

# Oprofile/Perf

Felix Grabowski

8grabows@informatik.uni-hamburg.de  
Universität Hamburg

27. März 2012

# Gliederung

1. Was ist Profiling (Profiling)
2. Welche Rolle spielt die Hardware
  1. Allgemeine Hardware Features
  2. Hardware Performance Counter
3. OProfile
4. Perf
5. Alternativen
6. Fazit
7. Fragen?
8. Literatur

# Profiling

# Profiling

- Welche Funktionen, Anweisungen, etc. benötigen wieviel Zeit
- Wie oft werden Funktionen aufgerufen
- Welche Funktionen werden von welchen Funktionen aufgerufen (Aufrufgraph)
- Speicherverhalten (Speicherzugriffe, Cache, etc)

# Profiling

## Invasives Profiling:

- Modifizierter Programmcode
- Einfügen von Protokollfunktionen

# Profiling

## Invasives Profiling:

- Vorteile:
  - sehr genau
  - theoretisch auf instruktionszebene
  - präziser Aufrufgraph
- Nachteile:
  - hoher Overhead
  - abhängig vom messcode
  - kann keine bereits laufenden Prozesse profilieren
  - beschränkt auf Anwendungen

# Profiling

## Nichtinvasives Profiling:

- Hardware Performance Counter (CPU Feature)
- Statistische Analyse (Analyse des PC und des Stacks bei Timer Interrupt)
  - ▶ Histogramm, Sampling

# Profiling

## Nichtinvasives Profiling

- Vorteile:
  - Geringer Overhead (steuerbar)
  - Systemperformance messbar
- Nachteile:
  - Nicht exakt
  - Möglicherweise unvollständiger Aufrufgraph

# Profiling

## Profilformen:

- “Flaches Profil” (flat profile)
- “Aufrufgraph” (call graph)
- “Kommentierte Quellen” (annotated source)

# Hardware

# Hardware

## Prozessor Architektur

- kann entscheidend für Performance sein

### Gut:

- Programm verhält sich wie von den Prozessor Entwicklern erwartet

### Schlecht:

- Programm weicht signifikant von gemachten Annahmen ab

# Hardware

## Prozessor Architektur

- Caches
- Pipelining
- Superskalar Execution
- Out-of Order Execution

# Hardware

## Caches

- Zeitunterschiede
- räumliche Lokalität
- zeitliche Lokalität

# Hardware

## Pipelining

- Maschinenbefehle in Teilaufgaben zerlegt
- dadurch fast parallele Ausführung

# Hardware

## Superscalar Execution

- Erweiterung des Pipelining
- mehrere Ausführungs Einheiten
  - mehrere Instruktionen pro Zyklus
  - mit pipelining → mehrere Instruktionsgruppen

# Hardware

## Out-of-Order Execution

- Annahme sequentielle Ausführung
- Ausführung erscheint sequentiell

Hardware kann Ausführungsreihenfolge verändern

# Hardware

## Branch Prediction & Instruction Speculation

# Hardware

## Branch Prediction

- versucht Ziele von Verzweigungen vorherzusagen
- Prozessor kann mit ausführen beginnen bevor das Ziel tatsächlich bekannt ist
- funktioniert gut bei einfachen sich wiederholenden Mustern
- indirekte Verzweigungen Problematisch

# Hardware

## Instruction Speculation

- mögliche Pfade im voraus berechnen  
oder gar alle Verzweigungen
- falsche Berechnungen werden verworfen
- Nachteil:  
knappe Ressourcen könnten noch stärker eingeschränkt werden

# Hardware

## Performance Counter

- special purpose Register
- für jede CPU unterschiedlich
- zählen Hardware Ereignisse
- nicht Invasiv

# OProfile

- Basiswissen
- Setup
- Benutzung
- Vorteile/Nachteile

# OProfile

## Basiswissen

- Systemweiter Profiler
- Besteht aus zwei Teilen:
  - Kernel Modul
  - Userspace Daemon
- Verwendet Hardware Performance Counter!
- Gut für High Performance Computing
- Es gibt eine GUI
- Linux only

# OProfile

## Setup

- **Installation:**  
`sudo apt-get install oprofile`
- **vmlinux (optional)**
- **debugging symbole**

# OProfile

## Benutzung

### Befehle:

- `opannotate`
- `opcontrol`
- `ophelp`
- `opimport`
- `opreport`

# OProfile

## Benutzung

1. Program optional mit `-g` Option Kompilieren

2. Profiler starten (Rootrechte werden benötigt)

```
opcontrol --vmlinux=/path/to/vmlinux
```

```
opcontrol --start
```

