Hochverfügbarkeit nur unzureichend berücksichtigt, was sich anhand folgender Probleme zeigt:

##### 1. Asterisk besitzt keine HA-Option

Eine Hochverfügbarkeits-Option ist – im Gegensatz zu anderen Open Source Lösungen wie z.B. sipX -- nicht integraler Bestandteil von Asterisk. Bei Asterisk muss

auf externe Mittel zurückgegriffen werden. Dies führt zum zweiten Problem: der Dokumentation.

##### 2. Unzureichende Dokumentation

Eine eigenhändige Umsetzung eines hochverfügbaren Asterisk-Systems gestaltet sich schwierig, da es an Dokumentation mangelt. Dies betrifft sowohl die Übersicht der möglichen Szenarien als auch Details zur Umsetzung. Asterisk bzw. dessen Hersteller Digium liefert hierzu weder Anleitungen, wie im Fall von sipX, noch werden in der mitgelieferten Dokumentation Hinweise auf weitere Quellen aufgeführt. Recherchen im Internet zu dem Thema Asterisk und Hochverfügbarkeit verweisen meist auf <http://www.voip-info.org>. Der dortige Link zu High-Availability liefert jedoch nur eine unsortierte Liste von Stichwörtern, teilweise mit unvollständigen und/oder veralteten Informationen. Eine differenzierte Übersicht von Lösungen, inklusive Details zur Umsetzung, hingegen fehlt. Dies betrifft jedoch nicht nur Quellen im Internet, auch in der Literatur wird das Thema Hochverfügbarkeit mit Asterisk nahezu vollständig ausgeblendet.

#### HA-Szenarien mit Asterisk

Will man eine HA-Lösung für Asterisk im Unternehmen realisieren, sind verschiedene Lösungsansätze denkbar. Diese unter-

scheiden sich im Konzept, im technischen Aufwand und in der Eignung für unterschiedliche Anforderungen im Unternehmen. Im Folgenden sollen drei denkbare Szenarien näher erläutert werden.

#### Szenario 1: Failover mit DNS SRV-Records

Die laut Henning Schulzrinne und Kundan Singh, den Mitentwicklern von SIP, technisch sauberste und daher bevorzugte Art und Weise, eine redundante SIP-Umgebung zu schaffen, bedient sich des DNS-Protokolls.

Der Client nutzt hierzu Naming Authority Pointer (NAPTR) und DNS Service (SRV) Resource-Records, um den zuständigen SIP-Server zu lokalisieren (Abbildung 1). Ist dieser nicht erreichbar, wird automatisch ein anderer Server gewählt. Generell kann mittels SRV-Resource-Records per DNS propagiert werden, welche Dienste eine Domäne anbietet und unter welcher Adresse diese erreichbar sind.

Neben der DNS SRV-Unterstützung für ausgehende Gespräche (Abbildung 1) müssen die Clients auch eine Registrierung mittels DNS SRV ermöglichen (Abbildung 2).

Bevor ein Client eine Anfrage versenden

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

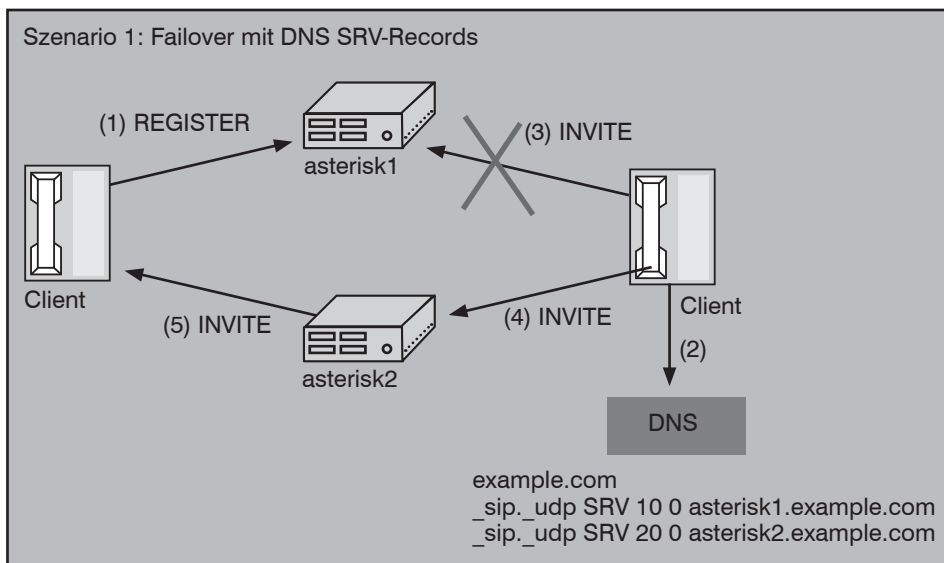


Abbildung 1: Failover mit DNS SRV-Records

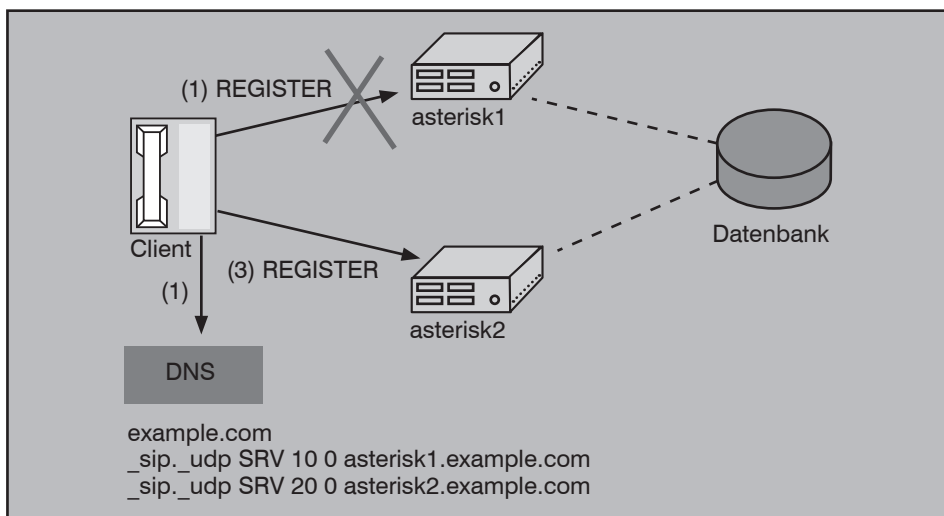


Abbildung 2: Registrierung mit DNS SRV-Records

möchte, welche eine SIP/SIPS-URI beinhaltet (z.B. INVITE), müssen folgende Schritte durchlaufen werden:

1. Auswahl des Transportprotokolls
2. Ermitteln von IP-Adresse und Port

Sofern im SIP-Header keine Angaben zum Protokoll oder der Portnummer angegeben sind, führt der Client eine NAPTR-Anfrage durch. Der Client erhält daraufhin eine Auswahl der zur Verfügung stehenden Transportprotokolle (Abbildung 3). Da Asterisk noch keine TCP-Unterstützung bietet, entfallen die Einträge für SIP über TCP und TLS. Als Transportprotokoll steht in diesem Fall nur UDP (SIP+D2U) zur Verfügung.

