Power Grid Exist:
Linux Virtual Desktop without Virtualization

User Distribution with Linux Virtual Server and
Replicated File Server with GlusterFS

Ziele, Architektur und Aufbau

S. Strack, G. Bengel

Informatik-Berichte
Hochschule Mannheim – Fakultät für Informatik

Computer Science Reports
Mannheim University of Applied Sciences – Computer Science Department

CSR 002.10
August 2010

URL: http://www.informatik.hs-mannheim.de/reports

hochschule mannheim

FAKULTÄT FÜR INFORMATIK

Power Grid Exist:

Linux Virtual Desktop without Virtualization

- User Distribution with Linux Virtual Server and Replicated File Server with GlusterFS -

Sylvia Strack, Günther Bengel

Stand: August 2010

Informatik - Berichte Hochschule Mannheim – Fakultät für Informatik

Hochschule Mannheim
Fakultät für Informatik

Paul-Wittsack-Str. 10
D-68163 Mannheim

E-Mail:

sylvia.strack@gmx.de
g.bengel@hs-mannheim.de

Abstract

Der vorliegende Bericht beschreibt die auf Power Grid Exist (PGE)

1. realisierte Benutzerverwaltung mit dem Linux Virtual Server (LVS) [LVS05] und

2. die fehlertolerante und hochverfügbare Auslegung eines konsistenten Dateisystems mit GlusterFS [GLU10].

Durch dieses Vorgehen erreicht man einen Linux Virtual Desktop, der fehlertolerant und hochverfügbar ausgelegt ist, und keine auf dem Server (Cluster) vorhandenen virtuelle Maschinen benötigt. LVS verteilt die Benutzer nach dem Round Robin-Verfahren auf die einzelnen Rechner des Clusters PGE (12 Workerknoten) und jedem Benutzer steht auf der Clientseite eine Linux-Maschine (Ubuntu) mit einer Remote Shell oder die graphische Oberfläche Gnome zur Verfügung.

Dieser Bericht ist der dritte einer Reihe von Berichten über das PGE. Um grundlegende Informationen über eine Master-Worker-Architektur und einen ersten Einblick in das PGE zu erhalten, wird empfohlen, „Power Grid Exist: A Dynamic Cluster with Hierarchic Master Worker Architecture“ von Günther Bengel [BEN08] zu lesen. Der Aufbau des PGE, der maßgebend für die Wahl der hier vorgestellten Technologien und Vorgehensweisen ist, wird weiterhin in „Power Grid Exist: Ziele, Architektur und Aufbau“ von Günther Bengel et al. [BEN10] beschrieben. Diese beiden Berichte sind für das Verständnis des vorliegenden Berichtes notwendig.

Inhaltsverzeichnis

1 Ansätze für High Performance Computing

Für das High Performance Computing (HPC) oder Supercomputing existieren zurzeit vier verschiedene Ansätze, die gemäß dem Overhead aus Betriebssystem und Virtualisierung geordnet sind:

1. Graphikprozessorberechnungen (GPU-Computing) basiert auf Graphikprozessoren mit aktuell 240 Recheneinheiten. Die Programmierung geschieht in C und dem parallelen Programmiermodell Compute Unified Device Architecture, kurz CUDA genannt. CUDA ist eine von Nvidia entwickelte Technik, die parallele Berechnungen auf dem Graphikprozessor auf einer Vielzahl von Recheneinheiten durchführt. Die Graphikprozessoren arbeiten dabei ohne den Overhead des Betriebssystems und der Virtualisierung [CUD10].

2. Clusterberechnungen (HPC-Cluster-Computing) zerlegen die parallelen Berechnungen in Jobs und verteilen diese auf ein Netz von Rechnern. Die Kommunikation zwischen auf verschiedenen Knoten laufenden Jobs geschieht in der Regel mittels dem Message Passing Interface (MPI) [MPI10]. Die Verteilung der Jobs übernimmt ein Job-Scheduler oder Load Balancer. Das Cluster läuft mit einem Betriebssystem auf den einzelnen Knoten, jedoch ohne Virtualisierungsschicht.

3. Gridberechnungen (Grid-Computing) Gridberechnungen laufen auf Clustern von verschiedenen räumlich verteilten Organisationen (virtuelle Organisationen). Die einzelnen Rechner im Netz werden mit dem Overhead des Betriebssystems betrieben, jedoch ohne Virtualisierung.

4. Cloudberechnungen (Cloud-Computing) bieten die parallelen Berechnungen als Webservice an. Der bekannteste parallele Webservice ist das MapReduce Framework von Google Inc. für parallele Berechnungen über große (mehrere Petabyte) Datenmengen auf Clustern [MAP10]. Cloudberechnungen basieren auf Rechner im Cluster mit einem auf jedem Knoten laufenden Betriebssystem und Virtualisierungssoftware.

2 Linux Virtual Desktop

Virtualisierungs-Desktops basieren auf Virtualisierungslösungen von z.B. VMware, Citrix, Parallels Virtuozzo Containers, XEN, Sun Microsystems, Wyse oder Microsoft, die jedem Arbeitsplatzrechner eine virtuelle Maschine auf einem Server zur Verfügung stellen. Das Betriebssystem läuft unabhängig von jedem Arbeitsplatzrechner in seiner eigenen dedizierten virtuellen Maschine. Damit sind die einzelnen Benutzer voneinander

abgeschottet und sie arbeiten dabei auf dem Server. Auf dem Thin Client der Benutzermaschine läuft nur die Remote Shell oder die graphische Oberfläche.

Zur Vermeidung des Virtualisierungsoverheads ist auf den beiden Masterknoten des PGE der Linux Virtual Server (LVS) installiert [LVS05].

LVS erlaubt einem Remote-Benutzer, sich per Round Robin-Verfahren auf dem Cluster einzuloggen. In diesem Zusammenhang ist Round Robin ein Lastverteilungsverfahren für Netzwerkdienste, bei dem eine Liste von Netzwerkressourcen vorliegt, aus der bei jeder Anfrage jeweils die nächste Ressource ausgewählt wird. Das bedeutet in diesem Fall, dass der Benutzer beim Einloggen auf das Cluster via virtuelle IP-Adresse mit diesem Verfahren auf einen damit ausgewählten Workerknoten verbunden wird. Dabei wählt LVS bei jeder neuen Loginanfrage aus der Liste der verfügbaren Workerknoten den nächsten aus, sodass die Benutzer gleichmäßig auf das Cluster verteilt sind.

Durch dieses Vorgehen kann man sich in transparenter Weise von jedem Rechner der Hochschule und von außerhalb innerhalb des Hochschulnetzes auf dem Cluster einloggen. Jedem Benutzer steht damit auf seinem Remote-Rechner ein Linux Virtual Desktop zur Verfügung und der Benutzer arbeitet auf den Workerknoten des Clusters PGE. Der Linux Virtual Desktop bietet einem Benutzer eine Linux Softwareentwicklungsumgebung, mit dem er die Erstellung der parallelen Software vornehmen kann. Dies geschieht, wie schon erwähnt, mit Betriebssystemoverhead, jedoch ohne eine Virtualisierungsschicht auf dem Cluster.

