Profiling CPU and memory on Linux, with opensource graphical tools

Embedded Linux Conference and Open Source Summit, Lyon, October 29, 2019

Presented by David Faure
Heaptrack

A heap memory profiler for Linux

- In-depth information about allocation patterns:
  - Counts allocations and finds temporary allocations
  - Aggregates requested memory sizes
  - Backtraces for every allocation
- Less overhead than Valgrind’s massif
- Supports runtime-attaching
- Works on any Linux, independent of architecture

Caveats:
- Still significant overhead for every allocation
- Debug symbols are resolved during recording

[GitHub](https://github.com/KDE/heaptrack)
- No need to compile anything, just use the AppImage:

[GitHub](https://github.com/KDAB/heaptrack/releases/tag/continuous)

Heaptrack: Building

Building heaptrack from sources

- Required Dependencies:
  - C++11 enabled GCC/Clang
  - libunwind (preferably from git master for performance reasons)
  - elfutils, esp. dwarf.h
  - Boost 1.41 or higher

- Optional dependencies for heaptrack_gui:
  - Qt 5
  - KF5 & KChart
  - zstd for faster (de)-compression

- CMake build process:

```
1. git clone git://anongit.kde.org/heaptrack
2. mkdir build-heaptrack
3. cd build-heaptrack
4. cmake ../heaptrack -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=...
5. make install
```

Heaptrack: Recording Data

Record heap profile data using heaptrack with:

```
1. $ heaptrack ./ex_string_comparison
2. heaptrack output will be written to "heaptrack.ex_string_comparison.24590.gz"
3. starting application, this might take some time...
```

```
5. heaptrack stats:
6. allocations: 1001145
7. leaked allocations: 16
8. temporary allocations: 1000003
```

Or attach to a running process:

```
$ heaptrack --pid $(pidof <your application>)
# Ctrl + C after some time to detach
```

Visualize the profile data using heaptrack_gui or heaptrack_print.

```
$ heaptrack_gui heaptrack.APP_PID.gz
$ heaptrack_print heaptrack.APP_PID.gz
```

Demo: profiling/ex_string_comparison
Heaptrack: Diffing Results

Use the diff mode to compare data files.

- Supported by both heaptrack_print and heaptrack_gui
- Usage: `heaptrack.FIRST.gz heaptrack.SECOND.gz`

```
$ heaptrack_print -d heaptrack.ex_string_comparison.24590.gz heaptrack.ex_string_comparison.22087.gz
...
MOST TEMPORARY ALLOCATIONS
-1000000 temporary allocations of -1000000 allocations in total (100%) from
QArrayData::allocate(unsigned long, unsigned long, unsigned long, QFlags<>)
in /usr/lib/libQt5Core.so.5
-1000000 temporary allocations of -1000000 allocations in total (100%) from:
QString::QString(int, Qt::Initialization)
in /usr/lib/libQt5Core.so.5
0x7fb2b20189cb
QString::fromUtf8_helper(char const*, int)
in /usr/lib/libQt5Core.so.5
QString::fromAscii_helper(char const*, int)
in /usr/lib/libQt5Core.so.5
main
at ../../ex_string_comparison/ex_string_comparison.cpp:15
in /path/to/ex_string_comparison/ex_string_comparison
```

Heaptrack: Pros and Cons

Pros:
- Fast heap memory profiling
- Tracks number of (temporary) allocations
- Can attach to a running process
- Supports diffing of results

Cons:
- Only available on Linux
- Incomplete support for cross-machine analysis
- GUI features around charts and timeline needs to be extended

Heaptrack

Demo: profiling/ex_string_comparison

Hotspot

GUI to replace the common perf report workflow.

- R&D project by KDAB, available on GitHub: github.com/KDAB/hotspot.
- Depends on elfutils, Qt 5.6 and some KDE Frameworks.
- Notable features:
  - Easier to use, no arcane command line switches.
  - Tries to give context sensitive information.
  - Good support for the common embedded workflow, i.e. record on ARM without debug symbols, report on x86-64 with sysroot and debug symbols.
  - Shows multiple cost types side-by-side.
- Does not support many of the more advanced perf features.
Hotspot demo

- Profiling on-CPU time (cycles spent)
- Profiling off-CPU time (thread blocked waiting)

Conclusion

- Download, compile, and try Heaptrack and Hotspot
- Book KDAB's 3 days Debugging and Profiling training
- Questions?
- If you think of one later, email me: david.faure@kdab.com