

The slide features a light pink background. In the top left corner is the Lunds Universitet logo and the text "LUNDS UNIVERSITET" and "Lunds Tekniska Högskola". The main title "EDAA35 – Föreläsning 5" is centered in a white box, with the presenter's name "MARTIN HÖST" below it. The bottom half of the slide shows a black and white photograph of a classical building facade with statues on top, and a large, semi-transparent gold seal of the University of Lund on the right side.

LUNDS UNIVERSITET
Lunds Tekniska Högskola

EDAA35 – Föreläsning 5

MARTIN HÖST

1

Innehåll

- Kursinfo
- Exekveringsmiljö och lite om resten av datorn
 - Varför tar ett program inte lika lång tid varje gång?
- Mätning av exekveringstid
- Lite om lab 5

2

Kursinformation: tidigare laborationer som inte är godkända

- Det finns tid att bli godkänd (och att ställa frågor) på laborationen i denna veckan
- All extra information som behövs för att göra lab 3 själv finns på kursens hemsida <http://cs.lth.se/edaa35/lab3/>
 - Randomisering om du inte redan estimerat: Om du inte redan fått en blankett, välj alternativ A om du är född i en udda månad, annars alternativ B
 - Tips: titta på blanketterna för att förstå experiment-design

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

3

Kursinformation: Presentation av lab 4

- Görs individuellt på övningstid
- Anmäl dig till en av fyra möjliga tider i SAM (se kursens hemsida).
 - Kontrollera att du inte redan har tiden bokad, t ex av laboration i denna kursen
- Ta med dig dina fyra sidor utskrivna och var beredda att hålla presentationen för 4-5 andra personer
- 15 min inkl. frågor (5 min) och feedback (ca 2-3 min)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

4

Kursinformation: Lab 5

- Förberedelseuppgifter
- Troligt att ni måste arbeta med labb-uppgifterna själva före eller efter labb-tiden
- Presentation: Individuell rapport som ska granskas och godkännas i kursen
 - Inlämning av rapport i moodle
 - Inlämning kommentarer i moodle (på tre rapporter)
 - Alla får kommentarer från tre granskare. Om mer än en granskare klagar på något så måste ni uppdatera för att bli godkända
 - Inlämning av slutgiltig version (i SAM)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

5

Kursinformation: Projektet

- Görs i nästa läsperiod
- Grupper om 3 personer
- Ni får gärna komma med egna idéer, men det finns förslag
- Individuellt teori-quiz i Moodle
- Gruppen presenterar på "kurskonferens" på övningstid i sista veckan
- Tidplan enligt kursprogram
- Mer information på föreläsning 7

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

6

Kursinformation: nästa föreläsning

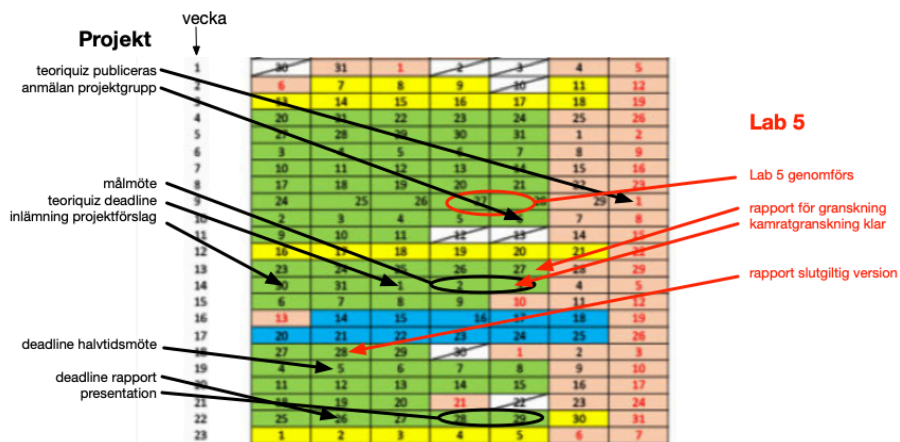
- Om att rapportera skriftligt
- Om litteratursökning och referenshantering
- Halva tiden gästföreläsning från UB
- I E:A

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

7

LP 4

EDAA35: Lab 5 och projekt 2020



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

8

Varför tar ett program inte lika lång tid varje gång?