Um das Cluster PGE hochverfügbar den Benutzern zur Verfügung zu stellen, benötigt man das Einrichten eines fehlertoleranten und konsistenten Dateiservers. Dies beinhaltet das Einrichten je eines Dateiservers auf beiden Masterknoten des PGE, wobei einer von beiden als Backup-Server dient. Außerdem sind die Daten in Echtzeit auf beiden Servern gespiegelt, sodass im Falle eines Ausfalls einer der beiden Server die Daten immer noch zur Verfügung stellt. Alle Workerknoten haben dadurch ständig Zugriff auf die gemeinsamen Daten.

3 Linux Virtual Server

Dieses Kapitel stellt den Linux Virtual Server (LVS) vor, wobei zuerst auf dessen Grundlagen und danach auf die Komponenten eingegangen wird, die für die Installation auf dem PGE benötigt werden.

3.1 Einführung und Grundlagen

Um ein hochverfügbares Cluster einzurichten, gibt es mehrere Möglichkeiten. Aus den vorhandenen Möglichkeiten werden in den folgenden Unterkapiteln diese vorgestellt, die für das Cluster ausgewählt worden sind.

3.1.1 Definition Hochverfügbarkeit

Bevor es darum geht, konkrete Technologien vorzustellen, ist es notwendig, den Begriff der Hochverfügbarkeit zu klären:

Innerhalb eines Computernetzwerkes werden unterschiedliche Ressourcen angeboten. Einige dieser Ressourcen müssen die Forderung erfüllen, immer bzw. mit sehr großer Wahrscheinlichkeit[1] verfügbar zu sein. Wenn ein Computer, der eine Ressource anbietet, ausfällt, findet ein sogenanntes Failover statt, bei dem diese Ressource vom einen Computer auf einen anderen verlagert wird. Dadurch gibt es keinen Single Point of Failure mehr, da nicht nur ein Computer für eine Ressource verantwortlich ist. Ein Single Point of Failure ist dabei eine Komponente des Systems, deren Ausfall zur Folge hat, dass das komplette System ausfällt.

3.1.2 Einführung in LVS

Die Realisierung von LVS verwendet die Cluster-Loadbalancing-Software IPVS, welche in Form von Linux-Kernel-Patches existiert. Durch diese Patches kann ein Computer als Cluster-Loadbalancer agieren. Zusammen mit den Knoten eines Clusters bildet diese Loadbalancing-Software den LVS.

Der Loadbalancer nimmt alle Benutzeranfragen von Client-Computern entgegen und entscheidet, welcher Clusterknoten die Anfrage entgegennehmen soll.

Es existieren drei verschiedene Arten von LVS-Clustern, die sich in der Paketweiterleitung unterscheiden:

- Network Address Translation (LVS-NAT)-Cluster: Wenn ein Paket den Loadbalancer erreicht, übersetzt dieser die Netzwerk-IP-Adresse und den Port dieses Pakets, bevor er es an einen anderen Clusterknoten weiterreicht. Bei diesem Clustertyp werden alle einkommenden und ausgehenden Pakete des Clusters über den Loadbalancer gereicht.
- Direct Routing (LVS-DR)-Cluster: Bei dieser Variante leitet der Loadbalancer alle einkommenden Pakete an das Cluster an einen anderen Clusterknoten weiter. Der Clusterknoten sendet diesmal allerdings sein Paket direkt zurück an den Client, der zuvor das Paket an das Cluster gesendet hat.

[1] Eine solche Wahrscheinlichkeit liegt bei mindestens 99,9%. Die restlichen 0,1% ziehen die Möglichkeit in Betracht, dass Ressourcen aufgrund eines Stromausfalls ausfallen können.

- IP Tunneling (LVS-TUN)-Cluster: Dies ist eine Erweiterung eines LVS-DR-Clusters, bei dem sich die einzelnen Clusterknoten in einem anderen physischen Netzwerk befinden können, als der Loadbalancer. Hierbei wird das eigentliche Paket in einem weiteren untergebracht, sodass es in einem Intranet oder sogar dem Internet transportiert werden kann [KOP05].

Im weiteren Verlauf dieses Berichts wird nur auf das LVS-NAT-Cluster eingegangen, da dieses für das PGE implementiert worden ist.

Damit der Loadbalancer bestimmte Ressourcen anbieten kann, werden virtuelle IP-Adressen benötigt, wobei für jede Ressource eine eigene virtuelle IP-Adresse verwendet wird. Dieses Thema wird im nächsten Abschnitt behandelt.

3.1.3 Virtuelle IP-Adressen

Um die Transparenz des Clusters gegenüber einem Benutzer zu erreichen, wird eine virtuelle IP-Adresse benötigt. Solch eine IP-Adresse ist unter Linux auf zweierlei Weisen erreichbar: Einmal als sekundäre IP-Adresse und einmal als IP-Alias.

Sekundäre IP-Adressen

Sekundäre IP-Adressen werden neben einer primären IP-Adresse erstellt.

Während unter Linux die primären IP-Adressen mit dem Kommando `ifconfig` angezeigt werden können, funktioniert das bei den sekundären IP-Adressen nur mit `ip addr sh`.

Erstellt werden kann eine solche IP-Adresse mit dem Befehl `ip addr add`, wobei noch entsprechende Parameter für die gewählte Adresse angegeben werden müssen.

Gelöscht wird sie mit `ip addr del`, genau wie beim vorigen genannten Befehl mit den entsprechenden Parametern.

IP-Aliase

IP-Aliase werden zu einer physischen Ethernetschnittstelle hinzugefügt, mit dem bereits eine IP-Adresse verknüpft ist. Aliase werden dargestellt, indem das Interface und die Angabe darüber, das wievielte Alias es ist, mit einem Doppeltpunkt voneinander getrennt werden. Bei dem eth0-Interface wird z.B. der erste Alias mit `eth0:0` und der zweite mit `eth0:1` angegeben.

Im Gegensatz zu den sekundären IP-Adressen können diese über den Befehl `ifconfig` angezeigt und erzeugt werden [KOP05].

Für das PGE wurde als virtuelle IP-Adresse eine sekundäre IP-Adresse gewählt.

3.2 Hochverfügbarkeit mit Heartbeat

Mit Heartbeat ist es möglich, ein Failover einer Ressource von einem Computer auf einen anderen zu erreichen, sodass diese hochverfügbar wird. Im Zusammenhang mit LVS bedeutet dies, die Funktionen des Loadbalancers auf einen anderen Computer zu übertragen, sodass dieser nun als Loadbalancer fungiert. Dabei werden auch die Aufgaben von ldirectord, was im anschließenden Kapitel erklärt wird, auf einen anderen Computer verlegt. Das Failover geschieht über ein IP Takeover, bei dem die IP-Adresse, die der Ressource zugeteilt ist, vom einen auf den anderen Computer übertragen wird [KOP05].

Die Funktionsweise von Heartbeat ist die, dass Heartbeat ein primärer und ein Backup-Server bekannt gemacht werden. Der primäre Server ist dabei Inhaber einer Ressource, die hochverfügbar sein soll. Der primäre Server sendet regelmäßig ein sogenanntes `heartbeat` an den Backup-Server, sodass dieser weiß, ob der primäre Server noch verfügbar ist oder nicht. Wenn er keine Nachricht mehr erhält, geht er davon aus, dass der primäre Server ausgefallen ist, und übernimmt über ein Failover dessen Aufgaben und wird somit selbst zum primären Server. Wenn der ausgefallene Server wieder aktiv ist, wird dieser zum nächsten Backup-Server und empfängt zukünftig die `heartbeats` vom neuen primären Server [LVS05].

