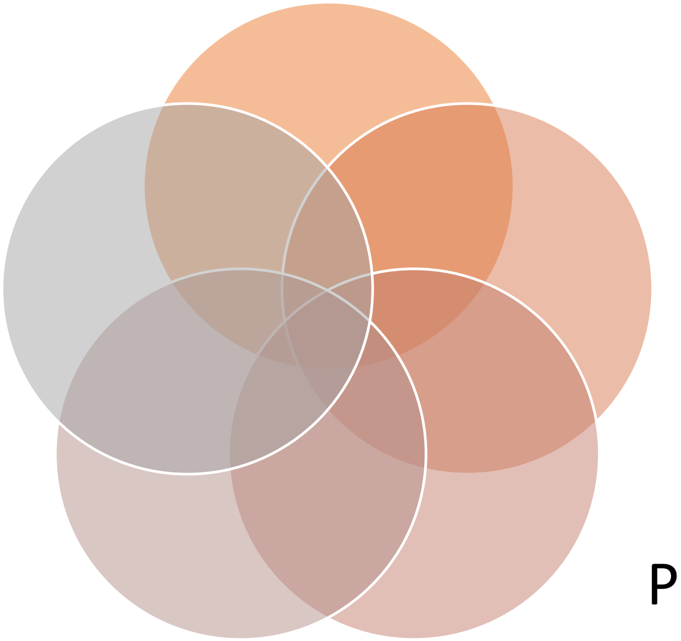
PerfView

Measure and Improve Your App’s Performance for Free

Sasha Goldshtein

CTO, Sela Group

@goldshtn



Performance metrics and simulations

Development-time profiling

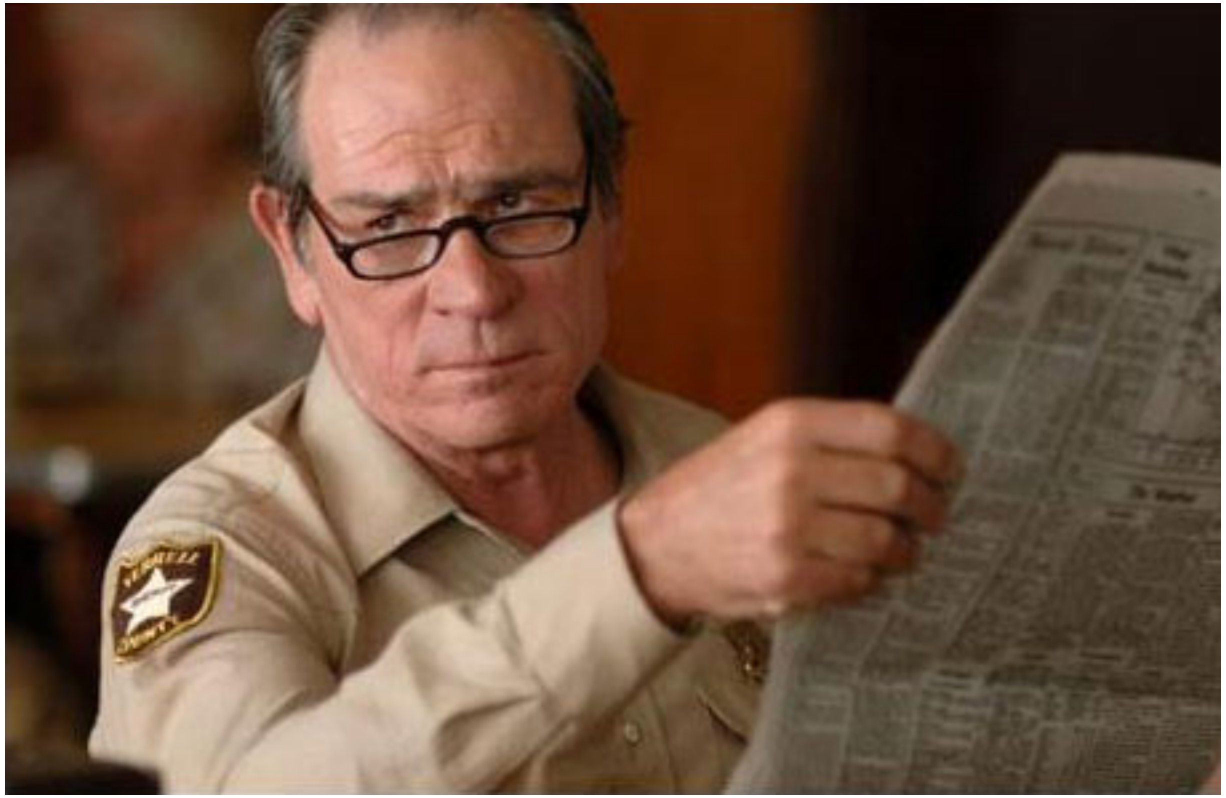
Performance and load tests

Production-time performance investigations

Continuous low-overhead monitoring

One Production Performance Investigation

- Customer pain point:
  - Very infrequently, user requests that save and load files are failing
- “Detailed” performance monitoring report (1 day):
  - Average storage array latency is 5ms, so there’s no problem
- “Super-detailed” performance monitoring report (3 days):
  - Per-minute average storage array latency numbers
  - Average of averages is 5ms, so there’s no problem



