Premature optimization is a root of all evil

Проблемы при профилировании

Скорость работы зависит от множества факторов

- Статистическая обработка разультатов (делаем 5 тестов, отбрасываем самый медленный и самый быстрый)
- Контролируйте нагрузку на процессор и его тактовую частоту

Детерминированное и вероятностное профилирование

- Детерминированное встраивание счетчиков в код
- Вероятностное периодическая остановка программы и анализ времени останова (периодическая остановка по исчерпанию люого счетчика, а не только времени)

top/htop/atop/iotop/nettop/time & sysinternals

timeit

Используется для профилирования быстрых конструкций

```
import timeit
print timeit.timeit("a + b", "a,b = 1,2") # 0.0374751091003
ipython - %timeit expression
```

timeit

```
import timeit

for pow in range(6, 8):
    print timeit.timeit("a + b", "a,b = 1,2", number=10 ** possible

# 2.5 E-8
# 2.14E-8
# 2.1 E-8
```

timeit

```
zero_time = timeit.timeit("pass", "", number=number) / number
timeit.timeit(..., number=number) / num - zero_time
```

profile cProfile

```
import re
     import cProfile
2
3
     cProfile.run("re.compile("a|b|" * 100 + "c")")
4
  4913 function calls (4712 primitive calls) in 0.002 seconds
   Ordered by: standard name
  ncalls tottime percall cumtime percall filename: lineno (functio
         0.000 0.000
     1
                       0.002
                               0.002 <string>:1(<module>)
         0.000 0.000 0.002
                                0.002 re.py:188(compile)
                                0.002 re.py:226( compile)
         0.000 0.000
                       0.002
     1
         0.000 0.000
                       0.000
                                0.000 sre compile.py:179( compile
                                0.000 sre compile.py:208(_optimize
      2
         0.000 0.000
                       0.000
         0.000
                0.000 0.000
    408
                                0.000 sre compile.py:25 (identity
                                0.000 sre compile.py:33( compile)
         0.000
                0.000
                        0.000
                                0.000 sre_compile.py:360(_compile_
         0.000
               0.000
                       0.000
     1
         0.000
               0.000
                                0.000 sre compile.py:473 (isstring)
                       0.000
     1
         0.000
               0.000
                       0.000
                                0.000 sre compile.py:479( code)
                                0.002 sre compile.py:494(compile)
         0.000
     1
                0.000
                        0.002
```

203 0.000 0.000 0.000 0.000 sre_parse.py:127(__len__)
404 0.000 0.000 0.000 0.000 sre_parse.py:131(__getitem_