ITDevCon

Delphi European Conference

Pleasures and Pitfalls of Profiling

Primož Gabrijelčič

October 27/28 2011 VERONA

bit Time software

What? When? How?

ITDevCon

What Is profiling?

- A form of dynamic analysis that measures some aspect of the program execution, typically:
  - Memory usage
  - Resource usage
  - Frequency and duration of function calls

"We should forget about small efficiencies, say about 97% of the time: ***premature optimization is the root of all evil.***
Yet we should not pass up our opportunities in that critical 3%."

-Donald Knuth

Tools

- “Optimization by guesswork” – bad!
- Hardcoded time measurement and logging
- Profilers

Profilers

- Sampling (statistical)
- Instrumenting
  - Source instrumenting
  - Code instrumenting
- (Event based)
- (Hypervisor)

Tools

- smartbear.com/products/development-tools/performance-profiling/
- Delphi, C++ Builder, .NET (incl. Silverlight), Java …
- Integration with RAD Studio and Visual Studio – D2006 and newer
- 32- and 64- bit
- Comes with XE and XE2 (limited version)
- $ 599

- Performance profiler
- Allocation (memory) profiler
- Coverage profiler
- Static analysis profiler
- Load library tracer profiler
- More …

- www.prodelphi.de
- Delphi 5 – XE2
- 32- and 64- bit
- Very precise profiling
- Free version (20 procedures)
- Separate Ansi and Unicode version
- Separate 32- and 64- bit version
- 50 – 90 €

Sampling Profiler

- delphitools.info/samplingprofiler
- Delphi 5 – XE (officially), works with XE2
- Measures time spent in OS DLLs
- Works at line level
- Real-time monitor
- Free

- Home-brewed timing and logging

- GetTickCount
- Now
- timeGetTime
- QueryPerformanceCounter
- RDTSC

Fixing performance problems

ITDevCon

Fixing performance problems

- Better algorithm ☺
  - Less memory allocations
  - Less string manipulations
  - Using different Windows controls
- Faster code ☹
  - Code optimization
  - Handcrafted assembler; using MMX/SSE
- Assembler tricks will not make up for bad design, however, they can make good design go faster.

Profiler FAIL

- Distributed algorithms (GUI, messaging) are hard to profile
- Optimizing the inner code of an infinite loop doesn’t help
- If time is spent in kernel, reason may be hard to find

Hands-on!

ITDevCon

Questions?

ITDevCon