Übungsaufgabe #6: Last-of-Many Semantik

04.06.2003

In dieser Aufgabe sollen nun das RPC-System (falls noch nicht geschehen) um eine Multithreadfähigkeit des Servers erweitert werden. Außerdem soll eine RPC-Semantik implementiert werden, welche dem Ziel "exactly-once" möglichst nahe kommt.

Multithreaded Server
Das RPC-System soll nun erweitert werden, sodass mehrere Anfragen gleichzeitig bearbeitet werden können. D.h. trifft während der Bearbeitung einer Anfrage eine weitere Anfrage ein, so wird diese nicht verworfen oder zurückgestellt sondern gleichzeitig bearbeitet. Am einfachsten erreicht man dieses Verhalten, indem jede Anfrage von einem eigenen Thread bearbeitet wird.

Last-of-Many Semantik
Bisher wurde davon ausgegangen, dass die Kommunkation fehlerfrei und zuverlässig ist. Nun soll das RPC-System mit möglichen Fehlern (z.B. Verlust von Nachrichten) umgehen können um eine Last-of-Many Semantik zu implementieren.
Wie in der Übung vorgestellt soll dazu jeder Fernaufruf mit einer eindeutigen ID versehen werden. Kommt nach dem Absenden einer Anfrage innerhalb eines vordefinierten Zeitintervalls keine Antwort, so soll die Anfrage nochmals versendet werden. Um verschieden Anforderungsnachrichten des selben Fernaufrufes unterscheiden zu können erhält jede Nachricht zusätzlich eine Sequenznummer welche bei wiederholtem versenden erhöht wird. Nach der Last-of-Many Semantik soll das Ergebnis der letzten Ausführung verwendet werden. Ankommende Antwortnachrichten müssen demnach überprüft werden, ob sie zur letzten abgesendeten Anforderung passen. Auf Serverseite müssen veraltete Anforderungen verworfen werden.

Antworten Puffern (Annäherung an Exactly-Once)
Um der Exactly-Once Semantik näher zu kommen soll der Server nun jede Anfrage möglichst nur einmal bearbeiten. Dazu muss er anhand der Anfrage-ID und Sequenznummer folgende Fälle unterscheiden:

- *neue Anfrage*: eine neue Anfrage trägt eine unbekannte ID und führt zur Ausführung der entsprechenden Funktion
- *alte, noch in Bearbeitung befindliche Anfrage*: Sobald Bearbeitung fertig, Antwort mit neuester Sequenznummer zurücksenden
- *alte Anfrage, grössere Sequenznummer*: Erhält der Server eine Anfrage doppelt, z.B. aufgrund einer Timeoutüberschreitung beim Client, so kann er dies anhand einer bereits bekannten Anfrage-ID feststellen. Um eine erneute Ausführung zu vermeiden, soll das Ergebnis der vorherigen Ausführung zurückgegeben werden. Dazu müssen alle Ergebnisse auf Serverseite gespeichert werden.

Fehlerhaftes Kommunikationssystem
Testen Sie Ihr RPC-System indem Sie das Kommunikationssystem sabotieren. Folgende Fehlerquellen könnten eingebaut werden:

- zufälliger Verlust von Nachrichten
- extreme Verzögerung einzelner Nachrichten
- zufällige Verdopplung von Nachrichten

Abgabe: bis 18.06.2003 12:00 Uhr
Abgabe mittels /proj/i4vs/pub/abgabe (Abgabe in 2er oder 3er Gruppen möglich)