UNIVERSITÄT SIEGEN

Fakultät IV
Betriebssysteme und verteilte Systeme
Prof. Dr. rer. nat. Roland Wismüller

Aufgabenblatt 2

(Zu bearbeiten bis 08.05.2020)

Vorlesung Verteilte Systeme
Sommersemester 2020

Aufgabe 1: Kommunikationsformen in verteilten Systemen

Man unterscheidet grundsätzlich zwei Arten von Kommunikation in verteilten Systemen: gesicherte Nachrichtenströme und ungesicherte Datagramme. Welches sind die wesentlichen Charakteristiken beider Kommunikationsformen und welche Vor- und Nachteile ergeben sich daraus? Nennen Sie mögliche Einsatzgebiete.

Aufgabe 2: Datenformat-Konvertierung

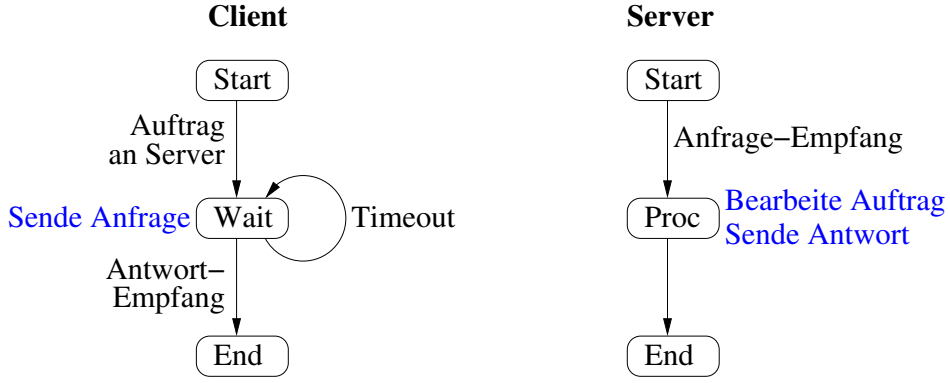
Bei der Kommunikation in verteilten Systemen ist es im Allgemeinen notwendig, intern in den Prozessen gespeicherte Datenstrukturen in ein zur Übertragung geeignetes Format zu konvertieren. Aus welchen Gründen ist dies unabdingbar? Diskutieren Sie notwendige Maßnahmen für unterschiedliche Datentypen.

Aufgabe 3: Semantik der Client/Server-Kommunikation

a) Bei der Client/Server-Kommunikation kann es zu Problemen kommen, wenn bei der Übertragung zwischen Client und Server Nachrichten verloren gehen. Die Behandlung solcher Fehler kann unterschiedlich sein, wodurch sich für die Kommunikation verschiedene Semantiken ergeben:

- *at least once* - der Auftrag wird mindestens einmal ausgeführt,
- *at most once* - der Auftrag wird höchstens einmal ausgeführt und
- *exactly once* - der Auftrag wird genau einmal ausgeführt.

Die folgende Abbildung zeigt Zustandsdiagramme für Client und Server bei einer Implementierung der *at least once* Semantik:



Skizzieren Sie analoge Zustandsdiagramme für Implementierungen der *at most once* und *exactly once* Semantiken, wobei wie oben eintreffende Nachrichten und Timeout-Ereignisse verwendet werden sollen. Verwenden Sie Sequenznummern zur Erkennung einzelner Anfragen für die *exactly once* Semantik.

b) Überlegen Sie für folgende Anwendungen, ob *at least once* oder *at most once* Semantik angebrachter ist:

- das Drücken eines Aufzugknopfs,

- das Übersetzen eines Programms,
- Daten in einer Datei schreiben/anhängen,
- das Bestellen einer Pizza,
- einen Kontoauszug holen,
- eine elektronische Überweisung tätigen,
- eine Stimme abgeben in einem elektronischen Wahlservice.

Aufgabe 4: Request/Reply-Protokoll

Für ein typisches Client-Server-Protokoll im Request/Reply-Stil bietet es sich an, ein auf Datagrammen basierendes Protokoll zu verwenden. Warum? Entwerfen Sie ein solches Protokoll.

Beachten Sie insbesondere nach Möglichkeit alle denkbaren Fehlerszenarien und wie diese behandelt werden sollten. Dabei sollte das Protokoll möglichst leichtgewichtig sein. Die Datagrammschicht bietet als Dienst beispielsweise folgende Operationen an:

- *send(in address, in data)* - asynchrones, ungesichertes Versenden der Daten,
- *recv(out address, out data, in timeout)* - blockierendes Empfangen eines Pakets mit Timeout.

Implementiert werden sollen Operationen der Art:

- *doOperation(in address, in request, out reply)* - synchrones Versenden eines Requests an den Server,
- *getRequest(out address, out request)* - blockierendes Empfangen eines Requests (Server),
- *sendReply(in address, in reply)* - Versenden der Antwort (Server)

Sie dürfen davon ausgehen, dass eine Fragmentierung der zu übermittelnden Nachrichten nicht notwendig ist. Weiterhin wird garantiert, dass keine Datenpakete verfälscht werden. Der eventuelle Verlust sowie Nichteinhaltung der Paketreihenfolge müssen hingegen berücksichtigt werden.

Aufgabe 5: Middleware

Ein zuverlässiger Multicast-Dienst erlaubt es einem Sender, zuverlässige Nachrichten an mehrere Empfänger zu übergeben. Gehört ein solcher Dienst zu einer Middleware-Schicht oder sollte er Teil einer darunter liegenden Schicht sein?

Aufgabe 6: Transparenz des RPC

Bei Verwendung von *Remote Procedure Call* (RPC) ist es prinzipiell egal, ob sich die aufgerufene Prozedur (d.h. der Server-Prozeß) auf dem lokalen Rechnerknoten oder einem anderen Knoten befindet. Was aber passiert, wenn die Prozedur einen Systemaufruf ausführt? Welche Probleme könnte dies nach sich ziehen und wie könnte man ihnen begegnen?