3.3 Überwachung mit ldirectord

ldirectord (Linux Director Daemon) ist ein Programm, mit dem die Zustände von Clusterknoten überwacht werden können. Wenn ein Knoten ausfällt, wird dieser automatisch auf dem Loadbalancer aus der Liste der vorhandenen Knoten entfernt. Wenn er wieder verfügbar ist, wird er wieder darin aufgenommen [KOP05].

Um dies zu erreichen, hat ldirectord eine Konfigurationsdatei, die die virtuellen Dienste und die mit ihnen verknüpften realen Rechner angibt. Während dem Betrieb von LVS überwacht ldirectord den Status der Knoten im Cluster, indem es regelmäßig versucht, über HTTP eine bekannte URL aufzurufen und zu testen, ob die Antwort der erwarteten Antwort entspricht. Wenn dies nicht der Fall ist, wird dieser Knoten entfernt und wieder reaktiviert, wenn er wieder wie erwartet antwortet. Die Statusüberprüfung kann auch über eine TCP/IP-Verbindung erfolgen, falls kein HTTP-Dienst installiert ist. Wenn keiner der Knoten im LVS-Verbund erreichbar ist, wird ein Fallback-Server in die Menge der Knoten eingefügt, an den zukünftige Anfragen weitergeleitet werden. In den meisten Fällen ist der Fallback-Server der lokale Host [LDI10].

3.4 Konfiguration

Mit der Lastverteilungssoftware LVS kann sich ein Benutzer per SSH auf das Cluster einloggen und wird dabei gleichmäßig auf die Workerknoten verteilt. Um diese Funktionalität hochverfügbar zu machen, wird Heartbeat und um ein dynamisches Cluster,

bei dem die einzelnen Knoten bei einem Ausfall entfernt und bei erneutem Vorhandensein hinzugefügt werden können, ldirectord eingesetzt. In den folgenden Unterkapiteln wird als erstes das LVS auf dem PGE, danach die Konfiguration von Heartbeat und schließlich diejenige von ldirectord beschrieben.

3.4.1 Konfiguration von LVS

Im Allgemeinen kann ein Benutzer sich nur auf das PGE einloggen, wenn er sich im Hochschulnetz befindet. Wie in [BEN10] nachzulesen ist, ist das Cluster über zwei IP-Adressen von außerhalb ansprechbar. Abbildung 1 zeigt, dass es durch die Architektur von LVS für einen normalen Benutzer nur noch die Möglichkeit gibt, das Cluster über die virtuelle IP-Adresse anzusprechen. Master 1 ist hier der aktive Loadbalancer und Master 2 der Backup-Server. Über `heartbeat` erkennt Master 2, ob Master 1 noch funktionstüchtig ist. Der aktive Master überprüft in bestimmten Abständen mit ldirectord, ob die einzelnen Workerknoten, die in einer entsprechenden Konfigurationsdatei auf dem Master aufgelistet sind, noch verfügbar sind.

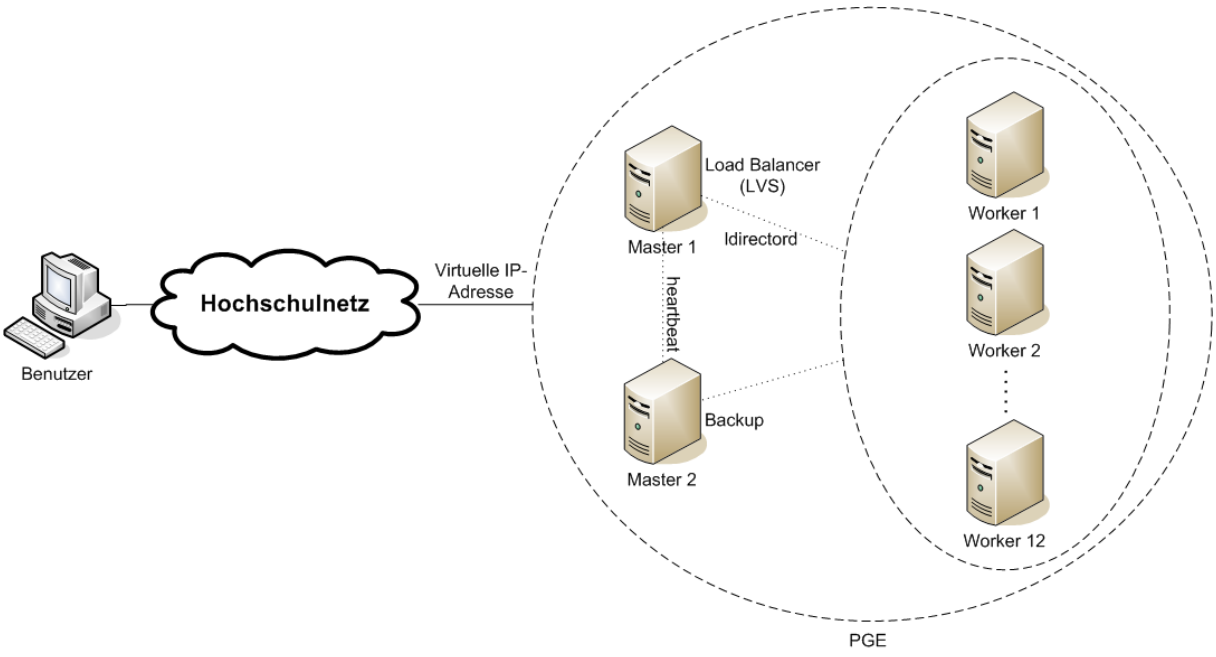

Abbildung 1: LVS-Architektur auf dem PGE

Auf dem PGE liegt aufgrund seiner physischen Netzwerkarchitektur LVS als LVS-NAT-Cluster vor. Das bedeutet, dass sowohl alle einkommenden als auch ausgehenden Pakete über den Loadbalancer gereicht werden. Voraussetzung dafür ist, dass auf allen Workern als Gateway die internen IP-Adressen der Master eingestellt sind. In Abbildung 2 sieht man den Verlauf des Request-Response-Modells innerhalb des PGE. Die blauen Kästchen stehen für das Request-Paket, die orangen Kästchen für das Response-Paket. Begonnen wird mit dem Request-Paket auf der Seite des Benutzers. Dieses Paket wird über das Hochschulnetz an das PGE weitergereicht, wo es vom Loadbalancer, hier Master 1, entgegengenommen wird.

Dieser wählt per Round Robin einen Worker aus und ändert lediglich die Ziel-IP-Adresse und eventuell den Zielport des Pakets von der virtuellen IP in die reale IP-Adresse des Workers. In diesem Fall fällt die Wahl auf Worker 1. Daraufhin sendet er das Paket weiter an Worker 1, der es bearbeitet und ein Response-Paket generiert. Die Ziel-IP-Adresse des Response-Pakets ist die des Clients, der anfangs den Request gesendet hat, und die Quell-IP-Adresse ist die reale IP von Worker 1. Der Loadbalancer wiederum empfängt das Paket und ändert die Quell-IP-Adresse in die virtuelle IP und sendet das Paket weiter an den Client.

