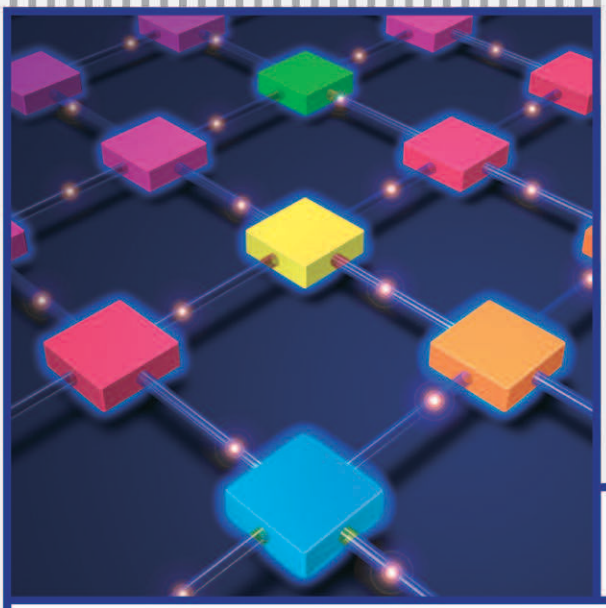

Andrew S. Tanenbaum
Maarten van Steen

Verteilte Systeme

Prinzipien und Paradigmen

2., aktualisierte Auflage

ALWAYS LEARNING

PEARSON

Andrew S. Tanenbaum
Maarten van Steen

Verteilte Systeme

Prinzipien und Paradigmen

2., aktualisierte Auflage

ein Imprint von Pearson Education
München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

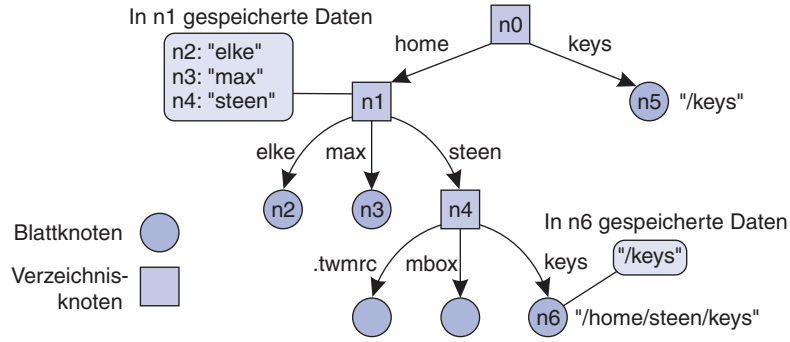

Abbildung 5.11: Das Konzept eines symbolischen Links, erklärt in einem Namensgraphen

Die Namensauflösung, wie wir sie bis hierhin beschrieben haben, vollzieht sich vollständig innerhalb eines einzelnen Namensraumes. Dennoch kann eine Namensauflösung auch dazu verwendet werden, verschiedene Namensräume in transparenter Weise miteinander zu verschmelzen. Zunächst wollen wir ein eingehängtes Dateisystem betrachten. In unserem Benennungsmodell entspricht ein eingehängtes Dateisystem einem Verzeichnisknoten, der den Bezeichner eines Verzeichnisknotens aus einem anderen Namensraum speichert, den wir als fremden Namensraum bezeichnen. Der Verzeichnisknoten, der den Knotenbezeichner speichert, wird Mountpoint genannt, ebenso auch der Verzeichnisknoten im fremden Namensraum. Der Mountpoint ist im Normalfall der Stamm eines Namensraumes. Während der Namensauflösung wird zuerst der Mountpoint nachgeschlagen und dann auf seine Verzeichnistabelle zugegriffen.

Das Prinzip des Mounting kann auch auf andere Namensräume verallgemeinert werden. Insbesondere wird ein Verzeichnisknoten benötigt, der als Mountpoint funktioniert und alle nötigen Informationen zur Erkennung des Mountpoint im fremden Namensraum und den Zugriff darauf speichert. Diesem Ansatz wird in vielen verteilten Systemen gefolgt.