- Beror på indata
- Flera saker händer samtidigt i datorn
- Programmet beter sig olika varje gång, t ex
 - Optimering
 - "Garbage collection"

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

9

Hur körs "ett vanligt" program?

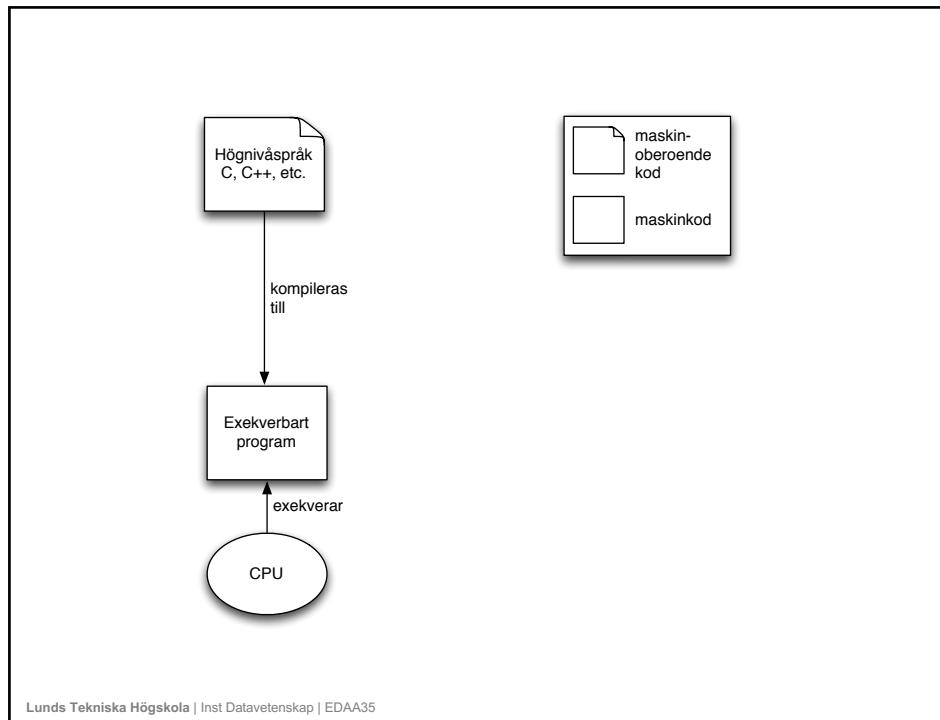
```
> ls
forelasning.c
> cc forelasning.c
> ls
a.out          forelasning.c
> ./a.out
Hello, EDAA35
>
```

```
#include <stdio.h>

int main() {
    printf("Hello, EDAA35 \n");
}
```

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

10



11

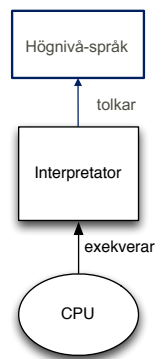
forts

- Fördelar:
 - Slipper skriva assembler direkt
 - Exekverar maskinkod, blir snabbt
- "Problem":
 - Maskinkoden anpassad för specifik datortyp, inte portabelt

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

12

Mer portabelt alternativt (interpretator)



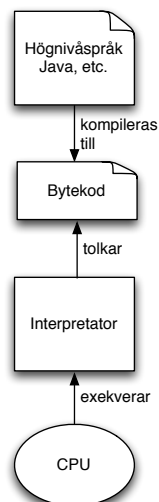
Exempel (R):
`>source("calculate.R")`

Fördel:
 - Portabelt
 Nackdel:
 - Långsamt

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

13

Bättre interpreterande alternativ



Text
`>javac Lab.java`
`>java -cp <classpath> Lab`

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

14

forts

- Fördelar
 - Portabelt, den kompillerade bytekoden kan flyttas mellan olika datorer
- Problem
 - Blir förhållandevis långsamt (men bättre än förra alternativet)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

