Massif-Visualizer
Memory Profiling UI

Milian Wolff
mail@milianw.de
Desktop Summit 2011
08.08.2011
Agenda

1. Valgrind
2. Massif
3. Massif-Visualizer
1  Valgrind

2  Massif

3  Massif-Visualizer
1 Valgrind
   ■ Introduction
   ■ Tool Suite
   ■ Usage
   ■ Caveats
Valgrind Introduction

- **Versatile:** currently nine debugging and profiling utilities
- **Open:** licensed under the GPL
- **Proven:** under active development since 2000
- **Cross Platform:** Linux and MacOS X
- **Cross Platform:** x86, AMD64, ARM, …
Valgrind Introduction

- **Versatile:** currently nine debugging and profiling utilities
- **Open:** licensed under the GPL
- **Proven:** under active development since 2000
- **Cross Platform:** Linux and MacOS X
- **Cross Platform:** x86, AMD64, ARM, ...
Valgrind Introduction

- **Versatile**: currently nine debugging and profiling utilities
- **Open**: licensed under the GPL
- **Proven**: under active development since 2000
  - **Cross Platform**: Linux and MacOS X
  - **Cross Platform**: x86, AMD64, ARM, …
Valgrind Introduction

- **Versatile**: currently nine debugging and profiling utilities
- **Open**: licensed under the GPL
- **Proven**: under active development since 2000
- **Cross Platform**: Linux and MacOS X
- **Cross Platform**: x86, AMD64, ARM, ...
Valgrind Introduction

- **Versatile**: currently nine debugging and profiling utilities
- **Open**: licensed under the GPL
- **Proven**: under active development since 2000
- **Cross Platform**: Linux and MacOS X
- **Cross Platform**: x86, AMD64, ARM, ...
1 Valgrind
   - Introduction
   - Tool Suite
   - Usage
   - Caveats
Tool Suite

- **Memcheck**: detect errors in memory management
  leaks, invalid read/writes, ...

- **Helgrind / DRD**: detect errors in multithreading
  race conditions, lock contention, ...

- **Callgrind**: CPU cache profiling
  hotspots, bottlenecks, cache issues, ...

- **Massif**: memory profiling
  total consumption, leaks, peaks, ...

- ...
Tool Suite

- **Memcheck**: detect errors in memory management leaks, invalid read/writes, . . .

- **Helgrind / DRD**: detect errors in multithreading race conditions, lock contention, . . .

- **Callgrind**: CPU cache profiling hotspots, bottlenecks, cache issues, . . .

- **Massif**: memory profiling total consumption, leaks, peaks, . . .

- . . .
Tool Suite

- **Memcheck**: detect errors in memory management
  - leaks, invalid read/writes, . . .

- **Helgrind / DRD**: detect errors in multithreading
  - race conditions, lock contention, . . .

- **Callgrind**: CPU cache profiling
  - hotspots, bottlenecks, cache issues, . . .

- **Massif**: memory profiling
  - total consumption, leaks, peaks, . . .

- . . .
Tool Suite

- **Memcheck**: detect errors in memory management
  - leaks, invalid read/writes, ...

- **Helgrind / DRD**: detect errors in multithreading
  - race conditions, lock contention, ...

- **Callgrind**: CPU cache profiling
  - hotspots, bottlenecks, cache issues, ...

- **Massif**: memory profiling
  - total consumption, leaks, peaks, ...

...
Tool Suite

- **Memcheck**: detect errors in memory management
  leaks, invalid read/writes, ...

- **Helgrind / DRD**: detect errors in multithreading
  race conditions, lock contention, ...

- **Callgrind**: CPU cache profiling
  hotspots, bottlenecks, cache issues, ...

- **Massif**: memory profiling
  total consumption, leaks, peaks, ...

