

What is the execution time of `spin(n)` when $n = 1\,000\,000$?

Function `spin(n)`:

```
void spin(int n) {  
    int i = n;  
    while (--i >= 0);  
}
```

Computer:

Intel Pentium 4 workstation with 1.5 GHz i686 processor (1st-level cache: 8 KB, 8-way associative 2nd-level cache: 256 KB, internal memory: 256 MB)

Operating system:

Red Hat Linux 7.1

Compiler:

g++ C++ compiler 3.0.4 (with option `-O3`)

The answer is ...

```
pc-014> python single_spin.py
```

```
E
```

```
=====
ERROR: __main__.spin_case
```

```
Less than 90% of the outcomes differ at most 20% from
the median; try again
```

```
-----
ERROR: __main__.spin_case
```

```
Traceback (most recent call last):
```

```
  File "benchmark.py", line 170, in run
```

```
    answer = self.output()
```

```
  File "single_spin.py", line 29, in output
```

```
    return (self.n, float(self.driver_output) / self.n)
```

```
ValueError: empty string for float()
-----
```

```
=====
: ()
-----
```

```
Ran 1 benchmark case in 44.804s
```

```
Errors: 2
```

```
pc-014> python single_spin.py
```

```
.
```

```
=====
: (1000000, 2.6322299999999998)
-----
```

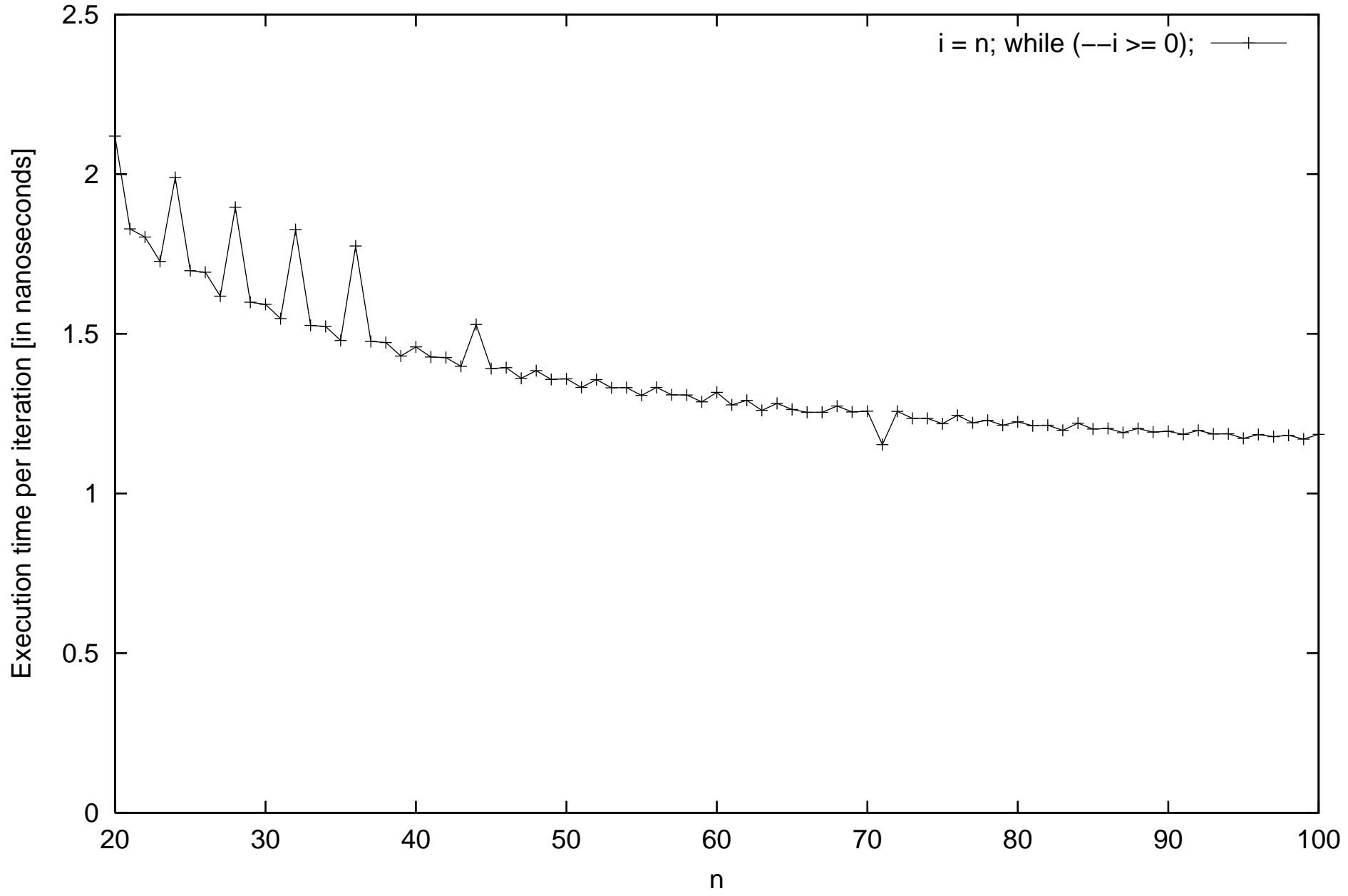
```
Ran 1 benchmark case in 3.602s
```

```
OK
```