Der Client nutzt das Ergebnis der NAPTR-

Anfrage und stellt anschließend eine DNS SRV-Anfrage an `_sip._udp.example.com` um die zuständigen Server zu ermitteln. Sofern keine NAPTR-Einträge vorhanden sind, werden direkt DNS SRV-Anfragen mit den vom Client unterstützten Transportprotokollen gesendet (Beispiel: `_sip._udp.example.com` und `_sip._tcp.example.com`). Nachdem der Client die DNS SRV-Antwort nach Priorität (Abbildung 4, dritte Spalte)

```
IN NAPTR 100 50 "s" "SIP+D2U" "" _sip._udp.example.com.
```

Abbildung 3: DNS NAPTR-Eintrag für Asterisk und SIP über UDP (SIP+D2U)

```
IN SRV 10 0 5060 asterisk1.example.coma
IN SRV 20 0 5060 asterisk2.example.com
```

Abbildung 4: DNS SRV-Antwort für eine Anfrage nach `_sip._udp.example.com`

sortiert hat, wird er versuchen den ersten Server zu kontaktieren. Für das Beispiel aus Abbildung 4 ist dies `asterisk1.example.com` auf Port 5060 über UDP. Ist der Server mit der höheren Priorität nicht erreichbar, wird der nächste Server kontaktiert. In diesen Fällen wird eine nahezu identische Anfrage an den nächsten Server geschickt. Für Abbildung 4 ist dies `asterisk2.example.com`.

Eine Lösung auf Basis von DNS SRV zeichnet sich durch folgende Eigenschaften aus:

- Einfacher Mechanismus: Im Vergleich zu anderen Lösungen ist die Architektur des DNS-basierten Ansatzes relativ einfach. Klassische Failover-Cluster haben eine deutlich höhere Komplexität. Da die DNS-Server üblicherweise redundant ausgelegt sind, wird durch den DNS SRV-Mechanismus keine zusätzliche Fehlerquelle hinzugefügt.
- Standortunabhängigkeit: Die Flexibilität bei der Wahl des Serverstandortes ist höher als bei einem HA-Cluster, der meist durch physikalische Voraussetzungen (Cluster-Interconnect, Netz-Topologie) an bestimmte räumliche Grenzen gebunden ist. Der Server kann bei einer DNS SRV-Konfiguration nahezu beliebig positioniert werden.
- Einfache Änderung der Client-Konfiguration: DNS NAPTR und SRV-Einträge ermöglichen es, alle verbindungsrelevanten Daten (Protokoll, Port, Hostname) ohne Modifikation der Clients zu ändern. Diese holen sich die aktualisierte Konfiguration automatisch bei der nächsten DNS NAPTR/SRV-Anfrage.
- Hohe Skalierbarkeit: Die Anzahl der Server lässt sich fast beliebig erhöhen. Man könnte z.B. einen weiteren Server `asterisk3.example.com` mit der Priorität 30 hinzufügen. Eine zusätzliche Konfiguration der Gewichtung ermöglicht eine statische Lastverteilung, wofür jedoch ein gemeinsames Backend für die SIP-Registrierung zwingend erforderlich ist. Ansonsten kann keine Kommunikation zwischen Clients stattfinden, die an

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

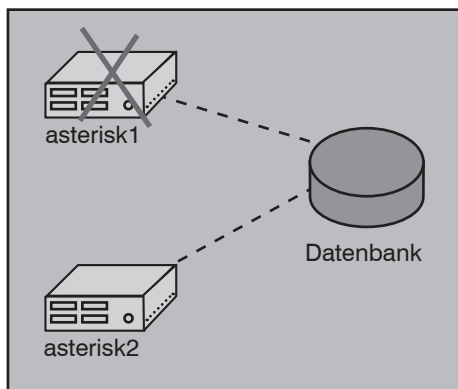


Abbildung 5: Gemeinsames Backend in Form einer Datenbank sorgt für einheitliche Konfiguration

unterschiedlichen Asterisk-Servern registriert sind.

- Konsistenter Datenbestand erforderlich: Kommt es zu einem Failover, wird eine konsistente Konfiguration auf dem zweiten Server erwartet. Fehlende Leistungsmerkmale, inkonsistente Wählpläne oder nicht vorhandene Teilnehmer-Konfigurationen sind inakzeptabel. Ein konsistenter Datenbestand muss, im Gegensatz zu typischen HA-Clustern, separat umgesetzt werden. Im einfachsten Fall geschieht dies über selbstprogrammierte Skripte. Besser geeignet ist ein gemeinsames Backend in Form einer Datenbank (Abbildung 5). Allerdings muss der Datenbank-Server zusätzlich redundant ausgelegt werden. Hierfür kann entweder ein bereits vorhandener Datenbank-Server bzw. -Cluster genutzt oder ein Master/Slave-Verbund auf den Asterisk-Knoten realisiert werden. Asterisk nutzt für den Zugriff auf die Datenbank die Asterisk RealTime Architecture (ARA). Diese ermöglicht es Asterisk, beliebige Konfigurationsdateien in eine Datenbank auszulagern. Für alle weiteren Daten, wie beispielsweise Verbindungsdaten (Call Detail Records, abgekürzt CDRs), Voice-Mails oder die Endgerätekonfiguration, muss eine andere Lösung gefunden werden.

- PSTN-Anbindung: Wird zusätzlich die Anbindung an das PSTN über analoge oder digitale Leitungen (BRI/PRI) gewünscht, wird häufig ein Layer1 Failover-Switch in Kombination mit einer TDM-Schnittstellenkarte eingesetzt. In einem HA-Cluster sorgt die Failover Management Software (FMS) für die Zuweisung des PSTN-Anschlusses an den jeweils aktiven Server. Dies ist bei der DNS SRV-Variante nicht möglich. Eine Lösung ist die Ausgliederung der PSTN-Funktionalität in ein separates Gateway, welches ebenfalls redundant ausgelegt

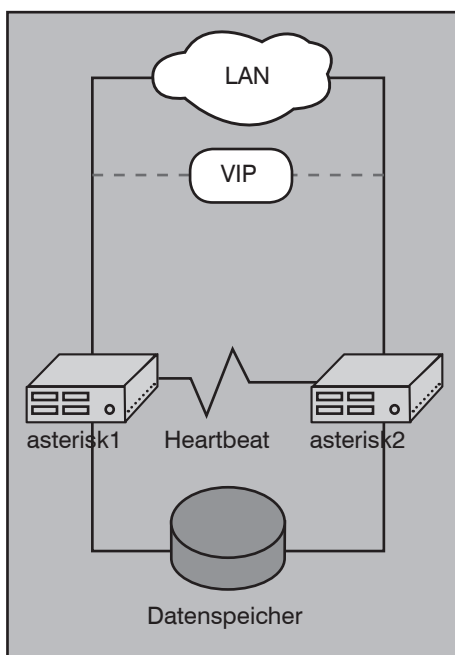


Abbildung 6: Aufbau eines typischen 2-Knoten HA-Clusters

und angebunden werden muss. Dieses wird anschließend über den Wählplan der Asterisk-Server adressiert.

**Szenario 2:  
HA-Cluster - Der Klassiker**

Eine klassische Maßnahme zur Steigerung der Verfügbarkeit ist der Einsatz von HA-Clustern (Abbildung 6). Auch wenn die Abbildung, der Übersicht halber, keine redundanten Anbindungen zeigt, ist dies jedoch

eine wichtige Bedingung eines Clusters, um einzelne Fehlerstellen (Single Point Of Failures, abgekürzt SPOFs) zu vermeiden.

Hier wird die Redundanz auf Server-Ebene hergestellt. Kommt es zum Ausfall eines Servers (Primärsystem), übernimmt automatisch ein zweiter Server die Aufgaben (Standby-System). Es empfiehlt sich eine Aktiv/Passiv-Konfiguration, so dass der zweite Server nur als Standby-System dient und keine weiteren Dienste zur Verfügung stellt. Dies erhöht die Chancen für einen erfolgreichen Failover.

