Entwurf und Implementierung eines Frameworks für plattform- und geräteübergreifendes verteiltes Rechnen

Finn Teegen
Christian-Albrechts-Universität zu Kiel
fte@informatik.uni-kiel.de

Einleitung

Im gleichen Maße wie die Verbreitung von Geräten wie Smartphones oder Tablets in den letzten Jahren gestiegen ist, hat gleichzeitig die Rechenleistung selbiger seit Jahren stetig zugenommen. Ziel der Arbeit war es, eine Möglichkeit zu schaffen, diese Rechenleistung zur Berechnung komplexer Probleme heranziehen zu können. Dabei sollte das geplante Framework sowohl einfach und intuitiv zu verwenden als auch auf einer möglichst großen Zahl von Plattformen und Geräten lauffähig sein. Um diesem Anspruch gerecht zu werden, wurden die ausdrückbaren Probleme für den Anfang auf solche eingeschränkt, die sich in unabhängig lösbare Teilprobleme splitten lassen.

Eingesetzte Technologien

Das Framework basiert auf Java, nicht zuletzt, weil Java für eine Vielzahl von Plattformen verfügbar ist. Dies schließt insbesondere auch Android ein, welches wiederum auf vielen Smartphones oder Tablets installiert ist. Zur Kommunikation zwischen dem Server und den Clients wird eine Bibliothek namens SIMON[1] verwendet. SIMON stellt eine Alternative zu der Java-eigenen Remote Method Invocation (RMI) dar, welche unter Android nicht verfügbar ist. Darüber hinaus bietet SIMON auch weitere, nützliche Vorteile, zum Beispiel werden sämtliche Aufrufe über eine einzelne TCP-Verbindung abgehandelt, welche auch für Callbacks genutzt werden kann. Als Entwicklungsumgebung wurde Eclipse zusammen mit dem Android SDK eingesetzt.

Prozess

Da das Projekt eine Einzelarbeit ist, fand keine Aufgabenverteilung oder ähnliches statt. Die Entwicklung begann nach der Ideenfindung zunächst mit der Erstellung von Prototy-

[1] Simple Invocation of Methods Over Network, http://dev.root1.de/projects/simon

pen, um auszutesten, inwieweit sich das Gedachte umsetzen ließe. Anschließend wurden die Interfaces spezifiziert, auf deren Basis der Benutzer später eigene Probleme beschreiben und programmieren können sollte. Schließlich entstanden parallel das eigentliche Serverprogramm, ein konsolenbasiertes Steuerungsprogramm für diesen, ein ebenfalls konsolenbasierter PC-Client und ein Android-Client. Zum Schluss wurde noch eine grafisches Kontrollprogramm für den Server ergänzt, welches eine - im Vergleich zur konsolenbasierten Alternative - bequemere Steuerung ermöglicht.

Erfahrungen

Im Wesentlichen hat sich bestätigt, dass es sinnvoll und nützlich ist, zu Beginn der Entwicklung Prototypen zu implementieren, zu testen und zu perfektionieren. Nicht nur, dass mit ihrer Hilfe erprobt werden kann, inwieweit sich bestimmte Ansätze umsetzen lassen, sie lassen sich darüber hinaus auch schnell und einfach in das Projekt überführen und integrieren. Durch die vorangegangenen Tests entfielen die einzelnen Modultests oder vereinfachten sich zumindest, sodass die Integrationstests schneller stattfinden konnten.

Rahmendaten

Die Idee kam im April 2013 im Rahmen der Themenfindung für eine Bachelorarbeit auf, wurde aber zugunsten eines anderen Themas zunächst verworfen. Dennoch begann im Mai 2013 aufgrund des ungebrochenen Interesses bereits die Implementierung erster, kleinerer Prototypen. Aufgrund der eigentlichen Bachelorarbeit kam es dann zu einer längeren Pause, bis das Projekt im Oktober 2013 wieder aufgegriffen wurde. Bis Ende November 2013 entstand neben dem begonnenen Masterstudium schließlich die aktuelle Umsetzung, welche eine erste, vollständige Realisierung der ursprünglichen Planung darstellt. Auch wenn das eigentliche Ziel somit bereits erreicht ist, ist die Entwicklung dennoch nicht abgeschlossen. Neben weiteren Optimierungen könnten zukünftige Arbeiten an dem Projekt die Implementierung weiterer Funktionen beinhalten, beispielsweise die Möglichkeit, dass Clients auch mit anderen Clients kommunizieren können. Auf diese Art und Weise könnten auch Probleme berechnet werden, die sich nicht in unabhängig lösbare Teilprobleme aufteilen lassen.

Präsentationsformat

Eine eventuelle Präsentation würde zunächst mit Folien erfolgen, mithilfe derer der Aufbau des Frameworks und dessen Verwendung erläutert werden würden. Dabei würde auch auf einige Designentscheidungen sowie technische Hintergründe eingegangen werden. Auch zukünftige Arbeiten könnten Erwähnung finden. Anschließend wäre eine kleine Demonstration vorgesehen. Dabei würde das Serverprogramm, welches die Arbeit auf die angemeldeten Clients verteilt, ein Client für Desktop-Rechner sowie innerhalb eines Emulators ein Client für Android-Smartphones auf dem Präsentationsrechner gestarten werden. Abhängig von der Verfügbarkeit einer drahtlosen Netzwerkverbindung könnte

zusätzlich das eigene Smartphone verwendet werden. Als Beispielproblem würde dann das Knacken eines einfachen Passworts mithilfe einer Brute-Force-Attacke, die auf alle angemeldeten Clients aufgeteilt wird, gezeigt werden.

Weitere Angaben

Da bisher keine Webseite zu dem Projekt existiert, sind an dieser Stelle einige inhaltliche Aspekte aufgeführt, um einen kurzen Eindruck über die Arbeit zu vermitteln. Zentrale Komponente ist der Server. Er hat die Aufgabe, alle vorhandenen Probleme auf angemeldete Clients zu verteilen, deren Ergebnisse zu verwalten und am Ende zu einem Gesamtergebnis zu verschmelzen. Um den Server zu kontrollieren, gibt es Steuerungsprogramme, wahlweise grafisch oder konsolenbasiert. Durch sie können neue Probleme geladen, angemeldete Clients angezeigt und Ergebnisse eingesehen werden. So wie sich mehrere Clients mit dem Server gleichzeitig verbinden können, ist dies auch bei den Steuerungsprogrammen möglich. Auf diese Art und Weise können mehrere Nutzer die Ergebnisse von verschiedenen Orten aus einsehen oder neue Probleme hinzufügen. Um eigene Probleme zu definieren, müssen nur zwei simpel gehaltene Interfaces implementiert werden, wahlweise in einer einzigen Klasse oder in zwei verschiedenen. Das eine Interface spezifiziert, wie das Problem in seine Teilprobleme gesplittet werden kann, während das andere angibt, wie ein solches Teilproblem gelöst werden kann. Um die Probleme geräteunabhängig verteilen zu können, werden sie zu Bytecode kompiliert, der dann zu den einzelnen Clients geschickt wird. Auf Clientseite wird dieser Bytecode dann mithilfe eines Class-Loaders in den Speicher geladen und per Reflection werden die im Interface definierten Methoden aufgerufen, um das Problem zu berechnen.

Sprache

Zu guter Letzt sei erwähnt, dass eine Ausarbeitung sowohl auf Englisch als auch auf Deutsch verfasst werden kann. Dies gilt auch für die Präsentationsfolien.