

```
# DURATION TID FUNCTION
[16354] | OPENSsl_cpuid_setup() {
720.142 us [16354] |   getenv();
1.513 ms [16354] | } /* OPENSsl_cpuid_setup */
776.470 us [16354] | _register_async_wrap();
[16354] | main() {
[16354] |   node::Start() {
771.219 us [16354] |     getenv();
11.638 us [16354] |     uv_setup_args();
[16354] |     node::Init() {
[16354] |       uv_default_loop() {
[16354] |         uv_signal_global_once_init() {
[16354] |           pthread_once() {
[16354] |             uv_signal_global_init() {
[16354] |               uv_make_pipe() {
10.825 us [16354] |                 uv_pipe2();
12.375 us [16354] |               } /* uv_make_pipe */
19.914 us [16354] |             } /* uv_signal_global_init */
24.885 us [16354] |           } /* pthread_once */
25.904 us [16354] |         } /* uv_signal_global_once_init */
575.198 us [16354] |       } /* uv_default_loop */
[16354] |     uv_disable_stdio_inheritance() {
10.148 us [16354] |       ioctl();
33.548 us [16354] |     } /* uv_disable_stdio_inheritance */
[16354] |     v8::V8::SetFlagsFromString() {
[16354] |       v8::internal::FlagList::SetFlagsFromString()
10.870 us [16354] |         operator new[]();
29.957 us [16354] |       } /* v8::internal::FlagList::SetFlagsFromString */
769.573 us [16354] |     } /* v8::V8::SetFlagsFromString */
[16354] |     v8::Isolate::New() {
[16354] |       v8::internal::Isolate::Isolate() {
[16354] |         v8::internal::Heap::Heap(Namhyung Kim <namhyung.kim@lge.com>
[16354] |         v8::internal::GCTracer::GCTracer() {
[16354] |           v8::base::OS::TimeCurrentMillis() {
793.049 us [16354] |             v8::base::Time::ToJsTime();
804.646 us [16354] |           } /* v8::base::OS::TimeCurrentMillis */
806.201 us [16354] |         } /* v8::internal::GCTracer::GCTracer */
[16354] |       v8::internal::MarkCompactCollector::MarkCompactCollector() {
[16354] |         v8::base::Semaphore::Semaphore() {
10.680 us [16354] |           sem_init();
12.711 us [16354] |         } /* v8::base::Semaphore::Semaphore */
13.890 us [16354] |       } /* v8::internal::MarkCompactCollector::MarkCompactCollector */
3.773 ms [16354] |     } /* v8::internal::Heap::Heap */
10.419 us [16354] |     v8::base::RecursiveMutex::RecursiveMutex();
27.511 us [16354] |     v8::internal::Counters::Counters();
8.498 ms [16354] |   } /* v8::internal::Isolate::Isolate */
```

# uftrace

Function graph tracer  
(not only) for userspace

Namhyung Kim <namhyung.kim@lge.com>

Tracing Summit 2016

# Who am I?

- Namhyung Kim
  - ◆ Open source developer
  - ◆ Linux kernel contributor
    - › Mostly for perf and ftrace
  - ◆ Works for LG Electronics

# Introduction

- Function graph tracer
  - ◆ Inspired by ftrace in Linux kernel
  - ◆ Show execution time of each function (in C/C++)
  - ◆ Focus on single target process
  - ◆ Need support from compiler
  - ◆ Support x86\_64 and ARMv7 running Linux
  - ◆ Trace library calls and kernel functions

<https://github.com/namhyung/uftrace>

# How it works

- Compiler instrumentation
  - ◆ Target program built with **-pg**
  - ◆ (or **-finstrument-functions**)
- Using LD\_PRELOAD
  - ◆ libmcount.so and friends
- Shared memory
  - ◆ Target process writes to memory
  - ◆ ufttrace reads and writes it to file

# uftrace commands

`record` : runs a program and saves the trace data

`replay` : shows program execution from the data

`report` : shows performance statistics in the data

`live` : `record` + `replay` without saving data

`info` : shows system and program info in the data

`dump` : shows low-level trace data (and convert it)

`graph` : shows a function call graph from the data

`recv` : saves trace data from network

# uftrace record

```
$ cat hello.c
```

```
#include <stdio.h>
```

```
int main(void) {  
    hello("world");  
    return 0;  
}
```