**Ist Asterisk clustertauglich?**

Prinzipiell eignet sich jede Applikation zum Einsatz auf einem HA-Cluster. Damit aber ein sinnvoller Einsatz möglich ist, müssen einige Bedingungen erfüllt werden. Man unterscheidet daher zwischen clusterfähigen und nicht-clusterfähigen Applikationen. Eine Applikation gilt als vollständig clusterfähig, wenn folgende Bedingungen zutreffen:

- Unterstützung für Shared Storage: Die Applikation muss in der Lage sein, von einem SAN oder NAS zu starten.
- Dienstprogramme zum Starten/Stoppen/Prüfen: Es müssen Dienstprogramme für die Kommandozeile zur Verfügung stehen, die es ermöglichen, die Applikation einfach zu starten, zu stoppen, sowie den Status der Applikation zu prüfen.

**Jetzt Leser werden**

**Der Netzwerk Insider**

Der Netzwerk Insider erscheint 12 Mal im Jahr im PDF-Format und informiert Sie per eMail über die Hintergründe aktueller Netzwerk-Technologien. Jeden Monat werden zwei Themen gewählt, über die in ausführlicher Form topaktuelle Insider-Informationen gegeben werden. Der Netzwerk-Insider vertritt die Sichtweise von Technologie-Anwendern und bewertet Produkte und Technologien im Sinne der wirtschaftlichen und erfolgreichen Umsetzbarkeit in der täglichen Praxis. Durch seine strenge wirtschaftliche Unabhängigkeit (keine Hersteller-Anzeigen) kann er es sich leisten, Schwachstellen und Nachteile offen anzusprechen. Der Netzwerk-Insider ist bekannt für seine kritische, herstellerneutrale und fundierte Technologie-Bewertung.

Hier können Sie sich zum Netzwerk Insider kostenlos und ohne jede Verpflichtung registrieren lassen:

<http://www.comconsult-akademie.de/de/Registrierung.php>

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

- **Zustandsspeicherung:** Der Zustand einer Applikation sollte weitestgehend auf nichtflüchtigem Datenspeicher (z.B. Shared Storage) festgehalten werden. Kommt es zu einem Failover, sollte die Applikation im Idealfall den vorherigen Zustand wiederherstellen.
- **Keine Datenkorruption:** Sofern die Applikation abrupt gestoppt wird, sei es z.B. durch ein KILL-Signal oder einen Stromausfall, darf dies nicht zu einer Datenkorruption führen. Dies bedeutet, ein anschließender Neustart der Applikation muss ohne zusätzliche (manuelle) Wartungsaufgaben durchführbar sein.

Eine clusterfähige Applikation kann weiterhin unterteilt werden in cluster-aware oder cluster-unaware, je nachdem ob der Applikation bekannt ist, dass sie in einem Cluster gestartet wurde oder nicht.

Asterisk ist dabei als (eingeschränkt) clusterfähig und cluster-unaware zu bezeichnen. Die Einschränkung betrifft die Zustandsspeicherung: Wichtige Daten werden nicht auf einem Festspeicher vorgehalten. Kommt es zu einem Failover, gehen diese Informationen zwangsläufig verloren. Dazu gehört beispielsweise, unter welcher IP-Adresse ein Endgerät erreichbar ist. Diese Information erhält Asterisk im Rahmen der REGISTER-Methode zusammen mit einer optionalen Authentifizierung. Für die Endgeräte bedeutet dies einen Ausfall, solange der Zähler für die erneute Registrierung (registration timer) nicht abgelaufen ist. Eine denkbare Lösung hierfür ist die Verwendung einer SQL-Datenbank mittels Asterisk RealTime. Die weiteren, oben aufgeführten, Bedingungen werden hingegen erfüllt.

**Datenspeicher: Shared-Nothing vs. Shared-Disk**

Als Datenspeicher eignet sich sowohl eine Shared-Nothing-, als auch eine Shared-Disk-Konfiguration.

Ist bereits eine hochverfügbare SAN-Umgebung (Abbildung 7) vorhanden, bietet sich eine Mitbenutzung für die Asterisk-Lösung an. Die Anbindung an das SAN sollte ebenfalls redundant ausgelegt sein (Multipathing), d.h. jeder Knoten benötigt zwei Host Bus Adapter (HBA).

Im Rahmen einer TK-Installation ist eine SAN-Investition jedoch übertrieben. Besser geeignet ist eine Replikation auf Blockgerät-Ebene mittels Distributed Replicated Block Device (DRBD).

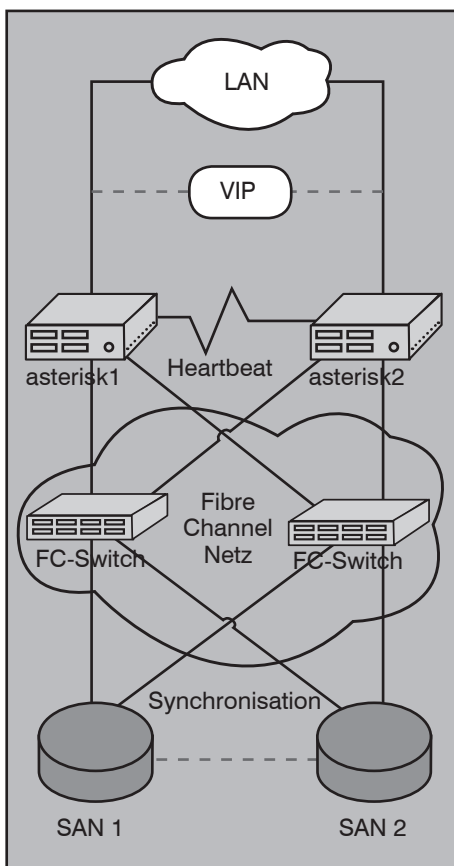


Abbildung 7: Hochverfügbare SAN-Umgebung für einen Asterisk HA-Cluster

**DRBD - Festplattenspiegelung übers Netzwerk**

Die Open Source Software DRBD von Philipp Reisner gilt in der Linux-Welt als erprobtes und etabliertes Werkzeug, wenn es um Replikation auf Blockgerät-Ebene geht. Hierbei werden alle Schreibzugriffe in Echtzeit via TCP/IP auf einen zweiten Server übertragen. Kommt es zu einem Ausfall des Primärservers, kann der zweite Server mit den gleichen Daten weiterarbeiten.

Die Anzahl der benötigten Infrastruktur-Komponenten ist im Gegensatz zu einer redundanten SAN-Infrastruktur überschaubar. DRBD benötigt nur eine IP-Verbindung zwischen den Knoten. Hierfür genügt im einfachsten Fall eine Direktverbindung zwischen den Knoten, die durch ein Crossover-Kabel hergestellt wird. Dies erhöht zum einen die Ausfallsicherheit, da weitere Fehlerquellen wie Hubs oder Switches vermieden werden. Zum anderen bietet es Schutz vor unerlaubtem Zugriff.

