

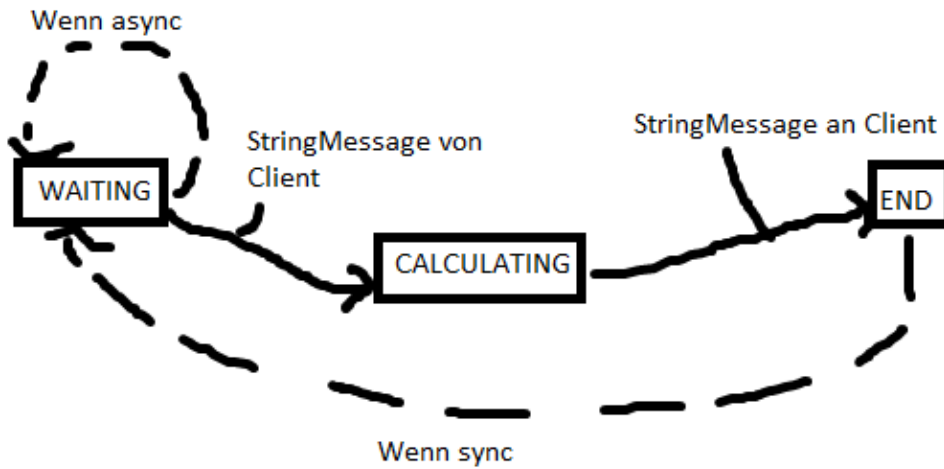
Protokoll:

Client schickt StringMessage an Server

Server schickt StringMessage zurück

Zustandsdiagramm:

Server



Client



Test:

```
Please choose an action [ time | random ]: time
Mon Mar 06 00:01:40 CET 2017
You have been waiting for 8094 milliseconds
```

Ausführung:

Einfach über CLI mit `java -jar <jarname>` starten (zuerst Server, danach Client)
Der Client muss dann entweder "time" oder "random" eingeben, um die Funktion auszuwählen.

Antworten:

Offensichtlich ist die parallele Berechnungsmethode zwar nicht schneller, senkt aber Wartezeiten von anderen Clients, die nach dem ersten Client starten. Bei der sequentiellen Abarbeitung müssen diese erst auf die Beendigung der vorherigen warten, bei der parallelen Berechnung kommen alle sofort dran. Allerdings ist die maximale Anzahl an Threads begrenzt. Zu viele Threads behindern den Prozessor zu stark und könnten sogar zum Absturz des Programms führen.