Betrachten wir eine Gruppe von Namensräumen, die über mehrere Rechner verteilt ist. Insbesondere soll dabei jeder Namensraum über einen anderen Server laufen, möglicherweise auf einem anderen Rechner. Wenn wir einen fremden Namensraum NR_2 in einen Namensraum NR_1 bereitstellen wollen, kann es dementsprechend erforderlich sein, über ein Netzwerk mit dem Server für NR_2 zu kommunizieren, da dieser Server auf einem anderen Computer laufen könnte als der für NS_1. Das Mounting eines fremden Namensraumes in einem verteilten System erfordert zumindest folgende Informationen:

1. Den Namen eines Zugriffsprotokolls
2. Den Servernamen
3. Den Namen des Mountpoint im fremden Namensraum

Beachten Sie, dass jeder dieser Namen aufgelöst werden muss. Der Name des Zugriffsprotokolls muss zur Implementierung eines Protokolls aufgelöst werden, über das die Kommunikation mit dem Server des fremden Namensraumes erfolgen kann. Der Name des Servers muss zu einer Adresse aufgelöst werden, unter der dieser Server erreicht werden kann. Als letzter Teil in der Namensauflösung muss der Name des Mountpoint nach einem Knotenbezeichner in dem fremden Namensraum aufgelöst werden.

In nicht verteilten Systemen wird möglicherweise keiner dieser drei Punkte benötigt. In UNIX beispielsweise gibt es kein Zugriffsprotokoll und keinen Server. Auch der Name des Mountpoint ist nicht erforderlich, da er einfach das Wurzelverzeichnis des fremden Namensraumes ist.

Der Name des Mountpoint muss von dem Server des fremden Namensraumes aufgelöst werden. Wir brauchen jedoch auch Namensräume und Implementierungen für das Zugriffsprotokoll und den Servernamen. Eine Möglichkeit besteht darin, die drei oben aufgeführten Namen als URL darzustellen.

Zur Veranschaulichung betrachten wir eine Situation, in der ein Benutzer mit einem Laptop auf Dateien zugreifen will, die auf einem entfernten Dateiserver gespeichert sind. Der Client und der Dateiserver sind beide mit dem Network File System *(NFS)* von Sun konfiguriert, das wir in *Kapitel 11* eingehend erläutern werden. NFS ist ein verteiltes Dateisystem mit einem Protokoll, das genau beschreibt, wie ein Client auf eine Datei zugreifen kann, die auf einem (entfernten) NFS-Dateiserver liegt. Um NFS zu ermöglichen, über das Internet zu arbeiten, kann ein Client mithilfe eines NFS-URL insbesondere genau spezifizieren, auf welche Datei er zugreifen möchte, beispielsweise *nfs://flits.cs.vu.nl//home/steen*. Dieser URL benennt eine Datei (zufällig ein Verzeichnis) mit dem Namen `/home/steen` auf einem NFS-Dateiserver `flits.cs.vu.nl`, auf den ein Client über das NFS-Protokoll zugreifen kann (Shepler et al., 2003).

Der Name `nfs` ist ein Standardname (*Well-Known Name*) in dem Sinne, dass es ein weltweites Übereinkommen gibt, wie dieser Name zu interpretieren ist. Da wir es mit einem URL zu tun haben, wird der Name `nfs` zu einer Implementierung des NFS-Protokolls aufgelöst. Der Server-Name wird unter Verwendung von DNS zu seiner Adresse aufgelöst. Das werden wir in einem späteren Abschnitt erläutern. Wie bereits gesagt, wird `/home/steen` vom Server des fremden Namensraumes aufgelöst.

▶Abbildung 5.12 zeigt die Teile der Struktur eines Dateisystems auf dem Client. Das Wurzelverzeichnis hat eine Reihe benutzerdefinierter Einträge, darunter ein Unterverzeichnis namens `/remote`. Dieses Unterverzeichnis soll Mountpoints für fremde Namensräume enthalten, wie zum Beispiel das Benutzerverzeichnis an der Vrije Universiteit. Zu diesem Zweck wird ein Verzeichnisknoten namens `/remote/vu` zum Speichern des URL *nfs://flits.cs.vu.nl//home/steen* verwendet.

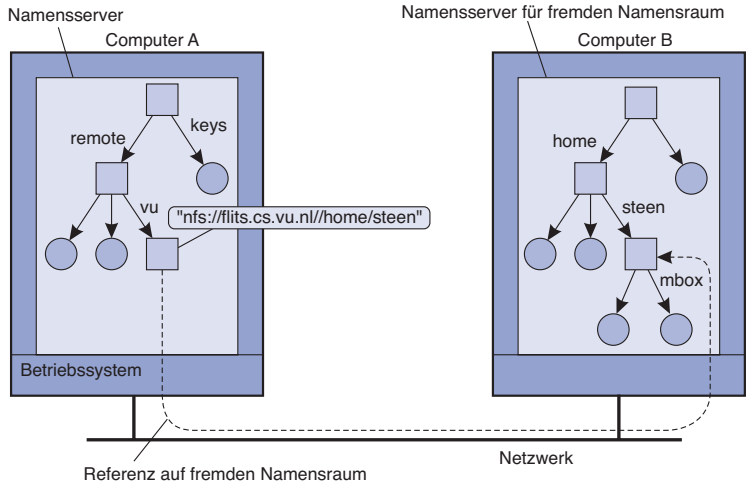

Abbildung 5.12: Mounting entfernter Namensräume durch ein besonderes Zugriffsprotokoll

