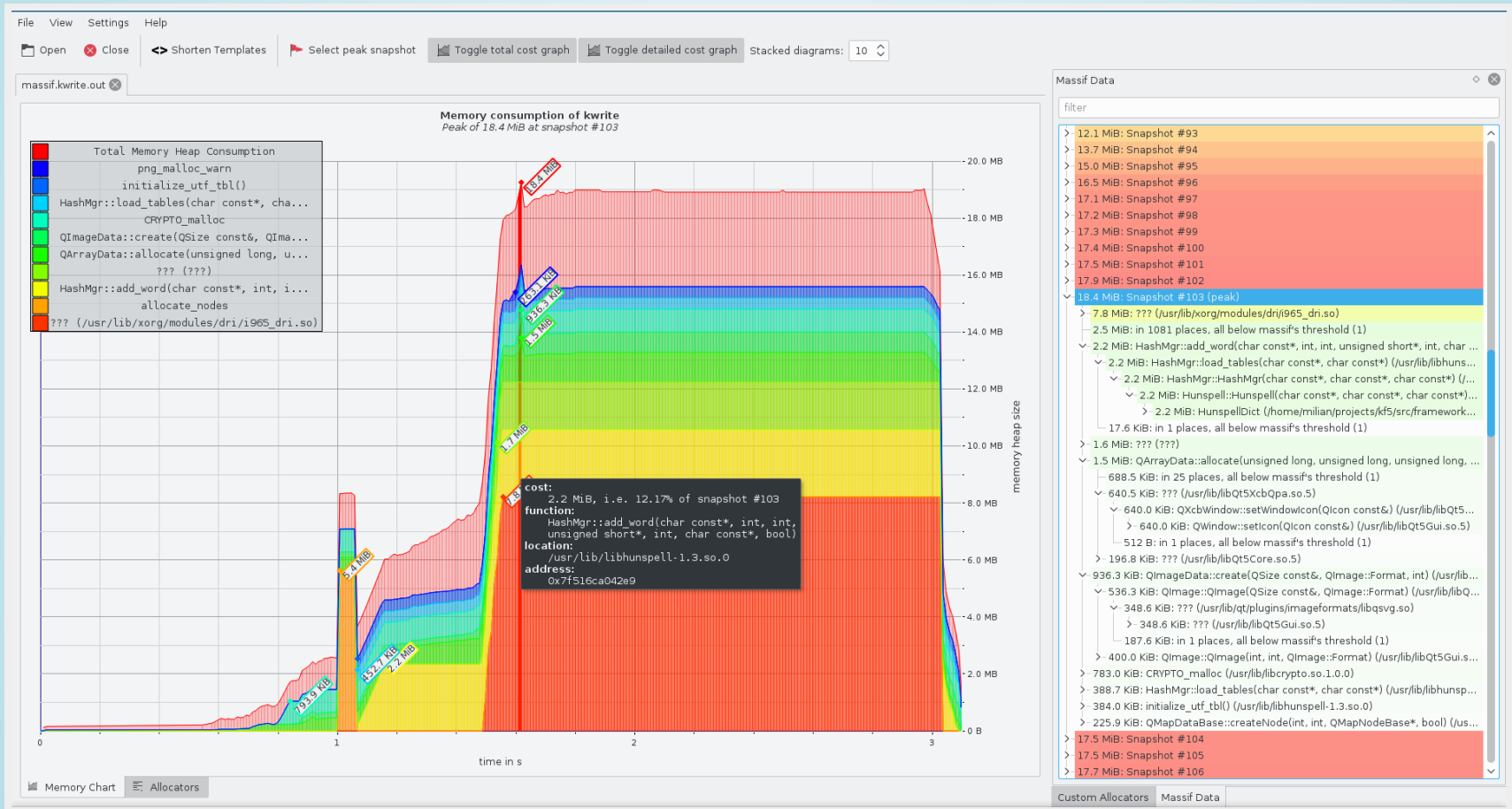


# HEAPTRACK

A HEAP MEMORY PROFILER FOR LINUX

Milian Wolff / [www.kdab.com](http://www.kdab.com)

# MEMORY PROFILING



Valgrind's massif tool & Massif-Visualizer

# THE GOAL

- a faster massif
- collect more data

# THE SOLUTION

```
$ heaptrack someapp
starting application, this might take some time...
output will be written to /home/milian/heaptrack.someapp.1234.gz
...
Heaptrack finished! Now run the following to investigate the data:

  heaptrack_gui /home/milian/heaptrack.someapp.1234.gz | less
```

# ANALYSIS

**debuggee:** kwrite

**total runtime:** 3.097s

**bytes allocated in total** (ignoring deallocations): 78.88 MiB (25.5 MiB/s)

**calls to allocation functions:** 433122 (139852/s)

**peak heap memory consumption:** 18.4 MiB

**total memory leaked:** 1.2 MiB

Bottom-Up Top-Down Leaked Allocations Allocated Flame Graph

filter by function...

filter by file...

filter by module...