Weiterhin benötigen die beteiligten Knoten eine separate Partition für die Nutzung von DRBD, bevorzugt ein RAID-System. Dieses wird zu einem logischen Blockgerät /dev/

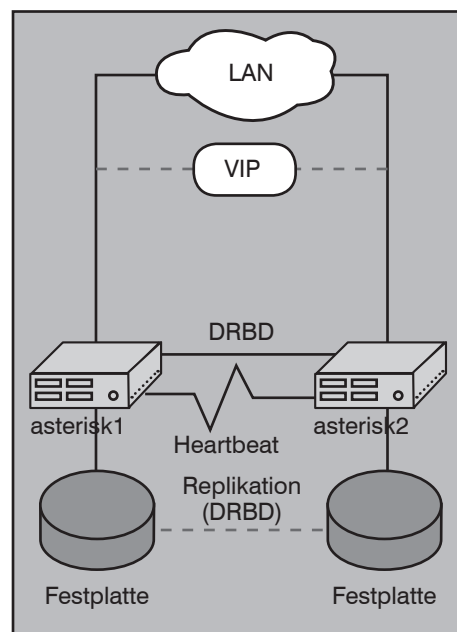


Abbildung 8: Hochverfügbare Shared-Nothing-Umgebung mit DRBD

drbdX verknüpft und kann anschließend mit einem Dateisystem versehen und in das System eingebunden werden.

**Cluster-Administration: ein Fall für „Heartbeat“**

Die Überwachung und Verwaltung des Clusters wird von der Failover Management Software übernommen. Im Fall von Linux gilt die Software „Heartbeat“ des Linux-HA Projekts als De-facto-Standard. Heartbeat steht unter der GPL-Lizenz und wird vorrangig auf Linux-Systemen eingesetzt. Dank eines portablen Designs läuft die Software jedoch auch unter Solaris, FreeBSD, OpenBSD und Mac OS X. Das Projekt wird dabei von Unternehmen wie IBM, Intel, Novell/SuSE Linux oder SGI unterstützt.

Die Heartbeat-Software liegt mittlerweile in Version 2 vor und leistet neben der Lebendüberwachung anhand von Heartbeat-Nachrichten weitaus mehr, als der Name vermuten lässt; ihr obliegt die gesamte Verwaltung und Überwachung des Clusters. Darunter finden sich in Heartbeat Version 2 die folgenden Punkte (s.a. Tabelle 1):

- **Ressourcen-Überwachung:** Neben der Überwachung, ob ein Knoten aktiv ist, erlauben sogenannte Ressourcen-Agenten (RA) auch eine Überprüfung, ob ein Dienst korrekt arbeitet.
- **Fencing-Mechanismen:** Heartbeat unterstützt in Umgebungen mit Zugriff auf gemeinsame Ressourcen (z.B. einem

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

Kriterium	Heartbeat Version 1	Heartbeat Version 2
Anzahl Knoten	max. 2	kein Maximum
Überwachung der Knoten	Ja	Ja
Überwachung der Ressourcen	Nein	Ja
Unterstützung von Master/Slave-Ressourcen	Nein	Ja
Unterstützung von Randbedingungen	gering	umfangreich
Unterstützung von Abhängigkeiten	gering	umfangreich
Unterstützung für Self-Fencing Ressourcen	Ja	Ja
Unterstützung für STONITH	Ja	Ja
Unterstützung durch Entwickler	gering	umfangreich
Verwaltung über GUI	Nein	Ja
Zeitbasierte Regeln	Nein	Ja
Architektur	monolithisch	modular
Replizierte Konfiguration	Nein	Ja
Komplexität	gering	hoch

Tabelle 1: Funktionsübersicht Heartbeat Version 1 vs. Version 2

SAN) das Aussperren von Ressourcen bzw. Knoten (engl.: resource fencing bzw. node fencing). Auf Knoten-Ebene hat sich Shoot The Other Node In The Head (STONITH) als Fencing-Methode bewährt; STONITH sorgt bei Erkennung eines unbekanntenen Status des anderen Knotens für Gewissheit, indem die Stromzufuhr des betroffenen Knotens gekappt wird (Abbildung 9). Eine entsprechende Heartbeat-API erlaubt die Programmierung eigener STONITH-Plugins.

- Zeitbasierte Regeln: Zeitbasierte Regeln ermöglichen es, einen genauen Zeit-

plan darüber zu erstellen, wann welche Ressource wo aktiv ist.

- Randbedingungen: Durch die Angabe von Randbedingungen kann z.B. die Platzierung einer Ressource definiert werden.
- Ressourcen-Abhängigkeiten: Die Konfiguration von Abhängigkeiten zwischen den Ressourcen gewährleistet die korrekte Reihenfolge beim Starten und Stoppen der Ressourcen.

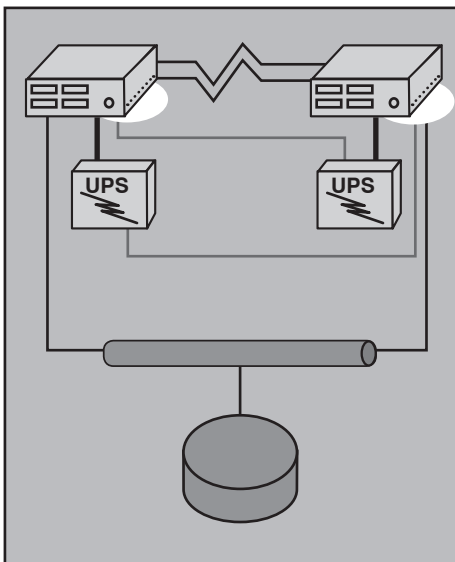


Abbildung 9: STONITH verhindert Datenkorruption in einer Shared-Disk-Architektur (z.B. SAN)

Zutaten für Asterisk HA-Cluster

Für einen Asterisk HA-Cluster sind vor allem vier wichtige Ressourcen zu behandeln:

1. Virtuelle IP (VIP)
2. Datenspeicher (z.B. Shared-Disk oder DRBD)
3. Dateisystem
4. Asterisk

Dabei ist besonders die Ressourcen-Überwachung von Bedeutung. Heartbeat Version 2 ermöglicht neben der Knoten-Überwachung durch Heartbeat-Nachrichten auch die Verwendung von Ressourcen-Agenten, welche dienstspezifische Überprüfungen durchführen können.

Asterisk abgestürzt?

Der Ressourcen-Agent überprüft anschließend in konfigurierbaren Zeitintervallen den Zustand von Asterisk. Kommt es zu einem Fehler, wird automatisch ein Neustart von Asterisk veranlasst. Die Heartbeat-Software erlaubt auch die Konfiguration eines Fehlerzählers, so dass z.B. nach dem zehnten Absturz von Asterisk auf Knoten 1 automatisch ein Failover zu Knoten 2 durchgeführt wird.

Überwachung durch Agenten

Neben dem Asterisk-RA werden weitere Ressourcen-Agenten benötigt, die für folgende Aufgaben zuständig sind:

## Jetzt Leser werden

**Der Netzwerk Insider** Mag. 2008

**Virtualisierung: Sicherheit in virtuellen Server-Umgebungen**  
von Dipl.-Inform. Oliver Fric, Dr. Simon Hoff, Dipl.-Inform. Daniel Meinhald

**IT-Verkabelung im Rechenzentrum: Beginn einer neuen Epoche?**  
von Dipl.-Inform. Hartmut Kall

**OCS-Forum 2009**  
**IT-Sicherheits-Forum 2009**

### Der Netzwerk Insider

Der Netzwerk Insider erscheint 12 Mal im Jahr im PDF-Format und informiert Sie per eMail über die Hintergründe aktueller Netzwerk-Technologien. Jeden Monat werden zwei Themen gewählt, über die in ausführlicher Form topaktuelle Insider-Informationen gegeben werden. Der Netzwerk-Insider vertritt die Sichtweise von Technologie-Anwendern und bewertet Produkte und Technologien im Sinne der wirtschaftlichen und erfolgreichen Umsetzbarkeit in der täglichen Praxis. Durch seine strenge wirtschaftliche Unabhängigkeit (keine Hersteller-Anzeigen) kann er es sich leisten, Schwachstellen und Nachteile offen anzusprechen. Der Netzwerk-Insider ist bekannt für seine kritische, herstellerneutrale und fundierte Technologie-Bewertung.

