

PE Taak: Performance onderzoek

Daan Bergmans

Sander Elsen

Arno Bruynseels

3AONC/D

2016-2017

# Wat is peformance analysis?

## Inleiding

Indien we opzoek gaan naar wat "performance analysis" betekent, waarvoor het gebruikt wordt en het nut ervan, stellen we vast dat dit concept gekend is en gebruikt wordt in talrijke domeinen:

- Software & Algoritmes - Human resource management

- Sport - Verkoop

- Economie - Marketing

- Verkeer - ...

In deze analyse gaan we ons echter focussen op het domein van software en algoritmes waaronder de programmeertaal java.

## Beschrijving

Performance analysis is eigenlijk een discipline waarbij systematische observaties worden gedaan om prestaties en besluitvorming te verbeteren. Hierbij wordt alle informatie hoofdzakelijk verstrekt door middel van objectieve statistische (Data Analysis) en visuele feedback (Video Analysis).

Het doel van performance analysis is dus registreren wat de huidige situatie is en hoe deze zo optimaal mogelijk kan verbetert worden.

Hiervoor onderschiedt het performance analysis proces 3 verschillende stappen:

### 1. Verzamelen van gegevens

Het verzamelen van gegevens is het proces waarbij data, over de prestaties van programma's, wordt verkregen uit een uitvoerend programma.

Om dit te kunnen realiseren worden er observatietools gebruikt die gegevens kunnen meten, registreren en visueel voorstellen.

Dit proces heeft een zeer uiteenlopende tijdsduur en is vrijwel afhankelijk van de software samen met welke doeleinden er geraliseerd moeten worden.

Doorgaans is dit een langdurig en intensief gebeuren waarbij de hoeveelheid data die verzamelt wordt zal verschillen per analsye.

### 2. Gegevens transformeren

Om data goed te kunnen bestuderen is het nodig dat de gegevens zodanig in een vorm worden omgezet die het mogelijk maken antwoorden te bieden. Nieuwe verzamelde gevegens voldoen zelden aan deze eisen en is het dus noodzakelijk om een transformatie te verwezenlijken.

Vaak wordt het volume van gegevens verminderd en gestructureerd tot een bestudeerbaar resultaat.

### 3. Gegevens visualiseren

Nadat de data een transformatie ondergaan heeft, zou het een enorme bijdrage zijn voor de analyse dat alle data visueel voorgesteld kan worden. Gegevens visualiseren kan vele vormen aannemen. Een van de meest voorkomende vormen zijn: grafieken, diagrammen en statistieken. Het is dus belangrijk dat er goed nagedacht wordt over welke visualisatietechnieken er gebruikt zullen worden in het analyse proces.

Visualisatie bevorderd dus het inzicht in de gegevens waardoor er sneller observaties kunnen worden gedaan en het nemen van beslissingen bekrachtigt.

## Waarom performance anylyse?

Het uitvoeren van een performence analyse brengt vele voordelen met zich mee. Zoals eerder vermeldt is het een grote aanwinst om een goed inzicht te krijgen in de performantie van de software waardoor toekomstige prestaties geoptimaliseerd kunnen worden en het nemen van beslissingen vergemakkelijk.

# Tools voor performance analysis

In het software & algoritme domein wordt er voornamenlijk gebruikt gemaakt van observatietools die de prestaties van lopende software kunnen gaan registeren, meten en visualiseren.

Bij het selecteren van welke tool je gaat gebruiken is het belandrijk om volgende aspecten te beschouwen:

* **Accuraatheid**: De tijdspanne waarin data wordt verzamelt. Als dit realtime moet gebeuren is het belangrijk dat de tool dit ondersteunt. Indien er enkel periodiek data moet geregistreerd worden kan men naar een tool opzoek gaan die hier beter voor geschikt is.
* **Flexibiliteit**: Hoe belangrijk is het voor de analyse dat de tool uitgebreid kan worden om bijkomende informatie te verzamelen? Anderzijds kan het ook van beland zijn dat de toepassing verschillende visualisatie technieken bevatten.
* **Abstractie:** Welke data ga je analyseren met welke graad van accuraatheid.

Voor java zijn vlogende observatie tools beschikbaar:

* NetBeans Profiler
* JProfiler
* GC Viewer
* VisualVM
* Java Performance Analysis Tool
* Jrockit
* Eclipse Memory Analyzer
* Java Interactive Profiler
* Profiler4J
* JConsole

Voor de uitwerking van onze analysa hebben we gekozen voor (TOOL) omdat deze tool aan al onze belangen voldoet met bovenstaande aspecten in beraad genomen.

(UITLEG OVER TOOL)

WEBSITE VAN ALLE TOOLS:

https://blog.idrsolutions.com/2014/06/java-performance-tuning-tools/

# Sudokusolver-project op Github

# Installatiegids

# Korte uitleg oplossingalgoritme

Solve Button

1. **Check of de sudoku vol is**

* *Uitleg***:** Controleer of elke cel een waarde > 0 bevat.
* *Ja* : Toon bericht dat de sudoku vol is, stop algoritme
* *Neen*: Ga naar punt 2.

1. **Check of de sudoku geldig is om op te lossen**

* *Uitleg***:** Check eerst in de 3x3 box of er geen 2 zelfde getallen in staan, daarna hetzelfde voor de rij en kolom.
* *Ja* : Toon bericht dat de sudoku niet geldig is, stop algoritme
* *Neen*: Maak de ingegeven getallen onbewerkbaar en maak ze grijs. Ga naar punt 3.

1. **Los de sudoku op (recursief = methode die zeichzelf aanroept)**

* *Uitleg***:**

1. *Check of elke cel is opgelost*

*Ja* : *einde algoritme*

*Neen*: Stap 2

1. *Check of de cell al een preset was of een waarde bevat*

*Ja* : Ga naar voglende cell

*Neen*: Stap 3

1. Maak een lijst aan met cijfers aan van 1 t.e.m. 9 en shud deze door elkaar. (schudden om een meer willekeurige uitkomst te bekomen).
2. Ga elk nummer in deze lijst af en test totdat nummer niet voorkomt en de 3x3box, rij en kollom. Check dat nummer
3. *Lukt* : Ga naar voglende cell en vul dit nummer in de cel in.

*Faalt*: Stap 5

1. Voer deze methode opnieuw uit met de volgende cel totdat elke cel ingevult is.

Verwijzing naar beschrijving van het algoritme of oplossingsmethode

(Wetenschappelijk paper of online artikel)

# Onderzoek

# Performantieverbeteringen (4)

# Verslag bevindingen (analyse, testen, cijfergegevens,…)

# Voorstellen verbeteringen aan auteur