3. Tests durchführen

4. Bericht generieren

```
opcontrol --stop
```

```
opcontrol --dump
```

```
opcontrol -l /path/to/binary
```

# OProfile

## Benutzung

### 4. Bericht generieren:

```
opcontrol --stop
```

```
opcontrol --dump
```

```
opcontrol -l /path/to/binary
```

**oder wenn bei -g Option:**

```
opcontrol --source --output-dir=/path/to/annotated/source /path/to/binary
```

**oder einen allgemeinen Bericht:**

```
opreport
```

### 5. Daemon beenden

```
opcontrol --shutdown
```

# OProfile

## Benutzung

### Konfigurationsmöglichkeiten:

```
opcontrol --list-events
```

```
opcontrol --setup
```

```
opcontrol -c / oprofile -c
```

```
oprofile --assembly
```

### Bsp:

```
opcontrol --setup --no-vmlinux \
```

```
--seperate-library \
```

```
--event=GLOBAL_POWER_EVENTS:700000:0x1:1:1
```

```
opcontrol --start
```

# OProfile

Benutzung

Vorführung

# OProfile

## Vorteile/Nachteile

### Vorteile:

- geringer Overhead durch Hardware Interrupts
- Kein Neukompilieren nötig
- Kann Hardware Effekte wie Cache Misses messen
- Kann Quellcode kommentieren
- Profil für jedes Binary seperat
- Gut für Systeme die man nicht unterbechen möchte

# OProfile

## Vorteile/Nachteile

### Nachteile:

- benötigt Rootrechte
- kann ein Sicherheitsrisiko darstellen
- Ereignisse und Instruktionen nicht unbedingt korreliert

# Perf

- Basiswissen
- Setup
- Benutzung
- Vorteile/Nachteile

# Perf

## Basiswissen

- Basiert auf dem `perf_events` Interface aktueller Linux Kernel
- Simple Command Line Interface
- Verwendet ebenfalls Performance Counter

# Perf

## Setup

- Benötigt die Pakete

linux-tools-common

linux-tools-3.0.0-16

# Perf

## Benutzung

- **Generisches Perf Programm**

- **Befehle:**

stat, record, report,...

- **Verwendung:**

perf record ./binary |

perf list

perf stat -e cache-misses [...]

perf top

# Perf

## Benutzung

# Beispielausgaben

# Perf

## Vorteile/Nachteile

### Vorteile

- Offizieller Profiler für den Linux Kernel
- keine Konfiguration nötig

# Alternativen

- Intel VTune (Linux, Windows)
- Instruments (OSX)
- Viele Andere mehr oder weniger gute

# Zusammenfassung

- **Grundlagen**
  - Profiling: Invasiv vs. nicht-Invasiv
  - Hardware: Prozessorarchitektur, Performance Counter
- **OProfile/Perf:**
  - Vorstellung von OProfile/Perf
  - Herausstellung der Vor- und Nachteile
  - Praxisbeispiel OProfile
  - Beispielausgaben von Perf

# Fazit

- Wenn Hardware unterstützt wird nützlich
- Leider wird nicht jede Hardware unterstützt
- Beide Profiler gut um Bottlenecks aufzuspüren

# OProfile/Perf

Felix Grabowski

[8grabows@informatik.uni-hamburg.de](mailto:8grabows@informatik.uni-hamburg.de)

Universität Hamburg

27. März 2012

# Quellen/Literatur

- <http://courses.engr.illinois.edu/ece498/SL/S11-archive/lectures/ece498sl-lecture-13.pdf> 19. 20. - März 2012 13:02
- <http://de.wikipedia.org/wiki/Pipeline-Architektur> - 19. März 2012 11:52
- <http://software.intel.com/en-us/articles/using-intel-vtune-performance-analyzer-events-ratios-optimizing-applications/> - 13. März 2012 16:16
- <http://www.osreviews.net/reviews/devel/oprofile> - 29. Februar 2012 23:21
- <http://kenshin579.tistory.com/entry/XenOprofile-OProfile> - 29. Februar 2012 22:59
- <http://www.dedoimedo.com/computers/oprofile.html> - 29. Februar 2012 22:51
- <http://blogs.epfl.ch/article/24223> - 29. Februar 2012 22:45
- <http://people.redhat.com/wcohen/Oprofile.pdf> - 1. März 2012 20:01
- <http://kwangwoo.blogspot.com/2011/11/performance-monitoring-on-arm.html> - 7. März 2012 12:09
- <http://kwangwoo.blogspot.com/2011/07/profiling-tools-fttrace-perf-and.html> - 7. März 2012 12:09
- <http://oprofile.sourceforge.net> - 12. März 2012 14:45
- <http://ssvb.github.com/2011/08/23/yet-another-oprofile-tutorial.html> - 12. März 2012 15:40