

*Universität Karlsruhe (TH)
Fakultät für Informatik*

Architektur von Webservern

Seminararbeit
von

Aysen Shibla

Sommersemester 2004

Betreuer: Jürgen Vogel

Universität Mannheim

Inhaltsverzeichnis

1. Einleitung.....	3
2. Motivation.....	5
2.1. Verfügbarkeit.....	5
2.1.1. Hardwareausfälle.....	5
2.1.2. Softwareausfälle.....	6
2.1.3. Serverausfälle.....	6
2.2. Skalierbarkeit.....	6
2.3. Leistung.....	7
3. Cluster.....	7
4. Clustering von Webservers.....	9
4.1. Hardwaregestützte Clustering-Lösungen.....	9
4.2. Softwaregestützte Clustering-Lösungen.....	11
4.2.1. Was ist DNS?.....	11
4.2.2. Round Robin Domain Name System Algorithmus.....	12
4.2.3. Primär/Backup Konfigurationen.....	13
5. Zusammenfassung.....	14
6. Literaturverzeichnis.....	15

1. Einleitung

Das World Wide Web ist ein grosses verteiltes System, das aus Millionen von Clients und Servern besteht, die auf der ganzen Welt verteilt sind und auf verknüpfte Dokumente zugreifen. Die Server verwalten die Dokumente, jedes Dokument ist als Datei gespeichert, während die Clients den Benutzern eine einfache Schnittstelle für die Darstellung und den Zugriff auf diese Dokumente bereitstellen.

Das Web begann seine Existenz als Projekt am CERN, dem European Particle Physics Laboratory in Genf, und man wollte damit einer grossen und geografisch verteilten Forschergruppe Zugriff auf gemeinsam genutzte Dokumente unter Verwendung eines einfachen Hypertext-Systems bieten.

Das Web wuchs langsam auf weltweite Grösse an. Wirklich populär ist es erst geworden, als grafische Benutzeroberflächen verfügbar wurden, insbesondere Mosaic. Seit 1994 werden die Webentwicklungen vom World Wide Web Consortium initiiert und gesteuert, einer Zusammenarbeit zwischen CERN und M.I.T. Dieses Konsortium ist dafür verantwortlich, Protokolle zu standardisieren und das Web weiter zu verbessern.

Ein Server im Web ist eine Maschine, auf der Web Dokumente gespeichert sind. Auf dieser Maschine ist ein spezielles Software installiert, ein Webserver, das die Clientanforderungen, ein Dokument zu laden, akzeptiert und überträgt es an den Client. Der Verweis auf ein Dokument erfolgt mithilfe einer Referenz, bezeichnet als URL (Uniform Resource Locator). Das URL gibt an, wo sich ein Dokument befindet, häufig unter Angabe des DNS-Namens des zugehörigen Servers sowie eines Dateinamens, nachdem der Webserver als Dokument in das lokale Dateisystem suchen kann. Das URL spezifiziert das Protokoll, mit dem das Dokument über das Netzwerk übertragen wird. Im Web kommunizieren Client und Server miteinander über das http Protokoll (HyperText Transfer Protocol). Zuerst baut der Client eine TCP Verbindung mit dem Server und fordert seine Anforderung am Server, der dieselbe Verbindung benutzt, um auf die Anfrage zu antworten und die gewünschten Daten dem Client zurück zu geben, nachdem die Verbindung beendet wird. Die aktuelle Version ist heute HTTP/1.1.

Seit seiner Einführung vor etwa 11 Jahren hat sich das Web ständig weiterentwickelt. Heute gibt es eine Vielzahl von Methoden und Werkzeugen, um Informationen zu erzeugen, die von Web-Clients und Webservern verarbeitet werden können. Die Web-Seiten sind mit der Dokumentenbeschreibungssprache HTML (Hypertext Markup Language) programmiert. Hierbei wird ASCII Text mit Hilfe von Steuerzeichen (Tags) formatiert. Web-Seiten mit diesen Elementen werden als statisch bezeichnet, da vorher bereits fest vorbereitete Abarbeitungspfade und Auswahlmöglichkeiten nicht verlassen werden können. Eine Web-Seite kann nur durch einen erneuten Zugriff auf den Webserver verändert werden. Abbildung 1 visualisiert den Ablauf einer Anforderung nach statischen Dateien. Heutzutage werden die meisten Webdokumente statisch erzeugt, aber mehrere neue Techniken (wie zum Beispiel XML, PHP, .NET oder JAVA,) dienen zur Dynamisierung der Web-Seiten, die durch Interaktion mit dem Benutzer erstellt

werden, die in Abhängigkeit von bestimmten Zuständen eine Änderung der Anzeige bewirken kann.