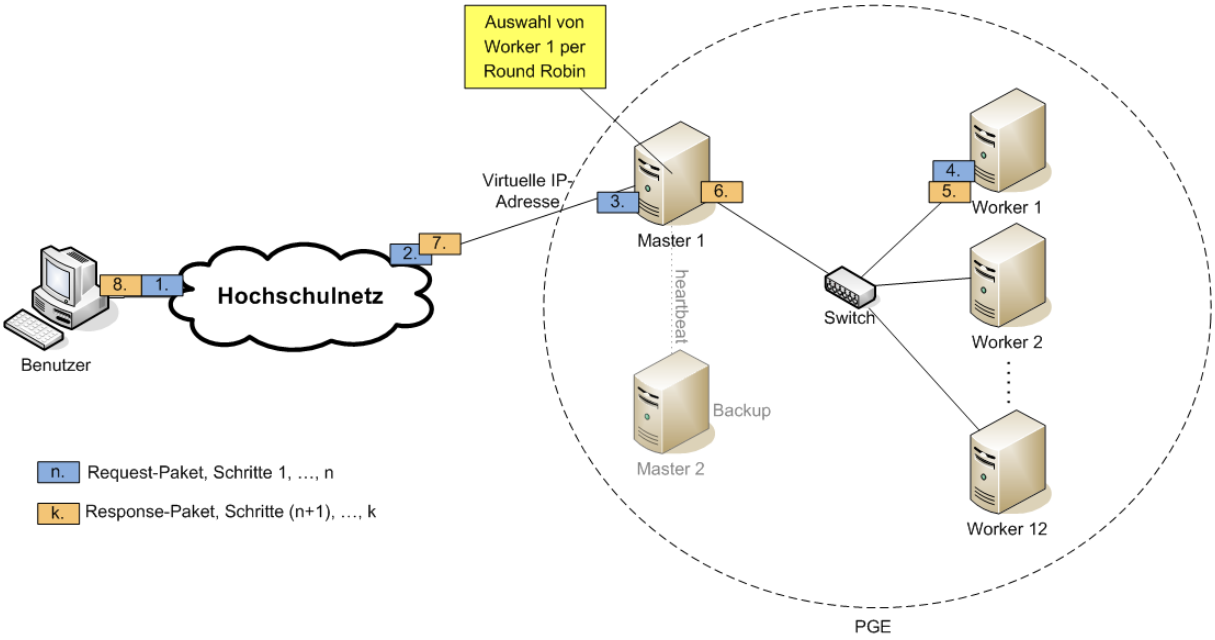

Abbildung 2: LVS-Architektur mit Network Address Translation (NAT)

3.4.2 Konfiguration von Heartbeat

Abbildung 3 stellt die allgemeine Architektur von Heartbeat auf dem PGE dar. Auch in diesem Fall ist Master 1 der aktive Loadbalancer und Master 2 ist der Backup-Server. Beide besitzen weiterhin jeweils eine IP-Adresse, die über das Hochschulnetz zu erreichen sind. Auch wenn diese IP-Adressen für die Funktionsweise von LVS keine Bedeutung haben, wurden sie der Vollständigkeit wegen in die Graphik mit aufgenommen. Innerhalb des lokalen Netzwerks des PGE besitzen die beiden Master jeweils noch eine interne IP-Adresse, die zur Kommunikation mit den Workerknoten dient. Master 1 verständigt Master 2 über `heartbeat` über seinen Zustand. Des Weiteren ist Master 1 im Besitz zweier Dienste, die beide ihre eigene virtuelle IP-Adresse zugewiesen bekommen haben. Über diese virtuellen IPs sind sie über das Hochschulnetz erreichbar. Mögliche Dienste können hier beispielsweise HTTP-Dienste, SSH-Dienste oder FTP-Dienste sein. Für das PGE gibt es zum jetzigen Zeitpunkt nur die Option, sich als Benutzer per SSH auf das Cluster einzuloggen.

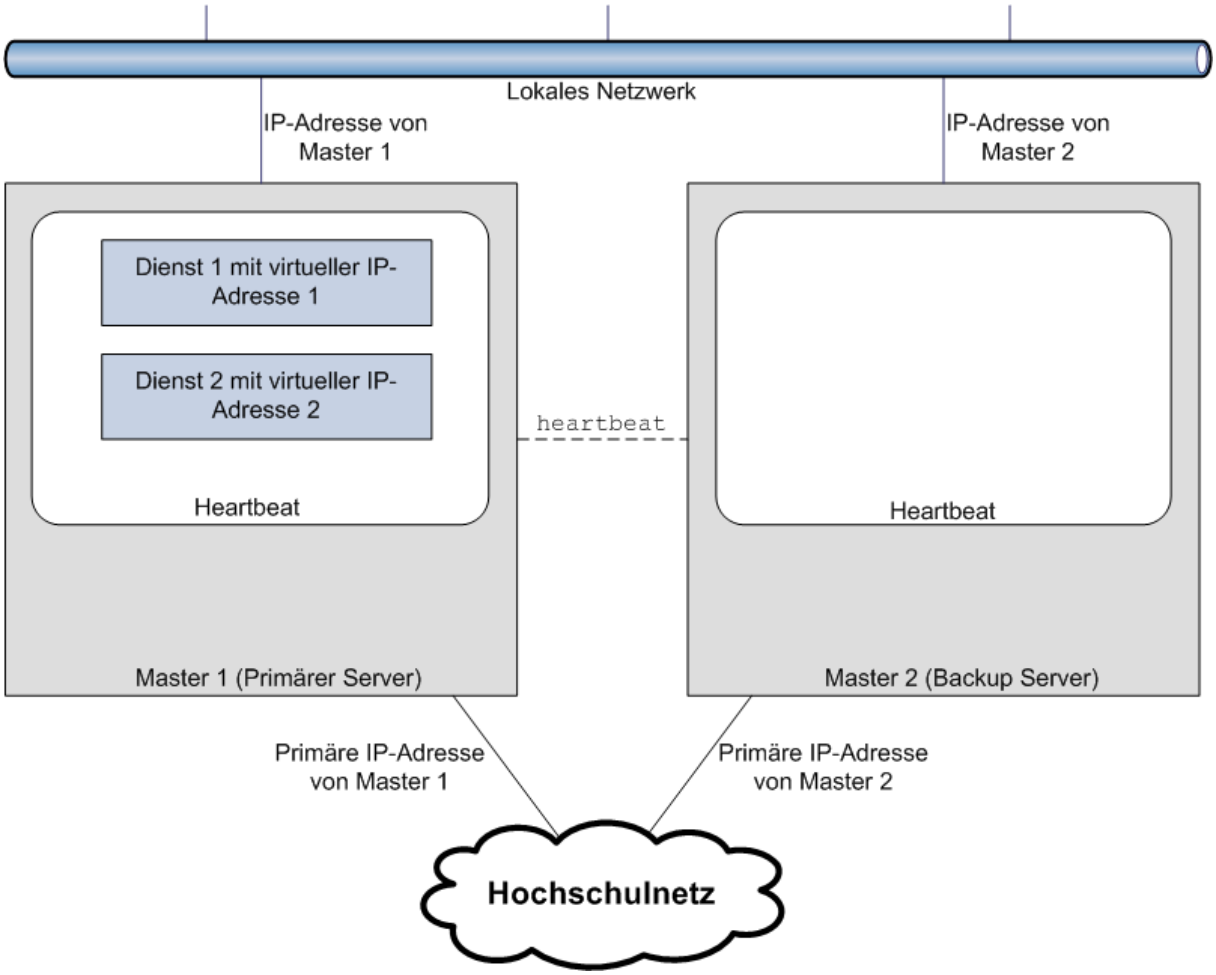

Abbildung 3: Hochverfügbarkeit mit Heartbeat

Wenn Master 1 ausfällt, übernimmt, wie schon in Kapitel 3.2 angesprochen, Master 2 die Dienste von Master 1, indem ein IP-Address Takeover von Heartbeat stattfindet. Abbildung 4 zeigt den Zustand des Systems nach einem solchen Failover. In dieser Situation ist Master 2 nun im Besitz der beiden virtuellen IPs und zukünftig für Anfragen an das PGE verantwortlich.