Betrachten wir nun den Namen `/remote/vu/mbox`. Die Auflösung dieses Namens beginnt im Wurzelverzeichnis auf dem Client und wird fortgeführt, bis der Knoten `/remote/vu` erreicht ist. Die Namensauflösung setzt sich mit der Rückgabe des URL *nfs://flits.cs.vu.nl//home/steen* fort. Der Client kontaktiert dabei abwechselnd mithilfe des NFS-Protokolls den Dateiserver `flits.cs.vu.nl` und greift nachfolgend auf das Verzeichnis `/home/steen` zu. Die Namensauflösung kann dann fortgeführt werden, indem die Datei `mbox` in diesem Verzeichnis gelesen wird, wonach der Auflösungsprozess endet.

Verteilte Systeme, die das gerade beschriebene Mounting eines entfernten Dateisystems ermöglichen, lassen einen Client zum Beispiel folgende Befehle ausführen:

```
cd /remote/vu
ls -l
```

Dieser führt daraufhin die Dateien im Verzeichnis `/home/steen` auf dem entfernten Dateiserver auf. Der Reiz der Sache liegt darin, dass dem Benutzer die Einzelheiten des eigentlichen Zugriffes auf den entfernten Server erspart bleiben. Im Idealfall ist im Vergleich zum Zugriff auf lokal verfügbare Dateien nur eine gewisse Leistungseinbuße erkennbar. Für den Client sieht es so aus, als bildeten der Namensraum auf dem lokalen Computer und der unter `/home/steen` auf dem entfernten Rechner einen einzigen Namensraum.

5.3.3 Die Implementierung eines Namensraumes

Ein Namensraum bildet das Herz eines Namensdienstes, also eines Dienstes, der es Benutzern und Prozessen ermöglicht, Namen hinzuzufügen, zu entfernen und nachzuschlagen. Ein Namensdienst wird durch Namensserver implementiert. Wenn ein verteiltes System auf ein lokales Netzwerk begrenzt ist, lässt sich ein Namensdienst meist mithilfe eines einzigen Namensservers verwirklichen. In umfangreichen verteilten Systemen mit vielen Entitäten, die vielleicht auch noch geografisch weit auseinanderliegen, ist es hingegen erforderlich, die Implementierung eines Namensraumes auf mehrere Namensserver zu verteilen.

Verteilung des Namensraumes

Namensräume für ein großes, eventuell weltweit verteiltes System werden gewöhnlich hierarchisch strukturiert. Nehmen Sie wie zuvor an, dass ein derartiger Namensraum nur einen einzigen Wurzelknoten aufweist. Um einen solchen Namensraum zu implementieren, ist es praktisch, ihn in logische Schichten zu unterteilen. Cheriton und Mann (1989) unterscheiden die folgenden drei Schichten.

Die globale Schicht wird von den Knoten auf der höchsten Ebene gebildet, also dem Wurzelknoten und anderen Verzeichnisknoten, die diesem logisch nahe stehen, insbesondere seinen Kindknoten. Knoten in der globalen Schicht zeichnen sich oft durch ihre Stabilität aus, in dem Sinne, dass die Verzeichnistabellen nur selten geändert werden. Solche Knoten können Organisationen oder Gruppen von Organisationen abbilden, für die Namen im Namensraum gespeichert sind.

Die Administrationsschicht wird von Verzeichnisknoten gebildet, die innerhalb einer Organisation verwaltet werden. Ein typisches Merkmal der Verzeichnisknoten in der Administrationsschicht ist, dass sie Gruppen von Entitäten darstellen, die zur selben Organisation oder Verwaltungseinheit gehören. Es kann zum Beispiel einen Verzeichnisknoten für jede Abteilung einer Organisation geben oder einen Verzeichnisknoten, über den sich alle Hosts finden lassen. Ein anderer Verzeichnisknoten kann als Ausgangspunkt zur Benennung aller Benutzer dienen usw. Die Knoten in der Administrationsschicht sind relativ stabil, auch wenn Änderungen im Allgemeinen häufiger auftreten als bei Knoten in der globalen Schicht.

