Beenden und Neustarten

Dieses Dokument umfasst das Beenden und Neustarten des Apache auf Unix-ähnlichen Systemen. Anwender von Windows NT, 2000 und XP sollten Betreiben des Apache als Dienst [1] lesen, während hingegen Anwender von Windows 9x sowie ME Betreiben des Apache als Konsolenanwendung [2] lesen sollten, um mehr Informationen zur Handhabung des Apache auf diesen Systemen zu erhalten.

Themen

Siehe auch

- httpd[3]
- apachectl[4]

Einleitung

Um den Apache zu stoppen oder neu zu starten, müssen Sie ein Signal an den laufenden `httpd`-Prozess senden. Es gibt zwei Möglichkeiten, diese Signale zu senden. Zum einen können Sie den Unix-Befehl `kill` verwenden, um den Prozessen direkt Signale zu senden. Sie werden feststellen, dass auf Ihrem System mehrere `httpd`-Programme laufen. Sie sollten jedoch nicht jedem dieser Prozesse ein Signal senden, sondern nur dem Elternprozess, dessen PID im `PidFile` steht. Das heißt, Sie sollten es niemals nötig haben, einem anderen Prozess, als dem Elternprozess, ein Signal zu senden. Es gibt drei Signale, die Sie an den Elternprozess senden können: `TERM`, `HUP` und `USR1`, die nachfolgend beschrieben werden.

Um dem Elternprozess ein Signal zu senden, verwenden Sie einen Befehl wie z.B.:

```
kill -TERM `cat /usr/local/apache2/logs/httpd.pid`
```

Die zweite Methode, dem `httpd`-Prozess zu signalisieren, ist die Verwendung der `-k`-Befehlszeilenoptionen `stop`, `restart` und `graceful`, wie unten beschrieben. Dies sind Argumente des httpd[3]-Programms, es wird jedoch empfohlen, sie unter Verwendung des Steuerskripts apachectl[4] zu senden, welches diese an `httpd` durchreicht.

Nachdem Sie `httpd` signalisiert haben, können Sie dessen Fortschritt beobachten, indem Sie eingeben:

```
tail -f /usr/local/apache2/logs/error_log
```

Passen Sie diese Beispiele entsprechend Ihren `ServerRoot`- und `PidFile`-Einstellungen an.

Beenden

Signal: TERM

```
apachectl -k stop
```

Das Senden des `TERM`- oder `stop`-Signals an den Elternprozess veranlasst diesen, sofort zu versuchen, alle seine Kindprozesse zu beenden. Es kann einige Sekunden dauern, bis alle Kindprozesse komplett

beendet sind. Danach beendet sich der Elternprozess selbst. Alle gerade bearbeiteten Anfragen werden abgebrochen. Es werden keine weiteren Anfragen mehr bedient.

Unterbrechungsfreier Neustart

Signal: USR1

```
apachectl -k graceful
```

Das `USR1`- oder `graceful`-Signal veranlasst den Elternprozess, die Kinder *anzuweisen*, sich nach Abschluß ihrer momentanen bearbeiteten Anfrage zu beenden (oder sich sofort zu beenden, wenn sie gerade keine Anfrage bedienen). Der Elternprozess liest seine Konfigurationsdateien erneut ein und öffnet seine Logdateien neu. Wenn ein Kindprozess stirbt, ersetzt der Elternprozess ihn durch ein Kind der neuen Konfigurations-*Generation*. Dieses beginnt sofort damit, neue Anfragen zu bedienen.

> Auf bestimmten Plattformen, welche kein `USR1` für einen unterbrechungsfreien Neustart erlauben, kann ein alternatives Signal verwendet werden (wie z.B. `WINCH`). Der Befehl `apachectl graceful` sendet das jeweils richtige Signal für Ihre Platform.

Der Code ist dafür ausgelegt, stets die MPM-Direktiven zur Prozesssteuerung zu beachten, so dass die Anzahl der Prozesse und Threads, die zur Bedienung der Clients bereitstehen, während des Neustarts auf die entsprechenden Werte gesetzt werden. Weiterhin wird `StartServers` auf folgende Art und Weise interpretiert: Wenn nach einer Sekunde nicht mindestens `StartServers` neue Kindprozesse erstellt wurden, dann werden, um den Durchsatz zu beschleunigen, entsprechend weitere erstellt. Auf diese Weise versucht der Code sowohl die Anzahl der Kinder entsprechend der Serverlast anzupassen als auch Ihre Wünsche hinsichtlich des Parameters `StartServers` zu berücksichtigen.

Benutzer von `mod_status` werden feststellen, dass die Serverstatistiken **nicht** auf Null zurückgesetzt werden, wenn ein `USR1` gesendet wurde. Der Code wurde so geschrieben, dass sowohl die Zeit minimiert wird, in der der Server nicht in der Lage ist, neue Anfragen zu bedienen (diese werden vom Betriebssystem in eine Warteschlange gestellt, so dass sie auf keinen Fall verloren gehen) als auch Ihre Parameter zur Feinabstimmung berücksichtigt werden. Um dies zu erreichen, muss die *Statustabelle* (Scoreboard), die dazu verwendet wird, alle Kinder über mehrere Generationen zu verfolgen, erhalten bleiben.

Das Statusmodul benutzt außerdem ein `G`, um diejenigen Kinder zu kennzeichen, die noch immer Anfragen bedienen, welche gestartet wurden, bevor ein unterbrechungsfreier Neustart veranlaßt wurde.

