

WS 25/26 Verteilte Systeme

Praktikumsaufgabe 4

Als Teil dieser Aufgabe soll der Datastore aus Aufgabe 1 um Replikation erweitert werden. Die Kommunikation von Client und Server soll weiterhin über Remote Procedure Calls (RPC) gekapselt werden. Die Aufgabe soll in Teams mit zwei Personen bearbeitet werden. Bei Praktikumsgruppen mit ungerader Personenzahl gibt es ein dreier Team.

Das Datastore Interface bleibt zu Aufgabe 2 identisch mit folgenden Methoden:

void write(int index, String data)

String read(int index) throws NoSuchElementException

Es sollen beliebig viele Datastore Server gestartet werden können. Eine Liste aller verfügbaren Server wird dem Client beim Start übergeben.

Aufrufe von „write“ sollen vom ClientStub auf allen Servern ausgeführt werden. Die Methode terminiert sobald alle verfügbaren Server den Vorgang bestätigt haben. Ein Aufruf der „read“-Methode soll auf einem der Server nach dem Round Robin Methode ausgeführt werden.

- Die verwendete Programmiersprache kann frei gewählt werden. Eine Möglichkeit wäre die Implementierung in Java. Die Methodensignaturen müssen entsprechend an die Programmiersprache angepasst werden.
 - Fehler bei der Übertragung zwischen Client und Server müssen als Exceptions (je nach Programmiersprache) gekapselt werden.
 - Die Stubs sollen nicht automatisch generiert werden, sondern selbst implementiert werden. Das heißt diese müssten bei jeder Änderung des Interfaces angepasst werden.
 - Die Kommunikation soll über das Netzwerk laufen. Es muss möglich sein Client und Server Anwendung auf verschiedenen Rechnern auszuführen.
 - Die Kommunikation soll über ein standardisiertes menschenlesbares Nachrichtenformat, wie zum Beispiel JSON oder XML, laufen.
 - Für die Implementierung dürfen keine externen Bibliotheken verwendet werden. Insb. sind keine RPC-Bibliotheken oder fertige Protokolle erlaubt.
Lediglich für das Marshalling und Unmarshalling dürfen Bibliotheken verwendet werden.
- 1.) Implementiert die Datastore Server und den ClientStub in der Programmiersprache eurer Wahl
 - 2.) Messt die Dauer für Schreibvorgänge abhängig von der Anzahl der Replicas mit 1 bis 4 Servern.
 - 3.) Bei Ausfall eines beliebigen Servers soll der Client möglichst wenig mitbekommen. Testet den Fehlerfall. Welche Auswirkungen hat der Ausfall eines Servers auf die Dauer von Lese- und Schreibzugriffen?
 - 4.) Passt die Beschreibung der Verteilungstransparenz aus Aufgabe 1 an die erweiterte Implementierung an.

Abzugeben ist ein Bericht in Form eines Text-Dokuments mit allen relevanten Design-Entscheidungen (z.B. Wahl der Programmiersprache) und Ergebnissen der Aufgaben. Der Source-Code muss während der Abgabe erklärt werden können. Die Implementierung muss auf den Pool Rechnern, ggf. über das Netzwerk, lauffähig sein.

Die Aufgabe sollte während des vierten Praktikumstermins bearbeitet und zum Ende des Termins präsentiert werden. Spätester Abgabepunkt ist der 25.01.26 per E-Mail.