15

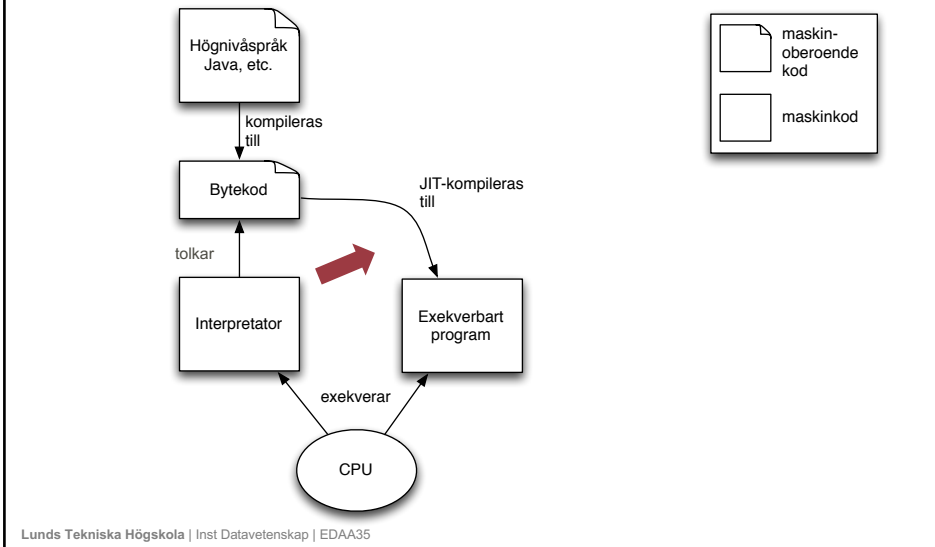
Exempel

- "Vanliga"
 - C, C++, Pascal, Cobol, Fortran
- Interpreterande
 - Java (såhär fungerade det i början)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

16

Alt 3: både snabbt och portabelt



17

Just-in-time kompilering (JIT)

- Används i t ex Java idag
- Små delar av bytekoden kompileras efterhand
- Avvägning
 - Ju mer som kompileras desto snabbare exekvering
 - Ju mer som kompileras desto längre kompileringstid
- Fördelar:
 - Blir både portabelt och snabbt
 - Ger även möjlighet att optimera koden

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

18

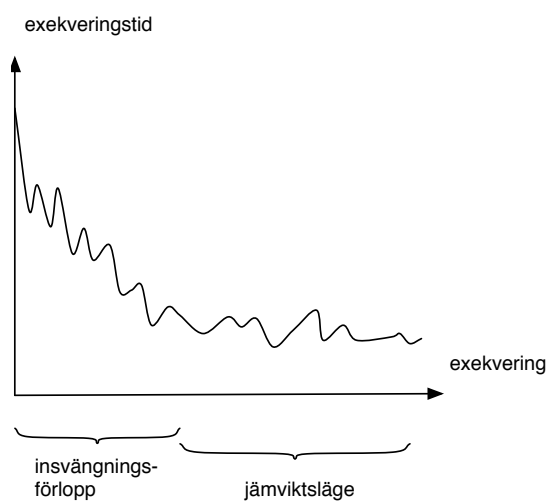
Optimering

- Anpassa den kompilerade koden efter programmet
 - T ex byta ut loopar mot konsekutiva satser
- All information finns dock inte i början och det tar tid att optimera
 - Programmet optimeras efter hand
 - » → Programmet blir snabbare och snabbare
- Om man vill så kan man stänga av JIT-kompilatorn
 - java -Xint fil