- ...
1 Valgrind
- Introduction
- Tool Suite
- Usage
- Caveats
### Usage

- no special compilation flags required
- but: debug symbols very useful
- running valgrind:
  ```bash
  valgrind --tool=... [tool options]
  myApp [your options]
  ```
- errors are outputted to CLI
  → human readable
- profiling data written to file
  → visualization UI required
Usage

- no special compilation flags required
- but: debug symbols very useful
  - running valgrind:
    ```
    valgrind --tool=... [tool options]
    myApp [your options]
    ```
  - errors are outputted to CLI → human readable
  - profiling data written to file → visualization UI required
Usage

- no special compilation flags required
- but: debug symbols very useful
- running valgrind:
  `valgrind --tool=... [tool options] myApp [your options]`
- errors are outputted to CLI
  → human readable
- profiling data written to file
  → visualization UI required
Usage

- no special compilation flags required
- but: debug symbols very useful
- running valgrind:
  ```
  valgrind --tool=... [tool options]
  myApp [your options]
  ```
- errors are outputted to CLI
  \[\rightarrow\] human readable
- profiling data written to file
  \[\rightarrow\] visualization UI required
Usage

- no special compilation flags required
- but: debug symbols very useful
- running valgrind:
  valgrind --tool=... [tool options]
  myApp [your options]
- errors are outputted to CLI
  → human readable
- profiling data written to file
  → visualization UI required
1 Valgrind

- Introduction
- Tool Suite
- Usage
- Caveats
Caveats

- slowdown by factor 5 to 100
- increased memory consumption
- interpretation of results requires knowledge and experience
Caveats

- slowdown by factor 5 to 100
- increased memory consumption
- interpretation of results requires knowledge and experience
Caveats

- slowdown by factor 5 to 100
- increased memory consumption
- interpretation of results requires knowledge and experience
1. Valgrind

2. Massif

3. Massif-Visualizer
2 Massif
  ■ Introduction
  ■ Output
  ■ Useful Options
  ■ ms_print
Massif Introduction

- mainly aimed at heap profiling
- optionally also stack profiling possible
- tracks memory allocations and deallocations
- caveats: slowdown by 20x, high memory overhead
Massif Introduction

- mainly aimed at heap profiling
- optionally also stack profiling possible
  - tracks memory allocations and deallocations
  - caveats: slowdown by 20x, high memory overhead
Massif Introduction

- mainly aimed at heap profiling
- optionally also stack profiling possible
- tracks memory allocations and deallocations
- caveats: slowdown by 20x, high memory overhead
Massif Introduction

- mainly aimed at heap profiling
- optionally also stack profiling possible
- tracks memory allocations and deallocations
- caveats: slowdown by 20x, high memory overhead
2 Massif

- Introduction
- Output
- Useful Options
- ms_print
Output

snapshots of memory consumption over time

- simple snapshots: total memory consumption
- detailed snapshots: backtraces to significant memory allocations
snapshots of memory consumption over time

- simple snapshots: total memory consumption
- detailed snapshots: backtraces to significant memory allocations
Output

snapshots of memory consumption over time

- simple snapshots: total memory consumption
- detailed snapshots: backtraces to significant memory allocations
2 Massif

- Introduction
- Output
- Useful Options
- ms_print
Useful Options

- `--max-snapshots=N` [Default: 100]
  increase for long running profiling sessions

- `--detailed-freq=N` [Default: 10]
  decrease for short running, increase for long running profiling sessions

- `--threshold=<m.n>` [Default: 1.0]
  increase in case your app mainly consists of allocations below the default threshold

- Output: `--massif-out-file=...`
  [Default: massif.out.%p]
Useful Options

- **--max-snapshots=N** [Default: 100]
  increase for long running profiling sessions

- **--detailed-freq=N** [Default: 10]
  decrease for short running, increase for long running profiling sessions

- **--threshold=<m.n>** [Default: 1.0]
  increase in case your app mainly consists of allocations below the default threshold

- **Output: --massif-out-file=...**
  [Default: massif.out.%p]
Useful Options

- **--max-snapshots=N** [Default: 100]
  increase for long running profiling sessions