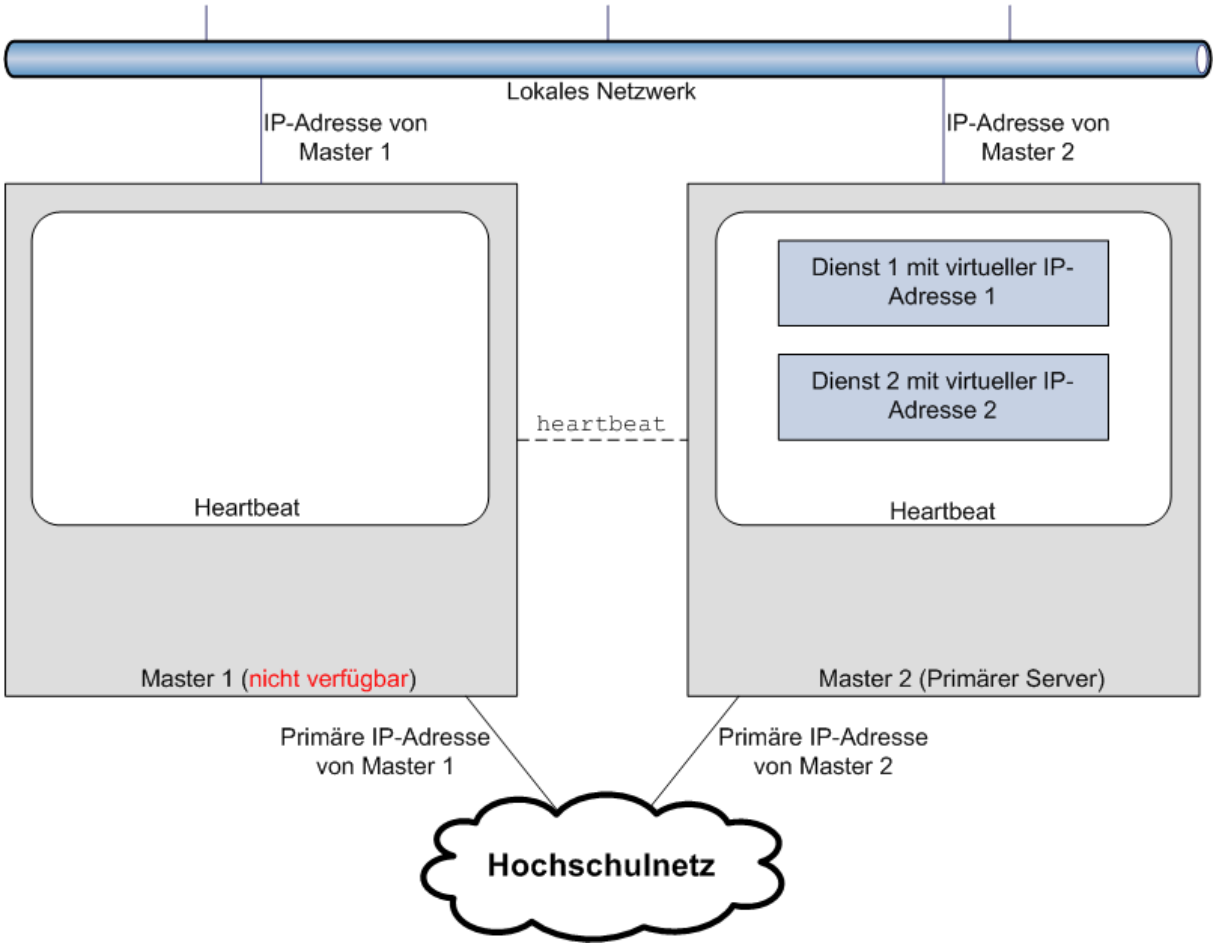

Abbildung 4: Heartbeat – Failover mit IP-Address Takeover

Heartbeat ist auf beiden Mastern so konfiguriert, dass ldirectord dann gestartet oder beendet wird, wenn auch Heartbeat gestartet oder beendet wird. Deswegen sind entsprechende Eintragungen in der Konfigurationsdatei `/etc/ha.d/haresources` auf beiden Mastern von Nöten. Um die Kommunikation zwischen den beiden Masterknoten zu ermöglichen, gibt es eine Datei mit dem Namen `/etc/ha.d/authkeys`.

Mit dem Start von Heartbeat erstellt dieses Programm automatisch die virtuelle IP in Form einer sekundären IP-Adresse, sodass sich der Administrator des PGE nicht darum kümmern muss, diese zu generieren.

3.4.3 Konfiguration von ldirectord

Für das dynamische Cluster liegt eine entsprechende Konfiguration von ldirectord vor.

In der Datei `/etc/ha.d/ldirectord.cf` sind alle Worker aufgelistet, sodass ein Benutzer per Round Robin bei einer Anfrage auf einen von ihnen verbunden werden kann. Außerdem beinhaltet diese Datei eine Einstellung, nach der ldirectord den Status der Worker überprüft, indem er testet, ob eine TCP/IP-Verbindung darauf möglich ist. Dieser Sachverhalt ist in Abbildung 5 erkennbar.

