

.NET Profiling

2024/11/29 dotnet new

@tanaka_733

自己紹介 @tanaka_733

- New Relic株式会社
テクニカルサポート 部部長
- **C# Tokyo**運営メンバー
- Microsoft MVP for Developer Technologies

- 最近の趣味
 - 草ソフトボール
 - ゴルフ
 - ピアノ

.NET Profilingとは

- アプリケーションのコードを開発者が変更せずに実行を監視するツール
- CLR（共通言語ランタイム）の機能
- 他の言語では(New Relicの場合)...
 - Javaは比較的似た機能 (-javaagent)で計測
 - スクリプト系言語はコードの自動書き換えで計測
 - Goでは開発者自身がコードを追加して計測

.NET Profilingの実例

- CLR Profiler
 - <https://github.com/microsoftarchive/clrprofiler>
- PerfView
 - <https://github.com/microsoft/perfview>
- OpenTelemetry .NET Automatic Instrumentation
 - <https://github.com/open-telemetry/opentelemetry-dotnet-instrumentation>
- New Relic .NET Agent
 - <https://github.com/newrelic/newrelic-dotnet-agent>
- 問題を起こすプロファイラーのサンプル
 - <https://jpdscore.github.io/blog/dotnet/common-issues-caused-by-dotnet-profiler/MyNetFxProfiler.zip>

New Relic Agentに見る、できること

- 特定のメソッドの実行時間を計測する
 - ControllerクラスであればWebの1リクエストあたりの時間
 - HttpClientやDBのクラスであれば外部通信の時間
- .NET 仮想マシンのメトリクスを取得する
- ログライブラリのAPIに介入して自動的に構造化したログイベントを記録する
- これらの動作をカスタマイズするAPIを提供する
 - APIはAgentが動作しないときは何もしない（エラーにならない）
 - 上の「特定のメソッド」を任意に指定できるようにする

.NET Profilingを有効にするには

- Profiling APIを実装したライブラリ（DLL）を用意する
- .NET プロセス起動時に環境変数を設定する
 - COR_ENABLE_PROFILING もしくは
CORECLR_ENABLE_PROFILING
 - 1を指定するとProfilingを有効にする
 - COR_PROFILER もしくは
CORECLR_PROFILER
 - プロファイラー固有のGUIDを指定
 - COR_PROFILER_PATH もしくは
CORECLR_PROFILER_PATH
 - プロファイラーのDLLのパスを指定
 - .NET Frameworkだとレジストリから検索することも可能らしい

.NET CLRの制約上、
1つのプロセスに読み込める
Profilerは高々1つ

.NET Profilerの起動の仕組み

- .NET CLRが環境変数を調べて有効になっていればDLLを見つけてロードする
- ロードしたDLLをCOMライブラリとしてインスタンス化
- 指定されたCorProfilerCallbackImplをインスタンス化
- CorProfilerCallbackImpl.Initializeを呼び出す

