

Aufgabenblatt 1

(Zu bearbeiten bis 30.10.)

Vorlesung Verteilte Systeme Wintersemester 2025/26

Aufgabe 1: Heterogenität und Offenheit

Ein Server-Programm, das in einer bestimmten Sprache (z.B. C++) geschrieben wurde, stellt Dienste (in Form von Methoden) bereit, die von Clients genutzt werden können, die in einer anderen Sprache geschrieben sind (z.B. Java). Client- und Server-Rechner können unterschiedliche Hardware und Betriebssysteme haben, sind aber beide an das Internet angeschlossen.

- a) Welche Probleme entstehen durch die Heterogenität und müssen gelöst werden, damit der Client eine Methode des Servers aufrufen kann?
- b) Ein offenes verteiltes System erlaubt das Hinzufügen neuer Dienste, die von einer Vielzahl von Clients genutzt werden können. Beschreiben Sie im Kontext des obigen Beispiels, wie sich die Anforderungen für die Offenheit von denen für die Heterogenität unterscheiden.
- c) Angenommen, der Server stellt neben öffentlichen Methoden, die von jedermann genutzt werden können, auch geschützte Methoden bereit, die nur gesondert angegebenen Benutzern zur Verfügung stehen. Zählen Sie alle Probleme auf, die gelöst werden müssen, damit nur berechtigte Nutzer die geschützten Methoden aufrufen können. Gehen Sie davon aus, daß die Methoden Information liefern, die nicht für alle Nutzer vorgesehen ist und überlegen Sie, welche weiteren Probleme daraus entstehen.

Aufgabe 2: Migration von Ressourcen

Ein Dienst, der Informations-Ressourcen (Daten) verwaltet, wird von mehreren Servern implementiert. Aus welchen Gründen kann es sinnvoll sein, diese Ressourcen zwischen den Servern zu migrieren?

Aufgabe 3: Architektur bekannter Client/Server-Anwendungen

Informieren Sie sich über die Client/Server-Architektur einer oder mehrerer großer Internet-Anwendungen (Web, Email, News) und beschreiben / illustrieren Sie diese Architektur (welche verschiedenen Server sind involviert, wie arbeiten diese zusammen, wie sind die Daten auf die Server verteilt bzw. repliziert, welche Informationen werden ausgetauscht?)

Aufgabe 4: Sitzungsverfolgung

In einfachen Web-Anwendungen können zwei verschiedene Techniken zur Verfolgung von Sitzungen verwendet werden: Der Server kann dem Client eine Sitzungs-ID in einem Cookie übergeben oder die Sitzungs-ID wird in die URL encodiert. Informieren Sie sich über die Details beider Techniken und überlegen Sie sich die Vor- und Nachteile.

Aufgabe 5: Antwortzeit von Client-Anwendungen

Welche Faktoren beeinflussen die Antwortzeit von Client-Anwendungen, die auf von einem Server verwaltete, gemeinsam genutzte Daten zugreifen? Welche (sinnvollen) Ansätze gibt es, um die Antwortzeit zu verkürzen?

Aufgabe 6: Server-Implementierung

In dieser Aufgabe soll das Lesen und Schreiben einer Datei unter der Verwendung eines single-threaded Servers und unter der Verwendung eines multi-threaded Servers untersucht werden. Es dauert 15 Millisekunden, eine Anforderung für Arbeit zu erhalten, sie weiterzugeben und die restliche erforderliche Verarbeitung vorzunehmen, vorausgesetzt, die benötigten Daten befinden sich im Datei-Cache im Hauptspeicher. Ist eine Festplattenoperation erforderlich, wie es in einem Drittel der Fälle vorkommt, sind weitere 75 Millisekunden erforderlich, während der Thread schläft.

- a) Wie viele Anforderungen/Sekunde kann der Server verarbeiten, wenn er single-threaded ist, wie viele, wenn er multi-threaded ist?
- b) Ist es sinnvoll, die Anzahl der Threads in einem Server-Prozess zu begrenzen?