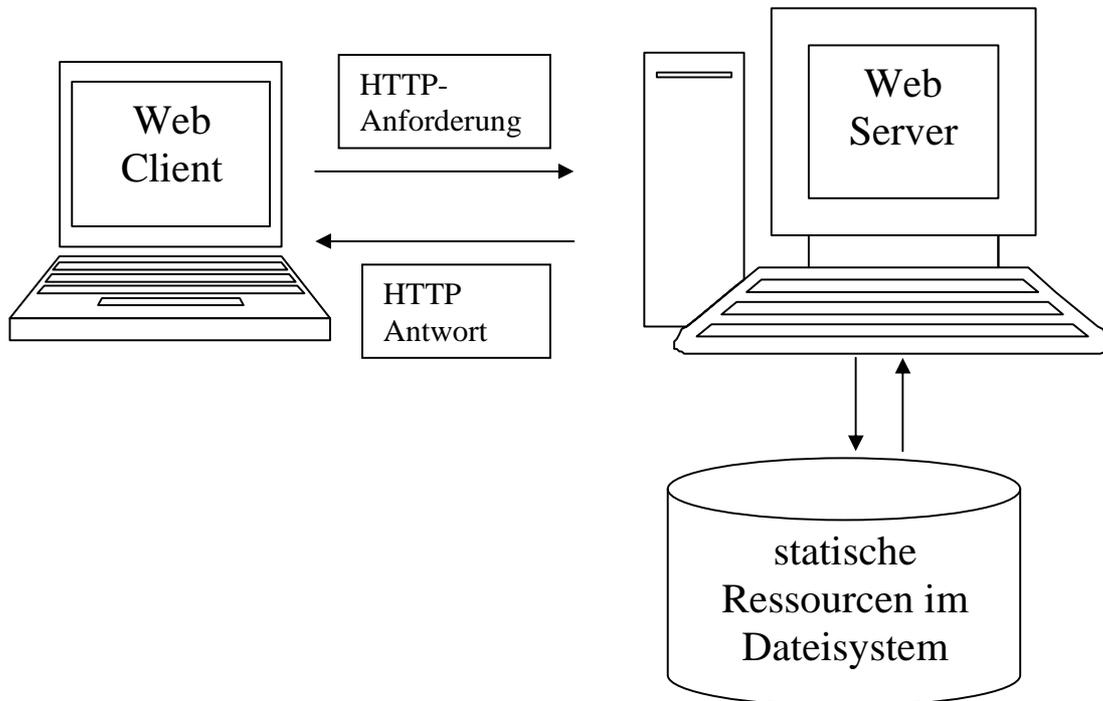


Abbildung 1: Anforderung nach statischen Ressourcen

Um ein konkretes Beispiel zu geben, wie ein Webserver konfiguriert wird und wie er funktioniert, betrachten wir die Architektur des berühmtesten Webserver, nämlich des Apache-Servers. Er besteht aus mehreren Modulen, die durch ein einzelnes Kernmodul gesteuert werden. Das Kernmodul akzeptiert die ankommenden HTTP - Anforderungen und gibt sie an die anderen Module weiter, somit bestimmt er den Steuerfluss für die Verarbeitung einer Anforderung. Das Kernmodul reserviert für jede Anforderung einen Datensatz mit Feldern für die Dokument Referenz, die in der HTTP - Anforderung enthalten ist. Jedes Modul arbeitet mit dem Datensatz, indem es die entsprechenden Felder liest und ändert. Nachdem alle Module gemeinsam die Anforderung verarbeitet haben, gibt das letzte Modul das angeforderte Dokument an den Client zurück. Jedes Modul muss eine oder mehrere Verarbeitungsroutinen bereitstellen, die vom Kernmodul aufgerufen werden können. Die Verarbeitungsroutinen sind sich alle ähnlich und nehmen einen Zeiger auf einen Anforderungsdatensatzes einzigen Eingabeparameter. Um die richtige Verarbeitungsroutine aufzurufen, wird die Verarbeitung von HTTP - Anforderungen in mehrere Phasen unterteilt. Ein Modul kann eine Verarbeitungsroutine für eine bestimmte Phase registrieren. Immer wenn eine Phase erreicht ist, überprüft das Kernmodul, welche Verarbeitungsroutinen für diese Phase registriert sind und ruft eine davon auf. Eine Verarbeitungsroutine kann eine Anforderung entweder zurückweisen, sie verarbeiten oder einen Fehler zurückgeben. Weist sie sie zurück, gibt sie damit an, dass sie sie nicht verarbeiten kann, sodass das Kernmodul eine andere Verarbeitungsroutine auswählen muss, die für die aktuelle Phase registriert ist. Kann eine Anforderung

verarbeitet werden, wird dann die nächste Phase gestartet. Wird ein Fehler zurückgemeldet, wird die Verarbeitung der Anforderung unterbrochen und dem Client wird eine Fehlermeldung zurückgegeben. Welche Module Teil des Apache-Webserver sind, wird bei der Konfiguration festgelegt. Damit mehrere Anforderungen gleichzeitig verarbeitet werden können, startet das Kernmodul für jede eingehende Anforderung einen neuen Prozess.

2. Motivation

Ein wichtiges Problem der Client-Server-Natur des Webs ist, dass ein Webserver einfach überlastet werden kann. Er muss tausende von gleichzeitigen Clientanforderungen bedienen und muss gemeinsam mit dem raschen Wachstum von Benutzerpopulation skalieren. Der Webserver muss schnell auf die Anforderungen der Clients antworten, indem er ihnen sichere Dienste und hohe Verfügbarkeit der Dienste anbieten kann. Denn ein Dienstausfall kann im heutigen kommerziellen Markt der Hochkonkurrenz zu ungewünschten Kunden- und Umsatzverlusten führen. Eine praktische Lösung, die in vielen Entwürfen verwendet wird, ist die Replikation eines Webserver auf einem Cluster aus Workstations. Es ist wichtig, dass im Fall eines Serverausfalls oder Serverüberlastung die Anforderungen, die zu dem ausgefallenen Webserver geschickt wurden, nicht verloren gehen, sondern von einem anderen Webserver ausgeführt werden, damit die Webseiten zu jeder Zeit für ihre Kunden verfügbar bleiben, auch dann, wenn es ein Ausfall im System vorhanden ist. In diesem Kapitel werden wir die wichtigsten Eigenschaften der Webdienste und der Webserver, nämlich Hochverfügbarkeit und Skalierbarkeit, betrachten