Use a Profiler

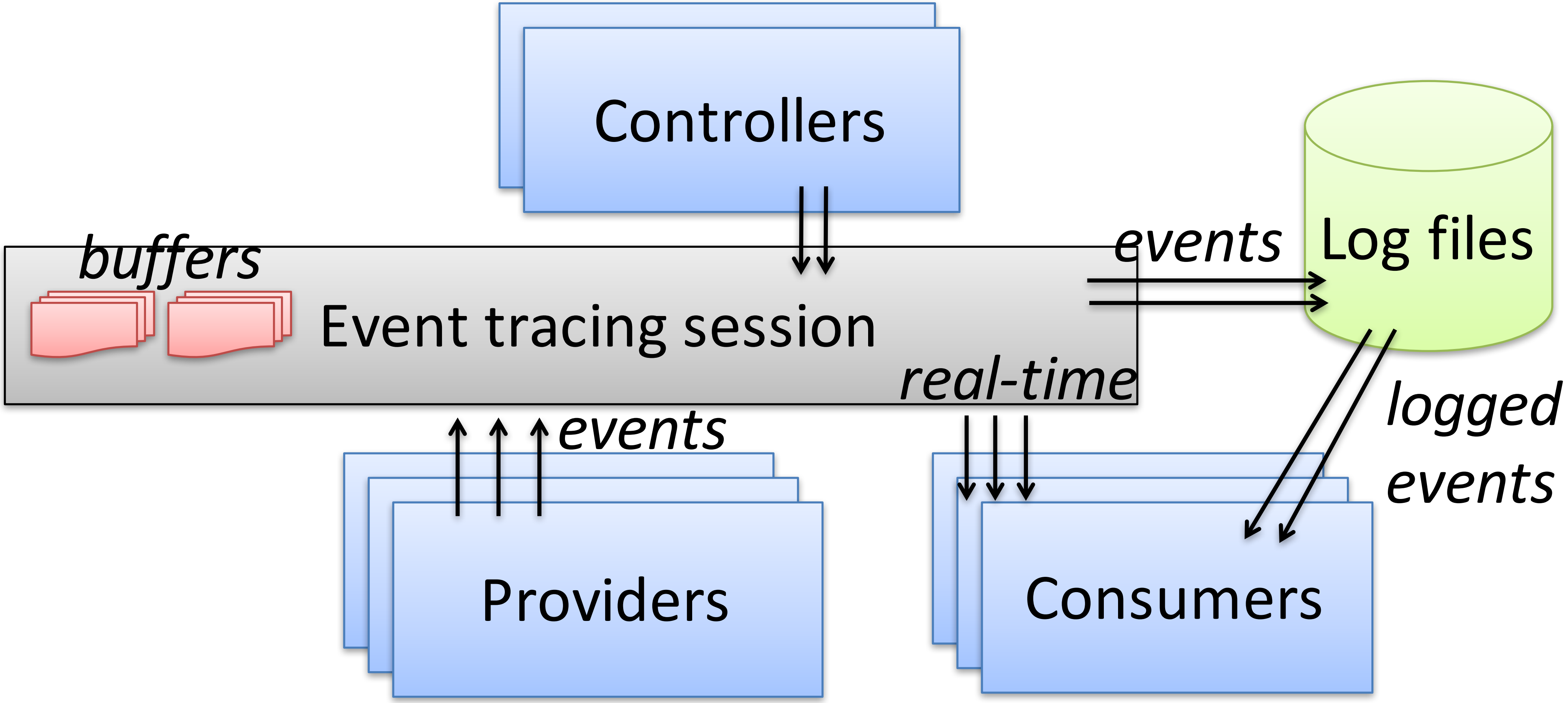
Problems with Traditional Profilers

- Invasiveness
- Overhead
- Trace size
- Licensing costs

Event Tracing for Windows (ETW)

- High-performance logging framework with structured payloads and stack trace support