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

19

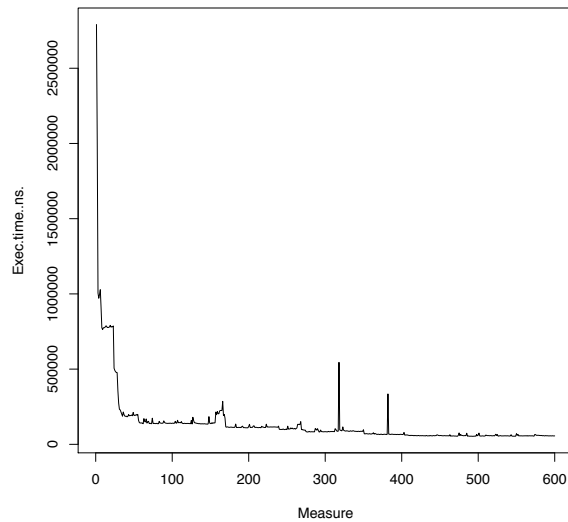
Snabbare och snabbare



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

20

Riktigt exempel



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

21

Andra anledningar till att det tar olika lång tid varje gång

- Olika indata, olika förutsättningar → olika tid
- Andra saker händer i datorn → försök stänga av så mycket som möjligt när ni mäter
- Programmet kan innehålla flera "parallella" trådar
- "Garbage collection"

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

22

Prestanda och mätning av exekveringstid



23

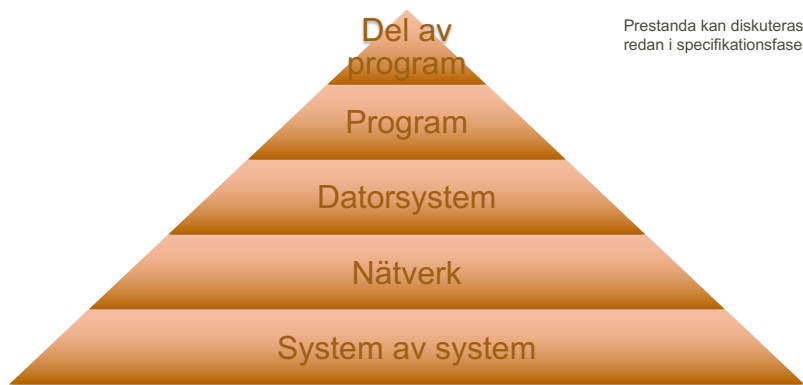
Prestanda

- Exekveringstid
- Minne som krävs
- Energi som går åt
- ...
- ISO9126:
 - Efficiency: “The capability of the software product to provide **appropriate performance**, relative to the amount of resources used, under stated conditions.”
 - Time behaviour: “The capability of the software product to provide appropriate **response and processing times** and throughput rates when performing its function, under stated conditions.”