Hier können Sie sich zum Netzwerk Insider kostenlos und ohne jede Verpflichtung registrieren lassen:

<http://www.comconsult-akademie.de/de/Registrierung.php>

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

- Überwachung des Datenspeichers
- Überwachung des Dateisystems
- Überwachung der VIP
- Überwachung der Netzanbindung

Sofern weitere Ressourcen überwacht werden müssen, ist auch hierfür ein entsprechender Ressourcen-Agent nötig.

Im ComConsult IT-Labor wurde ein Asterisk HA-Cluster mit DRBD realisiert und getestet. Abbildung 10 zeigt exemplarisch die grafische Oberfläche der Cluster-Verwaltung.

**Konfiguration der Endgeräte**

An die Endgeräte werden keine besonderen Anforderungen gestellt, wie es bei der DNS SRV-Lösung der Fall ist. Die Endgeräte konfigurieren als zuständigen SIP-Server die VIP des Clusters bzw. den dafür hinterlegten DNS-Namen. Damit ist die Konfiguration abgeschlossen.

**IP-Takeover mit Gratuitous ARP**

Kommt es zu einem Failover, wird die Ressourcengruppe auf den zweiten Knoten migriert. Dazu gehört neben dem Datenspeicher und Asterisk auch die VIP. Da keine virtuellen MAC-Adressen eingesetzt werden, wie dies z.B. beim Virtual Router Redundancy Protocol (VRRP) der Fall ist, muss der Server die neue Zuordnung zwi-

schen VIP und physikalischer MAC durch einen Gratuitous ARP (GARP) bekanntgeben. Sind GARP-Nachrichten zum Beispiel aus Sicherheitsgründen deaktiviert, wird der neue Server erst nach Ablauf des ARP-Caches erreichbar sein. In beiden Fällen wird es zu einer Unterbrechung kommen, bis neue Anrufe getätigt werden können. Im ersten Fall – mit GARP – ist diese allerdings kürzer.

Die folgenden Punkte charakterisieren eine Asterisk-Lösung auf Basis eines HA-Clusters:

- Konsistenter Datenbestand: Im Gegensatz zur DNS SRV-basierten Lösung gibt es beim HA-Cluster einen gemeinsamen Datenbestand, welcher eine zentrale Verwaltung der Konfiguration ermöglicht, ohne auf zusätzliche Komponenten wie Datenbanken angewiesen zu sein.
- HA für weitere Dienste: Neben der zentralen Konfiguration, können auch alle weiteren Daten an einer gemeinsamen Stelle, dem Shared Storage, abgelegt werden. Dazu zählen sowohl Asterisk-spezifische Daten wie z.B. Voice-Mails, als auch Konfigurationsdateien und Daten anderer Dienste, die nicht in einer Datenbank gespeichert werden können. Beispiele für zusätzliche Dienste wären ein HTTP-Server für eine webbasierte

Benutzeroberfläche oder FTP/TFTP-Server für die Konfiguration der Endgeräte.

- Transparenz gegenüber Clients / Interoperabilität: Der gesamte Cluster erscheint den Clients als eine Einheit, die unter einer gemeinsamen Adresse – der VIP – erreichbar ist. Somit liegt die Intelligenz und Komplexität allein beim Cluster. Dies ermöglicht den Einsatz jedes SIP-fähigen Endgerätes und anderer Netzelemente, die nicht über DNS SRV-Unterstützung verfügen. Diese tragen die VIP als SIP-Proxy/Registrar ein und können somit den Dienst in Anspruch nehmen. Diese Transparenz erfordert jedoch eine Layer2-Verbindung zwischen den Clusterknoten (s.a. Standortabhängigkeit).
- PSTN-Anbindung: Wie auch im DNS SRV-Szenario sollte die Anbindung an das öffentliche Telefonnetz mithilfe eines Layer1 Failover-Switch redundant ausgelegt werden (Abbildung 11). Im Gegensatz zu DNS SRV übernimmt hier die Failover Management Software die Verwaltung in Form einer zusätzlichen Ressource. Hierzu gehört auch eine Überwachung der PSTN-Anbindung durch einen Ressourcen-Agenten, um ggfs. Wiederherstellungs-Aktionen einzuleiten. Auf diese Weise ist das Primärsystem auch zuständig für die Anbindung an das öffentliche Telefonnetz.