- Used widely across Windows, .NET, drivers, services, and more

ETW Roles

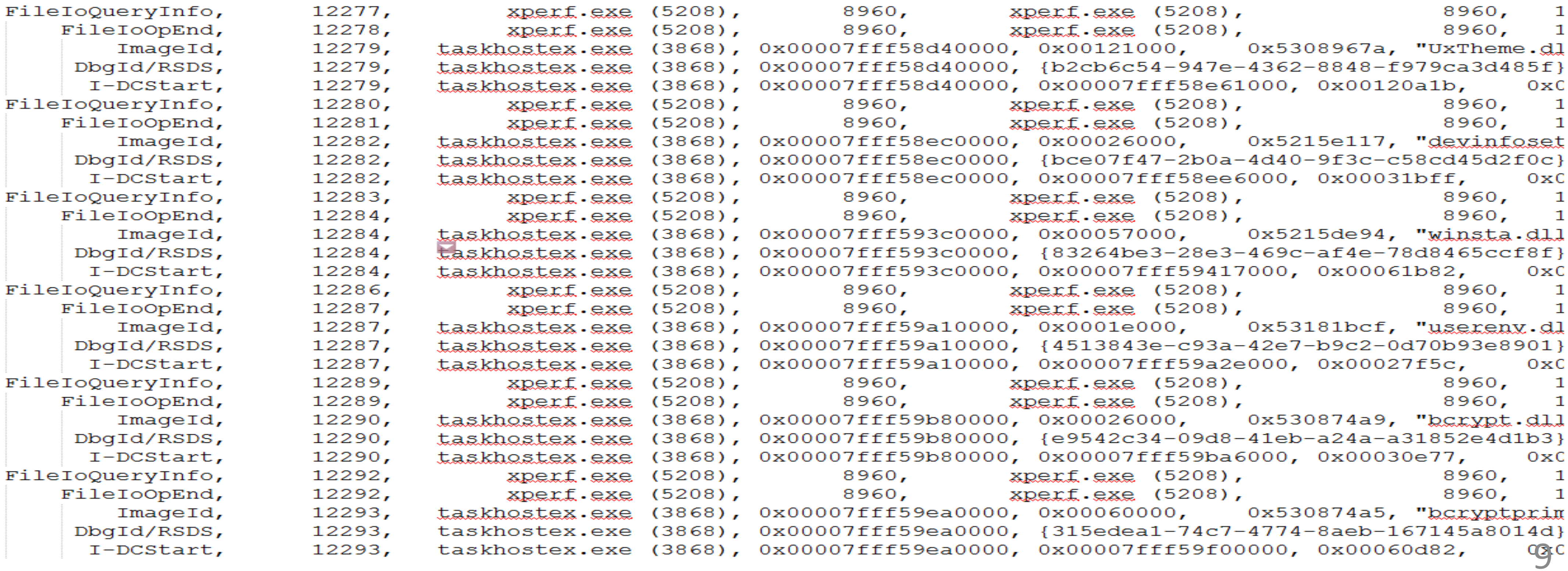

Controllers
buffers
Event tracing session
events
Log files
real-time
events
logged
events
Providers
Consumers