2.1. Verfügbarkeit

Webdienste hochverfügbar anzubieten, bedeutet, dass die verwendeten Maschinen gar nicht oder möglichst wenig ausfallen. Man muss die Zuverlässigkeit der Seite sicherstellen, indem man Strategien für Lastausgleich plant, konstruiert und einsetzt, mit denen gewährleistet wird, dass die Server zu jeder Zeit von jedem Ort für die Kunden funktionsfähig und verfügbar sind. Ein Serverausfall kann von einem Fehler verursacht werden, der sowohl in der Hardware, als auch in der Software auftreten kann. In beiden Fällen soll die Fehlerrate möglichst gering gehalten und der Fehler schnell entfernt werden.

2.1.1. Hardwareausfälle

Hardwareausfälle kommen wenig häufiger als Softwareausfälle vor. Als Ursache können sie u. a. defekte Festplatten, durchgebrannte Prozessoren und defekte Netzwerkkarten haben. Die Diagnose und Beseitigung dieser Probleme kann lange dauern, da die Beschaffung von Ersatzteilen und das Reparieren viel Zeit erfordern. Wenn die Webanwendung für das Unternehmen von grundlegender Bedeutung ist, sollte man sicherstellen, dass durch eine solide Strategie zur Hardware-Redundanz kostspielige Ausfallzeiten vermieden werden. Zu einer soliden Strategie gehören mindestens zwei oder besser drei Webserver.

2.1.2. Softwareausfälle

Softwareausfälle, die sich mit hoher Wahrscheinlichkeit auf eine Webanwendung auswirken, stehen in Zusammenhang mit dem Betriebssystem des Webservers, der Webserversoftware oder der Webanwendungssoftware. Wenn das Betriebssystem abstürzt oder fehlerhaft läuft, kann der Webserver ebenfalls nicht korrekt (oder überhaupt nicht) funktionieren, was sich negativ auf Verfügbarkeit, Zuverlässigkeit und Leistung der Webanwendung auswirken kann. Falls die Webserversoftware abstürzt oder fehlerhaft läuft, kann dies dazu führen, dass der Webserver seine Funktion einstellt. Auf Softwareausfälle kann man sich nur schwer vorbereiten; sofern für Ausfälle gespiegelte sekundäre Hardwaresysteme vorgesehen sind, können die Ausfallzeiten der Webanwendungen minimiert werden.

2.1.3. Serverausfälle

Außer dem Webserver können auch andere Server ausfallen, von denen die Webanwendungen abhängen, wodurch für die Seite entweder Ausfallzeiten oder verminderte Funktionsfähigkeit entstehen können. Beispielsweise kann bei verteilten Anwendungen ein Proxy Server ausfallen, was dazu führt, dass Anfragen an die Dienste der Webanwendungen nicht beantwortet werden. Der Datenbankserver kann ebenfalls abstürzen, wodurch es für Anwender unmöglich wird, Informationen an eine Datenbank zu schicken oder daraus abzufragen. Und bei einem Absturz des Mailservers sind Anwender nicht in der Lage, E-Mails zu verschicken.

Gegen diese Ausfälle muss man verschiedene Maßnahmen verwenden, um ihre Wirkung auf die Verfügbarkeit zu kontrollieren.

2.2. Skalierbarkeit

Gute Skalierbarkeit eines Webangebotes heißt, dass der eingesetzte Webserver kostengünstig und schnell in seiner Kapazität erweitert werden kann. Optimalerweise kann das durch einfaches Einsetzen von weiteren Servern oder durch Austausch von Maschinen durch stärkere ohne Dienstunterbrechung geschehen. Die Anzahl der gleichzeitigen Verbindungen zu dem Webserver hat eine direkte Auswirkung auf die Leistung. Aus diesem Grund müssen die Leistungsziele zwei Dimensionen berücksichtigen: die Geschwindigkeit der Transaktion eines einzelnen Anwenders und den Grad der Leistungsabnahme im Vergleich zur wachsenden Anzahl gleichzeitigen Anwender des Webservers. Deshalb muss man Werte für die Antwortzeit ermitteln und die höchste Anzahl der gleichzeitig mit der Seite verbundenen Anwender bestimmen und dann die Web Seite durch Hinzufügen von weiteren Webservern skalieren. Sobald die Web Seite auf mehreren Webservern läuft, müssen die Last und der Netzverkehr überwacht und verwaltet werden.

2.3. Leistung

Auf die Leistung und die Skalierbarkeit einer Web Anwendung wirken sich Engpässe negativ aus. Typische Engpässe sind:

- Prozessorkapazität – Selbst eine gut aufgebaute und programmierte Webanwendung kann eine schlechte Leistung zeigen, wenn die CPU des Webservers nicht in der Lage ist, ausreichende Rechenkapazität zur Verfügung zu stellen.
- Speicher – Unzureichender Arbeitsspeicher (RAM) begrenzt die Menge der Anwendungsdateien, die zwischengespeichert werden kann.
- Serverüberlastung – Serverüberlastung bezieht sich auf alle Arten von Servern, nicht nur den Webserver. Die Anwendung, Proxy, Such-, Index- und Backoffice Server können zeitweilig einem hohen Datenvolumen ausgesetzt sein, wodurch sich die Leistung der Webanwendungen indirekt verschlechtert.

