

Jak měřit výkon .NET kódu správně

Jiří Činčura

www.tabsoverspaces.com

@cincura_net

MVP

Profiling vs. benchmarking vs. microbenchmarking

- Profiling: Zjišťování co je pomalé (nebo rychlé)
 - dotTrace, dotMemory, ANTS Profiler, PerfView, VS Profiling Tools, ...
- Benchmarking: Zjišťování jak dlouho to trvá (metrika)
 - Stejný SW, různý HW
 - Různé verze SW
 - ...
- Microbenchmarking: Měření konkrétního malého kusu kódu

Není hned třeba...

- Přemýšlejte o architektuře
- Používejte rychlé algoritmy (big O)
- Volte správné datové struktury
 - List, hash tabulka, B tree, B+ tree, Red-black tree, ...
- Přemýšlejte o využití paměti

- I/O, caching, ...

Je (micro)benchmarking kódu jednoduchý?

```
var start = DateTime.Now;  
method();  
var duration = DateTime.Now - start;
```





Je (micro)benchmarking kódu jednoduchý?

- Přesnost a latence měření
- Warmup, cleanup
- (Ne)optimalizovaný build
- (Ne)připojený debugger
- Optimalizace měřeného kódu
 - Inlining, hoisting, lowering, unrolling, ...
- JIT
- GC
- Tiered Compilation
- Izolace prostředí
 - Power plan?
- Sběr a vyhodnocení měření
 - Modalita, odchylky, ...

Je (micro)benchmarking kódu jednoduchý?

```
static double Profile(string description, int iterations, Action benchmark) {  
    Process.GetCurrentProcess().PriorityClass = ProcessPriorityClass.High;  
    Thread.CurrentThread.Priority = ThreadPriority.Highest;  
    // warm up  
    benchmark();  
    var sw = new Stopwatch();  
    // clean up  
    GC.Collect(); GC.WaitForPendingFinalizers(); GC.Collect();  
    sw.Start();  
    for (var i = 0; i < iterations; i++) {  
        benchmark();  
    }  
    sw.Stop();  
    return watch.Elapsed.TotalMilliseconds;  
}
```


Nástroje

- Lepší je využít hotové nástroje
- BenchmarkDotNet
- NBench, MeasureIt, MS Xunit Performance, Lambda Micro Menchmarking, Etime.Benchmarks

BenchmarkDotNet

- Knihovna pro benchmarking
 - Abstrahuje komplexitu měření, takže je možné se soustředit jen na problém
- Podpora pro různé konfigurace
 - Windows, Linux, MacOS
 - x86/x64/...
 - LegacyJIT, RyuJIT
 - GC flags
 - .NET, .NET Core, Mono, CoreRT, ...
- Prezentace a validace výsledků
 - Modalita a odchylky
 - Grafy
 - Výstupy
- Diagnostika
 - Alokace, instrukce, ETW/hardware counters, ...

BenchmarkDotNet

- Operation: spuštění měřené metody
 - Iteration: několik Operations
 - Pilot: výběr počtu Operations
 - OverheadWarmup, OverheadWorkload: měření overhead BDN
 - ActualWarmup: warmup měřené metody
 - ActualWorkload: měření
-
- Výsledek: $\text{ActualWorkload} - \text{avg}(\text{overhead})$

BenchmarkDotNet

- Benchmark
 - Baseline
- Params
- GlobalSetup, GlobalCleanup

BenchmarkDotNet

- Validators
- Jobs
- Diagnosers
- Columns
- Exporters
- Loggers
- Analyzers

Demo & Fun