Derzeit gibt es keine Möglichkeit für ein Log-Rotationsskript, das `USR1` verwendet, sicher festzustellen, dass alle Kinder, die in ein vor dem Neustart geöffnetes Log schreiben, beendet sind. Wir schlagen vor, dass Sie nach dem Senden des Signals `USR1` eine angemessene Zeitspanne warten, bevor Sie das alte Log anfassen. Wenn beispielsweise die meisten Ihrer Zugriffe bei Benutzern mit niedriger Bandbreite weniger als 10 Minuten für eine vollständige Antwort benötigen, dann könnten Sie 15 Minuten warten, bevor Sie auf das alte Log zugreifen.

> Wenn Ihre Konfigurationsdatei Fehler enthält, während Sie einen Neustart anweisen, dann wird Ihr Elternprozess nicht neu starten, sondern sich mit einem Fehler beenden. Im Falle eines unterbrechungsfreien Neustarts läßt er die Kinder weiterlaufen, wenn er sich beendet. (Dies sind die Kinder, die sich "sanft beenden", indem sie ihre letzte Anfrage erledigen.) Das verursacht Probleme, wenn Sie versuchen, den Server neu zu starten -- er ist nicht in der Lage, sich an die Ports zu binden, an denen er lauschen soll. Bevor Sie einen Neustart durchführen, können Sie die Syntax der Konfigurationsdateien mit dem Befehlszeilenargument `-t` überprüfen (siehe auch httpd[3]). Das garantiert allerdings nicht, dass der Server korrekt starten wird. Um sowohl die

Syntax als auch die Semantik der Konfigurationsdateien zu prüfen, können Sie versuchen, `httpd` als nicht-root-Benutzer zu starten. Wenn dabei keine Fehler auftreten, wird er versuchen, seine Sockets und Logdateien zu öffnen und fehlschlagen, da er nicht root ist (oder weil sich der gegenwärtig laufende `httpd` bereits diese Ports gebunden hat). Wenn er aus einem anderen Grund fehlschlägt, dann liegt wahrscheinlich ein Konfigurationsfehler vor. Der Fehler sollte behoben werden, bevor der unterbrechungsfreie Neustart angewiesen wird.

Neustarten

Signal: HUP

```
apachectl -k restart
```

Das Senden des Signals `HUP` oder `restart` veranlaßt den Elternprozess, wie bei `TERM` alle seine Kinder zu beenden. Der Elternprozess beendet sich jedoch nicht. Er liest seine Konfigurationsdateien neu ein und öffnet alle Logdateien erneut. Dann erzeugt er einen neuen Satz Kindprozesse und setzt die Bedienung von Zugriffen fort.

Benutzer von `mod_status` werden feststellen, dass die Serverstatistiken auf Null gesetzt werden, wenn ein `HUP` gesendet wurde.

Wenn Ihre Konfigurationsdatei einen Fehler enthält, während Sie einen Neustart anweisen, dann wird Ihr Elternprozess nicht neu starten, sondern sich mit einem Fehler beenden. Lesen Sie oben, wie Sie das vermeiden können.

Anhang: Signale und Wettkampfsituationen

Vor der Version 1.2b9 des Apache existierten verschiedene *Wettkampfsituationen* (race conditions), die den Neustart und die Signale beeinflußt haben. (Eine einfache Beschreibung einer Wettkampfsituation lautet: es ist ein zeitabhängiges Problem; wenn etwas zum falschen Zeitpunkt erfolgt, wird es sich nicht wie erwartet verhalten.) Bei Architekturen mit dem "richtigen" Funktionsumfang haben wir so viele eliminiert wie wir nur konnten. Dennoch sollte beachtet werden, dass noch immer Wettkampfsituationen auf bestimmten Architekturen existieren.

Bei Architekturen, die ein `ScoreBoardFile` auf Platte verwenden, besteht die Gefahr, dass die Statustabelle beschädigt wird. Das kann zu "bind: Address already in use" ("bind: Adresse wird bereits verwendet", nach einem `HUP`) oder "long lost child came home!" ("Der verlorene Sohn ist heimgekehrt", nach einem `USR1`) führen. Ersteres ist ein schwerer Fehler, wärend letzteres lediglich bewirkt, dass der Server einen Eintrag in der Statustabelle verliert. So kann es ratsam sein, unterbrechungsfreie Neustarts zusammen mit einem gelegentlichen harten Neustart zu verwenden. Diese Probleme lassen sich nur sehr schwer umgehen, aber glücklicherweise benötigen die meisten Architekturen keine Statustabelle in Form einer Datei. Bitte lesen Sie für Architekturen, die sie benötigen, die Dokumentation zu `ScoreBoardFile`.

Alle Architekturen haben in jedem Kindprozess eine kleine Wettkampfsituation, welche die zweite und nachfolgende Anfragen einer persistenten HTTP-Verbindung (KeepAlive) umfaßt. Der Prozess kann nach dem Lesen der Anfragezeile aber vor dem Lesen der Anfrage-Header enden. Es existiert eine Korrektur, die für 1.2 zu spät kam. Theoretisch sollte das kein Problem darstellen, da der KeepAlive-Client derartige Ereignisse aufgrund von Netzwerk-Latenzzeiten und Auszeiten des Servers erwarten sollte. In der Praxis scheint keiner von beiden beeinflußt zu werden -- in einem Testfall wurde der Server zwanzig mal pro Sekunde neu gestartet, während Clients das Angebot abgegrast haben, ohne kaputte Bilder oder leere Dokumente zu erhalten.

URI-Referenzen

[1] http://httpd.apache.org/docs-2.1/platform/windows.html#winsvc
[2] http://httpd.apache.org/docs-2.1/platform/windows.html#wincons
[3] http://httpd.apache.org/docs-2.1/programs/httpd.html
[4] http://httpd.apache.org/docs-2.1/programs/apachectl.html