3. Cluster

Ein Cluster ist die Verknüpfung mehrerer Systeme zu einer Gruppe, die über eine Cluster-Software verwaltet und gesteuert wird. Physisch gesehen ist ein Cluster eine Gruppe von zwei oder mehreren unabhängigen Webservern, die auf dieselben Daten zugreifen können und dieselbe Gruppe von Clients bedienen. Dies bedeutet, dass die Webserver mit gebräuchlichen In/Out-Bussen und einem normalen Netzwerk für den Client-Zugriff verbunden sind.

Logisch betrachtet stellt ein Cluster eine einzelne Verwaltungseinheit dar, in der ein beliebiger Webserver jedem autorisierten Client jede verfügbare Leistung anbieten kann. Die Webserver müssen Zugriff auf dieselben Daten haben und ein gemeinsames Sicherheitssystem besitzen. Dies bedeutet, dass in der Regel Server in einem Cluster dieselbe Architektur besitzen und unter der gleichen Version des gleichen Betriebssystems laufen. In Abbildung 2 ist ein Webserver Cluster von 4 Knoten gegeben. Alle Knoten haben die gleiche Hostname aber verschiedene IP-Adressen.

Es gibt eine Menge verschiedener Möglichkeiten, Clusters aufzubauen, dabei hängt es von der Art des Verbindungssystems, die für die Speicher und die Clients benutzt wird, auch von der Ebene der Anordnungen der gemeinsam benutzten Ressourcen, von der maximalen Zahl der Knoten im Verbund und von der Möglichkeit, alle Verbundressourcen gleichzeitig zu benutzen. Man kann sie allgemein in Abhängigkeit von der Speicherarchitektur klassifizieren.

