Teachers and Contacts

• Petr Tůma
tuma@d3s.mff.cuni.cz

• Vojtěch Horký
horky@d3s.mff.cuni.cz

Room: MS205

Course Webpage: http://d3s.mff.cuni.cz/teaching/peva
Acknowledgments

- Vlastimil Babka
- Lubomír Bulej
- Tomáš Kalibera
- Peter Libič
Labs

SU1, Wednesday 14:00 (even weeks)
  • First lab – Introduction, simple measurement, R basics
  • Next 4 labs – Assignments
  • See calendar on the web

Credit
You need to finish all 4 assignments in the lab.
It is possible to re-do one assignment during the examination period.

Bringing your own laptop …

… is certainly possible but make sure you have all the necessary software installed (at least R, JDK, GCC, PAPI and perf).
It is much easier to use machines in Rotunda (remote access possible).
Lab Topics

0 Measurement examples, introduction to R (today)
1 Timers and counters
2 Instrumentation
3 Plotting, tests, experimental setup
4 Complete performance experiment
Measurement Example
Measurement Example: Sorting

Measure how long it takes to sort $n$ numbers in your favorite programming language (use the sorting implementation from the standard library).

Fast way to generate $\text{SIZE}$ random numbers:

\[
\text{echo } \ "\text{for}\(x=0;x<\text{SIZE};x++\)\text{random()}\;\text{\" | bc > data.\text{SIZE}}
\]
# The Right Way™ for Measurement

<table>
<thead>
<tr>
<th>Language</th>
<th>Construct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell</td>
<td><code>/usr/bin/time -f %e ...</code></td>
</tr>
<tr>
<td>C</td>
<td><code>clock_gettime(CLOCK_MONOTONIC_RAW) (&lt;time.h&gt;)</code></td>
</tr>
<tr>
<td></td>
<td><code>QueryPerformanceCounter (&lt;windows.h&gt;)</code></td>
</tr>
<tr>
<td>C++</td>
<td><code>std::chrono::steady_clock</code></td>
</tr>
<tr>
<td>Java</td>
<td><code>System.nanoTime()</code></td>
</tr>
<tr>
<td>C#</td>
<td><code>Stopwatch class</code></td>
</tr>
<tr>
<td>Ruby</td>
<td><code>Benchmark module</code></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Measurement Example: Common Issues

• Measurement of the whole program.
• Measurement of one call only.
• Small number of measurement loops.
• Dead code elimination.
• ...

Common Issues: Warm-up

![Graph showing time to sort (ms) vs. measurement index.]
Introduction to R
### Introduction to R

#### Vectors

```r
x <- c(2, 3, 4, 3, 3, 2, 5, 1, 2, 3)
x <- rnorm(5, 10, 2)  # (5 numbers from $\mathcal{N}(10, 4)$)
```

#### Printing

```r
print(x)
head(x)  # (just like the Unix command)
```

#### Data reading

```r
read.table, read.csv
x <- read.table("data.txt")
```
Plotting in R

Basics

plot(x)
boxplot(x)
hist(x)

Zoom-in

plot(x[2:20])
plot(x, xlim=c(9,11), ylim=c(3,8))
Plotting in R (cont.)

**Lines instead of dots**

```r
plot(x, type="l")
```

**Multiple categories**

```r
plot(x1, ylim=c(0, max(c(x1,x2,x3))), col="blue")
points(x2, col="red")
points(x3, col="green")
legend("topleft", legend=c("x1", "x2", "x3"),
       col=c("blue", "red", "green"), pch=20)
```
Measurement Example (cont.)
Measurement Example: Task # 2

- Run with different sizes of input.
- Plot the results.
- Compute (and plot) relative time needed to “sort” one element.
Measurement Example: Our Solution


**prepare.sh**
Generates random input data.
Compiles Java and C++ sources.

**run.sh**
Executes the C++ binary, Java class and sort(1).

**compare.r**
Plots comparison of execution times.

**relative-time.r**
Plots relative time needed for sorting one element.
Measurement Example: Plot by Input Size

Data size [millions of elements]

Time to sort [s]

- Java
- Unix
- C++
Measurement Example: Time to “Sort” One Element (Java)

- Time to sort one element [s]
- Data size [number of elements] (logarithmic)