Some Interesting Providers

- Kernel Sample (Profile) provider
  - Produces a call stack every 1ms of CPU execution time
- CLR GC and SampAlloc provider
  - Produces events for garbage collections and object allocation samples
- Kernel Context Switch provider
  - Produces events for context switches, enables tracking off- and on-CPU transitions (thread blocked time)
- Kernel FileIO provider
  - Produces events for each file I/O operation and its duration

What’s In An ETW Recording?

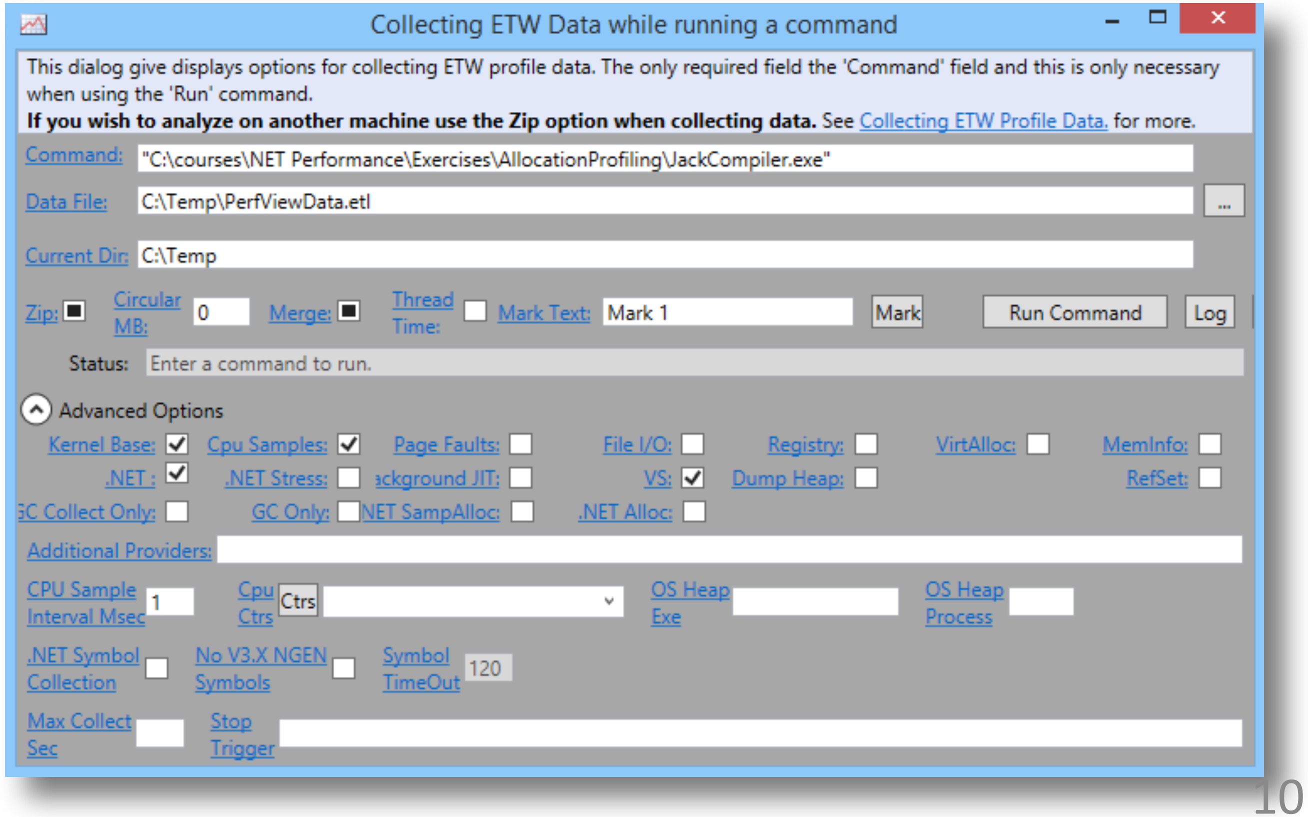
- A huge list of events, think millions
- Events have multiple columns (payload), think Excel spreadsheet
- Useless without additional processing



```
FileIoQueryInfo,      12277,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12278,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12279,    taskhostex.exe (3868), 0x00007fff58d40000, 0x00121000,     0x5308967a, "UxTheme.dl
     DbgId/RSDS,      12279,    taskhostex.exe (3868), 0x00007fff58d40000, {b2cb6c54-947e-4362-8848-f979ca3d485f}
      I-DCStart,      12279,    taskhostex.exe (3868), 0x00007fff58d40000, 0x00007fff58e61000, 0x00120a1b,    0xC
FileIoQueryInfo,      12280,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12281,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12282,    taskhostex.exe (3868), 0x00007fff58ec0000, 0x00026000,     0x5215e117, "devinfoset
     DbgId/RSDS,      12282,    taskhostex.exe (3868), 0x00007fff58ec0000, {bce07f47-2b0a-4d40-9f3c-c58cd45d2f0c}
      I-DCStart,      12282,    taskhostex.exe (3868), 0x00007fff58ec0000, 0x00007fff58ee6000, 0x00031bff,    0xC
FileIoQueryInfo,      12283,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12284,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12284,    taskhostex.exe (3868), 0x00007fff593c0000, 0x00057000,     0x5215de94, "winsta.dll