# uftrace record

```
$ cat hello.c
#include <stdio.h>
void hello(const char *whom) {
    printf("Hello %s\n", whom);
}
int main(void) {
    hello("world");
    return 0;
}
```

# uftrace record

```
$ gcc -pg hello.c
```

```
$ uftrace record a.out
```



# uftrace record

```
$ gcc -pg hello.c
```

```
$ uftrace record a.out
```

```
Hello world
```

```
$ ls
```

```
a.out  hello.c  uftrace.data
```

# uftrace replay

```
$ uftrace replay
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203] | __monstartup();
0.509 us [ 1203] | __cxa_atexit();
[ 1203] | main() {
[ 1203] |   hello() {
4.404 us [ 1203] |     printf();
4.972 us [ 1203] |   } /* hello */
5.258 us [ 1203] | } /* main */
```

# uftrace live

```
$ uftrace live a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
0.791 us [ 2981] | __monstartup();
0.623 us [ 2981] | __cxa_atexit();
[ 2981] | main() {
[ 2981] |   hello() {
4.128 us [ 2981] |     printf();
4.998 us [ 2981] |   } /* hello */
5.571 us [ 2981] | } /* main */
```

# uftrace live

```
$ uftrace a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
0.814 us [ 3279] | __monstartup();
0.577 us [ 3279] | __cxa_atexit();
[ 3279] | main() {
[ 3279] |   hello() {
4.432 us [ 3279] |     printf();
4.989 us [ 3279] |   } /* hello */
5.528 us [ 3279] | } /* main */
```

# uftrace report

```
$ uftrace report
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
5.258 us	0.286 us	1	main
4.972 us	0.568 us	1	hello
4.404 us	4.404 us	1	printf
0.811 us	0.811 us	1	__monstartup
0.509 us	0.509 us	1	__cxa_atexit

# uftrace report

```
$ uftrace report -s self
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
4.404 us	4.404 us	1	printf
0.811 us	0.811 us	1	__monstartup
4.972 us	0.568 us	1	hello
0.509 us	0.509 us	1	__cxa_atexit
5.258 us	0.286 us	1	main

# uftrace report

```
$ uftrace report --avg-total
```

Avg total	Min total	Max total	Function
=====	=====	=====	=====
5.258 us	5.258 us	5.258 us	main
4.972 us	4.972 us	4.972 us	hello
4.404 us	4.404 us	4.404 us	printf
0.811 us	0.811 us	0.811 us	__monstartup
0.509 us	0.509 us	0.509 us	__cxa_atexit

# uftrace report

```
$ uftrace report --avg-total -s max
```

Avg total	Min total	Max total	Function
=====	=====	=====	=====
5.258 us	5.258 us	5.258 us	main
4.972 us	4.972 us	4.972 us	hello
4.404 us	4.404 us	4.404 us	printf
0.811 us	0.811 us	0.811 us	__monstartup
0.509 us	0.509 us	0.509 us	__cxa_atexit



# uftrace graph

```
$ uftrace graph main
```

```
backtrace
```

```
=====
```

```
backtrace #0: hit 1, time    5.258 us
```

```
  [0] main (0x400660)
```

```
calling functions
```

```
=====
```

```
  5.258 us : (1) main
```

```
  4.972 us : (1) hello
```

```
  4.404 us : (1) printf
```

# uftrace graph

```
$ uftrace graph
```

```
backtrace
```

```
=====
```

```
backtrace #0: hit 1, time    5.258 us
```

```
  [0] main (0x400660)
```

```
calling functions
```

```
=====
```

```
  5.258 us : (1) main
```

```
  4.972 us : (1) hello
```

```
  4.404 us : (1) printf
```

# uftrace info

```
$ uftrace info
```

```
# system information
```

```
# =====
```

```
# program version      : uftrace v0.6-51-gc070
```

```
# recorded on          : Wed Oct  5 14:51:34 2016
```

```
# cmdline              : uftrace record a.out
```

```
# cpu info             : Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz
```

```
# number of cpus       : 4 / 4 (online / possible)
```

```
# memory info          : 8.9 / 15.5 GB (free / total)
```

```
# system load          : 1.04 / 1.05 / 1.06 (1 / 5 / 15 min)
```

```
# kernel version       : Linux 4.7.3-2-ARCH
```

```
# hostname              : danjae
```

```
# distro                : "Arch Linux"
```