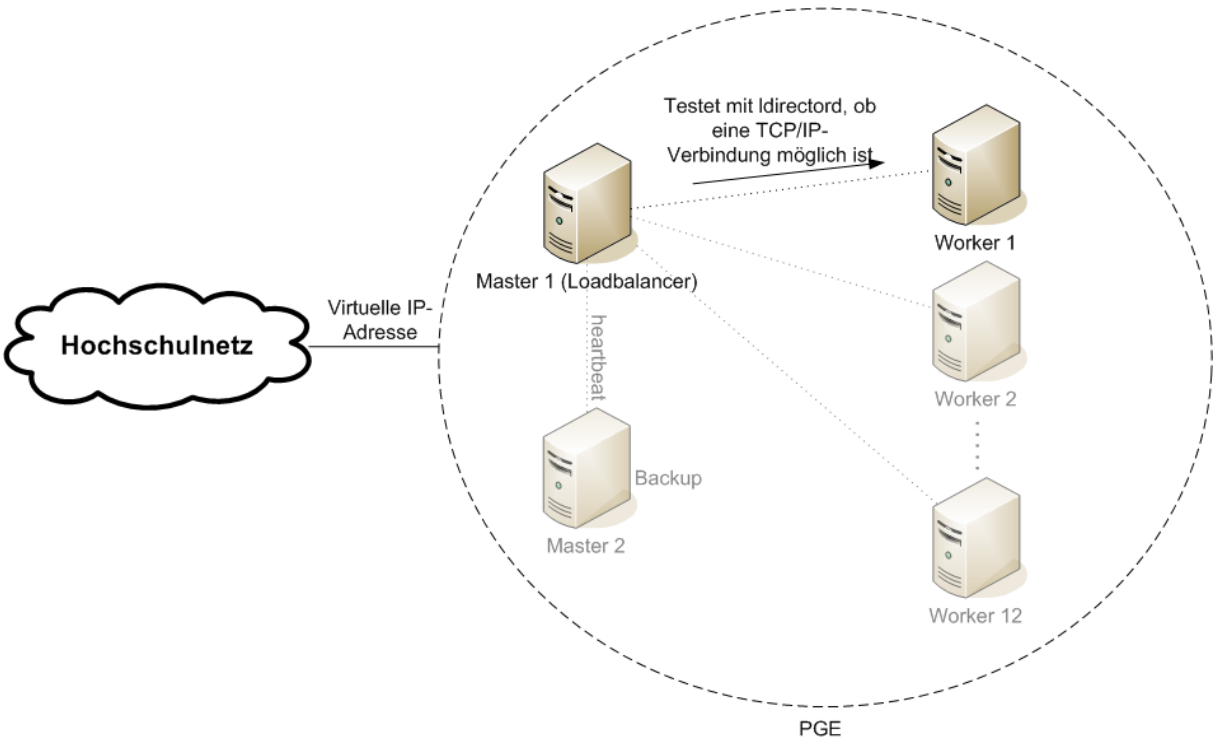

Abbildung 5: Status Check der Workerknoten mit ldirectord

Zum Starten von Heartbeat wird der Befehl

```
/etc/init.d/heartbeat start
```

und zum Beenden

```
/etc/init.d/heartbeat stopp
```

ausgeführt.

Der Status wird schließlich wie folgt abgefragt:

```
ipvsadm –Ln
```

Mit diesem Befehl kann unter anderem beobachtet werden, welche Worker aktiv und wie viele Verbindungen auf ihnen aktiv sind.

Durch den Start von Heartbeat wird die virtuelle IP in Form einer sekundären IP erzeugt und ein Benutzer kann nun über diese IP auf das PGE zugreifen.

4 GlusterFS

Auf dem PGE wurde ein NFS-ähnlicher, hochverfügbarer Dateiserver eingerichtet, welcher mit einem Dateisystem namens GlusterFS realisiert worden ist. In den folgenden zwei Abschnitten wird auf die Funktionsweise und die Möglichkeiten von GlusterFS

eingegangen, um in einem späteren Kapitel dessen konkrete Implementierung auf dem PGE zu erläutern.

4.1 Einführung

GlusterFS ist ein Dateisystem, mit dem Speicher von mehreren Servern zusammengefasst und als ein einziger Speicher gesehen werden kann. Dieses Dateisystem macht sich die Client-Server-Architektur zunutze, indem über eine TCP/IP Verbindung Speicheranfragen der Clients an Server gesendet werden. Änderungen, die am Dateiserver stattfinden, werden augenblicklich auf alle daran beteiligten Speichermedien weitergeschrieben [GLW10].

Beim Einsatz von GlusterFS gibt es verschiedene Betriebsmodi, die gefahren werden können. Hier werden die für das PGE am meisten relevanten Möglichkeiten vorgestellt:

- Standalone Storage: Dieser Modus ist einem NFS-Server sehr ähnlich, da hier ein einzelner Server das Dateisystem über das Netzwerk bereitstellt.
- Distributed Storage: Dieser Modus verteilt die Daten auf die Speichermedien von mehreren Servern. Somit kann man sich die Größe des dadurch entstehenden Gesamtspeichers zunutze machen.
- Replicated Storage: Die Daten sind auf mehreren Servern gespiegelt.
- Distributed Replicated Storage: Das ist ein Kompromiss aus Distributed Storage und Replicated Storage, bei dem die Daten sowohl verteilt gespeichert als auch gespiegelt werden.

Es existieren noch weitere Betriebsmodi, die in [GLU10] beschrieben sind.

4.2 Architektur auf dem PGE

Das Dateisystem GlusterFS ermöglicht, dass alle Workerknoten und auch die beiden Masterknoten des PGE auf denselben Dateiserver zugreifen und dieser hochverfügbar und konsistent vorliegt. Durch diese Technologie bilden alle vorhandenen Knoten eine Client-Server-Architektur, wie bereits in Kapitel 4.1 aufgegriffen. In diesem Zusammenhang sind auf dem PGE die beiden Masterknoten sowohl Server als auch Clients, da auch diese auf den Dateiserver zugreifen können sollen.

Für GlusterFS auf dem PGE fällt die Wahl auf den in Kapitel 4.1 vorgestellten Betriebsmodus Replicated Storage, wodurch ein Software-RAID1-Verbund vorliegt, um einen hochverfügbaren, konsistenten Dateiserver zu realisieren. Dieser Sachverhalt wird in Abbildung 6 ersichtlich, bei der man erkennen kann, dass auf beiden Masterknoten, Master 1 und Master 2, jeweils ein GlusterFS-Daemon installiert und aktiv ist. Die beiden Daemons spiegeln die Daten untereinander, sodass sie im Falle eines Ausfalls einer der beiden Computer immer noch verfügbar sind. Über das lokale Netzwerk haben alle vorhandenen

Knoten innerhalb des PGE Zugriff auf den Dateiserver, der durch Einhängepunkte (die dunklen Kreise) auf jedem Rechner ermöglicht wird. Mithilfe der Einhängepunkte wird das auf den beiden Mastern freigegebene Verzeichnis lokal auf jedem Computer gemountet. Die Einhängepunkte werden auf den Masterknoten aus dem Grund benötigt, da diese Computer nicht direkt auf die Dateien des Servers, sondern nur über die Einhängepunkte darauf zugreifen dürfen, da sonst GlusterFS nicht mehr ordnungsgemäß funktionieren könnte.