     DbgId/RSDS,      12284,    taskhostex.exe (3868), 0x00007fff593c0000, {83264be3-28e3-469c-af4e-78d8465ccf8f}
      I-DCStart,      12284,    taskhostex.exe (3868), 0x00007fff593c0000, 0x00007fff59417000, 0x00061b82,    0xC
FileIoQueryInfo,      12286,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12287,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12287,    taskhostex.exe (3868), 0x00007fff59a10000, 0x0001e000,     0x53181bcf, "userenv.dl
     DbgId/RSDS,      12287,    taskhostex.exe (3868), 0x00007fff59a10000, {4513843e-c93a-42e7-b9c2-0d70b93e8901}
      I-DCStart,      12287,    taskhostex.exe (3868), 0x00007fff59a10000, 0x00007fff59a2e000, 0x00027f5c,    0xC
FileIoQueryInfo,      12289,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12289,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12290,    taskhostex.exe (3868), 0x00007fff59b80000, 0x00026000,     0x530874a9, "bcrypt.dll
     DbgId/RSDS,      12290,    taskhostex.exe (3868), 0x00007fff59b80000, {e9542c34-09d8-41eb-a24a-a31852e4d1b3}
      I-DCStart,      12290,    taskhostex.exe (3868), 0x00007fff59b80000, 0x00007fff59ba6000, 0x00030e77,    0xC
FileIoQueryInfo,      12292,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
    FileIoOpEnd,      12292,         xperf.exe (5208),       8960,       xperf.exe (5208),               8960,   1
        ImageId,      12293,    taskhostex.exe (3868), 0x00007fff59ea0000, 0x00060000,     0x530874a5, "bcryptprim
     DbgId/RSDS,      12293,    taskhostex.exe (3868), 0x00007fff59ea0000, {315edea1-74c7-4774-8aeb-167145a8014d}
      I-DCStart,      12293,    taskhostex.exe (3868), 0x00007fff59ea0000, 0x00007fff59f00000, 0x00060d82,    0xC
```

PerfView

- ETW collection and analysis tool authored by Vance Morrison of the Microsoft .NET team
- Huge variety of performance investigation scenarios
- Steep(ish) learning curve
    - Watch training videos!