- Skalierbarkeit: Während bei der DNS

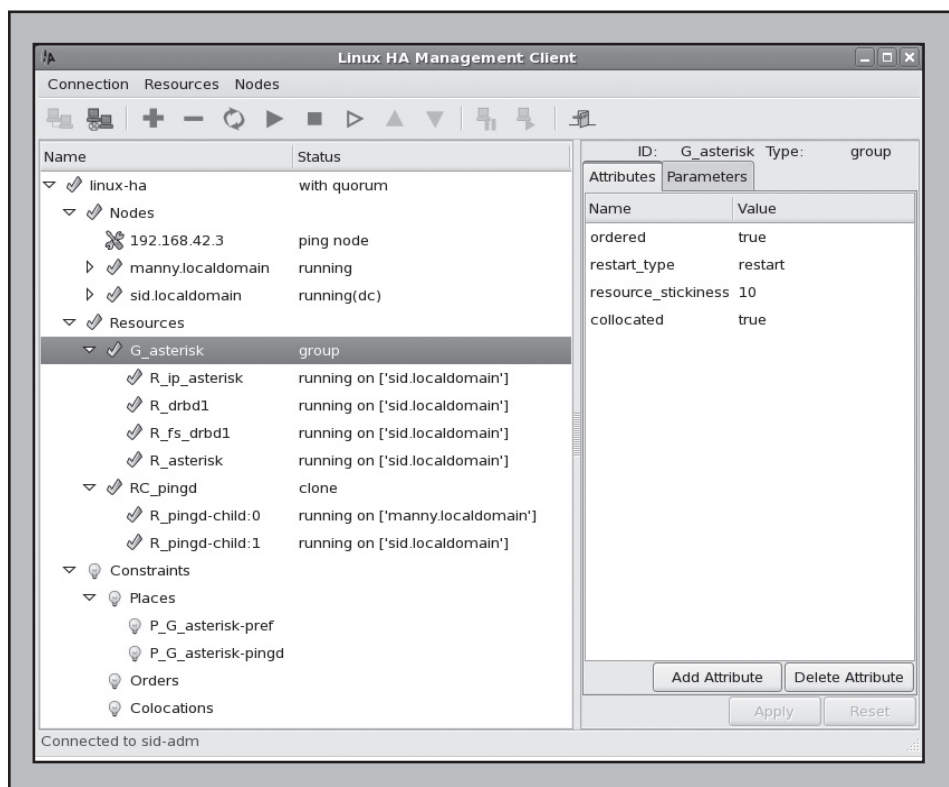


Abbildung 10: Asterisk HA-Cluster: Cluster-Administration mittels Heartbeat-GUI

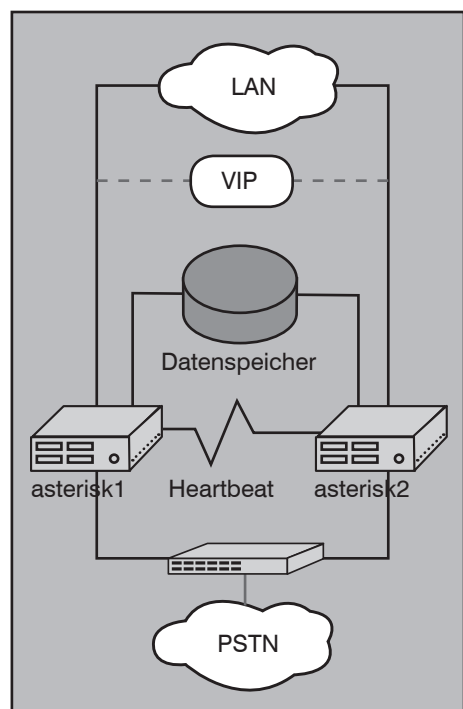


Abbildung 11: Asterisk HA-Cluster mit PSTN-Anbindung via Failover-Switch

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

SRV-basierten Lösung einfach weitere Server hinzugefügt werden können, um eine zusätzliche Lastverteilung zu erzielen, ist man bei einem Aktiv/Passiv-Cluster auf die Kapazität des einzelnen Servers angewiesen. In diesem Fall muss entweder die Hardware ausgetauscht werden oder es müssen Funktionen wie z.B. Voice-Mail oder Konferenzräume auf zusätzliche Server ausgelagert werden.

- Komplexität: HA-Cluster sind komplexe Systeme und erfordern eine genaue Planung und Umsetzung. Themen wie FMS, Split-Brain, Cluster-Interconnect, Shared-Disk und STONITH erfordern entsprechendes Know-how. Eine Umsetzung mittels DNS SRV-Einträgen ist in dieser Hinsicht weniger komplex, da hier vorwiegend die bereits vorhandene DNS-Konfiguration erweitert wird.

- Standortabhängigkeit: Cluster nutzen zur Kommunikation ein dediziertes Netz. Entsprechend sind diesem gewisse Grenzen auferlegt, auch wenn dies im Fall von Heartbeat IP-basiert ist und die Distanz damit theoretisch keine Rolle spielt. Zusätzliche Fehlerquellen und Verzögerungen sind jedoch vor allem bei HA-Clustern sehr kritisch und nach Möglichkeit zu vermeiden.

**Szenario 3: Load Balancing durch zusätzlichen SIP-Server**

Die Forderungen nach Ausfallsicherheit und Skalierbarkeit sind die Domäne der Load-Balancing-Cluster (LB-Cluster). Vor allem im World Wide Web sind diese vertreten; Auftritte von Amazon, Ebay oder Wikipedia wären ohne LB-Cluster undenkbar. Neben verteilten, DNS-basierten Lösungen (Abbildung 12) kommen zentralisierte Lösungen zum Einsatz. Dies ist der Zuständigkeitsbereich des Server Load Balancing (SLB).

Die zentrale Rolle nimmt dabei der Load-Balancer ein. Er dient als Zugriffspunkt für die Clients und verteilt, mittels statischer oder dynamischer Regeln, die Anfragen auf die realen Server. Im Zusammenhang mit den bisher vorgestellten Lösungen vereint Server Load Balancing den Vorteil der Transparenz eines HA-Clusters mit der Skalierbarkeit einer DNS SRV-basierten Lösung.

Besonderes Augenmerk gilt dabei folgenden Themen:

- Load-Balancer
  - Persistenz
  - Algorithmen und Metriken zur Ziel-

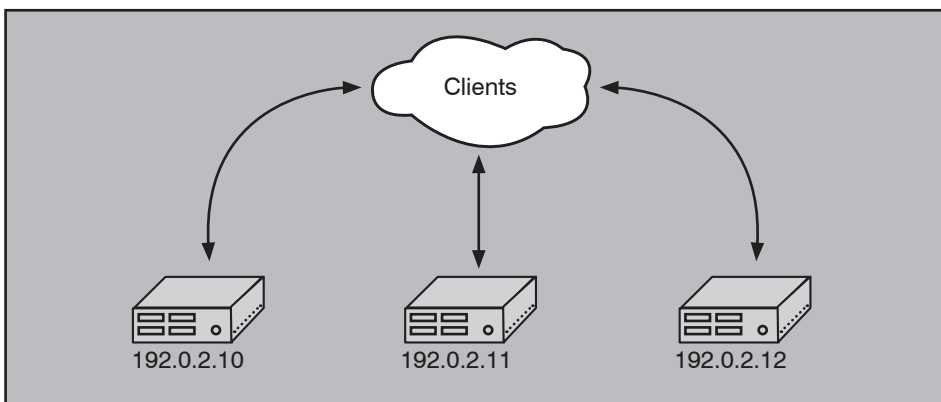


Abbildung 12: Dezentrale Lastverteilung mittels DNS Round-Robin

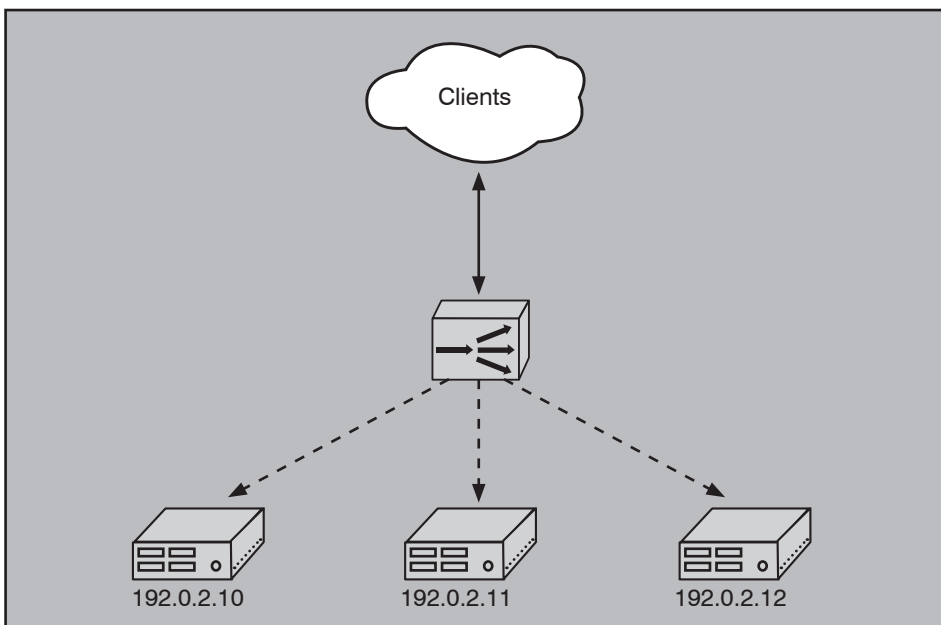


Abbildung 13: Zentrales Load Balancing mittels Load-Balancer

- auswahl
  - Erkennung von fehlerhaften Knoten
  - Gemeinsamer Datenbestand

**OpenSER als Load-Balancer**

Als Load-Balancer eignet sich die Open Source Software OpenSER. OpenSER ist ein robuster und performanter SIP-Server, der die Funktionen Registrar-, Location-, Proxy- und Redirect-Server beherrscht und sich durch weitere Funktionen auszeichnet, dazu gehören u.a.:

- SIP via UDP/TCP/TLS
- NAT-Traversal für SIP/RTP
- AAA via Datenbank, RADIUS und DIAMETER
- Least Cost Routing (LCR)
- Instant-Messaging- und Präsenz-Informationen via SIP (SIMPLE)
- ENUM (RFC3761)

Im Unterschied zu einem dedizierten Lay-

er/3/4 Load-Balancer ist OpenSER als Layer7-Load-Balancer somit in der Lage, zusätzliche Aufgaben zu übernehmen. Damit der Load-Balancer in diesem Szenario nicht zu einem Single Point Of Failure (SPOF) wird, sorgt ein zusätzlicher OpenSER-Server für die nötige Redundanz. Unter Verwendung der Heartbeat-Software, die auch für den Aktiv/Passiv-Cluster verwendet wurde, bilden diese einen HA-Cluster (Abbildung 14). Der OpenSER-Cluster ist jedoch nicht auf zwei Knoten beschränkt; weitere Knoten können bei Bedarf hinzugefügt werden. Laut Heartbeat-Homepage wurden Topologien mit 16 Knoten getestet.

