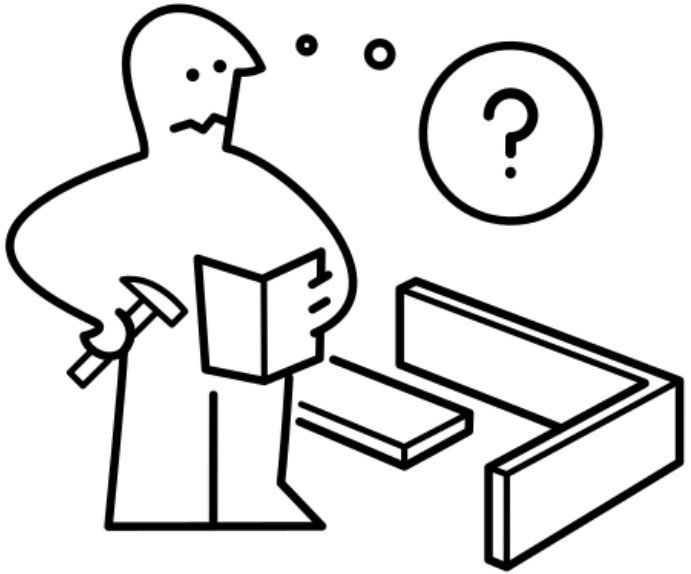
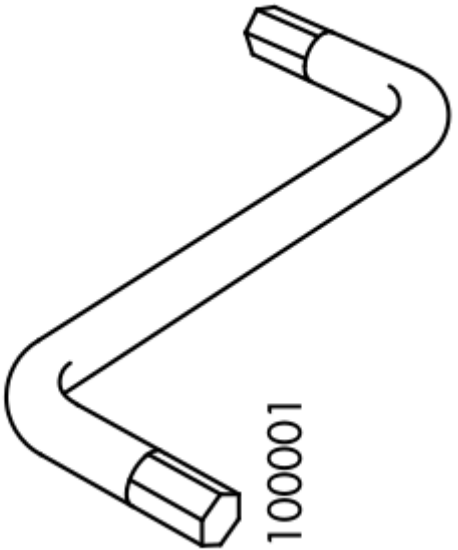
How worse can a stock perform?

Markus Bilz, 1705042



Greatest, likely loss?



Value at Risk
Simulation

(c) ikea.com

Value at Risk using a Monte Carlo simulation

```
repeat n times -> first kernel
    repeat t times
        generate normal distributed number
        update interim price
    save end price to path array
extract the nth rank -> second kernel
scale value at risk to holding period
print results
```

1st kernel

(generating random prices)

specification

trivial problem / no interaction between threads

All threads run the same code / no thread divergence...

{demo}

2nd kernel

(extracting the minimal price)

{demo}

specification

non trivial problem / dependence between threads

potential bank conflicts / idle threads

solved through efficient reduction

...