- **--detailed-freq=N** [Default: 10]
  decrease for short running, increase for long running profiling sessions

- **--threshold=<m.n>** [Default: 1.0]
  increase in case your app mainly consists of allocations below the default threshold

Output: **--massif-out-file=...**
  [Default: massif.out.%p]
Useful Options

- `--max-snapshots=N` [Default: 100]
  increase for long running profiling sessions

- `--detailed-freq=N` [Default: 10]
  decrease for short running, increase for long running profiling sessions

- `--threshold=<m.n>` [Default: 1.0]
  increase in case your app mainly consists of allocations below the default threshold

- Output: `--massif-out-file=...`
  [Default: massif.out.%p]
2 Massif

- Introduction
- Output
- Useful Options
- ms_print
ms_print Visualization

Command: kate .xsession-errors
Massif arguments: --detailed-freq=1
ms_print arguments: massif.out.kate

ms_print massif.out.PID | less
ms_print Visualization

```
ms_print massif.out.PID | less
```
1 Valgrind

2 Massif

3 Massif-Visualizer
3 Massif-Visualizer
- Massif-Visualizer
  - Features
  - Future Ideas
  - Packages
  - Resources
Massif-Visualizer

massif-visualizer massif.out.PID
Massif-Visualizer

interactive!
3 Massif-Visualizer
  - Massif-Visualizer
  - Features
  - Future Ideas
  - Packages
  - Resources
Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
- hiding functions
- shorten templates
- ...

Milian — Massif-Visualizer — Desktop Summit — Berlin
Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
- hiding functions
- shorten templates
- . . .
Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
  - hiding functions
  - shorten templates
- ...
Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
- hiding functions
  - shorten templates
- ...

Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
- hiding functions
- shorten templates
- . . .
Features

- vis. evolution of mem consumption
- vis. backtrace/callgraph
- custom allocators
- hiding functions
- shorten templates
- ...
3 Massif-Visualizer
   - Massif-Visualizer
   - Features
   - Future Ideas
   - Packages
   - Resources
Future Ideas & Todo’s

- Comparison/Diff of Massif data files
- Top-Down view of backtraces (subtree matching)
- Export Graphs
Future Ideas & Todo’s

- Comparison/Diff of Massif data files
- Top-Down view of backtraces (subtree matching)
- Export Graphs
Future Ideas & Todo’s

- Comparison/Diff of Massif data files
- Top-Down view of backtraces (subtree matching)
- Export Graphs
3 Massif-Visualizer
- Massif-Visualizer
- Features
- Future Ideas
- Packages
- Resources
Packages

- Arch, Slackware, ... have packages
- OpenSuse BuildService:
  http://kde-apps.org/content/show.php?content=122409
- MacOS X:
  https://trac.macports.org/ticket/27168
- Generally: **Please help with packaging!**
Packages

- Arch, Slackware, ... have packages
- OpenSuse BuildService:
  http://kde-apps.org/content/show.php?content=122409
- MacOS X:
  https://trac.macports.org/ticket/27168
- Generally: Please help with packaging!
Packages

- Arch, Slackware, ... have packages
- OpenSuse BuildService:
  http://kde-apps.org/content/show.php?content=122409
- MacOS X:
  https://trac.macports.org/ticket/27168
- Generally: Please help with packaging!
Packages

- Arch, Slackware, ... have packages
- OpenSuse BuildService:
  http://kde-apps.org/content/show.php?content=122409
- MacOS X:
  https://trac.macports.org/ticket/27168
- Generally: Please help with packaging!
3 Massif-Visualizer
- Massif-Visualizer
- Features
- Future Ideas
- Packages
- Resources
Resources

- Project Page:
  http://projects.kde.org/massif-visualizer

- Bugs and Wish Requests:
  http://bugs.kde.org

- Mailing List:
  massif-visualizer@kde.org
  https://mail.kde.org/mailman/listinfo/massif-visualizer

- KDE-Apps Page:
  http://kde-apps.org/content/show.php?content=122409
Questions?