Die Managementschicht schließlich besteht aus Knoten, die regelmäßig Änderungen unterliegen. Zu dieser Schicht gehören beispielsweise Knoten, die Hosts im lokalen Netzwerk darstellen. Aus dem gleichen Grund enthält die Schicht Knoten, die gemeinsam genutzte Dateien abbilden, wie solche für Bibliotheken oder ausführbare Dateien. Eine andere wichtige Art von Knoten umfasst solche, die benutzerdefinierte Verzeichnisse und Dateien darstellen. Im Gegensatz zu der globalen und der Administrationsschicht werden die Knoten in der Managementschicht nicht nur von Systemadministratoren verwaltet, sondern auch von einzelnen Endbenutzern eines verteilten Systems.

Zur Veranschaulichung zeigt ▶Abbildung 5.13 ein Beispiel der Unterteilung eines Teiles des DNS-Namensraumes, einschließlich der Namen von Dateien in einer Organisation, auf die über das Internet zugegriffen werden kann, wie zum Beispiel Internetseiten und übertragbare Dateien. Der Namensraum ist in überschneidungsfreie Teile eingeteilt, die in DNS *Zonen* genannt werden (Mockapetris, 1987). Eine Zone ist ein Teil des Namensraumes, der durch einen eigenen Namensserver implementiert wird. Einige dieser Zonen verdeutlicht Abbildung 5.13.

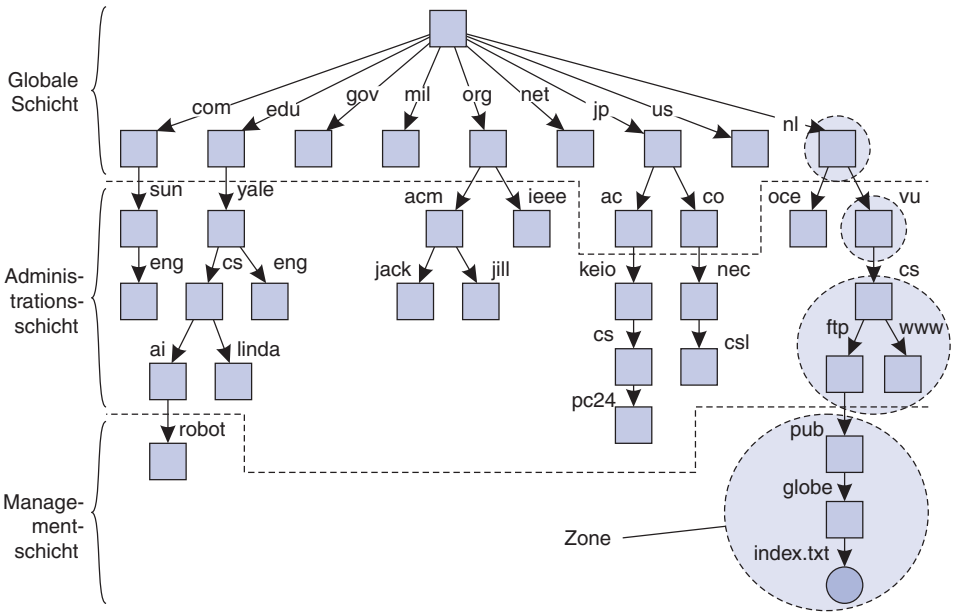

Abbildung 5.13: Beispiel für eine Partitionierung des DNS-Namensraumes in drei Schichten, einschließlich der über das Internet verfügbaren Dateien

In Bezug auf Verfügbarkeit und Leistung müssen Namensserver in jeder Schicht unterschiedlichen Anforderungen genügen. Hohe Verfügbarkeit ist von besonderer Bedeutung für Namensserver in der globalen Schicht. Wenn ein Namensserver ausfällt, ist ein großer Teil des Namensraumes nicht erreichbar, weil die Namensauflösung nicht über den ausgefallenen Server hinaus fortgesetzt werden kann.

Das Thema Leistung ist etwas kompliziert. Aufgrund der geringen Änderungsrate bei Knoten in der globalen Schicht gelten die Ergebnisse von Nachschlageoperationen im Allgemeinen für eine lange Zeit. Damit können die Clients solche Ergebnisse zwischenspeichern (also lokal speichern). Wenn die gleiche Lookup-Operation wieder ausgeführt wird, können die Ergebnisse aus dem Cache des Clients abgerufen werden, statt einer Lieferung der Ergebnisse durch den Namensserver. Das führt dazu, dass Namensserver in der globalen Schicht nicht schnell auf eine einzelne Nachschlageanforderung antworten müssen. Andererseits kann Durchsatz von Bedeutung sein, insbesondere in großen Systemen mit Millionen von Benutzern.