Allocations [-]	Peak [B]	^	Leaked [B]	Allocated [B]	Location
>- 9	11248056	0		16872084	allocate_nodes in ?? (/usr/lib/libhspell.so.0)
>- 35281	8464128	0		17194320	0x7f516c2b1ce0 in ?? (/usr/lib/xorg/modules/dri/i965_dri.so)
>- 136324	5197339	20435		15632136	QArrayData::allocate(unsigned long, unsigned long, unsigned long, QFlags<QA...
▼ 49150	2343968	0		2345290	HashMgr::add_word(char const*, int, int, unsigned short*, int, char const*, boo...
▼ 48743	2325963	0		2327233	HashMgr::load_tables(char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.s...
▼ 48743	2325963	0		2327233	HashMgr::HashMgr(char const*, char const*, char const*) in ?? (/usr/lib/libhun...
>- 48...	2325963	0		2327233	Hunspell::Hunspell(char const*, char const*, char const*) in ?? (/usr/lib/libhuns...
▼ 407	18005	0		18057	HashMgr::add_hidden_capitalized_word(char*, int, int, unsigned short*, int, ch...
>- 407	18005	0		18057	HashMgr::load_tables(char const*, char const*) in ?? (/usr/lib/libhunspell-1.3.s...
>- 11422	2000648	0		2000648	0x7f516c2b1ea1 in ?? ()

debuggee: kwrite  
total runtime: 3.097s  
bytes allocated in total (ignoring deallocations): 78.88 MiB (25.5 MiB/s)  
calls to allocation functions: 433122 (139852/s)  
peak heap memory consumption: 18.4 MiB  
total memory leaked: 1.2 MiB

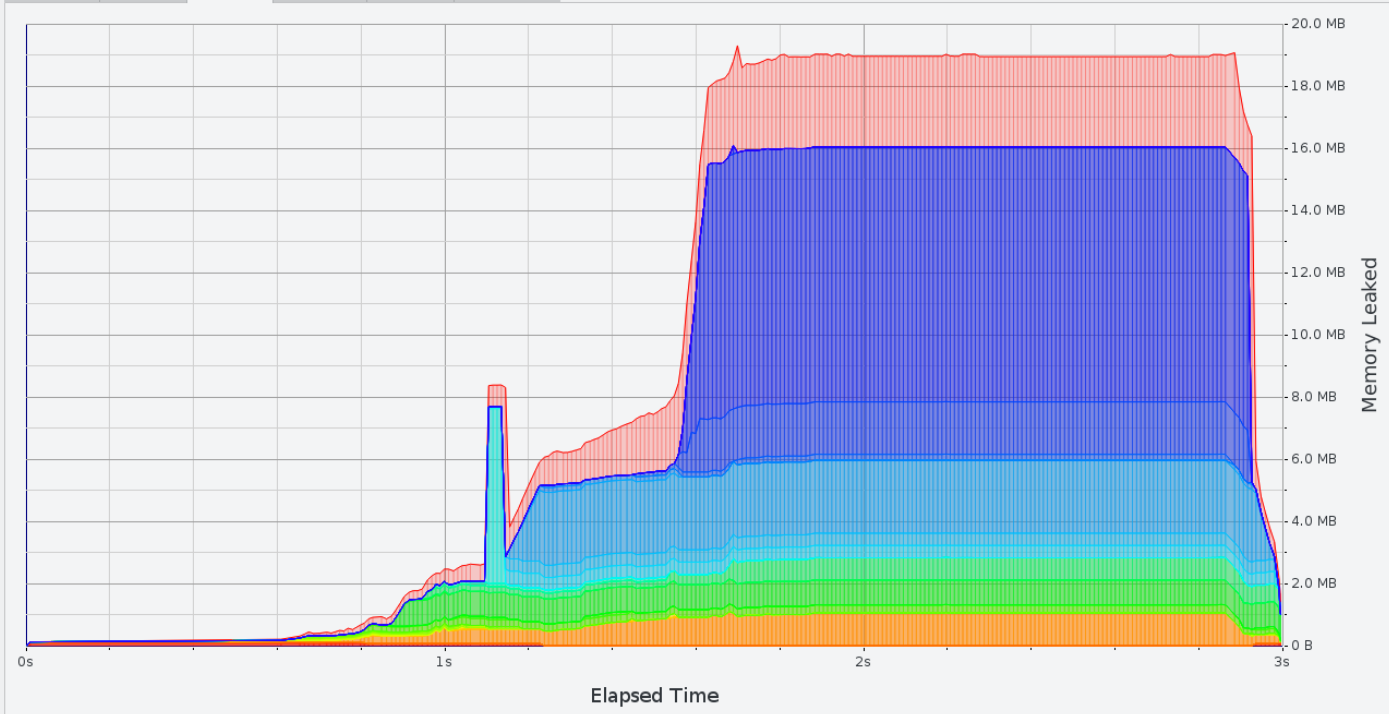
Bottom-UpTop-DownLeakedAllocationsAllocatedFlame Graph