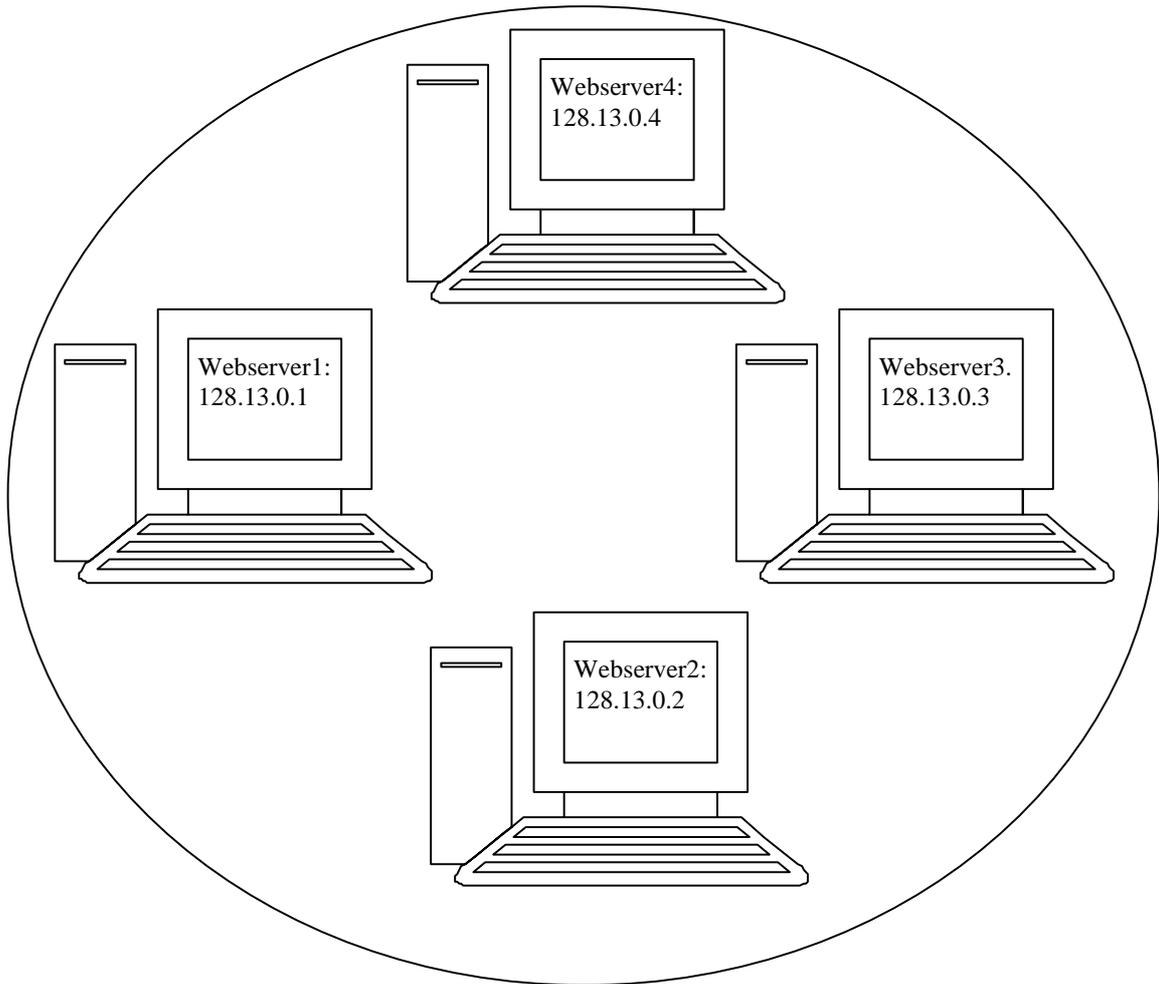


Abbildung2 : Webserver Cluster mit vier Knoten

Share-Nothing und Shared-Storage Architekturmodelle

In Abhängigkeit von der Speicherarchitektur teilen sich die Cluster in zwei Gruppen-Clusters mit share-nothing Architektur und mit shared-storage Architektur. Im ersten Fall hat jeder Knoten sein eigenes Speicher und Speicherressourcen. Es ist aber auch erlaubt, auf gemeinsame Ressourcen und Applikationen zuzugreifen, wenn diese Ressourcen zur gleichen Zeit nur von einem einzigen System verwendet und verwaltet werden, das heißt ein Knoten kann gleichzeitig von der gemeinsamen Festplatte lesen und schreiben. Ein Vorteil dieses Modells ist, dass der Ausfall eines einzigen Punktes, in diesem Fall der Ausfall eines gemeinsam benutzten Speicherressourcen, das ganze System zum Stillstand bringt und alle Knoten den Zugriff auf den Dateien verlieren. Ein weiterer Vorteil dieses Modells ist seine einfache Skalierbarkeit, weil Overhead und Auseinandersetzung (contention) fehlen. Es gibt einige Strategien, deren Ziel ist, den Bandbreitenoverhead (band width overhead) zu minimieren, indem die lokalen Festplatten durch das Netzwerk wieder spiegelt werden. Eine spezielle Verbindung kann benutzt werden, um die Widerspiegelung zu unterstützen, das ihren Verkehr von der

Verbindung des Clients isoliert. Im shared-storage Modell benutzen die Server gemeinsame Speicherressourcen. Es muss garantiert werden, dass sich nebenläufige Prozesse nicht wechselseitig stören. Es wird eine regelmässige Synchronisation der Festplattenzugriffe benötigt, um die Datenintegrität zu behalten.

4. Clustering von Webservers

Clustering für Skalierbarkeitszwecke funktioniert durch Verteilen der Last auf alle Webserver im Cluster, wobei eine nicht- intelligente, aber regelmässige Verteilfolge (Raund Robin Domain Name System oder Router) verwendet wird, ein zuvor definierter Schwellenwert, oder ein Algorithmus, den man für jeden Webserver im Cluster anpassen kann. Clustering zu Failover Zwecken beruht auf redundanten Webservern, die sicherstellen, dass geschäftskritische Anwendungen verfügbar bleiben, wenn ein Webserver im Cluster ausfällt. Clustering kann mit software- und auch mit hardwaregestützte Lösungen erreicht werden.

4.1. Hardwaregestützte Clustering-Lösungen

Eine häufig verwendete und zuverlässige Hardwarelösung für Clustering besteht aus einem Gerät, das als Paket-Router bezeichnet wird. Auf Abbildung 3 ist ein Router, der 3 Knoten eines Webserver Clusters verwaltet, zu sehen. Einer der am häufigsten verwendeten Router am Markt ist der LocalDirector von Cisco Systems. Ein Router ist einem Cluster aus Webservern vorgeschaltet und leitet eingehende HTTP-Anforderungen an verfügbare Webserver im Cluster weiter. Ein Router funktioniert so, dass er Geschwindigkeit und Volumen des Flusses an IP-Paketen von und zu den Webservern prüft und dann den für den aktuellen Netzverkehr am besten geeigneten Webserver auswählt. Dieser Prozess ist schnell und effizient. Weil nur die IP Adresse des Routers die öffentliche Adresse der Web Seite ist, hat Caching der IP Adressen kein Effekt auf dem Cluster. So können verschiedene Objekte einer Web Seite von verschiedenen Knoten im Cluster bedient werden. Um den Ausfall des Routers zu vermeiden, ist ein Mechanismus eingeführt, wo der verbundene Webserver die Antwort direkt dem Client übergibt, ohne weiteres Eingreifen durch den Routerknoten.

Router werden als halb-intelligente Geräte bezeichnet, da sie Serverausfälle erkennen und Anforderungen an andere Server umleiten können. Wenn ein Webserver ausfällt oder nicht mehr antwortet, sendet der Router keine Pakete mehr an den nicht reagierenden Webserver. Router werden aber nicht als vollständig intelligent betrachtet, denn obwohl sie beim Erkennen eines Ausfalls Anforderungen umleiten können, bieten sie keine Möglichkeit, für einzelne Webserver Schwellenwerte für die Umleitung zu konfigurieren. Sie sehen auch keinen Lastausgleich vor. Wenn der Router selbst ausfällt, dann wird Übernahme der IP Adresse benötigt. Die IP Adresse des ausgefallenen Routers muss den neuen Clients nicht bereitgestellt werden. Den Clients, die die Adresse des ausgefallenen Routers in ihrem Cache haben, wird die Adresse des Webservers übergeben, der die Aufgabe des ausgefallenen Routers übernimmt.