Die Anforderungen an Namensserver in der globalen Schicht hinsichtlich Verfügbarkeit und Leistung können durch Replikation von Servern in Verbindung mit Caching durch die Clients erfüllt werden. Wie wir in *Kapitel 7* erläutern werden, müssen Aktualisierungen in dieser Schicht im Allgemeinen nicht sofort in Kraft treten, was es sehr viel einfacher macht, die Konsistenz der Replikate sicherzustellen.

Die Verfügbarkeit eines Namensservers in der Administrationsschicht ist hauptsächlich von Belang für Clients in derselben Organisation wie der Namensserver. Wenn der Namensserver ausfällt, lassen sich viele Ressourcen innerhalb dieser Organisation nicht ansprechen, da sie nicht nachgeschlagen werden können. Andererseits kann es für Benutzer außerhalb dieser Organisation von geringerer Bedeutung sein, dass deren Ressourcen zeitweilig nicht verfügbar sind.

Was die Leistung angeht, haben Namensserver in der Administrationsschicht ähnliche Eigenschaften wie die in der globalen Schicht. Da Änderungen an Knoten nicht allzu oft auftreten, kann das Zwischenspeichern von Nachschlageergebnissen höchst effektiv sein, was Leistungsprobleme entschärft. Im Gegensatz zur globalen Schicht sollten in der Administrationsschicht aber Nachschlageergebnisse innerhalb von wenigen Millisekunden zurückgegeben werden, entweder direkt vom Server oder aus dem lokalen Puffer des Clients. Gleichermaßen sollten Aktualisierungen im Allgemeinen schneller bearbeitet werden als in der globalen Schicht. So sind stundenlange Wartezeiten, bis ein Konto für einen neuen Benutzer verwendet werden kann, nicht akzeptabel.

Diese Anforderungen lassen sich oft erfüllen, indem man Namensserver auf Hochleistungsrechnern ausführt. Zur Erhöhung der allgemeinen Verfügbarkeit sollte zudem das Caching durch die Clients in Kombination mit Replikation praktiziert werden.

Die Anforderungen an die Verfügbarkeit von Namensservern auf der Managementebene sind im Allgemeinen weniger anspruchsvoll. Auf die Gefahr eines zeitweiligen Ausfalls hin genügt es insbesondere zumeist, für Namensserver einen einzigen (dedizierten) Rechner zu nutzen. Leistung ist indessen ein entscheidendes Thema. Benutzer erwarten, dass Aktionen sofort vonstatten gehen. Wegen regelmäßiger Akualisierungen ist das Caching durch die Clients oft weniger nützlich, falls nicht besondere Maßnahmen ergriffen werden, die wir in *Kapitel 7* erörtern.

Aspekt	Globale Schicht	Administrations-schicht	Management-schicht
Geografische Ausdehnung des Netzwerkes	Weltweit	Organisation	Abteilung
Knotengesamtzahl	Wenige	Viele	Zahllose
Reaktionsfähigkeit auf Nachschlageanforderungen	Sekunden	Millisekunden	Sofort
Verbreitung von Aktualisierungen	Verzögert	Sofort	Sofort
Anzahl Replikate	Viele	Keine oder wenige	Keine
Caching durch Clients	Ja	Ja	Manchmal

Abbildung 5.14: Vergleich zwischen Namensservern zur Implementierung eines großen Namensraumes, partitioniert in eine globale Schicht, eine Administrationsschicht und eine Managementschicht

Einen Vergleich zwischen Namensserver auf unterschiedlichen Schichten zeigt ▶Abbildung 5.14. In verteilten Systemen sind Namensserver in der globalen Schicht und der Administrationsschicht am schwierigsten zu implementieren. Schwierigkeiten bereiten die Replikation und das Caching. Sie werden aus Gründen der Verfügbarkeit und der Leistung benötigt, führen aber zu Problemen in der Konsistenz. Manche Probleme werden dadurch verschärft, dass gepufferte Dateien und Replikate über ein weitläufiges Netzwerk verstreut werden, was zu langen Verzögerungen in der Kommunikation führt und damit die Synchronisierung noch zusätzlich erschwert. Replikation und Caching behandeln wir ausführlich in *Kapitel 7*.

Implementierung der Namensauflösung

Die Verteilung eines Namensraumes über mehrere Namensserver wirkt sich auf die Implementierung der Namensauflösung aus. Zur Erklärung wollen wir vorerst voraussetzen, dass der Namensserver nicht repliziert und auf der Client-Seite keine Caches verwendet werden. Jeder Client hat Zugriff auf einen lokalen Namensauflöser *(Resolver)*, der dafür verantwortlich ist, den Prozess der Namensauflösung sicherzustellen. Mit Blick auf ▶Abbildung 5.13 nehmen wir an, dass der (absolute) Pfadname

```
root:<nl, vu, cs, ftp, pub, globe, index.html>
```

aufzulösen ist. In URL-Notation entspricht dieser Pfadname *ftp://ftp.cs.vu.nl/pub/globe/index.html*. Es gibt nun zwei Möglichkeiten, die Namensauflösung umzusetzen.