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

24

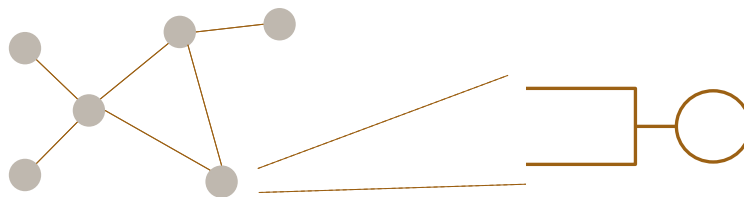
System – ett sätt att se på det



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

25

exempel

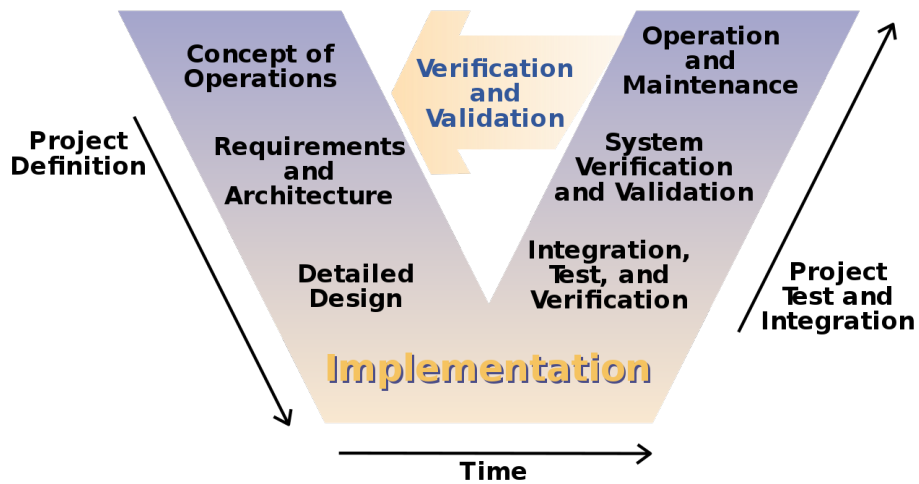


Enkel modell:
 $T = 1/(\mu - \lambda)$ om $\mu > \lambda$
 (se EITF95 kösystem)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

26

Utveckling av programvara – ett sätt att se på det



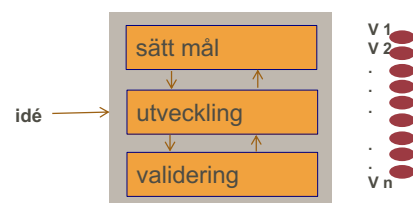
Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

(Wikipedia V-model public domain)

27

Utveckling av programvara (2)

- Mer eller mindre iterativt
 - Steg efter steg
 - Inkrementellt
 - Iterativt



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

28

Egenskaper hos prestandamått

- Linjäritet
- Tillförlitlighet
- Lätt att mäta
- Konsistens
- Oberoende

(Lilja, 2000)

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

29

Klockfrekvens

- hur många klockcykler som processorns inbyggda klocka gör per sekund, t ex 1,6 GHz
- relaterat till hur många instruktioner som utförs per tidsenhet, men tar inte hänsyn till hur många klockcykler som krävs för varje instruktion
- enkelt att redogöra för för olika datorsystem
- ...många andra saker påverkar upplevelsen av att använda en dator, t ex lagringenhetens snabbhet, minnets snabbhet
- ...10% snabbare klockfrekvens behöver inte upplevas som just 10% snabbare i olika datorer

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

30

MIPS (Million Instructions Per Second)

- hur många instruktioner en processor i ett datorsystem exekverar per sekund
- Jmf med klockfrekvens bättre eftersom det tar hänsyn till att varje instruktion kräver olika många klockcykler
- Dock så genererar instruktioner olika mycket värde i olika processorer...

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

31

MFLOPS (Millions of floating-point instructions per second)

- Istället för att fokusera på instruktioner, fokuserar på flyttalsoperationer
 - Flyttal: reellt tal med begränsat antal siffror, t ex float
- Mer "rättvist" mått än MIPS eftersom flyttalsoperationer är mer lika mellan olika datorer
- Alla applikationer handlar dock inte om att räkna med flyttal...

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

32

Testuppsättningar, t ex SPEC



33

SPECjvm2008: för Java exekveringsmiljöer

- kompilering av java-filer
- komprimering av data
- kryptering och dekryptering av data
- användning av en java-databas för affärssystem
- mp3-avkodning
- numeriska metoder, baserat på SciMark 2.0
- serialisering och av-serialisering av Java-objekt
- grafisk visualisering
- XML-hantering

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

34

Exekveringstid för programdel

- Ofta intressant att mäta hur lång tid en del av ett program tar, t ex i Java:

...

```
long startTime = System.nanoTime();
```

```
// measured code
```

```
long endTime = System.nanoTime();
```

```
long measuredTime = endTime - startTime;
```