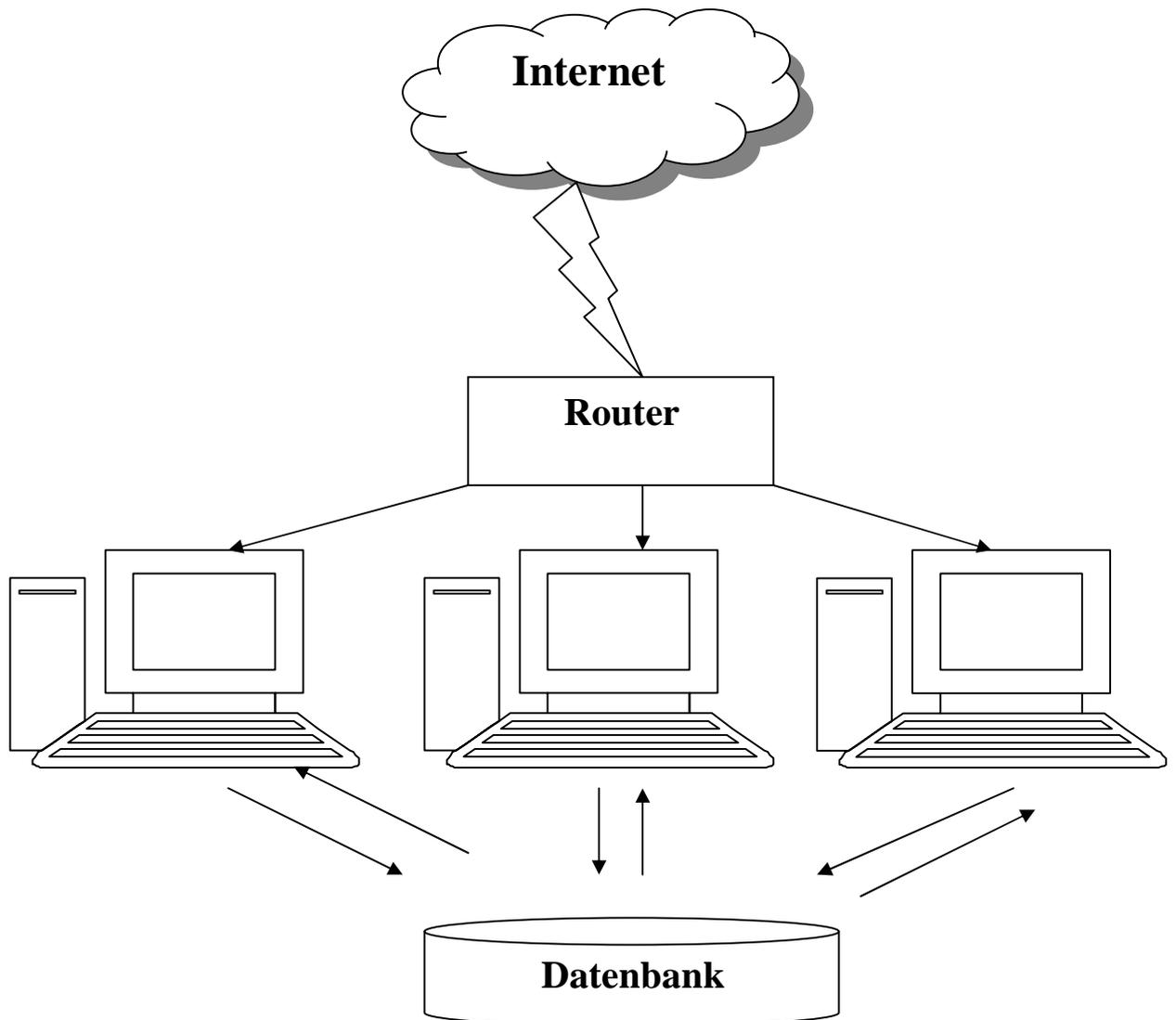


Abbildung 3: Router

In einem Cluster kann man auch so genannte Heartbeats zwischen den verteilten Komponenten verwenden. Dadurch können die Knoten effektiv den Zustand ihres Nachbarn überwachen und insgesamt die aktuelle Mitgliedschaft des Clusters feststellen, damit ausgefallene Webserver keine Anforderungen bekommen. Das Einsetzen von mehrfachen Heartbeats kann eine zuverlässige Information darüber geben, ob der Ausfall im Netzwerk oder im Prozessor ist. Zwei Knoten können in einem zweifachen Netzwerk nur dann miteinander kommunizieren, falls die beiden vital sind. Falls ein Ereignis registriert wird, indem man diese zuverlässigen Kanäle ausnutzt, dann kann jeder Knoten dem Clusterstatus trauen und mit seiner Wiederherstellungsaktion beginnen. Diese beinhalten Aufrufe der unterliegenden hochverfügbaren Infrastruktur auf niedriger Ebene, oder auch Aufruf von Programmen und Skripten auf höhere Ebene.

4.2. Softwaregestützte Clustering-Lösungen

Es sind verschiedene Arten von softwaregestützten Clustering-Lösungen erhältlich. Ähnlich wie hardwaregestützte Clustering-Lösungen haben auch softwaregestützte Lösungen ihre Stärken und Schwächen. Mögliche Softwarelösungen sind:

- Round-Robin-DNS – Diese Lösung wird sehr häufig verwendet, da sie relativ einfach und kostengünstig zu implementieren ist. Sie bietet jedoch keine Intelligenz zum Lastausgleich bzw. Failover.
- Clustering mit Primär/Backup-Systemen – Zwei identische Systeme sorgen für gegenseitige Redundanz. Dieser Clustering-Typ bietet keinen parallelen Server-Lastausgleich.

Zuerst betrachten wir das Round-Robin-DNS Verfahren, indem wir zuerst das Domain Name System kennenlernen.

4.2.1. Was ist DNS?

DNS sind Protokolle und Dienste in einem TCP/IP-Netzwerk, die es Netzwerkbenutzern ermöglichen, bei der Suche im Netzwerk nach anderen Hosts (Servers) hierarchische Namen in für Menschen lesbare Namen anstelle von IP-Computeradressen zu verwenden. DNS wird sowohl im Internet als auch in privaten Firmennetzwerken einschließlich LANs und WANs verwendet. Die primäre Fähigkeit des DNS ist die Möglichkeit, Hostnamen auf IP-Adressen abzubilden und umgekehrt. Beispielsweise könnte der Webserver von Macromedia die IP-Adresse 216.35.148.103 haben. Die meisten Benutzer würden für die Verbindung mit diesem Server den Domänennamen (www.macromedia.com) anstelle der umständlichen IP-Adresse eingeben. Der Name ist nicht nur leichter zu merken, er ist auch zuverlässiger, da sich die numerische Adresse aus einer Reihe von Gründen ändern kann, während der Name durch Reservierung erhalten bleibt.