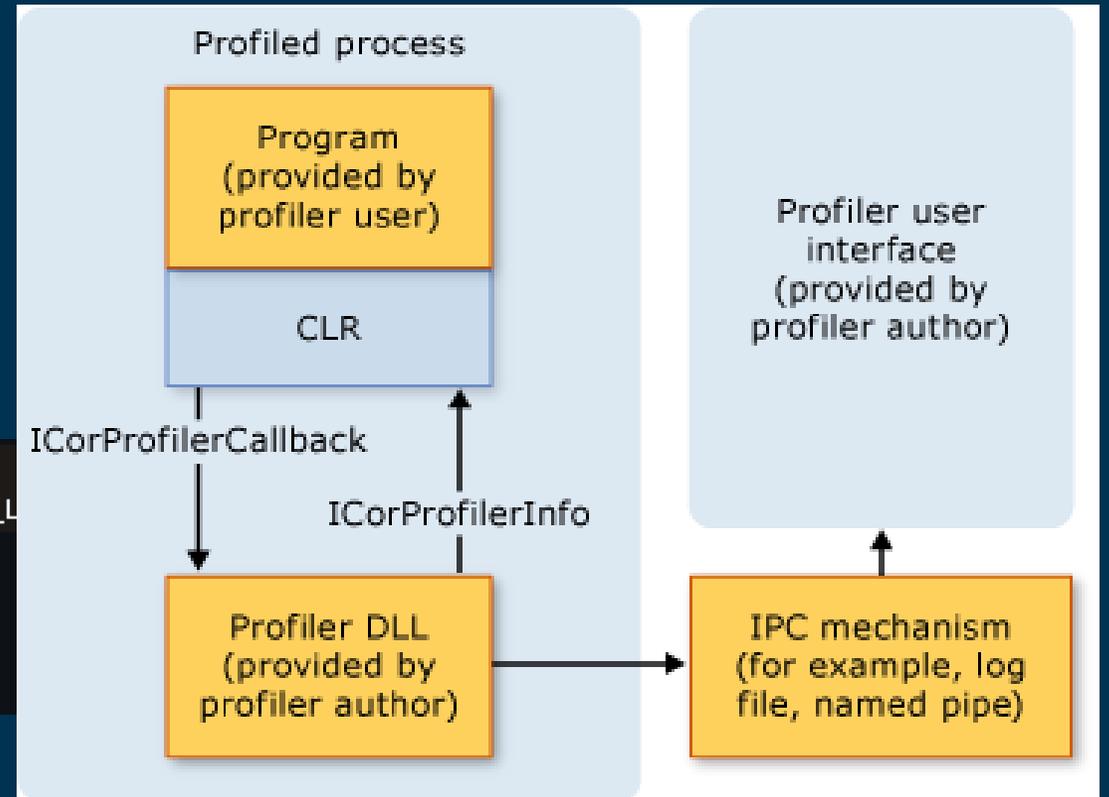
上のいずれかのプロセスで条件を満たさない（例:S_OKを返さない）とそのProfilerは解除される

情報取得

- ICorProfilerCallbackを実装
- Initializeメソッド内で購読したいイベントに応じて COR_PRF_MONITOR フラグを設定する
- ICorProfilerInfo インターフェース経由で情報を取得する

```
874     DWORD _eventMask = OverrideEventMask(  
875         COR_PRF_MONITOR_JIT_COMPILATION | COR_PRF_MONITOR_MODULE_L  
876  
877     xstring_t _productName = _X("");  
878     xstring_t _agentCoreDllPath = _X("");  
879
```

<https://github.com/newrelic/newrelic-dotnet-agent/blob/v10.34.0/src/Agent/NewRelic/Profiler/Profiler/CorProfilerCallbackImpl.h#L874-L875>



参考資料[1] より引用

.NET Profilerでコードを書き換える

- イベントが発生しJITCompilationStartedコールバックを受け取る
- 対象のメソッドをチェックし書き換えるかどうか判定する
- 書き換える場合、reJITを要求する
- reJITが開始されるとイベントが発生しコールバックを受け取る
- 元のメソッドをJITコンパイル結果したバイトコードを取得する
- 書き換えたいコードに変換する
 - **New Relic**の場合、計測するためのコードでラップする
- 変換後のコードをJITコンパイルしたバイトコードをCLRに伝える

参考資料

- [1] プロファイリングの概要
 - https://learn.microsoft.com/ja-jp/dotnet/framework/unmanaged-api/profiling/profiling-overview?WT.mc_id=AZ-MVP-5000211
- [2] .NET Framework のプロファイル機能を利用する製品に起因する一般的な問題について
 - <https://jpdscore.github.io/blog/dotnet/common-issues-caused-by-dotnet-profiler/>
- [3] Start a journey into the .NET Profiling APIs
 - <https://chnasarre.medium.com/start-a-journey-into-the-net-profiling-apis-40c76e2e36cc>
- [4] NewRelic Profiler README
 - <https://github.com/newrelic/newrelic-dotnet-agent/blob/main/src/Agent/NewRelic/Profiler/README.md>