**Persistente Bindung zwischen Client und Asterisk**

Für die Lastverteilung der SIP-Nachrichten muss sichergestellt werden, dass Gespräche auf denselben physikalischen Asterisk-Server geleitet werden (Abbildung 15).

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

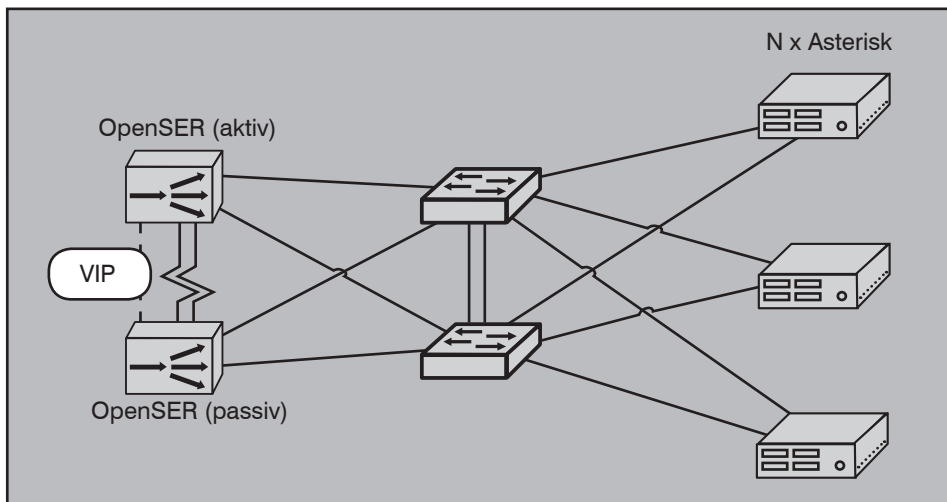


Abbildung 14: Redundante Auslegung des Load-Balancers eliminiert einzelne Fehlerstelle (SPOF)

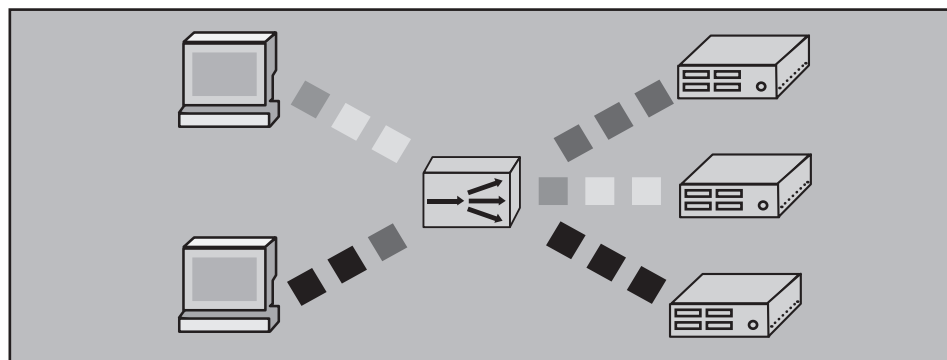


Abbildung 15: SIP-Nachrichten eines Gespräches müssen auf denselben Asterisk-Server geleitet werden

Wird dies bei der Verteilung der Anfragen nicht beachtet, funktionieren bestimmte vermittlungsspezifische Leistungsmerkmale nicht (z.B. Call Transfer). Für SIP eignet sich zur Identifizierung von Gesprächen das Call-ID Feld.

**Algorithmen und Metriken zur Zielauswahl**

Die eigentliche Verteilung der Client-Anfragen übernimmt das OpenSER Dispatcher-Modul. Dies ermöglicht eine Lastverteilung basierend auf Layer7-Merkmalen. Hierfür stehen folgende Algorithmen bereit:

- Algorithmus 0: Hash über die Call-ID
- Algorithmus 1: Hash über die From-URI
- Algorithmus 2: Hash über die To-URI
- Algorithmus 3: Hash über die Request-URI (R-URI)
- Algorithmus 4: Round-Robin

Neben der Call-ID als Auswahlkriterium für eine persistente Bindung zwischen Client und Asterisk-Server ist auch die From-URI für diesen Zweck geeignet.

**Erkennung von fehlerhaften Knoten**

Fehlerhafte Knoten werden nicht durch dienstspezifische Überprüfungen identifiziert, sondern anhand eines Timers. Wird eine bestimmte Zeit beim Verbindungsaufbau zum Asterisk-Server überschritten, kommt es zum Timeout. Damit wird der fehlerhafte Knoten automatisch aus der Liste der verfügbaren Asterisk-Server entfernt und der nächste Server gewählt. Dies wird solange fortgesetzt, bis die Liste keine weiteren Server enthält.

**Datenpflege: kein Chaos dank Datenbank**

Der Load-Balancer wird über eine einzige Konfigurationsdatei gesteuert. An dieser sind keine häufigen Änderungen zu erwarten, so dass keine Shared-Disk Architektur oder Replikationstechniken vonnöten sind. Im Regelfall reicht das manuelle Kopieren auf den zweiten Knoten.

Dies steht in Kontrast zu den Asterisk-Servern, bei denen die Anzahl der Änderungen am Datenbestand deutlich höher ausfällt. Wie auch bei den vorangegangenen

Lösungen muss sichergestellt werden, dass die Konfiguration auf allen Asterisk-Servern identisch ist. Für einen Load-Balancing-Cluster mit Asterisk empfiehlt sich die Variante: Shared-Nothing + Asterisk RealTime.

**Load Balancing: nur mit Asterisk RealTime**

Jeder Knoten verfügt über eine lokale Asterisk-Instanz (Shared-Nothing), die zugehörige Konfiguration wird aus einer zentralen Datenbank gelesen. Asterisk nutzt für den Zugriff auf die Datenbank die Asterisk Realtime Architektur (ARA). Die Nutzung einer Datenbank ist für einen Asterisk Load-Balancing-Cluster unabdingbar. Andernfalls wäre es möglich, dass sich Clients an unterschiedlichen Asterisk-Servern registrieren. Jeder Asterisk-Server würde dann nur den Standort der eigenen Clients kennen, da kein gemeinsames Backend zur Verfügung steht. Die Konsequenz wäre, dass die Clients nur Kontakt zu anderen Clients des gleichen Asterisk-Servers aufnehmen könnten.

Sofern weitere Daten synchron gehalten werden müssen, kommen angepasste Skripte oder Software-Pakete zum Einsatz. Eine geeignete Software, welche Dateien in einem Cluster synchronisiert und bei Bedarf auch mit Aktionen (z.B. dem Starten und Stoppen von Applikationen) verknüpft, liefert z.B. Clifford Wolf mit csync2.

Ein vollständiges Szenario ist in Abbildung 16 dargestellt.

