Technologisches Gewerbe Museum

DezSys

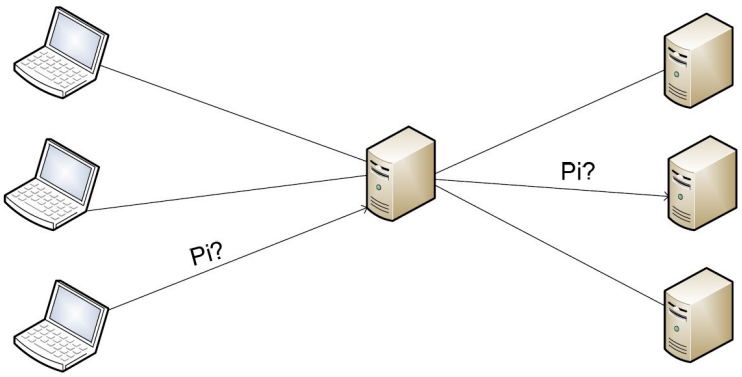
PI Calculator

Author: Pöcher René & Steinkellner Sebastian

January 8, 2015

Contents

1 Aufgabe



Als Dienst soll hier die beliebig genaue Bestimmung von pi betrachtet werden. Der Dienst stellt folgendes Interface bereit:

```
    // Calculator.java
public interface Calculator {
public BigDecimal pi (int anzahl_nachkommastellen);
}
```

Ihre Aufgabe ist es nun, zunächst mittels Java-RMI die direkte Kommunikation zwischen Klient und Dienst zu ermöglichen und in einem zweiten Schritt den Balancierer zu implementieren und zwischen Klient(en) und Dienst(e) zu schalten. Gehen Sie dazu folgendermassen vor:

Entwicklen Sie ein Serverprogramm, das eine CalculatorImpl-Instanz erzeugt und beim RMI-Namensdienst registriert. Entwicklen Sie ein Klientenprogramm, das eine Referenz auf das Calculator-Objekt beim Namensdienst erfragt und damit pi bestimmt. Testen Sie die neu entwickelten Komponenten.

Implementieren Sie nun den Balancierer, indem Sie eine Klasse CalculatorBalancer von Calculator ableiten und die Methode pi() entsprechend implementieren. Dadurch verhält sich der Balancierer aus Sicht der Klienten genauso wie der Server, d.h. das Klientenprogramm muss nicht verändert werden. Entwickeln Sie ein Balanciererprogramm, das eine CalculatorBalancer-Instanz erzeugt und unter dem vom Klienten erwarteten Namen beim Namensdienst registriert. Hier ein paar Details und Hinweise:

Da mehrere Serverprogramme gleichzeitig gestartet werden, sollten Sie das Serverprogramm so erweitern, dass man beim Start auf der Kommandozeile den Namen angeben kann, unter dem das CalculatorImpl-Objekt beim Load-Balancer gespeichert wird. Damit soll der Server nun seine exportierte Instanz an den Balancer übergeben, ohne es in die Registry zu schreiben. Verwenden Sie dabei ein eigenes Interface des Balancers, welches in die Registry gebinded wird,

um den Servern das Anmelden zu ermöglichen.

Das Balancierer-Programm sollte nun den Namensdienst in festgelegten Abständen abfragen um herauszufinden, ob neue Server Implementierungen zur Verfügung stehen. Java-RMI verwendet intern mehrere Threads, um gleichzeitig eintreffende Methodenaufrufe parallel abarbeiten zu können. Das ist einerseits von Vorteil, da der Balancierer dadurch mehrere eintreffende Aufrufe parallel bearbeiten kann, andererseits müssen dadurch im Balancierer änderbare Objekte durch Verwendung von synchronized vor dem gleichzeitigen Zugriff in mehreren Threads geschützt werden.

Beachten Sie, dass nach dem Starten eines Servers eine gewisse Zeit vergeht, bis der Server das CalculatorImpl-Objekt erzeugt und beim Namensdienst registriert hat sich beim Balancer meldet. D.h. Sie müssen im Balancierer zwischen Start eines Servers und Abfragen des Namensdienstes einige Sekunden warten.

2 Working time

2.1 Estimated Working time

Estimated working time

Task	**Person**	**Time in hours**
Planung	Pöcher Steinkellner	1
UML Design	Pöcher	1/2
PI Methode desig. & Implementieren	Pöcher	2
Total	Pöcher Steinkellner	2 1/2 1
Total Team		**3 1/2 hours**

Table 1: Estimated working time

3 Umsetzung

3.1 PI Methode

Zu PI Methode wird die Bailey-Borwein-Plouffe-Formel verwendet[1]

Als Datentyp wird BigDecimal verwendet. Bei der Umsetzung aufpassen:
Um mit BigDecimal zu rechnen müssen die Methoden verwendet werden, um mit diesen jedoch zu rechnen müssen BigDecimal Datentypen verwendet werden, die ganz einfach mit new BigDecimal("int") erstellt sind.

```
public BigDecimal pi(int anzahl_nachkommastellen) {

    BigDecimal sum2=new BigDecimal(0);
    int komma= 0;
    for (int i = 0; komma < anzahl_nachkommastellen; i++) {
        komma=sum2.toString().length();
        BigDecimal j = new BigDecimal(i);
        BigDecimal x = new BigDecimal(0);
        x=(new BigDecimal(1).divide(new BigDecimal(16).pow(i),anzahl
        sum2=x.add(sum2).setScale(anzahl_nachkommastellen,BigDecimal
    }
```

4 Easy Bibliography

List of Tables

List of Figures

References

[1] **Wikipedia,Bailey-Borwein-Plouffe**
http://de.wikipedia.org/wiki/Bailey-Borwein-Plouffe-Formel
last used: 5.01.1995, 16:00