filter by function...filter by file...filter by module...

Allocations [-]	^	Peak [B]	Leaked [B]	Allocated [B]	Location
└ 429286		43102487	1067236	82249354	main in /home/milian/projects/kf5/build/kde/applications/kat...
└└ 341362		37544663	1032930	66293107	kdmain in /home/milian/projects/kf5/src/kde/applications/k...
└└└ 180201		19402610	886343	30157358	KWrite in /home/milian/projects/kf5/src/kde/applications/kat...
└└└└ 94879		15541997	16421	21738595	KTextEditor::EditorPrivate::createDocument(QObject*) in /h...
└└└└└ 94610		15525625	16421	21722223	DocumentPrivate in /home/milian/projects/kf5/src/framework...
└└└└└└ 94552		15521813	16421	21716932	KTextEditor::DocumentPrivate::onTheFlySpellCheckingEnabl...
└└└└└└└ 94479		15515476	16421	21710595	KateOnTheFlyChecker in /home/milian/projects/kf5/src/fram...
└└└└└└└└ 94478		15515444	16421	21710563	Speller in /home/milian/projects/kf5/src/frameworks/sonnet/...
└└└└└└└└└ 90234		3581967	0	3586803	Sonnet::Speller::Private::init(QString const&) in /home/milia...
└└└└└└└└└└ 4244		11933477	16421	18123760	Sonnet::Speller::Private::init(QString const&) in /home/milia...

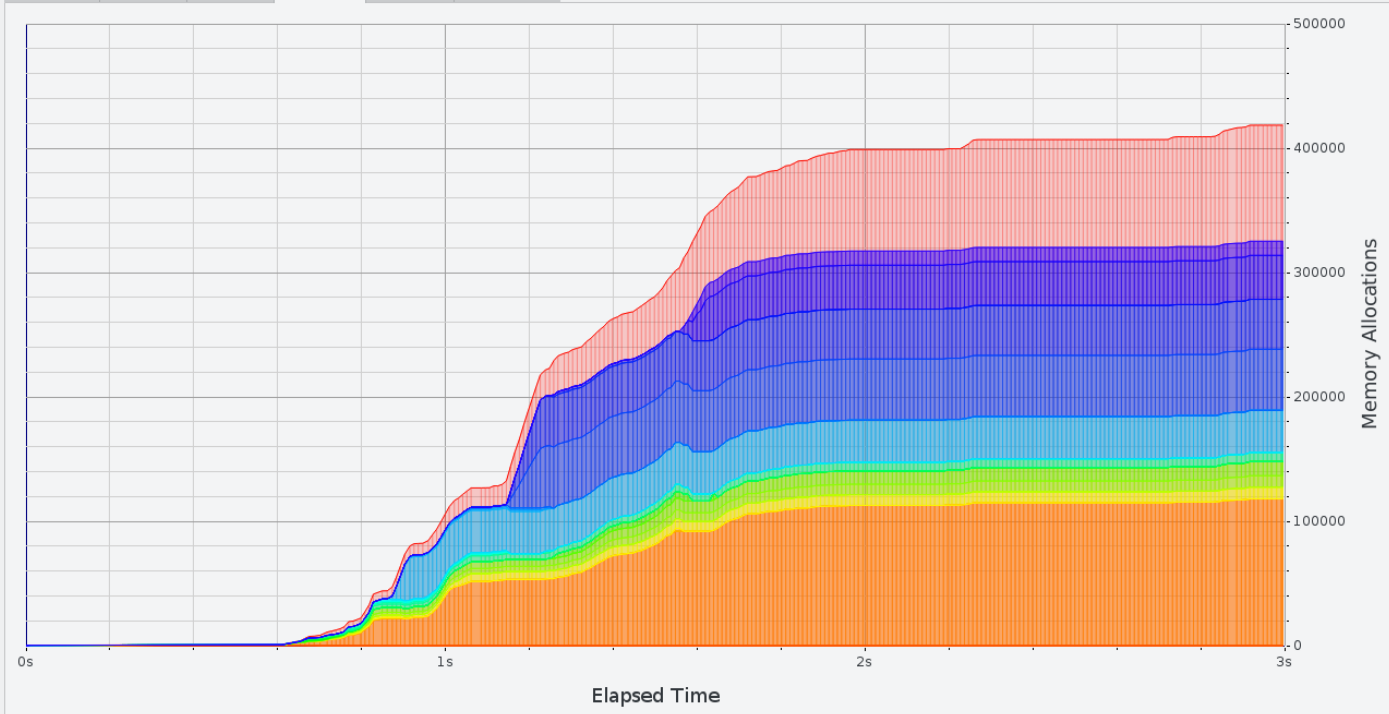
debuggee: kwrite  
total runtime: 2.996s  
bytes allocated in total (ignoring deallocations): 78.82 MiB (26.3 MiB/s)  
calls to allocation functions: 418513 (139690/s)  
peak heap memory consumption: 18.4 MiB  
total memory leaked: 1.2 MiB

