

Belegarbeit

im Wahlpflichtfach Verteilte Systeme

Berechnung der Mandelbrotmenge
mittels Bootstrap und Java-RMI

Vitali Tichonow, Daniil Podlubniy

23. Juni 2023

Inhaltsverzeichnis

Kurzbeschreibung der Belegarbeit.....	3
Architektur des Beleges.....	3
Beschreibung der Programme.....	4
Bootstrap.....	4
MandelbrotService.....	4
MandelbrotServer.....	4
Client.....	5
Bedienung.....	6

Kurzbeschreibung der Belegarbeit

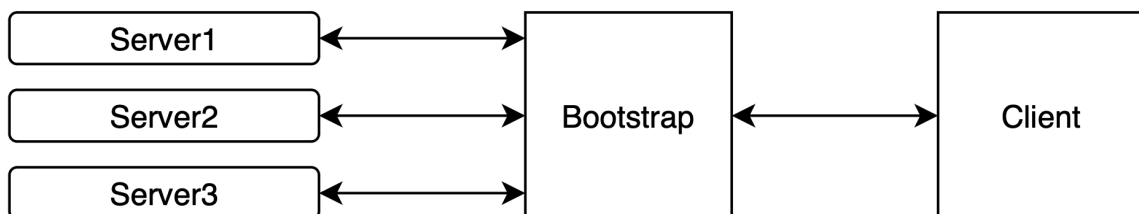
Im Rahmen des Wahlpflichtmoduls Verteilte Systeme soll ein Beleg angefertigt werden. Dabei soll ein Java-Programm entwickelt werden, das in der Lage ist, eine Mandelbrotmenge auf verteilten Systemen zu berechnen. Dabei sollen folgende Aspekte umgesetzt werden:

- schrittweiser Zoom zu einem ausgewählten Bildpunkt
- Nutzung von Java-RMI oder einer anderen geeigneten Technologie zur Verteilung der Aufgaben
- Architektur Model-View-Presenter für den Client "
- Die Erstellung eines Bootstrapservers
- Die Implementierung von Clients, welcher durch Verteilung von Aufgaben an Workern die entsprechenden Zoomfunktion bearbeiten
- Nutzung von Threads, um die Berechnung der Mandelbrotbilder zu beschleunigen
- Verteiltes Berechnen der Mandelbrotbilder auf mehreren Servern (Worker)

Architektur des Beleges

Das Programm wird nach Diskussion innerhalb der Gruppe mit folgenden Merkmalen realisiert:

- Transfer über Java-RMI, das auf TCP basiert.
- Um die RGB Pixelfarben zu übertragen, wird ein zweidimensionales Int-Array verwendet
- Für die Übertragung der RGB-Pixelfarben wird ein zweidimensionales Array verwendet
- Jeder Server (Worker) führt die Berechnung der Farbwerte unter Erzeugung benutzerdefinierter Anzahl an Threads
- Jeder Server berechnet allein das gesamte Bild einer Iteration



Beschreibung der Programme

Bootstrap

Der Bootstrapservers besteht aus 3 Dateien:

- Bootstrap.java
- BootstrapImpl.java
- BootstrapRegistration.java

Nach dem BootstrapRegistration.java gestartet wurde, wurde ein Server namens "Bootstrap" erstellt. Er ermöglicht, dass andere Server, die sogenannten Worker, sich auf dem Server registrieren und Aufgaben annehmen.

MandelbrotService

MandelbrotService ist ein Interface, das dem Server und dem Client ermöglicht, miteinander zu kommunizieren und Informationen auszutauschen. Dadurch wird zum Beispiel das berechnete zweidimensionale Array für die Darstellung der Mandelbrotfraktale an den Client geschickt.