performance evaluation

gpu specification

gpu specification

NVIDIA GeForce 940MX

3 Mb dedicated memory

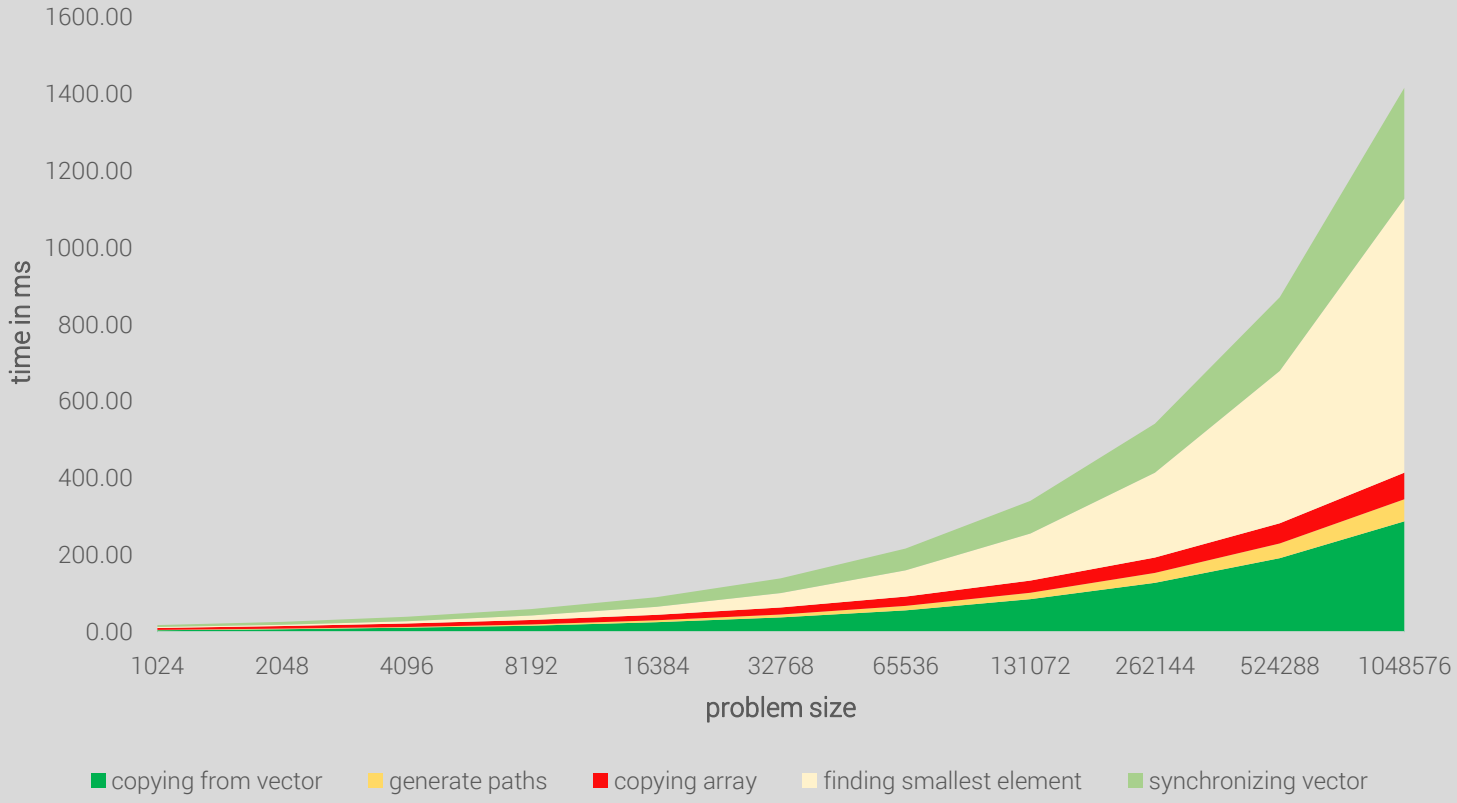
384 cores

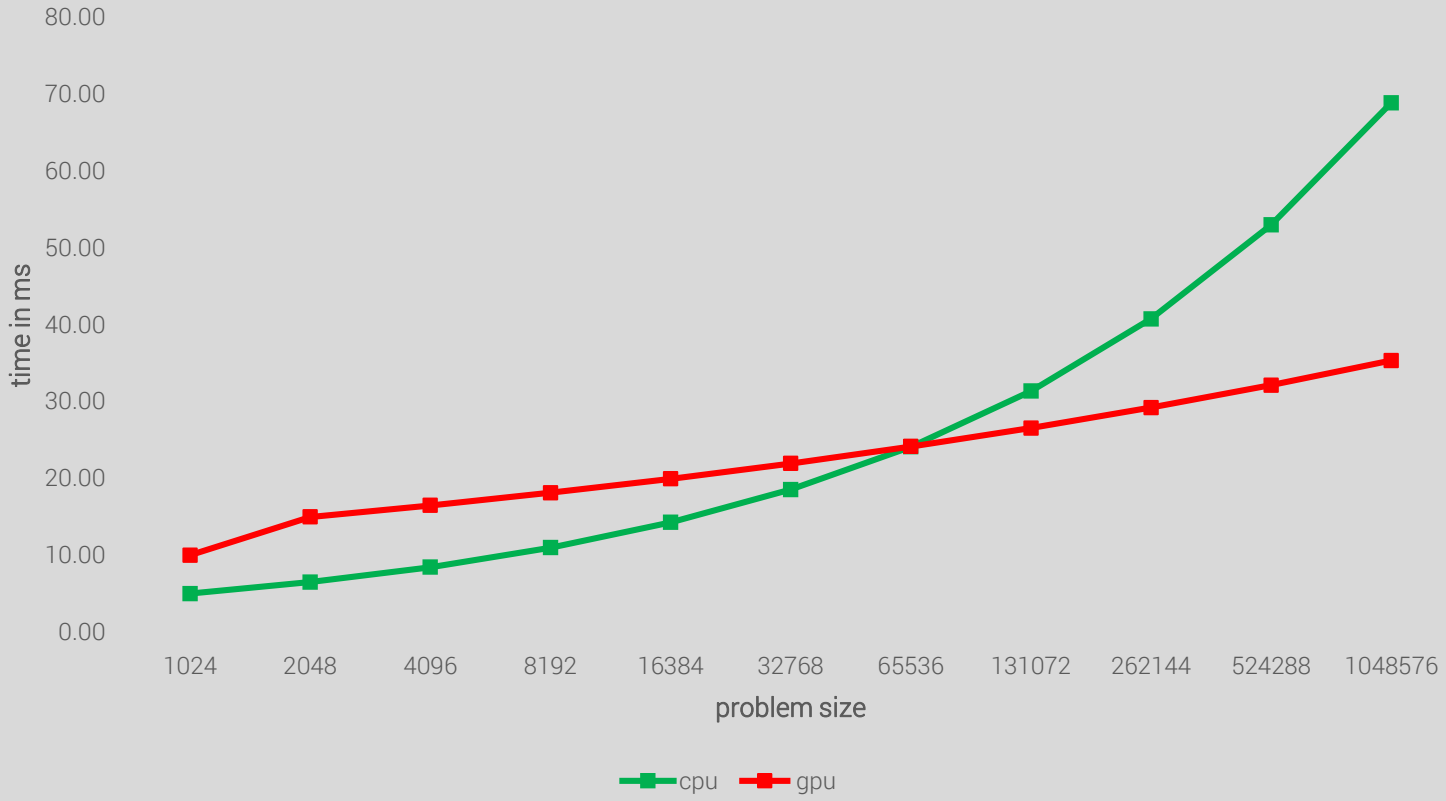
misc

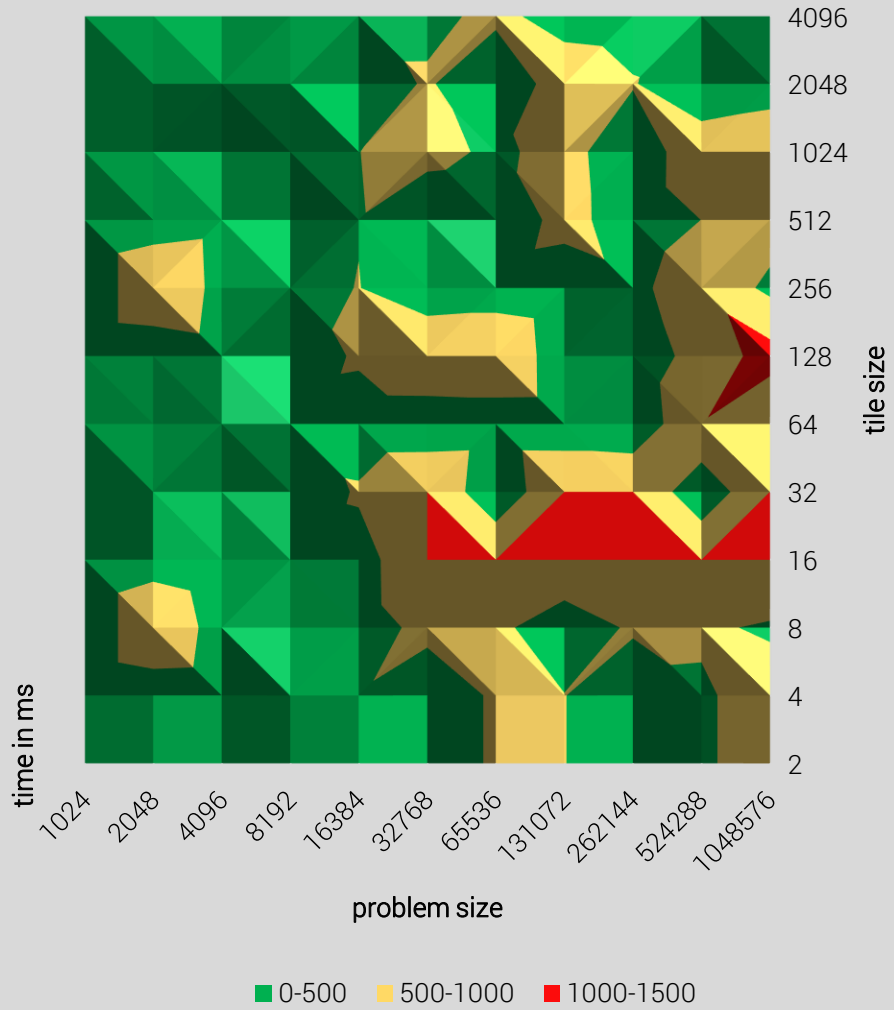
using wakeup call to prevent JIT and lazy initialization

using std::chrono::steady_clock

using tile_size with multiple of two


1600.00
1400.00
1200.00
1000.00
800.00
600.00
400.00
200.00
0.00
time in ms
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
problem size
copying from vector
generate paths
copying array
finding smallest element
synchronizing vector


80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
time in ms
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
problem size
cpu
gpu


time in ms
tile size
4096
2048
1024
512
256
128
64
32
16
8
4
2
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
problem size
0-500
500-1000
1000-1500

https://github.com/KarelZe/MC-VAR-Sim