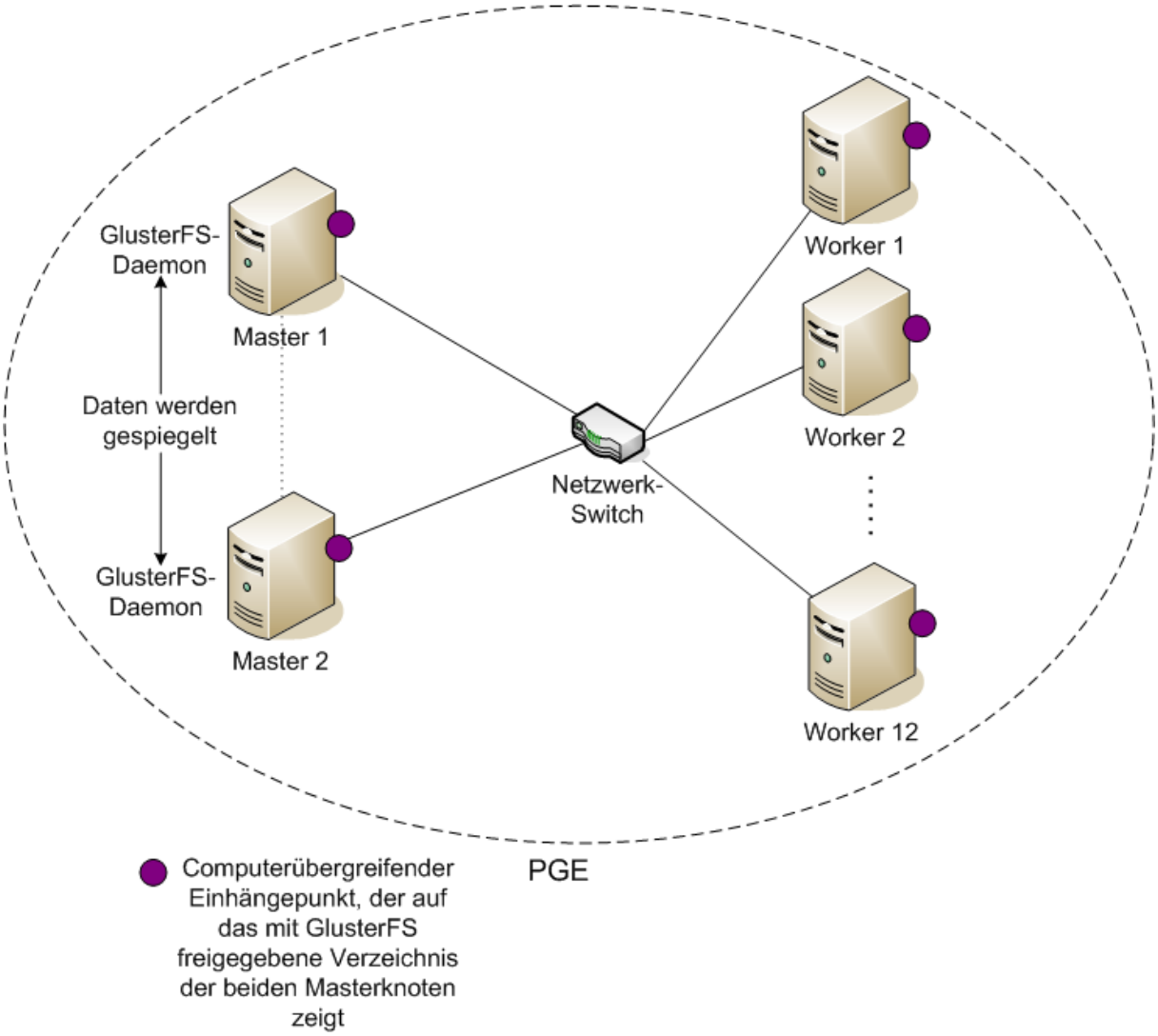

Abbildung 6: Ein hochverfügbares Replicated Storage mit GlusterFS auf dem PGE

4.3 Konfiguration

Dieser Abschnitt erläutert anhand der benötigten Verzeichnisse und der verwendeten Konfigurationsdateien die Konfiguration von GlusterFS. Die folgenden Schritte sind notwendig, um den Dateiserver zum Laufen zu bringen.

Erstellen der benötigten Verzeichnisse

Auf den beiden Masterknoten sind folgende Verzeichnisse eingerichtet:

```
/data/
/data/export/
/data/export-ns/
```

In diesen Verzeichnissen werden die tatsächlich vorhandenen Dateien bei Schreibzugriffen gespeichert.

Um auf die soeben angelegten Verzeichnisse zuzugreifen, müssen auf jedem Knoten des Clusters, also sowohl auf den beiden Masterknoten als auch auf allen zwölf Workerknoten, diese beiden Verzeichnisse vorhanden sein:

```
/mnt/glusterfs/
/etc/glusterfs/
```

Erstellen und Bearbeiten der Konfigurationsdateien

Um den GlusterFS-Server auf beiden Mastern zu konfigurieren, existiert auf beiden die Konfigurationsdatei `/etc/glusterfs/glusterfs-server.vol`, deren Inhalt auf beiden Mastern identisch und in Abbildung 7 ersichtlich ist. Die Datei selbst stammt aus [HOW09] und in [GLV08] gibt es einen etwas ausführlicheren Artikel über das Erstellen der Volume-Dateien.

```
# file: /etc/glusterfs/glusterfs-server.vol

volume posix
  type storage/posix
  option directory /data/export
end-volume

volume locks
  type features/locks
  subvolumes posix
end-volume

volume brick
  type performance/io-threads
  option thread-count 8
  subvolumes locks
end-volume

volume posix-ns
  type storage/posix
  option directory /data/export-ns
end-volume

volume locks-ns
  type features/locks
  subvolumes posix-ns
end-volume

volume brick-ns
```

```
  type performance/io-threads
  option thread-count 8
  subvolumes locks-ns
end-volume

volume server
  type protocol/server
  option transport-type tcp
  option auth.addr.brick.allow *
  option auth.addr.brick-ns.allow *
  subvolumes brick brick-ns
end-volume
```

Abbildung 7: Konfigurationsdatei für den GlusterFS-Server auf beiden Masterknoten. Quelle: [HOW09]

Der GlusterFS-Daemon wird danach über den Befehl

```
glusterfsd -f /etc/glusterfs/glusterfs-server.vol
```

gestartet, sodass auf den Knoten des Clusters nun das gemountete Verzeichnis, in dem die Dateien abgespeichert werden, auf die Daten zugreifen kann.

Wie bereits erwähnt, funktioniert GlusterFS nach dem Client-Server-Prinzip. Die gerade vorgestellte Konfigurationsdatei und das Starten des Daemons kreieren den Server. Um nun auf jedem Knoten jeweils einen Client zu erstellen, wird auch hier eine Konfigurationsdatei benötigt. Auf Grundlage des Konzepts, ein Replicated Storage zu schaffen, hat die Datei den Inhalt, wie er in Abbildung 8 zu sehen ist. Die Quelle der Datei befindet sich in [HOW09], wobei sie nachträglich auf das PGE angepasst worden ist.

```
### Add client feature and attach to remote subvolume of server1
volume brick1
 type protocol/client
 option transport-type tcp/client
 option remote-host 192.168.100.1    # IP address of remote brick
 option remote-subvolume brick        # name of the remote volume
end-volume

### Add client feature and attach to remote subvolume of server2
volume brick2
 type protocol/client
 option transport-type tcp/client
 option remote-host 192.168.100.3     # IP address of remote brick
 option remote-subvolume brick        # name of the remote volume
end-volume

### The file index on server1
volume brick1-ns
 type protocol/client
 option transport-type tcp/client
 option remote-host 192.168.100.1    # IP address of remote brick
 option remote-subvolume brick-ns        # name of the remote volume
```

```
end-volume

### The file index on server2
volume brick2-ns
 type protocol/client
 option transport-type tcp/client
 option remote-host 192.168.100.3      # IP address of remote brick
 option remote-subvolume brick-ns        # name of the remote volume
end-volume

#The replicated volume with data
volume afr1
 type cluster/afr
 subvolumes brick1 brick2
end-volume

#The replicated volume with indexes
volume afr-ns
 type cluster/afr
 subvolumes brick1-ns brick2-ns
end-volume
```

Abbildung 8: Konfigurationsdatei für den GlusterFS- Client auf allen Knoten des PGE. Quelle: [HOW09]

Danach ist es möglich, das Verzeichnis, auf dem sich die Dateien auf den Mastern befinden, zu mounten, indem der folgende Befehl verwendet wird:

```
glusterfs -f /etc/glusterfs/glusterfs-client.vol /mnt/glusterfs
```

Damit beim Systemstart das Verzeichnis automatisch gemountet wird, existiert außerdem in `/etc/init.d` ein Startscript, das diese Aufgabe erledigt.