Collecting ETW Data while running a command

This dialog give displays options for collecting ETW profile data. The only required field the 'Command' field and this is only necessary when using the 'Run' command.
If you wish to analyze on another machine use the Zip option when collecting data. See Collecting ETW Profile Data. for more.

Command: "C:\courses\NET Performance\Exercises\AllocationProfiling\JackCompiler.exe"

Data File: C:\Temp\PerfViewData.etl

Current Dir: C:\Temp

Zip: ■ Circular MB: 0 Merge: ■ Thread Time: □ Mark Text: Mark 1

Status: Enter a command to run.

Advanced Options

Kernel Base: ☑ Cpu Samples: ☑ Page Faults: □ File I/O: □ Registry: □ VirtAlloc: □ MemInfo: □

.NET : ☑ .NET Stress: □ ackground JIT: □ VS: ☑ Dump Heap: □ RefSet: □

GC Collect Only: □ GC Only: □ NET SampAlloc: □ .NET Alloc: □

Additional Providers:

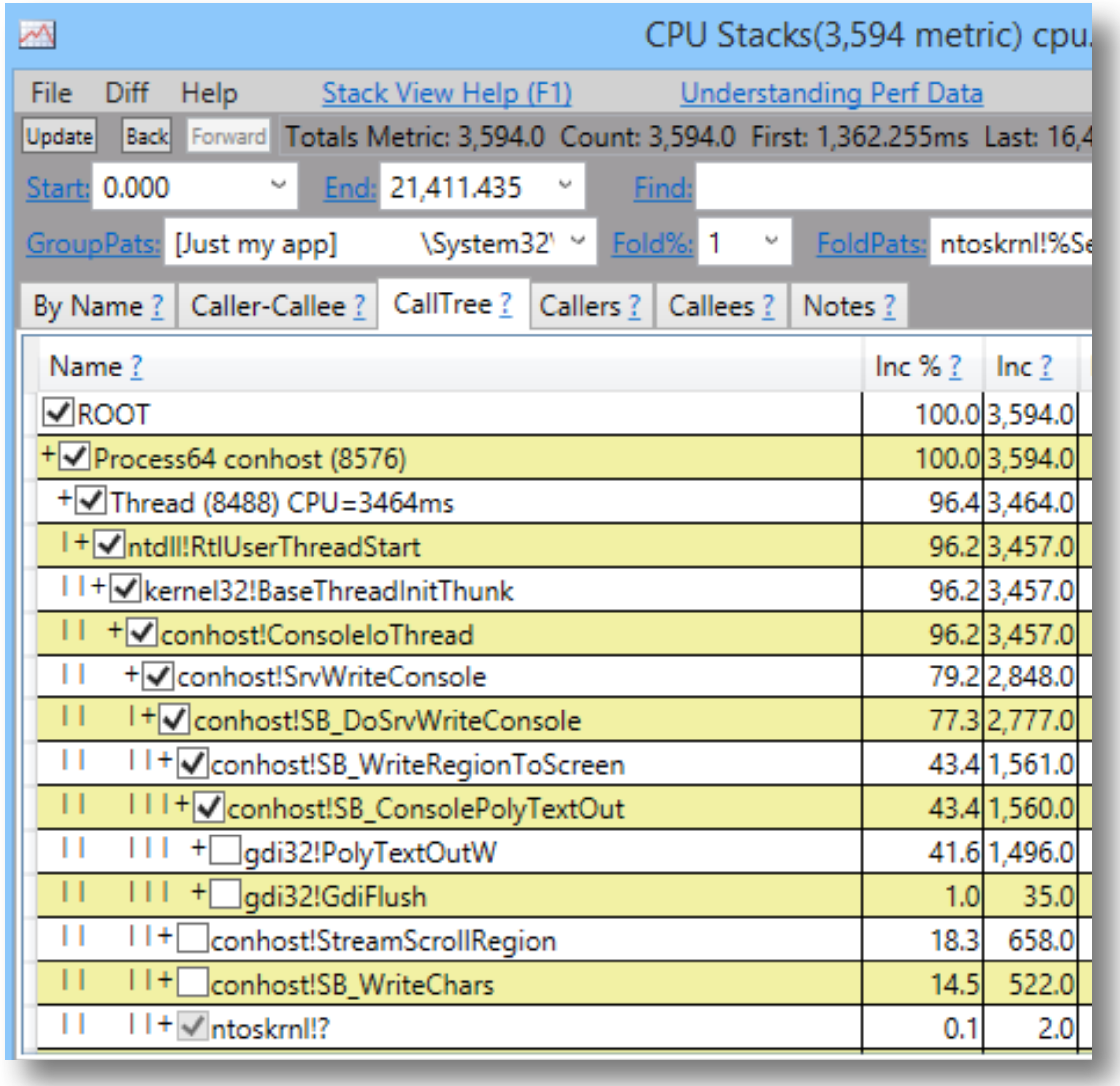
CPU Sample Interval Msec 1 Cpu Ctrs OS Heap Exe OS Heap Process

