

Bachelor's Thesis

**A 64-bit Windows Performance Profiler for Performance  
Analysis and Optimization of the BMD software NTCS**

Student: Florian Gatzweiler

Advisor: Dipl.-Ing. Dr. Markus Weninger, BSc (BMD: Dr. Horst Hagmüller)

Start date: May 2024

**Dipl.-Ing. Dr. Markus Weninger, BSc**

Institute for System Software

T +43-732-2468-4361

markus.weninger@jku.at

---

## Motivation und Ziel

Die kontinuierliche Verbesserung der Performance von Softwareanwendungen ist von entscheidender Bedeutung, insbesondere für Unternehmen wie BMD, die komplexe 64-Bit Anwendungen entwickeln.

Das Ziel dieser Arbeit ist es, ein effektives Performance-Profiling-Tool zu entwickeln, das eine detaillierte Analyse des Laufzeitverhaltens von 64-Bit-Anwendungen ermöglicht. Das Hauptziel ist es, ein Werkzeug zu schaffen, das in der Lage ist, den Prozessstack in kurzen Intervallen zu erfassen und aus Debug- und Map-Informationen die Position und Dauer der einzelnen Aktionen innerhalb der Anwendung zu bestimmen. Durch die Visualisierung dieser Daten soll ein detailliertes Bild des Programmablaufs entstehen, das es den Entwicklern ermöglicht, Performance-Engpässe zu identifizieren und zu optimieren.

## Schwerpunkte

- Im theoretischen Teil werden die bestehenden Methoden des Profilings analysiert und die Methode mittels Injektion im Vergleich zu autonomen Systemen bei der Durchführung von Leistungsanalysen untersucht.
- Entwicklung eines Performance-Profilers für 64-Bit-Anwendungen unter Windows.
- Implementierung der Funktionen zur Erfassung des Prozess-Stacks
- Echtzeitkorrelation von Codeausführung und Hardwareereignissen (CPU-/Speicherauslastung)
- Entwicklung eines Algorithmus zur Analyse der erfassten Daten und zur Erstellung einer übersichtlichen Darstellung des Programmablaufs.
- Integration einer Analyse zur Identifizierung von Stillständen und Endlosschleifen in der Anwendung.

## Anforderungen

- Das Programm sollte so entwickelt werden, dass keine zusätzliche Umgebung/Runtime (wie Java Runtime oder .NET Framework) neben dem Programm installiert werden muss. (Beispiel: Delphi/C++)
- Es wird eine leicht verständliche Benutzeroberfläche erwartet. Beim Start des Programms sollte es möglich sein, die zu analysierende Anwendung auszuwählen.
- Die Ergebnisse sollen übersichtlich und leicht verständlich dargestellt werden.

## Abgrenzung zu bestehenden Ansätzen

Im Vergleich zu bestehenden Profiling-Methoden, wie z.B. Flamegraphen, zeichnet sich das vorgeschlagene Performance-Profiler-Tool durch seine Fähigkeit aus, detaillierte Informationen über den Programmablauf zu liefern und eine direkte Korrelation zwischen dem Prozessstack und Echtzeit-Hardwareereignissen, wie z.B. CPU- und Speicherauslastung, herzustellen.

Ziel ist es, den Entwicklungsprozess zu beschleunigen und die Produktivität zu steigern, indem klare und aussagekräftige Informationen zur Verfügung gestellt werden, die direkt auf Optimierungsmaßnahmen hinweisen, ohne dass eine langwierige Einarbeitung in die Funktionsweise des Tools erforderlich ist.

Ein wesentlicher Vorteil dieses Performance-Profiler-Werkzeugs im Vergleich zu vielen existierenden Werkzeugen wie z.B. AQTime ist, dass es nicht notwendig ist, in den Code der zu analysierenden Anwendung einzugreifen.

Neben seiner Funktion als Performance-Profiler dient dieses Werkzeug auch als Fehleranalyse-Tool. Durch die detaillierte Erfassung des Prozessstacks und die Echtzeit-Korrelation von Codeausführung und Hardware-Ereignissen ermöglicht es die Identifikation von Stillständen, z.B. durch Endlosschleifen innerhalb der Anwendung.

Tool Unterschiede/Schwächen:

Flamegraphen - Detaillierte Informationen fehlen

- Keine Echtzeitkorrelation mit Hardware-Ereignissen

VTune Profiler

- Komplizierte Bedienung

Windows Performance Recorder

- Detaillierte Informationen fehlen
- Keine Echtzeitkorrelation mit Hardware-Ereignissen

AQTime & andere Performance Profiler (mit Injection)

- Betrieb mit Code Injection (Programm benötigt den Zugriff auf Source Code)

*(Angabe verfasst von BMD)*

#### Modalities:

The progress of the project should be discussed at least every four weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks and discussed with the advisor. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than 31.10.2024.