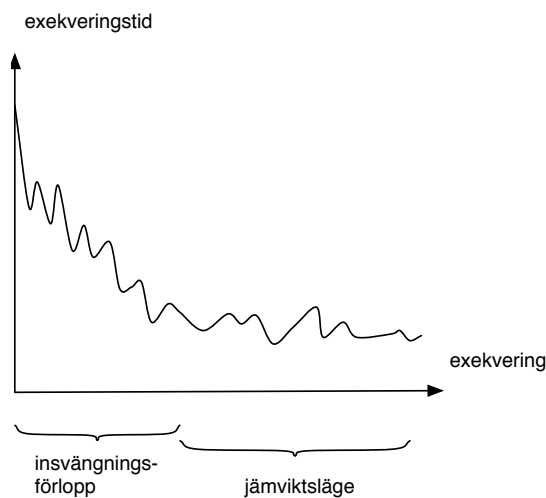
...

- Skulle också kunna använda `currentTimeMillis()`
- "Klocktid"

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

35

Vad vill vi mäta - tiden vid uppstart eller vid jämviktsläget?



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

36

Tiden vid uppstart

- Starta programmet p gånger och mät tiden för programdelen en gång per uppstart

– Data: x_1, x_2, \dots, x_p

- Beräkna medelvärdet $\bar{x} = \frac{1}{p} \sum_{i=1}^p x_i$

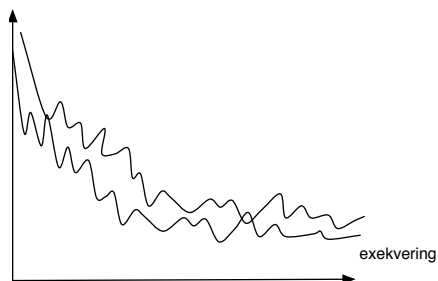
och räkna ut konfidensintervallet

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

37

Tid vid jämviktsläget

Två uppstarter:
exekveringstid



- Uppstarterna kan bete sig olika
- Jämviktslägena behöver inte starta på samma läge

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

38

Forts.

1. Studera p olika uppstarter av exekveringsmiljön, där man vid uppstart i gör s_i mätningar av samma sak.
 - x_{ij} är resultatet av mätning j vid uppstart i
 2. För varje uppstart bestäm var tröskeln, r_i , mellan insvängningen och jämvikten är. Man vill använda $s_i - r_i = k$ mätningar
 3. Bestäm medelvärdet av varje uppstart $\bar{x}_i = \frac{1}{k} \sum_{j=s_i-k+1}^{s_i} x_{ij}$
1. Bestäm medelvärdet av medelvärdena $\bar{\bar{x}} = \frac{1}{p} \sum_{i=1}^p \bar{x}_i$ och konfidensintervallet

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

39

Most Influential OOPSLA Paper Award

Presented annually to the author(s) of a paper presented at the OOPSLA held 10 years prior to the award year. The award includes a prize of \$1,000 to be split among the authors of the winning paper. The papers are judged by their influence over the past decade.

Selection Committee The award given in year N is for the most influential paper presented at the conference held in year $N-10$. The selection committee consists of the following members:

- the current SIGPLAN Chair, ex officio,
- a member of the SIGPLAN EC appointed as committee Chair by the SIGPLAN Chair,
- the General Chair and Program Chair for OOPSLA $N-10$,
- the General Chair and Program Chair for OOPSLA $N-1$, and
- a member of the SIGPLAN EC appointed by the committee Chair.

The SIGPLAN Chair shall adjudicate conflicts of interest, appointing substitutes to the committee as necessary.