Bei der iterativen Namensauflösung übergibt ein Resolver den vollständigen Namen an den Stammnamensserver. Das setzt die Kenntnis der Adresse voraus, an der der Root Server erreicht werden kann. Der Root Server löst den Pfadnamen auf, soweit er das kann, und gibt das Ergebnis an den Client zurück. In unserem Beispiel kann der Root Server nur die Bezeichnung `nl` auflösen und gibt die Adresse des zugehörigen Namensservers zurück.

An diesem Punkt gibt der Client den restlichen Pfadnamen (d.h. `nl:<vu, cs, ftp, pub, globe, index.html>`) an diesen Namensserver weiter. Dieser kann nur die Beschriftung `vu` auflösen und gibt die Adresse des zugehörigen Namensservers zusammen mit dem verbleibenden Pfadnamen `vu:<cs, ftp, pub, globe, index.html>` zurück.

Der Resolver des Clients kontaktiert daraufhin diesen nächsten Namensserver, der mit der Auflösung der Bezeichnung `cs` und nachfolgend auch `ftp` antwortet und die Adresse des FTP-Servers zusammen mit dem Pfadnamen `ftp :<pub, globe, index.html>` zurückgibt. Der Client spricht dann den FTP-Server an und fordert ihn auf, den letzten Teil des ursprünglichen Pfadnamens aufzulösen. Der FTP-Server löst daraufhin die Bezeichnung `pub`, `globe` und `index.html` auf und überträgt die angeforderte Datei (in diesem Fall über FTP). Diesen Prozess der iterativen Namensauflösung zeigt ▶Abbildung 5.15. (Die Schreibweise #<cs> bezeichnet die Adresse des Servers, der für den Knoten <cs> zuständig ist.)

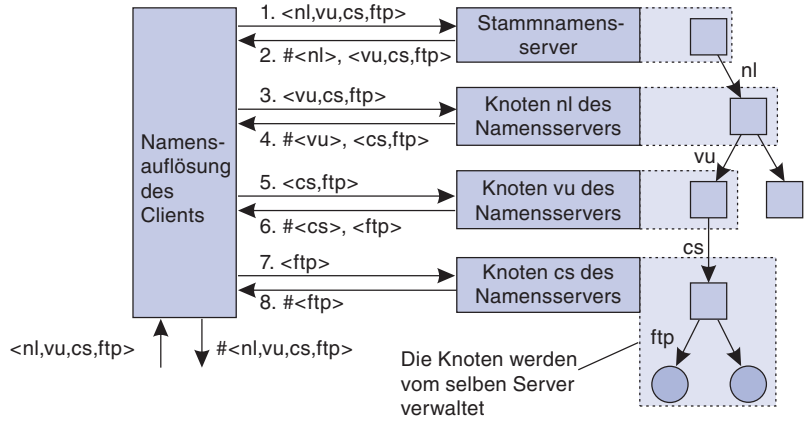

Abbildung 5.15: Das Prinzip der iterativen Namensauflösung

In der Praxis ist es der Client-Prozess, der den letzten Schritt separat ausführt, insbesondere die Kontaktaufnahme mit dem FTP-Server und die an ihn gerichtete Aufforderung, die Datei mit dem Pfadnamen `ftp:<pub, globe, index.html>` zu übertragen. Der Client reicht also im Normalfall nur den Pfadnamen `root:<nl, vu, cs, ftp>` an den Resolver weiter und erwartet, im Gegenzug die Adresse zu erhalten, unter der er den FTP-Server erreichen kann. Auch dies zeigt Abbildung 5.15.

Eine Alternative zur iterativen Namensauflösung ist die rekursive Namensauflösung. Statt jedes Zwischenergebnis an den Resolver des Clients zurückzugeben, sendet ein Namensserver das Ergebnis bei rekursiver Namensauflösung an den nächsten Namensserver, den er findet. Wenn der Stammnamensserver also zum Beispiel die Adresse des Namensservers findet, der für den Knoten `nl` zuständig ist, fordert er diesen Server auf, den Pfadnamen `nl:<vu, cs, ftp, pub, globe, index.html>` aufzulösen. Dieser nächste Server löst den vollständigen Pfad ebenfalls unter Verwendung rekursiver Namensauflösung auf und gibt schließlich die Datei `index.html` an den Root Server zurück, der diese wiederum an den Resolver des Clients weitergibt.