**Für die Lastverteilung auf Basis von OpenSER gelten folgende Vor- und Nachteile:**

- OpenSER als Load-balancer: Im Unterschied zu einem herkömmlichen Layer3/4 Load-Balancer ermöglicht OpenSER eine Behandlung der SIP-Nachrichten auf Layer7 des OSI-Referenzmodells. Doch auch im Vergleich zu (kommerziellen) Layer7 Load-Balancern bietet OpenSER als SIP-Server einen höheren Funktionsumfang. Dazu gehören neben AAA, NAT-Traversal und Protokollumsetzungen auch das SIP-Routing und Protokollausbesserungen, um die Zusammenarbeit zwischen verschiedenen Systemen (Stichwort Interoperabilität) zu verbessern oder erst zu ermöglichen.
- OpenSER Performance: OpenSER gilt als robuster und leistungsfähiger SIP-Proxy. Laut Projekt-Homepage erreicht das zustandsbehaftete Transaktionsmodul ca. 8000 Anrufe pro Sekunde (Calls Per Second, abgekürzt CPS).

Ein Asterisk für den Katastrophenfall - Hochverfügbare Telefonie auf Basis einer Open Source Lösung?

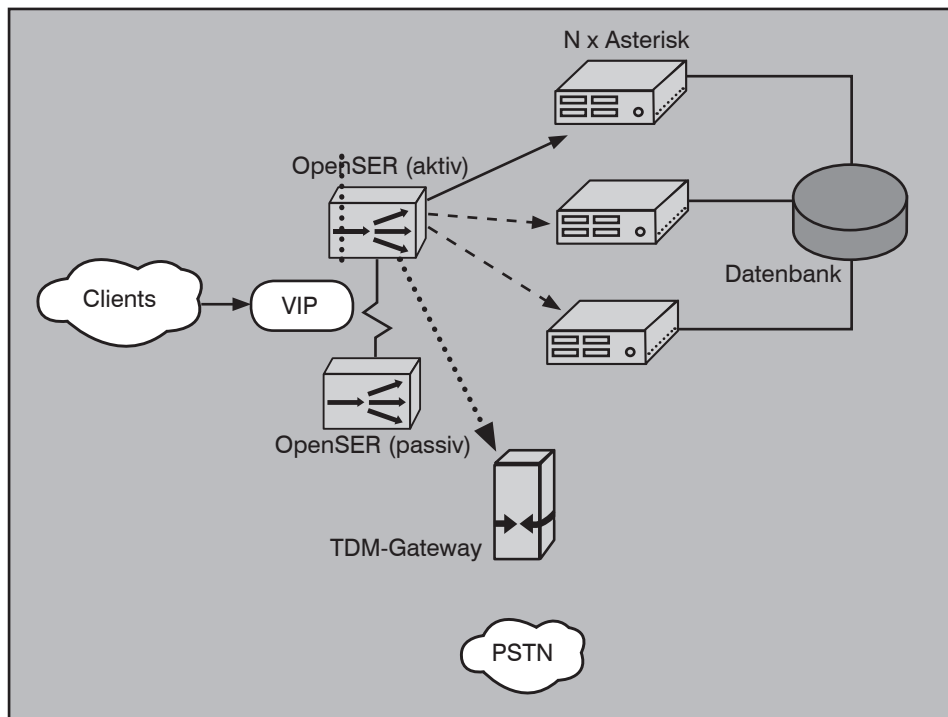


Abbildung 16: Load-Balancing Szenario mit OpenSER HA-Cluster und PSTN-Anbindung

- **Transparenz:** Im Gegensatz zu einer Lastverteilung mittels DNS SRV ist Server Load Balancing transparent gegenüber den Clients. Der Load-Balancer fungiert als Eintrittspunkt in den Cluster. Alle SIP-Elemente adressieren nur die virtuelle IP (VIP) des Load-Balancers.
- **Komplexität:** Im Vergleich zu den beiden vorherig beschriebenen Szenarien wird die Komplexität nochmals erhöht. Mit OpenSER als Load-Balancer kommt eine weitere zu pflegende Komponente hinzu. Um einen SPOF an dieser kritischen Position zu umgehen, muss OpenSER redundant ausgelegt werden. Die Redundanz wird über einen Aktiv/Passiv-Cluster mit der bereits erläuterten Heartbeat-Software umgesetzt.
- **Dokumentation:** Während die OpenSER-Dokumentation ausführlich einzelne Module und Funktionen beschreibt, fehlt es ihr dennoch an einem Gesamtüberblick der Szenarien und Lösungen. Im Internet finden sich oft nur Ansätze einer Lösung, so dass eine Implementierung durchaus zeitintensiv ausfallen kann.
- **Datenbestand:** Während der Datenbestand auf dem OpenSER-Cluster überschaubar ist und manuell synchronisiert werden kann, muss die Konfiguration der Asterisk-Server in ei-

ner Datenbank mittels Asterisk RealTime vorgehalten werden. Weitere Daten können skriptgesteuert (z.B. mittels rsync oder csync2) auf alle Knoten verteilt werden.

- **PSTN:** Wie in der DNS SRV-basierten Variante, muss auch hier die PSTN-Anbindung über ein Mediagateway erfolgen. Neben kommerziellen Media-

gateways kann auch ein zusätzlicher Asterisk Aktiv/Passiv-Cluster diese Funktionalität bereitstellen.

**Fazit**

Asterisk besitzt von Haus aus keinerlei Hochverfügbarkeits-Funktionen. Diese lassen sich dank Open Source Software in einem gewissen Rahmen nachrüsten. Dabei reicht das Spektrum vom Aktiv/Passiv-Cluster bis hin zum Load Balancing-Cluster für mehrere hundert Teilnehmer.

Sofern das für eine solche Lösung notwendige Know-How nicht im eigenen Haus vorliegt, wird eine eigenhändige Umsetzung nicht empfohlen. Aufgrund fehlender Dokumentation im Bereich Hochverfügbarkeit mit Asterisk kann eine Realisierung sehr zeit- und damit kostenintensiv ausfallen. Hier ist man auf Dritthersteller angewiesen, die entweder eine gehobene Lösung (IP-Centrex) samt Support und Haftung anbieten oder entsprechende Hochverfügbarkeits-Lösungen für Asterisk im Portfolio haben.

Laut Kevin Fleming, Senior Software Engineer bei Digium und Co-Maintainer von Asterisk, rücken die Themen Clustering und Failover erst mit Version 1.6 in den Vordergrund. Damit ist es nur noch eine Frage der Zeit, bis Asterisk auch in diesem Punkt mit kommerziellen Herstellern gleichzieht.

**Jetzt Leser werden**

**Der Netzwerk Insider**

Der Netzwerk Insider erscheint 12 Mal im Jahr im PDF-Format und informiert Sie per eMail über die Hintergründe aktueller Netzwerk-Technologien. Jeden Monat werden zwei Themen gewählt, über die in ausführlicher Form topaktuelle Insider-Informationen gegeben werden. Der Netzwerk-Insider vertritt die Sichtweise von Technologie-Anwendern und bewertet Produkte und Technologien im Sinne der wirtschaftlichen und erfolgreichen Umsetzbarkeit in der täglichen Praxis. Durch seine strenge wirtschaftliche Unabhängigkeit (keine Hersteller-Anzeigen) kann er es sich leisten, Schwachstellen und Nachteile offen anzusprechen. Der Netzwerk-Insider ist bekannt für seine kritische, herstellerneutrale und fundierte Technologie-Bewertung.



Hier können Sie sich zum Netzwerk Insider kostenlos und ohne jede Verpflichtung registrieren lassen:

<http://www.comconsult-akademie.de/de/Registrierung.php>