.NET Symbol Collection □ No V3.X NGEN Symbols □ Symbol TimeOut 120

Max Collect Sec Stop Trigger

Demo: Using PerfView for CPU investigations

Sidebar: It’s All In The Visualization

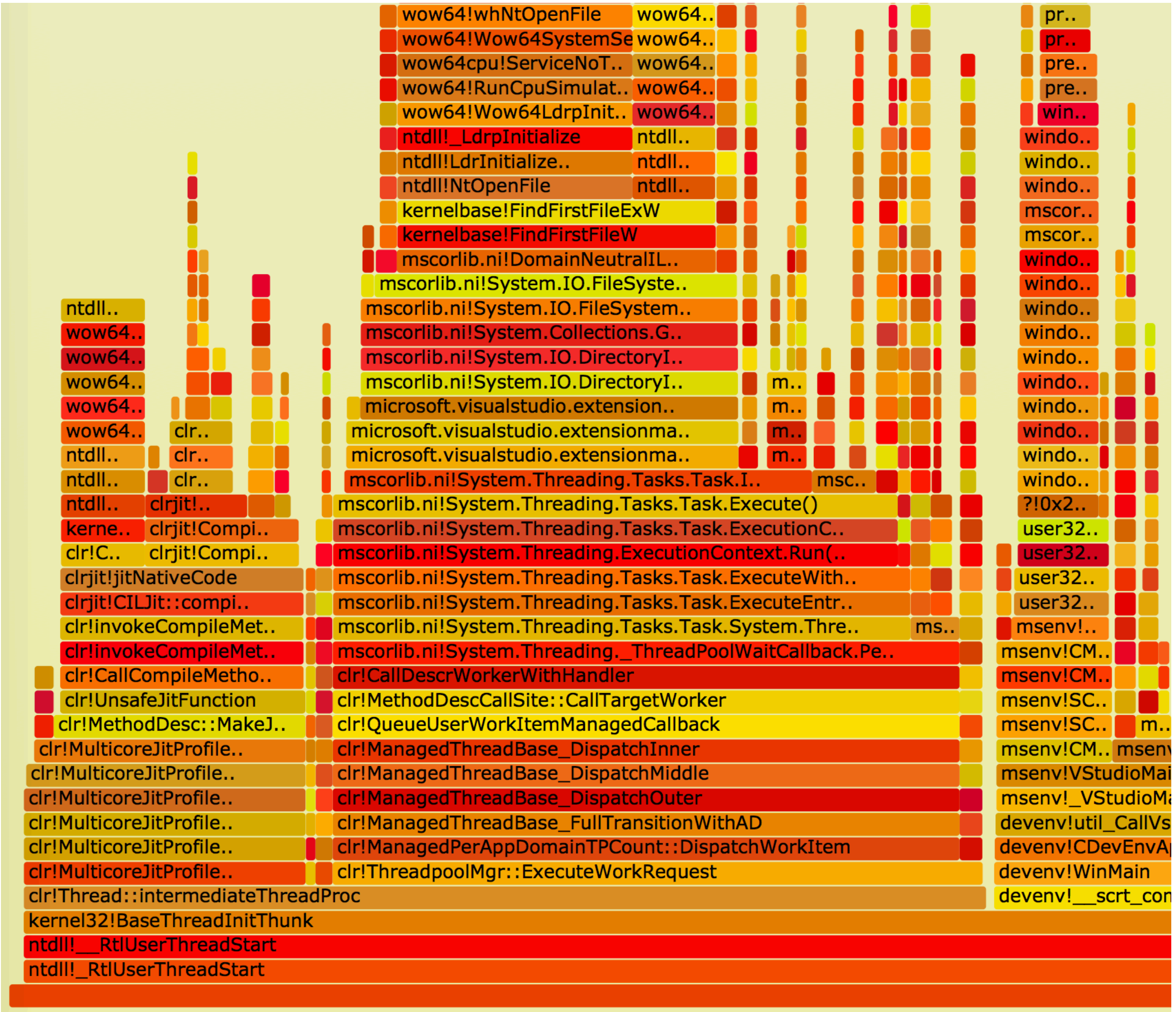

CPU Stacks(3,594 metric) cpu
File Diff Help Stack View Help (F1) Understanding Perf Data
Update Back Forward
By Name ? Caller-Callee ? CallTree ? Callers ? Callees ? Notes ?

Totals Metric: 3,594.0 Count: 3,594.0 First: 1,362.255ms Last: 16,4

Start: 0.000 End: 21,411.435 Find:

GroupPats: [Just my app] \System32' Fold%: 1 FoldPats: ntoskrnl!%Se

Name ?	Inc % ?	Inc ?
☑ROOT	100.0	3,594.0
+☑Process64 conhost (8576)	100.0	3,594.0
+☑Thread (8488) CPU=3464ms	96.4	3,464.0
|+☑ntdll!RtlUserThreadStart	96.2	3,457.0
| |+☑kernel32!BaseThreadInitThunk	96.2	3,457.0
| | +☑conhost!ConsoleIoThread	96.2	3,457.0
| | +☑conhost!SrvWriteConsole	79.2	2,848.0
| | |+☑conhost!SB_DoSrvWriteConsole	77.3	2,777.0
| | | |+☑conhost!SB_WriteRegionToScreen	43.4	1,561.0
| | | | |+☑conhost!SB_ConsolePolyTextOut	43.4	1,560.0
| | | | | +☐gdi32!PolyTextOutW	41.6	1,496.0
| | | | | +☐gdi32!GdiFlush	1.0	35.0
| | | |+☐conhost!StreamScrollRegion	18.3	658.0
| | | |+☐conhost!SB_WriteChars	14.5	522.0
| | | |+☑ntoskrnl!?	0.1	2.0



Demo: Using PerfView in continuous monitoring of ASP.NET

Demo: Using PerfView to reduce memory allocations

Memory Analysis with PerfView

- PerfView can capture optimized heap snapshots
    - Sampling, spanning tree generation, object content dropping
- Compare snapshots to determine which objects are being allocated and not freed
- Determine which GC root paths keep objects alive

Demo: Using PerfView to diagnose a memory leak

Summary

- ETW is the performance monitoring framework of today and the future
- PerfView is a free, universal analysis tool for CPU, memory, and thread time investigations
- Detailed performance monitoring in production is possible and <u>required</u>

Thank You!

Sasha Goldshtein

@goldshtn