Auswirkungen von DNS auf Leistung und Verfügbarkeit einer Website

DNS stellt im Internet einen leistungsfähigen und erfolgreichen Mechanismus dar, der es einer großen Anzahl von Einzelpersonen und Organisationen ermöglicht, leicht auffindbare Websites im Internet zu erstellen. Trotzdem kann DNS selbst der Grund dafür sein, dass eine Website nicht den Anforderungen des Clients an Leistung und Skalierbarkeit gerecht wird und daher nicht mehr verfügbar und damit unzuverlässig wird. Ob man DNS alleine zur Lastverteilung des eingehenden Netzverkehrs verwendet, hängt in erster Linie vom Zweck der Seite und der Menge der gleichzeitigen Aktivitäten ab, die man dort erwartet. Im Folgenden werden die wesentlichen Bestandteile von DNS beschrieben, die man konfigurieren muss, wenn die Webanwendungen gut mit DNS zusammenarbeiten sollen.

Zonen und Domänen

Ein Domänennamenssystem besteht aus einer verteilten Namensdatenbank. Durch die Namen in der DNS-Datenbank wird eine logische Baumstruktur namens Domain Name

Space (Domänennamensraum) aufgebaut. Im Internet wird das Stammverzeichnis (root) der DNS-Datenbank vom Internet Network Information Center (InterNIC) verwaltet. Die Top-Level-Domänen wurden ursprünglich nach Organisationen und Ländern zugewiesen. Für Länder werden aus zwei und drei Buchstaben bestehende Abkürzungen verwendet. Einige Abkürzungen sind für den Gebrauch durch Organisationen reserviert – z. B. .com, .gov und .edu für geschäftliche, behördliche bzw. schulische Einrichtungen.

Eine Domäne besteht aus einem Knoten in einem Netzwerk und allen darunter liegenden Knoten (Subdomänen), die in der Baumstruktur der DNS-Datenbank enthalten sind. Domänen und Subdomänen können zu Zonen gruppiert werden, um eine verteilte Verwaltung des Namensraums zu ermöglichen. Genauer gesagt ist eine Zone ein Bereich des DNS-Namensraums, dessen Ressourceneinträge in einer physischen Datei verwaltet werden. Ein DNS-Server kann dafür konfiguriert werden, eine oder mehrere Zonendateien zu verwalten. Jede Zone ist an einen bestimmten Domänenknoten gebunden. Zonen dienen dazu, Domänen in mehrere Segmente aufzuteilen, wenn die Verwaltung der Domäne auf mehrere Gruppen verteilt werden muss. Außerdem lassen sich Daten auf diese Weise effizienter replizieren.

DNS-Server speichern Informationen über den Domänennamensraum und werden als Namensserver bezeichnet. Namensserver besitzen normalerweise eine oder mehrere Zonen, für die sie verantwortlich sind. Der Namensserver besitzt die Autorität für diese Zonen und kennt alle anderen DNS-Namensserver in der gleichen Domäne. So kann sich ein Webbrowser zu einem Namensserver verbinden, um die IP Adresse einer Web Seite zu bestimmen.

4.2.2. Round Robin Domain Name System Algorithmus

Typischerweise verknüpfen die Namensserver einen Hostnamen zu einer einzigen IP Adresse. Um die Lastverteilung zu verbessern, benutzt man den Round Robin Domain Name System Algorithmus, das ein einziger Hostname zu mehreren IP Adressen verknüpft, das den verschiedenen Clients erlaubt, sich zu einem Clusterknoten anzubinden, der diesen Namen bedient. So wird eine einzige URL zu einer ganzen Gruppe von Webservern. Seiten wie Yahoo! Benutzen diese einfache Annäherung.

Ein grosses Problem dieses Algorithmus ist das Caching von Hostnamen und entsprechende IP Adressen in den Namensservern und auch auf Clientseite. Das kann zu grossen Ladeungleichgewichten im Cluster führen. Zwischen das RR-DNS und den Client gibt es eine ganze Reihe von Namensservern, die in ihre Caches die aufgelösten IP Adressen speichern. Um die Namensservern dazu zu zwingen, den Hostnamen zu verschiedenen IP Adressen aufzulösen, verwendet das RR-DNS TTL(time to live) Begrenzungen für die aufgelösten IP Adressen, so dass Anforderungen nach einer bestimmten Zeit nicht lokal verarbeitet werden, sondern werden zu dem entsprechenden RR-DNS weitergeleitet, um zu eine neue IP Adresse verknüpft zu werden. Namensauflösungsanforderungen, die in den TTL gesendet sind, werden alle mit der gleichen IP Adresse verknüpft, das wieder zu Probleme bei der Lastverteilung führen

kann, falls die TTL Periode zu knapp gewählt wird. Deshalb bestimmen alle Webserver alleine ihre TTL Periode.

Das zweite Problem bei dem RR-DNS Algorithmus ist das Clientcache und die dort gespeicherten Hostname-IP Adresse Paare. Auf diese Weise wird der Client immer zu demselben Webserver aus einem Webserver Cluster verbindet, das wieder zu Probleme bei der Lastverteilung führt, das Serverausfälle verursachen kann.