# uftrace info

```
. . .  
# process information  
# =====  
# number of tasks      : 1  
# task list           : 1203  
# exe image           : /home/namhyung/tracing-summit/a.out  
# build id            : ae443aeb9080a445bc308ec1eb186918bbeab581  
# exit status         : exited with code: 0  
# cpu time            : 0.000 / 0.000 sec (sys / user)  
# context switch      : 1 / 1 (voluntary / involuntary)  
# max rss             : 3100 KB  
# page fault          : 0 / 193 (major / minor)  
# disk iops           : 0 / 16 (read / write)
```

# Advanced Usage

- Using filters
- Using triggers
- Argument display
- Kernel tracing
- Visualization

# Task filter

```
$ uftrace replay --tid 1203
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203] | __monstartup();
0.509 us [ 1203] | __cxa_atexit();
[ 1203] | main() {
[ 1203] |     hello() {
4.404 us [ 1203] |         printf();
4.972 us [ 1203] |     } /* hello */
5.258 us [ 1203] | } /* main */
```

# Function filter

```
$ uftrace replay -F main
```

```
# DURATION      TID      FUNCTION
          [ 1203] | main() {
          [ 1203] |   hello() {
4.404 us [ 1203] |     printf();
4.972 us [ 1203] |   } /* hello */
5.258 us [ 1203] | } /* main */
```

# Function filter

```
$ uftrace replay -N hello
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203] | __monstartup();
0.509 us [ 1203] | __cxa_atexit();
5.258 us [ 1203] | main();
```



# Depth filter

```
$ uftrace replay -D 2
```

```
# DURATION      TID      FUNCTION
 0.811 us [ 1203] | __monstartup();
 0.509 us [ 1203] | __cxa_atexit();
          [ 1203] | main() {
 4.972 us [ 1203] |     hello();
 5.258 us [ 1203] | } /* main */
```

# Time filter

```
$ uftrace replay -t 1us
```

```
# DURATION      TID      FUNCTION
      [ 1203] | main() {
      [ 1203] |   hello() {
4.404 us [ 1203] |     printf();
4.972 us [ 1203] |   } /* hello */
5.258 us [ 1203] | } /* main */
```

# Depth trigger

```
$ uftrace replay -T hello@depth=1
# DURATION      TID      FUNCTION
  0.811 us [ 1203] | __monstartup();
  0.509 us [ 1203] | __cxa_atexit();
           [ 1203] | main() {
  4.972 us [ 1203] |     hello();
  5.258 us [ 1203] | } /* main */
```

# Trace-on trigger

```
$ uftrace replay --disable -T hello@trace-on
```

```
# DURATION      TID      FUNCTION
          [ 1203] | hello() {
4.404 us [ 1203] |   printf();
4.972 us [ 1203] | } /* hello */
5.258 us [ 1203] | } /* main */
```

# Trace-off trigger

```
$ uftrace replay -T printf@trace-off
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203] | __monstartup();
0.509 us [ 1203] | __cxa_atexit();
[ 1203] | main() {
[ 1203] |   hello() {
```

# Backtrace trigger

```
$ uftrace replay --disable \  
> -T printf@trace-on,backtrace  
# DURATION      TID      FUNCTION  
backtrace [ 1203] | /* [ 0] main */  
backtrace [ 1203] | /* [ 1] hello */  
4.404 us [ 1203] | printf();  
4.972 us [ 1203] | } /* hello */  
5.258 us [ 1203] | } /* main */
```

# Color trigger

```
$ uftrace replay -T hello@color=red
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203] | __monstartup();
0.509 us [ 1203] | __cxa_atexit();
[ 1203] | main() {
[ 1203] |     hello() {
4.404 us [ 1203] |         printf();
4.972 us [ 1203] |     } /* hello */
5.258 us [ 1203] | } /* main */
```

# Regex matching

```
$ uftrace replay -F ^[^_]
```

```
# DURATION      TID      FUNCTION
      [ 1203] | main() {
      [ 1203] |   hello() {
4.404 us [ 1203] |     printf();
4.972 us [ 1203] |   } /* hello */
5.258 us [ 1203] | } /* main */
```



# Specifying module

```
$ uftrace replay -T '.*@plt,color=red'
```

```
# DURATION      TID      FUNCTION
0.811 us [ 1203 ] | __monstartup();
0.509 us [ 1203 ] | __cxa_atexit();
[ 1203 ] | main() {
[ 1203 ] |   hello() {
4.404 us [ 1203 ] |     printf();
4.972 us [ 1203 ] |   } /* hello */
5.258 us [ 1203 ] | } /* main */
```

