

Web Engineering – 3. Übungsblatt

Name: Oliver Knörzer
E-Mail: novil@gmx.de

1) - Installation: Das MPM-Prozessmodell muss in der Konfigurationsphase vor der Kompilierung des Apache-Webrowsers ausgewählt werden, ansonsten werden für die verschiedenen Betriebssysteme die Standard-MPMs verwendet. Dies geschieht mit der Anweisung `./configure --with-mpm=NAME`, wobei NAME der Name des gewünschten MPMs ist, z. B. `worker`. Nach der Kompilierung kann das ausgewählte MPM mit `./httpd -l` abgefragt werden, wodurch alle installierten Module aufgeführt werden.

- Konfiguration: Die Einstellungen für die MPM-Module können einfach in der Hauptkonfigurationsdatei `httpd.conf` konfiguriert werden. Die Einstellungsdaten können dabei von einem `<IfModule prefork.c>...</IfModule>`-Tag eingeschlossen werden, wodurch sie ignoriert werden, wenn das entsprechende Modul, hier also Prefork, nicht installiert ist. Für Prefork sehen die Default-Einstellungen wie folgt aus:

```
StartServers          5
MinSpareServers      5 (Minimale Anzahl der unbeschäftigten Kindprozesse
des Servers)
MaxSpareServers      10 (Maximale Anzahl der unbeschäftigten Kindprozesse
des Servers)
MaxClients           150
MaxRequestsPerChild  0 (Obergrenze für die Anzahl von Anfragen, die ein
einzelner Kindprozess während seines Lebens bearbeitet; 0 = unendlich viele)
```

- Das Forker-Prozessmodell geht so vor, dass der Steuerprozess Kindprozesse generiert, die auf Anfragen warten und bedienen. Während sich der Steuerprozess normalerweise an Port 80 bindet, laufen die Kindprozesse in einem weniger privilegierten Modus. Der Hauptprozess leitet die Anfragen an die Kindprozesse weiter. Die Anzahl an Kindprozessen ist variabel, d. h. eine Lastanpassung durch Erzeugen und Löschen von Kindprozessen ist möglich. Dabei wird immer versucht, eine gewisse Anzahl an Kindprozessen frei zu halten, damit eine neue Anfrage nicht darauf warten muss, bis ein neuer Kindprozess gestartet oder frei wird.

- Das Worker-Prozessmodell weicht vom Forker-Prozessmodell dadurch ab, dass die Kindprozesse zusätzliche Threads (\triangleq leichtgewichtige Prozesse) enthalten. Einer dieser Threads ist als Listener-Thread dafür verantwortlich, Anfragen zur weiteren Bearbeitung an die Server-Threads weiterzureichen. Die Anzahl an Threads pro Kindprozess ist festgelegt, die Lastanpassung erfolgt durch das Erzeugen und Löschen von Kindprozessen. In der Konfiguration können die Anzahl an Threads pro Kindprozess und andere Einstellungen getätigt werden.

- Das Forker-Prozessmodell ist ein stabiles Modell, welches auf jedem Unix-System läuft. Aus Kompatibilitätsgründen ist es vor allem dann geeignet, wenn mit nicht Thread-sicheren Bibliotheken gearbeitet werden muss, so dass das Worker-Prozessmodell nicht zum Einsatz kommen kann. Außerdem ist es einfacher, eine strikte Trennung der einzelnen Kindprozesse zu gewährleisten als mit Threads.

- Das Worker-Prozessmodell verbraucht durch die Verwendung von Threads bei gleicher Anzahl an Anfragen deutlich weniger Systemressourcen. Gerade für stark frequentierte Webserver bietet sich daher der Einsatz des Worker-Prozessmodells an. Da trotzdem auch weiterhin mehrere Kindprozesse gleichzeitig verfügbar sind, geht dies in der Regel nicht zu Lasten der Stabilität.

- Probleme beim Einsatz von Prefork:

- + Die Wahl für `MaxClients` muss groß genug sein, um so viele gleichzeitige Anfragen wie erwartet zu bedienen, andererseits muss aber gewährleistet sein, dass auch genug Hauptspeicher für alle Prozesse vorhanden ist, ohne dass Teile auf den Hintergrundspeicher ausgelagert werden müssen.
- + Prozesse benötigen mehr Ressourcen, z. B. mehrere MByte Hauptspeicher pro Prozess, als leichtgewichtige Threads.
- + Auch die Generierung von Prozessen ist entsprechend aufwendiger.

- Probleme beim Einsatz von Worker:

- + Wenn ein Thread abstürzt, ist der gesamte Prozess davon betroffen. Daher ist die Stabilität (etwas) eingeschränkt.
- + Threads werden noch nicht von allen Betriebssystemumgebungen unterstützt.
- + Durch die Verwendung von Threads ist auch eine Thread-sichere Programmierung notwendig, die z. B. den konkurrierenden Zugriff auf gemeinsame Variablen verbietet. Insbesondere die mangelnde Thread-Unterstützung von CGI-Anwendungen wie PHP4 macht(e) dies problematisch.
- + In Ausnahmefällen kann es vorkommen, dass sehr viele Kindprozesse vorhanden sind, aber nicht gelöscht werden können, weil sie noch einen/wenige aktive Threads enthalten.

2) - Zuerst einmal ist als Vorteil des Web-Switching auf Layer 7 durch Redirects zu nennen, dass dadurch alle weiteren Zugriffe nicht mehr über den Load-Balancer laufen sondern direkt erfolgen, wodurch der Load-Balancer entlastet wird.

- Wenn nun aber der Fall eintritt, dass viele Kunden, die rechenaufwendige Routenberechnungen durchführen, auf einen einzelnen Webserver geleitet werden, so wird dieser überlastet, während die anderen Webserver, die nur Kunden bedienen, die statische Inhalte abfragen, noch Kapazitäten frei haben. Dieser Sachverhalt muss bei der Server-Auswahl durch den Load-Balancer beachtet werden, z. B. würde sich der Einsatz eines Response-Time-Verfahrens anbieten. Nicht gut geeignet wären dagegen Round-Robin- und Least-Connection-Verfahren.