

TK1 – Theorieübung 2

Aufgabe 1.

Die Nachrichten benötigen eine eindeutige Anfrage-Identifizierung um Duplikate aufzuspüren und fallen zu lassen. Der Client implementiert das Senden seiner Anfrage gekoppelt an einen Timeout nach dessen Ablauf er die Anfrage ein weiteres Mal sendet, sofern noch keine Antwort eingetroffen ist. Der Server führt die Berechnungen der Anfrage nur aus, falls er über die ID der Anfrage feststellt, dass er die Anfrage nicht bereits erhalten hat. Berechnungen müssen darüber hinaus gespeichert werden, bis die Übertragung an den Client abgeschlossen ist, deshalb wird RRA (request-reply-ack) verwendet und das Ergebnis erst aus dem Speicher gelöscht, wenn das ACK vom Client empfangen wird.

Aufgabe 2.

a)

Marshalling beschreibt die Konvertierung von Daten oder Datenstrukturen einer Sprache in ein Format, mit dem es möglich bzw. vereinfacht wird, Informationen zwischen zwei Maschinen zu senden. Dabei kann das Marshalling von Daten dafür sorgen, dass die Heterogenität von Netzen erhöht wird, da die konvertierten Daten über spezielle De-/Marshaller in verschiedene Sprachen importiert werden können. Darüber hinaus kann auch die Datenübertragung davon profitieren, da zum Beispiel Google's Protobuf die Daten komprimiert.

b)

Die Information (CORBA, XML/SOAP) über die Typen ist auf beiden Systemen (Empfänger und Sender) bekannt, deshalb werden die Typen nicht benötigt. Es reicht die Primitiven zu senden, da die Informationen wie die gesendeten Daten interpretiert und eingelesen werden, auf beiden Seiten vorhanden ist. Bei RMI wird mit jeder Nachricht die gesamte Klassendefinition mitgesendet. Hierbei sieht man die Vor- und Nachteile. CORBA und XML/SOAP benötigen keine Strukturinformationen über die Daten in ihren Nachrichten, was zu wesentlich kürzeren Nachrichten führt, während bei RMI ein größer Overhead an Daten für die Strukturbeschreibung benötigt wird. Der Nachteil ist jedoch, dass die Strukturbeschreibungen bei CORBA und XML/SOAP identisch sein müssen, da die Daten sonst nicht mehr interpretiert werden können.

Aufgabe 3.

a)

Der Vorteil von RRA ist, dass der Server so in der Lage ist Berechnungen für einzelne Anfragen nur ein einziges Mal zu berechnen, da er das Ergebnis solange speichert, bis er ein ACK vom Client empfängt. Zusammen mit eindeutigen ID's für Anfragen ist der Server somit in der Lage redundante Anfragen zu filtern, was die Last deutlich reduzieren kann. Der Nachteil ist, dass Clientabstürze bei RRA ein nicht zu unterschätzendes Problem werden, denn während bei RR pro Anfrage eine Berechnung ausgeführt und das Ergebnis nach dem Fire & Forget Prinzip gesendet wird, kann es passieren, dass ein Server kein ACK auf seine Ergebnismnachricht erhält. Wird nun der Client nach einem Absturz neu gestartet, erhält er womöglich die Ergebnismnachrichten vom Server, zu denen ihm nun der Kontext fehlt um sie zu verstehen, da sein Zustand zurückgesetzt wurde. Also müssen Routinen eingeführt werden, die bestimmen, wie bei Clientabstürzen vorgegangen wird.

b)

Ja, er sollte die Nachricht öfter herausenden, um eine ACK Nachricht zu erhalten. Jedoch sollte die Anzahl an Versuchen begrenzt sein, da Clientabstürze der Grund für die fehlende Antwort sein können. Werden dann x Versuche durchgeführt das Ergebnis an den Client zu senden und in dieser Zeit kommen von diesem Client keine neuen Anfragen, so kann das Ergebnis der Berechnung fallen gelassen werden. Deshalb sollte ein Client dem Server auch melden, wenn er mit dem Ergebnis einer Berechnung nichts anfangen kann.