Bottom-Up Top-Down Leaked Allocations Allocated Flame Graph



debuggee: kwrite  
total runtime: 2.996s  
bytes allocated in total (ignoring deallocations): 78.82 MiB (26.3 MiB/s)  
calls to allocation functions: 418513 (139690/s)  
peak heap memory consumption: 18.4 MiB  
total memory leaked: 1.2 MiB

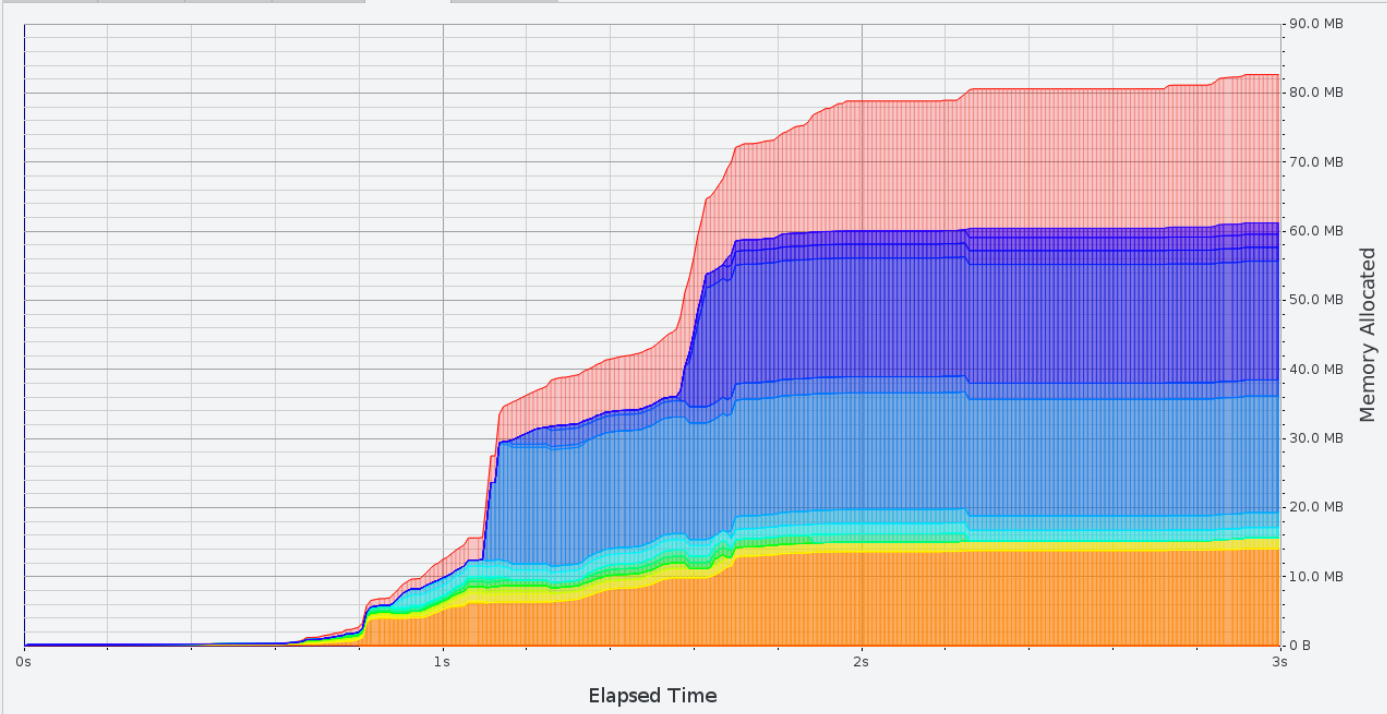
Bottom-Up Top-Down Leaked Allocations Allocated Flame Graph