1986-1996

To cover the years from the inception of OOPSLA (1986) to 1996, the OOPSLA steering committee formed a committee to select the three most influential OOPSLA papers that were presented during that time period. The three most influential OOPSLA papers from 1986-1996:

- Subject Oriented Programming: A Critique of Pure Objects, William Harrison and Harold Ossher
- Concepts and Experiments in Computational Reflection, Pattie Maes
- Self: The Power of Simplicity, David Ungar and Randall S. Smith

Recipients

Andy Georges, Dries Buytaert, Lieven Eeckhout 2017

(for 2007) *Statistically rigorous Java performance evaluation*

Citation:

The paper by Georges et al. has been widely influential in Java performance evaluation. Before this paper, Java performance evaluation methodologies had varied widely, leading to difficulty in comparisons and often misleading conclusions. The authors survey several past studies, carefully discuss discrepancies in experimental and reporting methodologies, and expose the pitfalls of benchmarking approaches. The fundamental problem is that without an appropriate statistical analysis, it is difficult to tell whether reported measurements in the literature reflect significant distinctions or just random fluctuations. The paper proposes a statistically rigorous approach for experimental measurements in the Java setting, ways to mitigate uncertainty, and even new reporting techniques. The paper also introduced JavaStats, publicly available software that monitors the variability in measurements to determine how many measurements will be needed in all to reach a desired level of statistical confidence.

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

40

Indata

- Exekveringstiden påverkas av vilken indata man har
- Definiera vilka indata som används i varje studie
- T ex för sortering
 - Hur många datapunkter som ska sorteras?
 - Hur sorterade / blandade de är från början?

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

41

R microbenchmark

```

unsorted <- order(runif(100))
> microbenchmark(insertionSort(unsorted), sort(unsorted))
Unit: microseconds
      expr      min       lq      mean     median      uq      max     neval
insertionSort(unsorted) 1510.668 1823.7300 1958.16018 1853.934 1943.3295 6391.995 100
      sort(unsorted)    19.735   24.5295  30.03572   27.115   32.1005  86.174  100
>

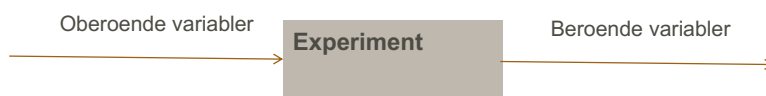
```

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

42

Experiment

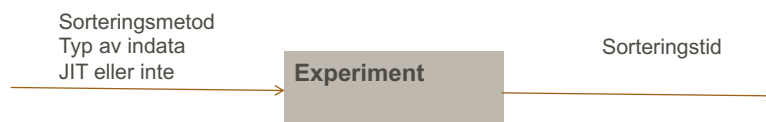
- Den uppmätta tiden påverkas alltså av andra saker i datorn
- ...men om det är en jämförande studie så kan det gå bra om alla saker som jämförs påverkas lika mycket



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

43

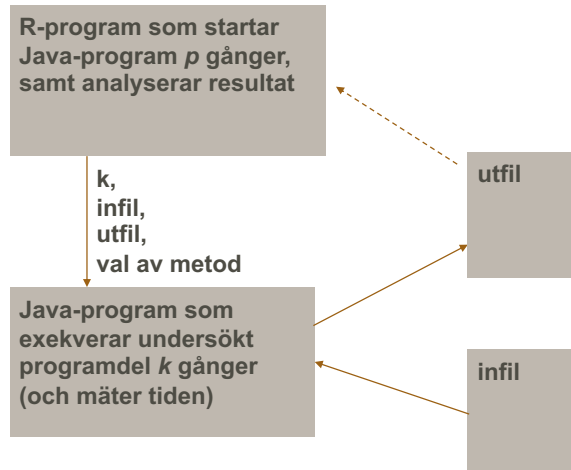
Lab 5



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

44

Lab 5: “labbuppställning”



Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

45

Argument till Java-program

```

public class ArgumentReader {
    public static void main(String[] args) {
        int nrArguments = args.length;
        System.out.println("Antal argument: " + nrArguments);
        if(nrArguments>0) {
            for(String arg: args) {
                System.out.println(arg);
            }
        }
    }
}
  
```

```

> java ArgumentReader
Antal argument: 0
> java ArgumentReader Hej 2018
Antal argument: 2
Hej
2018
  
```

Lunds Tekniska Högskola | Inst Datavetenskap | EDAA35

46