# Ignoring library calls

```
$ uftrace replay -N '*.plt'
# DURATION      TID      FUNCTION
           [ 1203] | main() {
4.972 us [ 1203] |   hello();
5.258 us [ 1203] | } /* main */

$ uftrace --no-libcall a.out
```

# Only library calls

```
$ uftrace replay --force /usr/bin/pwd
```

```
# DURATION      TID      FUNCTION
```

```
...
```

```
  2.929 us [ 3841] | getopt_long();
```

```
  5.476 us [ 3841] | getcwd();
```

```
  1.591 us [ 3841] | puts();
```

```
  0.424 us [ 3841] | free();
```

```
...
```

# Argument display

```
$ uftrace -A hello@arg1 a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
0.712 us [ 4778] | __monstartup();
0.602 us [ 4778] | __cxa_atexit();
[ 4778] | main() {
[ 4778] |   hello(0x400746) {
3.947 us [ 4778] |     printf();
4.214 us [ 4778] |   } /* hello */
5.174 us [ 4778] | } /* main */
```

# Argument format

```
$ uftrace -A hello@arg1/s a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
0.809 us [ 4923] | __monstartup();
0.487 us [ 4923] | __cxa_atexit();
[ 4923] | main() {
[ 4923] |   hello("world") {
4.132 us [ 4923] |     printf();
4.948 us [ 4923] |   } /* hello */
5.185 us [ 4923] | } /* main */
```

# Return value display

```
$ uftrace -R printf@retval a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
0.769 us [ 4984] | __monstartup();
0.446 us [ 4984] | __cxa_atexit();
[ 4984] | main() {
[ 4984] |   hello() {
4.320 us [ 4984] |     printf() = 12;
4.899 us [ 4984] |   } /* hello */
5.105 us [ 4984] | } /* main */
```

# Kernel tracing

```
$ sudo uftrace -k -F hello a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
           [ 8803] | hello() {
           [ 8803] |   printf() {
1.244 us [ 8803] |     sys_newfstat();
3.054 us [ 8803] |     __do_page_fault();
7.264 us [ 8803] |   } /* printf */
8.769 us [ 8803] | } /* hello */
```

# Kernel filtering

```
$ sudo uftrace -k -F hello \
```

```
> -F *page_fault@kernel a.out
```

```
Hello world
```

```
# DURATION      TID      FUNCTION
      [ 8887] | hello() {
      [ 8887] |   printf() {
3.243 us [ 8887] |     __do_page_fault();
6.788 us [ 8887] |   } /* printf */
7.942 us [ 8887] | } /* hello */
```



# Kernel depth

```
$ sudo uftrace -K 2 -F printf \  
> -F *page_fault@kernel a.out
```

Hello world

```
# DURATION      TID      FUNCTION  
      [ 9109] | printf() {  
      [ 9109] |   __do_page_fault() {  
0.268 us [ 9109] |     down_read_trylock();  
0.246 us [ 9109] |     find_vma();  
1.643 us [ 9109] |     handle_mm_fault();  
0.212 us [ 9109] |     up_read();  
3.340 us [ 9109] |   } /* __do_page_fault */  
6.412 us [ 9109] | } /* printf */
```

# Full kernel tracing

```
$ sudo uftrace -k --kernel-full -D 1 a.out
```

```
# DURATION      TID      FUNCTION
```

```
...
```

```
8.819 us [ 9286 ] | __do_page_fault();
```

```
2.685 us [ 9286 ] | sys_writev();
```

```
0.916 us [ 9286 ] | __monstartup();
```

```
0.595 us [ 9286 ] | __cxa_atexit();
```

```
8.427 us [ 9286 ] | main();
```

```
9.274 us [ 9286 ] | sys_munmap();
```

```
3.633 us [ 9286 ] | sys_writev();
```

```
...
```

# Visualizing with chrome

```
$ uftrace dump --chrome > hello.json
```

```
$ head hello.json
```

```
{"traceEvents": [  
  {"ts":92369113386,"ph":"B","pid":1203,"name":"__monstartup"},  
  {"ts":92369113387,"ph":"E","pid":1203,"name":"__monstartup"},  
  {"ts":92369113389,"ph":"B","pid":1203,"name":"__cxa_atexit"},  
  {"ts":92369113390,"ph":"E","pid":1203,"name":"__cxa_atexit"},
```

```
$ chrome
```