MandelbrotServer

Der Server implementiert folgende Klassen:

- MandelbrotServer-Klasse: Diese Klasse erbt von UnicastRemoteObject und implementiert MandelbrotService, was bedeutet, dass sie Methoden für RMI bereitstellt. Es gibt einige wichtige Felder in dieser Klasse, wie xpix und ypix, die die Dimensionen des generierten Bildes darstellen, bild, das das generierte Bild repräsentiert, und workerCount, das die Anzahl der Worker angibt.
- Konstruktor von MandelbrotServer: Der Konstruktor initialisiert xpix, ypix und das bild-Feld.
- main-Methode: In der main-Methode wird ein MandelbrotServer erstellt und mit dem Bootstrap-Service registriert. Die Bootstrap-Service-Adresse und der Worker-Name werden als Programmargumente bereitgestellt.
- apfel_bild Methode: Diese Methode berechnet das Mandelbrot-Set-Bild und gibt es zurück. Es verwendet Multithreading, um das Bild zu berechnen, wobei jeder Thread einen Teil des Bildes berechnet.
- ApfelWorker-Klasse: Dies ist eine innere Klasse, die Runnable implementiert und die Arbeit jedes Threads durchführt. Die run-Methode berechnet den Teil

des Bildes, der dem Thread zugewiesen ist. Es gibt auch Hilfsmethoden wie `calc`, die das Iterieren der Mandelbrot-Set-Formel durchführt, und `farbwert`, die den Farbwert für jede Iteration berechnet.

Client

- `main`: Hier verbindet sich der Client mit dem RMI-Server und erhält eine Liste aller verfügbaren Worker (Dienste). Dann sendet er jedem Worker die Anzahl der Worker und seine spezifische ID.
- `MandelbrotClient.run`-Methode: Hier wird das erste Bild der Mandelbrot-Menge generiert und auf dem Bildschirm angezeigt.
- `ApfelPresenter`-Klasse: Diese Klasse steuert den Prozess der Generierung von Mandelbrot-Bildern. Sie erzeugt Threads für jeden Worker und lässt diese das Mandelbrot-Bild berechnen. Die Ergebnisse werden dann gesammelt und auf dem Bildschirm angezeigt.
- `ApfelView`-Klasse: Diese Klasse stellt die Benutzeroberfläche dar. Sie hat ein Textfeld, in dem der Benutzer die Anzahl der Iterationen eingeben kann, die für die Generierung des Mandelbrot-Bildes verwendet werden sollen, und einen Button, um den Prozess zu starten. Es gibt auch einen "Stopp"-Button, um den Iterationsprozess zu stoppen, und einen "Reset"-Button, um das Bild zurückzusetzen.

•

Klasse `ApfelPresenter`:

- `MandelbrotService service`: Dies ist ein Remote-Service, der die Mandelbrot-Berechnung durchführt.
- `ApfelView v`: Dies ist die Ansicht, die die berechneten Bilder anzeigt.
- `setServiceAndView(MandelbrotService service, ApfelView v)`: diese Methode ist für Service und Ansicht verantwortlich. Dadurch können Berechnungen durchgeführt und Bilder angezeigt werden.
- `erstesImage()`: Diese Methode berechnet und zeigt das erste Bild der Mandelbrotmenge an.
- `apfel(int anzahlIteration)`: Diese Methode verwendet mehrere Threads und verteilt die Mandelbrot-Berechnungen auf mehrere Remote-Services (worker). Jeder Thread ruft die `apfel_bild`-Methode eines Workers auf, um ein Mandelbrot-Bild zu berechnen, und speichert das Ergebnis in einer Liste. Die Methode synchronisiert die Zugriffe auf diese Liste, um Race Conditions zu vermeiden. Sie stellt auch sicher, dass die Bilder in der richtigen Reihenfolge angezeigt werden, unabhängig davon, in welcher Reihenfolge die Berechnungen abgeschlossen wurden.
- `displayResults(List<Color[][]> results, AtomicInteger nextToDisplay)`: Diese Methode ist dafür da, um die berechneten Bilder in der richtigen Reihenfolge anzuzeigen. Sie ruft die `update`-Methode der Ansicht für jedes berechnete Bild auf und löscht das bereit angezeigte Bild aus der Liste.

Bedienung

- Es besteht die Möglichkeit, die Iterationsanzahl direkt im Panel zu ändern
- Der Mittelpunkt des Bildes kann entweder durch das Klicken auf den gewünschten Punkt oder durch manuelle eingabe in die dafür vorgesehene Felder erfolgen.
- Durch das Klicken auf Start, werden die Mandelbrotbilder berechnet und angezeigt.
- Durch das Klicken auf Stopp, wird die Berechnung unterbrochen
- Durch das Klicken auf Reset, wird wieder das erste Bild angezeigt