

Verteilte Systeme

Hausarbeit

Abgabe WS-22-II: 30.01.2022 9.00 Uhr

Abgabe SoSe-23-I: 27.03.2023 9.00 Uhr

Abgabe via Stud.IP-Upload, E-Mail, DVD / USB-Stick im Fach A-Gebäude (1.Stock)

Enthält: Quellcode (compilierbar), Doku als PDF

Doku

Umfang 2 bis max. 5 Seiten (ohne Abbildungen / SG ~10)

Konzept, Aufbau der Software

Eine halbe Seite je Funktionalität (s.u.)

Software

Ziel Ihres Verteilten Systems ist es, Probleme mittels verteiltem Rechnen zu lösen. Schreiben Sie dazu zwei Programme. Ein Programm als Knoten Ihres verteilten Systems, welches auf mehreren Rechnern ausgeführt Ihr verteiltes System bildet und eine Clientsoftware die sich mit dem System verbindet um das System zu bedienen.

Die Clientsoftware kann eine GUI- oder CLI-Anwendung sein. Sie soll möglichst wenig Logik enthalten. Der Client soll sich mit einem beliebigen Knoten verbinden können. Der Client soll außerdem Statusinformationen zum aktuellen Zustand des Systems anzeigen können. Es sollen sich zeitgleich mehrere Clients mit dem System verbinden/arbeiten können. Gehen Sie davon aus, dass sich normalerweise mehrere Clients mit dem System verbunden haben und damit arbeiten.

Die Software, die als Knoten Ihres verteilten Systems agiert, soll auf mehreren System eingesetzt werden. Gehen Sie davon aus, dass es sich um Mehrkernsysteme handelt. Sie können dabei von min. 3 max. 1000 ausgehen. Achten Sie dabei möglichst darauf, dass Ihr System robust und effizient arbeitet. Den Aspekt der Sicherheit können Sie, bis auf u.g. Ausnahme, bei der Implementation vernachlässigen.

Verwenden Sie für Client und Knoten ein geeignetes Logging.

Wenn Ihr verteiltes System läuft, soll es durch den Client um Module/Plugins zur Laufzeit ohne Neustart erweiterbar sein. Diese Module sollen die Funktionalität für konkrete Probleme beinhalten. Der Client soll also Module in das System hochladen können, damit dieses dann eine Aufgabe mit Hilfe der Module lösen kann. Bei diesem PlugIn-System sollen Sie auch den Aspekt der Sicherheit berücksichtigen.

Schreiben Sie zwei Module um zwei unterschiedliche Probleme mittels verteiltem Rechnen zu lösen. Sie können diese frei wählen. Im Folgenden finden Sie Beispiele aus denen Sie wählen können. Sie können den Funktionsumfang beliebig erweitern.

1. – WebCrawler

Das System bekommt eine Url / Liste von Urls, lädt diese und alle verlinkten Dateien (bis zu einer angegebenen Tiefe) herunter und speichert diese. Zu jeder Datei werden Hashwerte gebildet. Später kann der Client anhand eines Hashwertes oder Url eine Datei aus dem System downloaden.

2. – Hashwert-Kollision

Das System bekommt Daten (z.B. eine Datei). Es ermittelt den Hashwert. Nun soll es Daten ermitteln, die von den Ursprungsdaten abweichen, aber den gleichen Hashwert liefern.

3. – GitHubAnalyzer

Das System bekommt eine Liste von Github-Projekten. Es bildet zu jedem Projekt einen Graphen der Commits ab. Dabei werden zu jedem Commit weitere (Meta-) Daten erfasst; Z.B. Author, Message, AST, ...

4. – Berechnung von Zahlen

Primfaktorzerlegung, Pierpont-Primzahl, ... Hier gibt es im Netz viele Probleme mit Ansätzen zur Lösung