Somit können Dateien im Verzeichnis `/mnt/glusterfs` gelesen und geschrieben werden.

5 Ergebnisse und weiterführende Arbeiten

Abschließend werden in diesem Kapitel die erreichten Ergebnisse zusammengefasst und zukünftige Ideen und Erweiterungen dargestellt.

5.1 Linux Virtual Desktop

Mit dem Linux Virtual Desktop und dem verfügbaren Dateisystem GlusterFS steht eine Benutzerverwaltung und jedem Benutzer eine Unix-Benutzerumgebung auf dem Cluster zur Verfügung. Diese Benutzerverwaltung und der virtuelle Desktop wurden ohne den Overhead einer Virtualisierungsschicht auf dem Cluster erreicht. LVS zusammen mit Heartbeat und ldirectord machen aus dem PGE ein hochverfügbares Cluster. Durch diese Implementierung können Benutzer über eine virtuelle IP-Adresse eine SSH-Verbindung auf das Cluster aufbauen, indem sie per Round Robin auf die einzelnen Worker verteilt werden.

Ein Administrator kann zudem den Status von LVS abfragen. Auf jedem Knoten befindet sich die Desktop-Umgebung Gnome, sodass es für einen Benutzer des PGE möglich ist, Programme zu starten, die eine grafische Benutzeroberfläche voraussetzen.

Wie viele Benutzer sich an dem Cluster einloggen und auf dem Cluster arbeiten können, soll in Zukunft erprobt werden. Diese Erprobung soll zeigen, ob eventuell bezüglich der Benutzeranzahl durch die beiden vorhandenen Zugänge zum Cluster ein Engpass auftritt. Dazu steht den Studierenden, die eine Unix-Software-Entwicklungsumgebung benötigen, in einer Erprobungsphase der Linux Virtual Desktop zur Verfügung.

5.2 GlusterFS

Auf dem Cluster PGE liegt eine Dateiserverarchitektur mit GlusterFS vor, bei der die Dateien gespiegelt auf beiden Masterknoten gespeichert sind. Alle Worker und die beiden Master können auf die gemeinsamen Dateien sowohl lesend als auch schreibend zugreifen.

Zukünftige Projekte und Anforderungen benötigen möglicherweise eine andere Form der gemeinsamen Speichernutzung. Am wahrscheinlichsten ist es, dass mehr Speicher benötigt wird. Hierfür bietet sich das Distributed Storage aus Kapitel 4.1 an, bei dem Dateien nicht nur auf den Masterknoten, sondern verteilt auf mehreren Master- und Workerknoten vorliegen.

5.3 Cloud-Computing

Zur Unterstützung des Cloud-Computing und der Überführung von Power Grid Exist in ein Power Cloud Exist muss man auf das Cluster eine Virtualisierungsschicht aufpflanzen. Auf dieser Virtualisierungsschicht können dann unterschiedliche Betriebssysteme ablaufen. Inwieweit dann das Cluster eine Virtual Desktop-Infrastruktur zur Verfügung stellt und wie der Einsatz dann von LVS aussieht, ist zurzeit noch offen und bedingt weitere zukünftige Arbeiten.

Glossar

IPVS (IP Virtual Server)	Eine Ansammlung von Kernel Patches für Linux, die es einem Computer ermöglichen, als Cluster-Loadbalancer zu agieren.
LVS (Linux Virtual Server)	Eine Software zur Lastverteilung. Ermöglicht das Einrichten eines hochverfügbaren Clusters.
PGE (Power Grid Exist)	Ein Computercluster der Fakultät für Informatik an der Hochschule Mannheim.
Round Robin	Hier: Ein Lastverteilungsverfahren für Netzwerkdienste, bei dem eine Liste von Netzwerkressourcen vorliegt, aus der bei jeder Anfrage jeweils die nächste Ressource ausgewählt wird. Dadurch kann man ein gleichmäßig ausgelastetes System erreichen.
Single Point of Failure	Eine Komponente eines Systems, deren Ausfall zur Folge hat, dass das komplette System ausfällt.

Literatur

[BEN04] Bengel, Günther: Grundkurs – Verteilte Systeme: Grundlagen und Praxis des Client-Server-Computing – Inklusive aktueller Technologie wie Web-Services u.a. – Für Studenten und Praktiker. Vieweg+Teubner, 2004.

[BEN08] Bengel, Günther: Power Grid Exist: A Dynamic Cluster with Hierarchic Master Worker Architecture. Informatik-Berichte Hochschule Mannheim – Fakultät für Informatik, 2008.

[BEN10] Bengel, Günther et al.: Power Grid Exist: Ziele, Architektur und Aufbau. Informatik-Berichte Hochschule Mannheim – Fakultät für Informatik, 2010.

[BKS08] Bengel, Günther; Baun, Christian; Kunze, Marcel; Stucky, Karl-Uwe: Masterkurs Parallele und Verteilte Systeme. Vieweg+Teubner Verlag, 2008.

[CUD10] Artikel Compute Unified Device Architecture. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 10. August 2010. URL: http://en.wikipedia.org/w/index.php?title=CUDA&oldid=378151536

[GLU10] GlusterFS UserGuide – GlusterDocumentation. URL: http://www.gluster.com/community/documentation/index.php/GlusterFS_User_Guide (Abgerufen: 03. Juni 2010, 15:05)

[GLV08] Volfile dump – GlusterDocumentation. Bearbeitungsstand: 11. Dezember 2008. URL: http://europe.gluster.org/community/documentation/index.php?title=Volfile_dump&redirect=no

[GLW10] Artikel GlusterFS. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 11. April 2010. URL: http://de.wikipedia.org/w/index.php?title=GlusterFS&oldid=72950645

[HOW09] Howto install GlusterFS on Ubuntu | Share your knowledge!. Bearbeitungsstand: 19. Februar 2009. URL: http://blogama.org/node/78/

[KOP05] Karl Kopper: The Linux Enterprise Cluster: Build a Highly Available Cluster With Commodity Hardware And Free Software. No Starch Press, 2005.

[LDI10] ldirectord. Bearbeitungsstand: 28. Juli 2010. URL: http://horms.net/projects/ldirectord/

[LVS05] High Availability. Bearbeitungsstand: 12. Februar 2005. URL: http://www.linuxvirtualserver.org/HighAvailability.html

[MAP10] Artikel MapReduce. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 12. Juli 2010. URL: http://en.wikipedia.org/w/index.php?title=MapReduce&oldid=373016180

[MPI10] Artikel Message Passing Interface. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 01. August 2010. URL: http://en.wikipedia.org/w/index.php?title=Message_Passing_Interface&oldid=376627234