Multicore Programming
Final Remarks

Louis-Claude Canon
louis-claude.canon@univ-fcomte.fr

Bureau 414C

Master 1 computer science – Semester 8
Performance and Code Comparison

Compute the sum of cosinuses of all integers between 1 and 100 000 000.
Java:

```java
IntStream.rangeClosed(1, 100_000_000)
    .mapToDouble(x -> Math.cos(x))
    .sum();
```
Other Languages

C:

```c
double sum = 0;
for (int i = 1; i <= 100000000; i++)
    sum += cos(i);
```

R:

```r
sum(cos(1:1e8))
```

Rust:

```rust
(1..n + 1)
    .map(|x| (x as f64).cos())
    .sum();
```
Benchmarks

Sequential computation time:

<table>
<thead>
<tr>
<th>Implément.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>7.3s</td>
</tr>
<tr>
<td>C</td>
<td>3.5s</td>
</tr>
<tr>
<td>R</td>
<td>4.4s</td>
</tr>
<tr>
<td>Rust</td>
<td>4.0s</td>
</tr>
</tbody>
</table>

Parallel computation time (four multithreaded cores):

<table>
<thead>
<tr>
<th>Implément.</th>
<th>Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>3.2s</td>
<td>2.3x</td>
</tr>
<tr>
<td>Rust</td>
<td>1.5s</td>
<td>2.7x</td>
</tr>
</tbody>
</table>

Memory consumption: negligible except for R (around 1 GB).
Perspectives

Advanced multicore concepts:

- Java memory model.
- Thread pools under load: how to customize a ThreadPoolExecutor.
- Synchronization and locking: barriers, phases (java.util.concurrent.locks).
- Atomic data structures (java.util.concurrent.atomic).
- Concurrent data structures.
- Designing a custom collector for better parallelization.
- Practical study of reactive programming.
- Actors.
- ...

Louis-Claude Canon  
MCP – Final Remarks
Outline

References
Modern Java in Action, 2nd edition, 2018

- Covers recent Java additions (Java 8, 9 and 10).
- Presents streams, API improvements, lambdas, reactive programming and functional-style Java programming.
Effective Java, 3rd edition, 2018

- written by Joshua Bloch (lead developer of the Java Collection framework)
- 1 chapter on streams
- 1 chapter on concurrency
- many good advises on Java idioms (relevant for any Java programmer)
The art of multiprocessor programming, revised preprint, 2012

- technical/low-level and multi-languages (Java, C# and Pthreads)
- 1 chapter on future and work-stealing
- concurrent data structures
- concurrent synchronization mechanisms
Java Concurrency In Practice, 2006

- co-written by Joshua Bloch
- many code examples and advises for concurrency
- a bit outdated