Ein weiterer Nachteil des RR DNS Algorithmus ist, dass wenn ein Clusterknoten ausfällt, verteilt die RR DNS weiterhin die Anforderungen unter allen Webservern, die auf die Anforderungen nicht antworten können. Daraus folgt, dass es alleine für unternehmenskritische Seiten keine zuverlässige Lösung im Bezug auf Lastausgleich und Ausfallsicherheit darstellt. Das Internet-DNS bietet also nur beschränkte Möglichkeiten, und der Round- Robin-Mechanismus sieht keine intelligenten Verfahren vor, mit denen sich Überlastungen oder Ausfälle von Servern überwachen oder verwalten.

4.2.3. Primär/Backup Konfigurationen

In der Primär/Backup-Konfiguration ist eine Maschine vorhanden, auf der die wichtigsten Applikationen laufen und eine zweite Maschine, die ausgeschaltet ist. Die beiden müssen nicht absolut identisch sein. Der zweite Rechner, der Backuprechner, braucht nur die grundlegenden Ressourcen, wie Festplatte, Speicher und Verbindungen, um die Applikationen unterstützen zu können. Es genügt eigentlich, wenn die Maschine nur die Hauptapplikationen ausführen kann, während der immer aktive Rechner, der Primärrechner nach seinem Ausfall repariert wird.

Softwaregestütztes Clustering von Webservern hat viele Vorteile. Zuerst sind softwaregestützte Clusters billiger im Vergleich mit den Kosten für Hardwaregeräte wie Router und Switches. Einige Softwarelösungen bieten eine Intelligenz, die Präventivmaßnahmen für den Lastausgleich ermöglicht und dadurch die Verfügbarkeit der Server noch steigert. Falls ein Webserver doch überlastet wird oder sogar ausfällt, kann die Software das Problem automatisch erkennen und HTTP-Anforderungen an verfügbare Server im Cluster umleiten. Anstatt sich nur auf ein einzelnes Gerät zu verlassen, werden die Fähigkeiten zum Lastausgleich auf mehrere Webserver im Cluster oder auf mehrere Cluster verteilt, sodass kein einzelner Serverausfall die Anwendung schaden kann.

Softwaregestütztes Clustering unterscheidet sich in ihren Fähigkeiten und Funktionen. Beispielsweise bieten einige keine automatische Ausfallerkennung, Benachrichtigung oder Übernahme von IP-Adressen, andere haben eine bedeutend verzögerte Erkennungsfunktion. Bei einigen könnte man Lastschwellenwerte als Präventivmaßnahme konfigurieren, bei anderen nicht.

Einige softwaregestützte Clustering-Lösungen sind relativ einfach strukturiert. Andere bringen aufgrund der gebotenen Funktionen, des Umfangs der anfänglichen Konfiguration und der folgenden Administration oder des Ausmaßes der erforderlichen

Integration mit anderen Systemen und Geräten einen höheren Grad der Komplexität mit sich.

Es ist auch möglich, beide Clustering Mechanismen zu kombinieren. Eine Kombination aus Hardware- und Softwarelösungen bietet sicherlich die größten Fähigkeiten bezüglich Skalierbarkeit und Verfügbarkeit der Web Seite. Eine kombinierte Lösung stellt eine attraktive Option dar, wenn das Unternehmen bereits in einem dieser Bereiche Investitionen getätigt hat, aber nach einer umfassenderen Lösung sucht. Die Integration von Hardware und Software bedeutet, dass das Unternehmen nicht unbedingt Kapitalverluste für Investitionen in die vorher verwendete Technologie abschreiben muss, wenn man sich für den Kauf zusätzlicher Clustering-Technologie entscheidet.

5. Zusammenfassung

Das WWW ist momentan das wichtigste und populärste dokumentbasierte Client-Server Applikation, das sich heute schnell weiterentwickelt und verbessert, indem es seiner Benutzer sichere und flexible Dienste zur Verfügung stellt, was als Folge Erhöhung der Anzahl der Benutzer und damit auch der Anzahl, des Grads der Leistung, Sicherheit und Zuverlässigkeit der Webserver hat, wobei verschiedene Massnamen verwendet werden. Ein solches zufrieden stellendes Verfahren ist das Clustering von WWW-Servern. In grosse skalierbare verbundene Webserver Konfigurationen werden Verfahren wie das RR-DNS nicht effektiv arbeiten. Router dagegen haben eine relativ einfache Arbeitsweise und können für Lastausgleichszwecke Verwendung finden, indem sich die Zahl der verwalteten Knoten beliebig viel steigern lässt.

„For the Web specifically, we were quite aware that something very significant was going on, but I don't think anyone expected it to become so popular so quickly”

Bill Gates

Literaturverzeichnis:

1. „Clustering Algorithms“, John Hartigan
2. „Cluster-Formation und –Analyse“, Helmut Späth
3. „Cluster Computing“, S. Martens
4. „The Web Server Book; tools & Techniques for building your own internet information site“, Jonathan Magid, Douglas Matthews, Paul Jones
5. “Das World Wide Web: Web - Server und -Clients “, Rainer Klute
6. “Apache Bibel”, A. Mohammed
7. “How to set up and maintain a World Wide Web Site: the guide for information providers”, Lincoln Stein
8. “Distributed Systems”, A. Tanenbaum
9. “Web Engineering”, Albrecht Schmidt
10. <http://www.google.com>