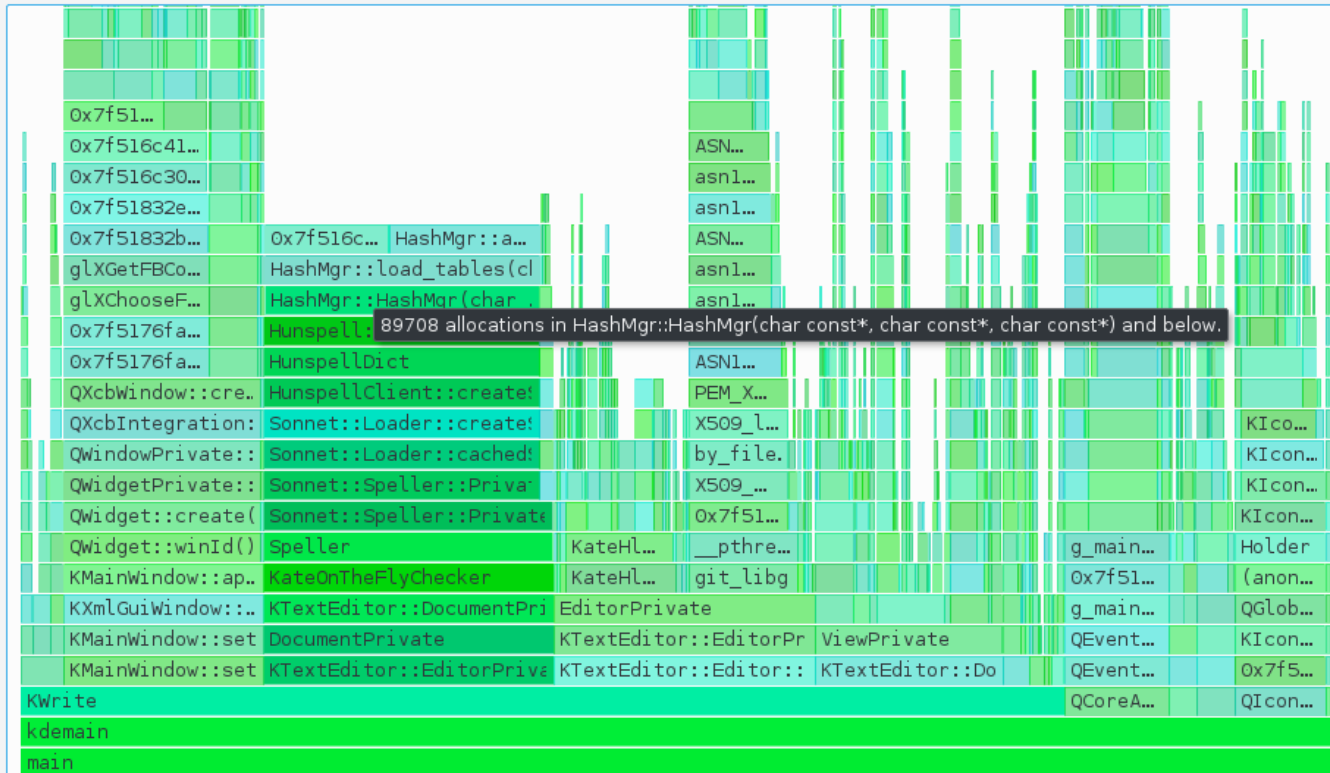


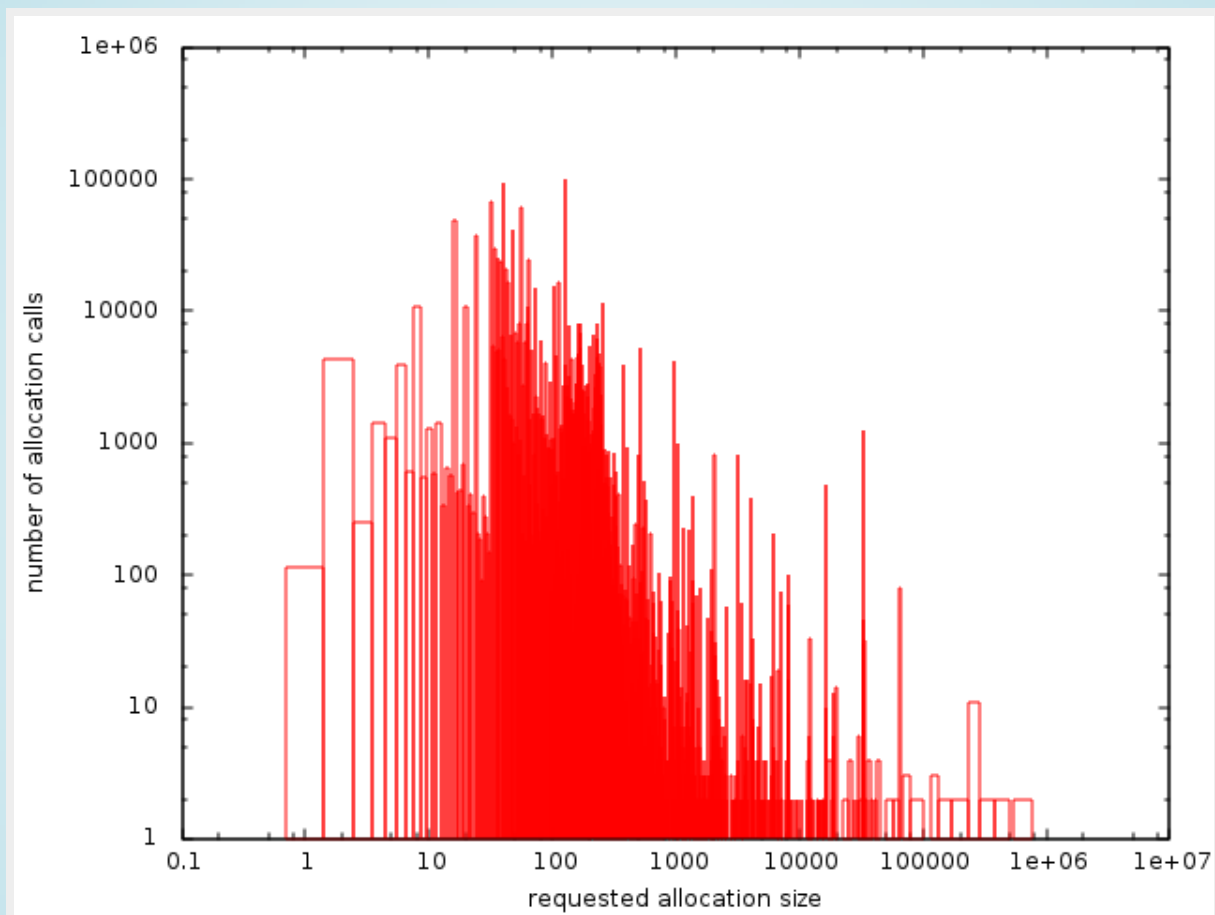
debuggee: kwrite  
total runtime: 2.996s  
bytes allocated in total (ignoring deallocations): 78.82 MiB (26.3 MiB/s)  
calls to allocation functions: 418513 (139690/s)  
peak heap memory consumption: 18.4 MiB  
total memory leaked: 1.2 MiB

Bottom-Up Top-Down Leaked Allocations Allocated Flame Graph



Bottom-Up Top-Down Leaked Allocations Allocated Flame Graph





# HOW?

- LD\_PRELOAD
- libunwind
- libbacktrace
- heaptrack\_gui

# INSTALLATION

```
git clone git://anongit.kde.org/heaptrack  
cd heaptrack  
mkdir build  
cd build  
cmake ..  
make install
```

# THE GOOD

- usable & often faster than massif
- lots of data leading to new insights
- foundation for more tools

# THE BAD

- `heaptrack_gui` unfinished
- more people should use it

# THE UGLY

- Linux platform specific
- cannot track stack memory



# THE END

## QUESTIONS?

Milian Wolff / [www.kdab.com](http://www.kdab.com)

```
git clone git://anongit.kde.org/heaptrack
```