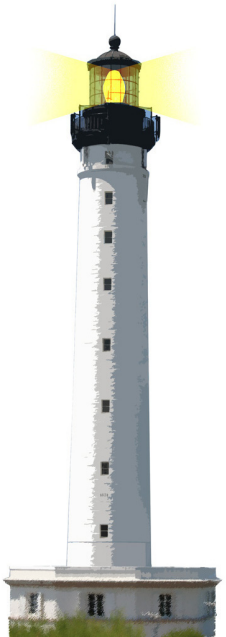
Benchmarking in Pharo

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W5S09

http://www.pharo.org



Common Wisdom

If you did not profile your code you may have 40-50% speed up waiting for you.

Measuring Execution Speed

We create an expression and use [expression] timeToRun

```
[ 1000 factorial ] timeToRun
```

Comparing Two Executions

Is select:, then collect: slower than select:thenCollect:?

timeToRun example

```
| coll |
coll := #(1 2 3 4 5 6 7 8 9 10) asOrderedCollection.
[ 1000000 timesRepeat: [
    (coll select: [ :each | each > 5]) collect: [ :i | i * i ]]
  ] timeToRun
> "0:00:00:00.517"
```

```
| coll |
coll := #(1 2 3 4 5 6 7 8 9 10) asOrderedCollection.
[ 1000000 timesRepeat: [
    coll
      select: [ :each | each > 5]
      thenCollect: [ :i | i * i ]]
    ] timeToRun
> "0:00:00:00.362"
```

bench

- Returns how many times the code can get executed in 5 seconds
- Answer a string with meaningful description

```
[ 1000 factorial ] bench
> '610.234 per second'
```

The higher the better!

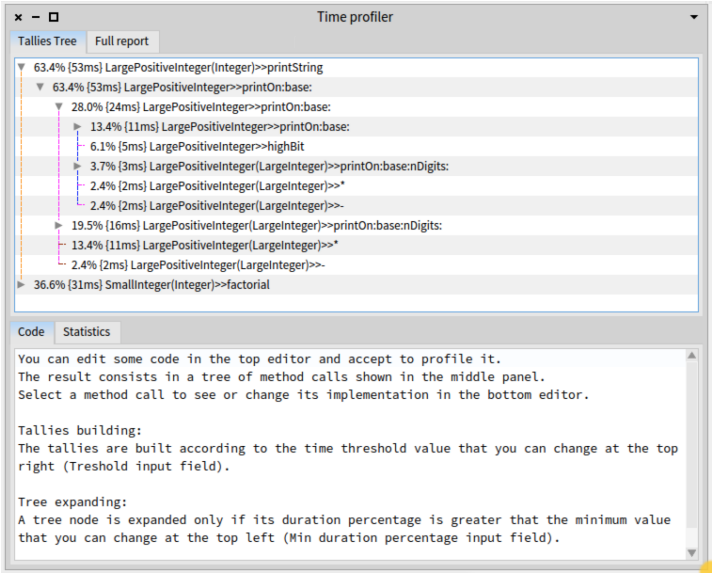
[1234 factorial] benchFor: 2 seconds

Time Profiler

- TimeProfiler: a sampling-based code profiler
- At regular interval, take information from the execution stack

```
TimeProfiler
 spyOn: [ 20 timesRepeat: [
        Transcript show: 1000 factorial printString ] ]
```

Time Profiler



Summary

- [anExpression] timeToRun
- [anExpression] bench
- TimeProfiler spyOn: [anExpression]
- Check *Profiling Applications* Chapter in **Deep into Pharo** (at `http://books.pharo.org`)

A course by

and

in collaboration with