▶Abbildung 5.16 zeigt die rekursive Namensauflösung. Wie bei der iterativen Namensauflösung erfolgt der letzte Schritt (der Kontakt mit dem FTP-Server und die an ihn gerichtete Anfrage, die genannte Datei zu übertragen) im Allgemeinen als separater Client-Prozess.

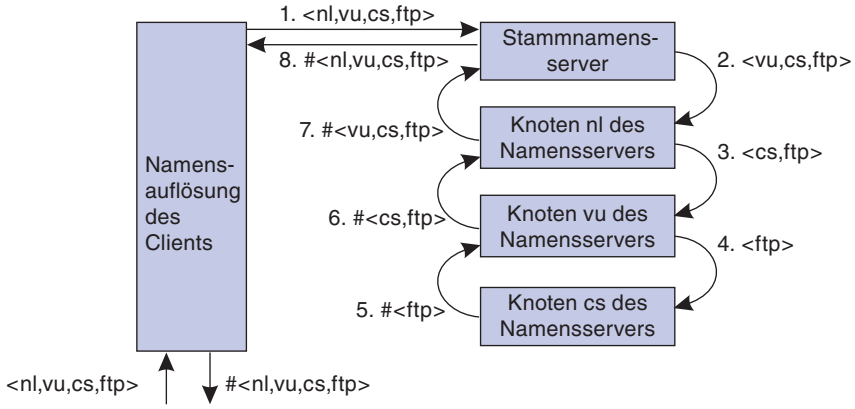

Abbildung 5.16: Das Prinzip der rekursiven Namensauflösung

Der Hauptnachteil der rekursiven Namensauflösung besteht darin, dass sie größere Anforderungen an die Leistung jedes Namensservers stellt. Im Grunde muss ein Namensserver die vollständige Auflösung eines Pfadnamens leisten, auch wenn er das gemeinsam mit anderen Namensservern tun kann. Diese zusätzliche Last ist gewöhnlich so groß, dass Namensserver in der globalen Schicht eines Namensraumes nur die iterative Namensauflösung unterstützen.

Die rekursive Namensauflösung weist zwei bedeutsame Vorteile auf. Der erste besteht darin, dass das Zwischenspeichern der Ergebnisse im Vergleich zur iterativen Namensauflösung effektiver ist. Der zweite Vorteil liegt in der möglichen Senkung der Kommunikationskosten. Um diese Vorteile zu erklären, setzen wir voraus, dass die Namensauflösung eines Clients nur Pfadnamen annimmt, die auf Knoten in der globalen Schicht oder der Administrationsschicht des Namensraumes verweisen. Um den Teil eines Pfadnamens aufzulösen, der Knoten in der Managementschicht entspricht, kontaktiert ein Client gesondert den Namensserver, den die Namensauflösung zurückgibt, wie wir es weiter vorn beschrieben haben.

Die rekursive Namensauflösung ermöglicht es jedem Namensserver, allmählich die Adressen der Namensserver zu lernen, die für die Knoten auf tieferen Ebenen zuständig sind. Infolgedessen kann das Caching erfolgreich zur Leistungssteigerung genutzt werden. Wenn der Root Server zum Beispiel aufgefordert wird, den Pfadnamen `root:<nl, vu, cs, ftp>` aufzulösen, wird er schließlich die Adresse des für den Knoten zuständigen Namensservers erhalten, auf den dieser Pfadname verweist. Um an diesen Punkt zu gelangen, muss der Namensserver für den Knoten `nl` die Adresse des Namensservers für den Knoten `vu` nachschlagen und Letzterer die Adresse des Namensservers für den Knoten `cs`.

Da Änderungen an Knoten in der globalen und der Administrationsschicht nicht oft auftreten, kann der Stammnamensserver die zurückgegebene Adresse zwischenspeichern. Weil die Adresse durch die Rekursion auch an die Namensserver zurückgegeben wird, die für die Knoten `vu` und `nl` verantwortlich sind, kann sie sogar auch von diesen Servern gepuffert werden.

Genauso können die Zwischenergebnisse beim Nachschlagen von Namen auch zurückgegeben und gepuffert werden. Beispielsweise muss der Server für den Knoten `nl` die Adresse des Knotenservers `vu` nachschlagen. Diese Adresse kann an den Root Server zurückgegeben werden, wenn der Server `nl` das Ergebnis der ursprünglichen Namenssuche liefert. Einen vollständigen Überblick über den Auflösungsprozess und die Ergebnisse, die von jedem Namensserver zwischengespeichert werden können, gibt ▶Abbildung 5.17.