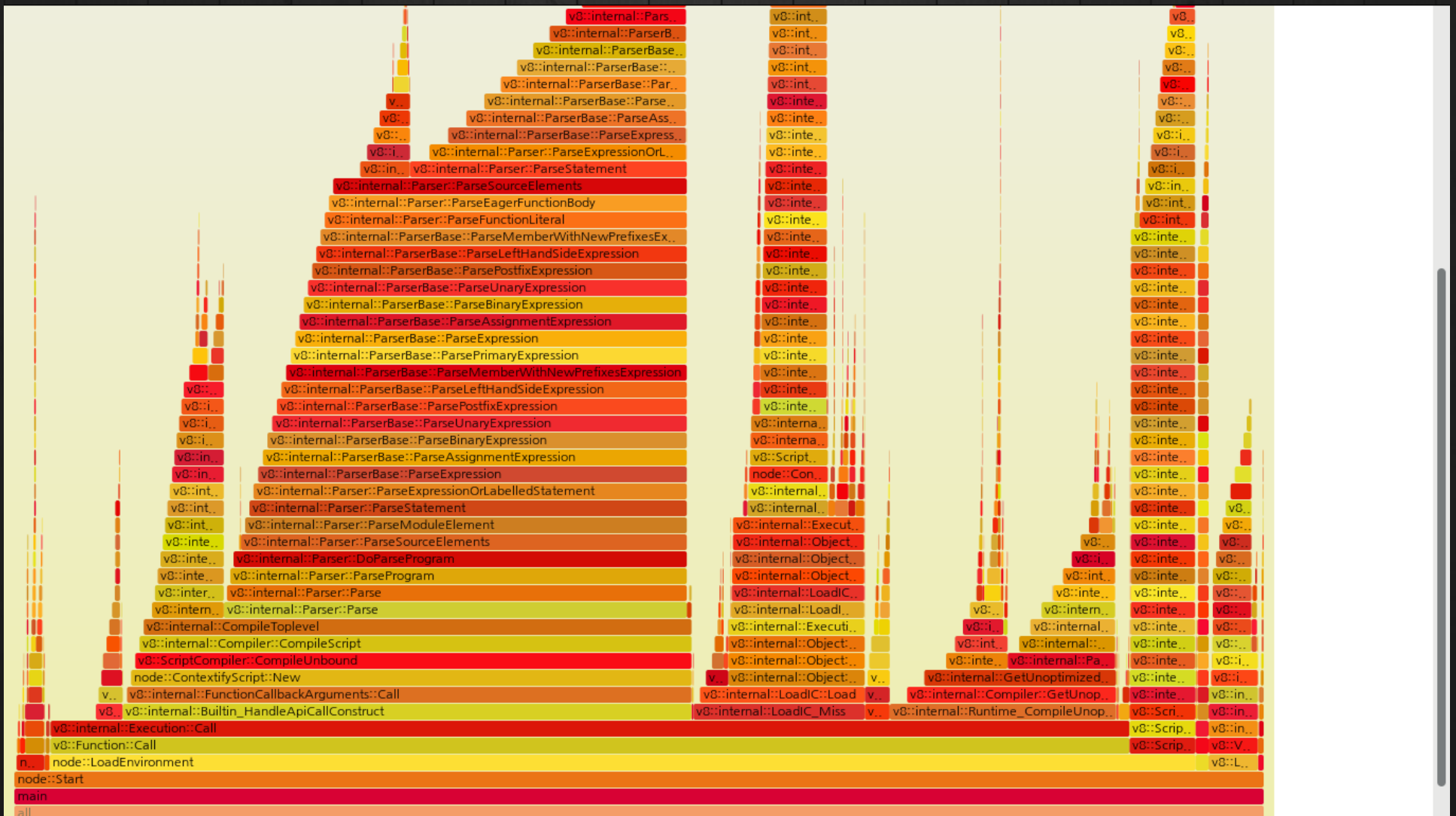


# Visualizing with flame-graph

```
$ uftrace dump --flame-graph | \  
> flamegraph.pl > hello.svg
```

```
$ firefox hello.svg
```

# Visualizing with flame-graph



# Future works

- Dynamic tracing
- SDT support
- Debug info
- More filtering
- User interface
- Performance optimization

# Questions?

Thanks!

<https://github.com/namhyung/uftrace>



**LG Electronics**