Jyrki's benchmark tool

- Write a C++ class describing the experiment
- Fill in a form to explain for which parameters the experiment is to be carried out (i.e., write a Python program)
- Send the form to the system (i.e., run the Python program)
- The system returns the answer in textual or graphical form

Execution time of an empty loop



C++ experiment class

```
/*  
Benchmark for spinning  
  
Author: Jyrki Katajainen  
Email: jyrki@diku.dk  
$Revision: 1.2 $  
$Date: 2003/02/05 12:29:06 $  
  
*/  
  
class spin {  
public:  
    spin(int n): n(n) {  
    }  
    void primal() {  
        int i = n;  
        while (--i >= 0);  
    }  
private:  
    int n;  
};
```

Python form

```
"""
Benchmarking the execution time of an empty loop

shell> echo $PYTHONPATH
/home/disk04/jyrki/CPHSTL/Tool/Benchmark/
"""

__author__ = "Jyrki Katajainen"
__email__ = "jyrki@diku.dk"
__version__ = "$Revision: 1.2 $"[11:-2]
__date__ = "$Date: 2003/02/05 12:29:06 $"[7:-2]

import benchmark
import os

class spin_case(benchmark.case):
    def __init__(self, n):
        benchmark.case.__init__(self)
        self.n = n
        self.computer = 'pc-014.diku.dk'
        self.compiler = '/scratch/g++/bin/g++'
        self.compiler_options.extend(['-O3'])
        self.include_files.extend(['spin_benchmark.c++'])
        self.constructor_call = 'spin(' + str(n) + ')'
        self.time_unit = 'ns'
        self.driver_file = self.generate_execution_time_driver()

    def output(self):
        return (self.n, float(self.driver_output) / self.n)

if __name__ == '__main__':
    benchmark.main(\
        task = spin_case(1000000),\
        runner = benchmark.text_runner\
    )
```

Got interested in Python?

Come to my lecture next Monday:

Speaker:

Jyrki Katajainen

Title:

Learning Python in two hours

Time:

Monday 10 February 2003 at 13.15–15.00

Place:

N026 at DIKU

What does the word “bachelorproject” mean for me?

- \leq 40 pages
- not much research
- not much programming
- practise writing
- practise information retrieval
- work individually

Possible topics under the umbrella of the CPH STL

- programming language issues
 - pure C decompiler
 - Meta C++
- algorithmic issues
 - stable sort
 - in-place sort
- tools
 - automatic testing
- web services
 - interactive documentation

For other proposals, see the CPH STL homepage <http://www.cphstl.dk> under menu item “New Developers”

Basic concept

- select a topic relevant for the CPH STL
- search for the best known solutions from the literature
- possibly do some implementation work
- report the outcome of your work

Relevant links

Resources for theses writers:

<http://www.diku.dk/undervisning/2002f/741/>

L^AT_EX style file DIKU-article:

<http://www.cphstl.dk/Report/>

Bibliography style file DIKU.bst:

<http://www.cphstl.dk/Report/>

CPH STL home page:

<http://www.cphstl.dk>

PE-labs home page:

<http://www.diku.dk/forskning/performance-engineering/>

How to get started?

Contact me personally or electronically!

Office hours:

Mondays 13.15 - 15.00 up to 16 June
2003

E-mail:

`jyrki@diku.dk`