Knoten, für den der Server zuständig ist	Aufzulösender Bestandteil	Nachgeschlagener Bestandteil	An Kindknoten weitergegebener Bestandteil	Empfangene und zwischengespeicherte Adresse	Rückgabe an die anfordernde Stelle
cs	<ftp>	#<ftp>	-	-	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs,ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
Wurzel-knoten	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Abbildung 5.17: Rekursive Namensauflösung von `<nl, vu, cs, ftp>` mit Pufferung der Zwischenergebnisse durch die Namensserver für spätere Suchvorgänge

Der Hauptvorteil dieses Ansatzes liegt darin, dass Nachschlageoperationen letztendlich recht effizient bearbeitet werden können. Lassen Sie uns zum Beispiel unterstellen, dass ein anderer Client zu einem späteren Zeitpunkt die Auflösung des Pfadnamens `root:<nl, vu, cs, flits>` anfordert. Dieser Name wird an den Wurzelknoten weitergeleitet, der ihn sofort an den Namensserver für den Knoten `cs` sendet und diesen auffordert, den restlichen Pfadnamen `cs:<flits>` aufzulösen.

Bei iterativer Namensauflösung ist das Zwischenspeichern zwangsläufig auf die Namensauflösung des Clients beschränkt. Wenn ein Client *A* und später ein Client *B* die Auflösung desselben Namens anfordern, muss die Namensauflösung dementsprechend zweimal über die gleichen Namensserver laufen. Viele Unternehmen verwenden als Kompromisslösung einen lokalen zwischengeschalteten Namensserver, der von allen Clients gemeinsam genutzt wird. Dieser lokale Namensserver bearbeitet alle Namensanforderungen und puffert die Ergebnisse. Ein solcher Zwischenserver ist auch aus Managementsicht praktisch. So muss nur dieser Server wissen, wo sich der Stammnamensserver befindet; andere Rechner benötigen diese Information nicht.

Der zweite Vorteil der rekursiven Namensauflösung besteht darin, dass sie oft kostengünstiger im Bereich Kommunikation ist. Betrachten wir wieder die Auflösung des Pfadnamens `root:<nl, vu, cs, ftp>` und unterstellen, dass sich der Client in San Francisco befindet. Unter der Annahme, dass der Client die Adresse des Servers für den Knoten `nl` kennt, verläuft die Kommunikation bei rekursiver Namensauflösung entlang der in ▶Abbildung 5.18 als *R1* bezeichneten Route vom Host des Clients in San Francisco zum Server `nl` in den Niederlanden. Von da an ist eine Kommunikation zwischen dem `nl`-Server und dem Namensserver der Vrije Universiteit auf dem Universitätscampus in Amsterdam erforderlich. Diese Kommunikation wird als *R2* dargestellt. Schließlich zeigt *R3* die zwischen dem `vu`-Server und dem Namensserver im Fachbereich Informatik erforderliche Kommunikation. Die Antwort folgt der gleichen Route in umgekehrter Richtung. Offensichtlich diktiert somit der Nachrichtenverkehr zwischen dem Host des Clients und dem `nl`-Server die Kommunikationskosten.

Bei der iterativen Namensauflösung muss der Host des Clients hingegen getrennt mit den Servern `nl`, `vu` und `cs` kommunizieren. Damit dürften die Gesamtkosten rund das Dreifache von denen der rekursiven Namensauflösung betragen. Die mit *I1*, *I2* und *I3* bezeichneten Pfeile in ▶Abbildung 5.18 zeigen den Weg der Kommunikation bei iterativer Namensauflösung.

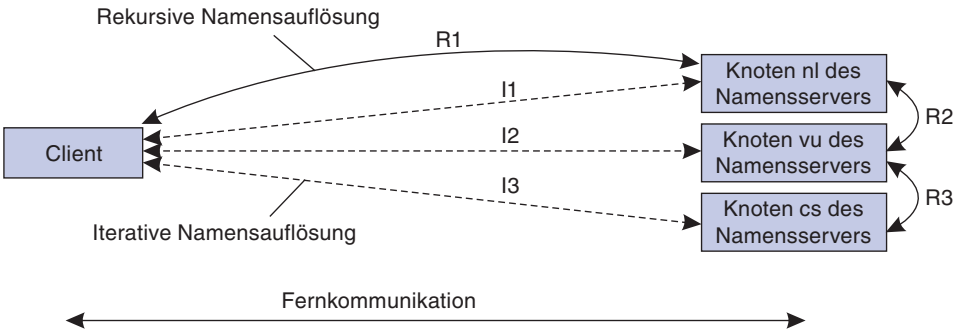

Abbildung 5.18: Vergleich der Kommunikationskosten bei rekursiver und iterativer Namensauflösung

Copyright