

# **Understanding the runtime behaviors of C++ programs using uftrace tool**

*CppCon 2017*

September 29, 2017

**Honggyu Kim (김홍규)**

LG Electronics

[honggyu.kp@gmail.com](mailto:honggyu.kp@gmail.com)

[hong.gyu.kim@lge.com](mailto:hong.gyu.kim@lge.com)

**uftrace**

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

# Introduction to ufttrace

- **ufttrace is a function tracer for C/C++ programs**
  - created by Namhyung Kim
    - one of active devs of Linux **perf** (profiling) tool
  - perf-like usage
    - record & (replay or report) model
- **ufttrace is able to trace**
  - C/C++ (user space) functions
    - compiled with **-pg** or **-finstrument-functions**
  - Library functions
  - Linux kernel functions
  - Some of system events

# Introduction to ufttrace

- **ufttrace is a function tracer for C/C++ programs**
  - created by Namhyung Kim
    - one of active devs of Linux **perf** (profiling) tool
  - perf-like usage
    - record & (replay or report) model
- **ufttrace is able to trace**
  - C/C++ (user space) functions
    - compiled with **-pg** or **-finstrument-functions**
  - Library functions
  - Linux kernel functions
  - Some of system events

# C/C++ (user) Function Tracing

- -pg or -finstrument-functions recompilation required

```
void bar() {  
  
}  
void foo() {  
    bar();  
}  
int main() {  
    foo();  
}
```

# C/C++ (user) Function Tracing

- -pg or -finstrument-functions recompilation required

```
$ gcc test.c
```

```
void bar() {  
  
}  
void foo() {  
    bar();  
}  
int main() {  
    foo();  
}
```

# C/C++ (user) Function Tracing

- -pg or -finstrument-functions recompilation required

```
$ gcc test.c
```

<pre>void bar() {</pre>	<pre>&lt;bar&gt;:</pre>
<pre>    }</pre>	<pre>    ret</pre>
<pre>void foo() {</pre>	<pre>&lt;foo&gt;:</pre>
<pre>    bar();</pre>	<pre>    call &lt;bar&gt;</pre>
<pre>}</pre>	<pre>    ret</pre>
<pre>int main() {</pre>	<pre>&lt;main&gt;:</pre>
<pre>    foo();</pre>	<pre>    call &lt;foo&gt;</pre>
<pre>}</pre>	<pre>    ret</pre>

# C/C++ (user) Function Tracing

- -pg or -finstrument-functions recompilation required

```
$ gcc -pg test.c
```

```
void bar() {
    }
void foo() {
    bar();
}
int main() {
    foo();
}

<bar>:
    call <mcount@plt>
    ret

<foo>:
    call <mcount@plt>
    call <bar>
    ret

<main>:
    call <mcount@plt>
    call <foo>
    ret
```



# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ ./a.out
```

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c  
$ uftrace record a.out
```

```
uftrace record
```

- Run a command and record its trace data

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c  
$ uftrace record a.out  
$ uftrace replay
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

```
uftrace replay
```

- Print recorded function trace

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

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

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

```
uftrace live
```

- Trace functions in a command during live execution same as uftrace record and replay

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

uftrace (live)

- Trace functions in a command during live execution same as uftrace record and replay

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */



# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

# C/C++ (user) Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	1.293 us	[11558]	__monstartup();
	0.814 us	[11558]	__cxa_atexit();
		[11558]	main() {
		[11558]	foo() {
	0.156 us	[11558]	bar();
	0.767 us	[11558]	} /* foo */
	1.140 us	[11558]	} /* main */

# Library Function Tracing

- Library Function Tracing works via PLT hooking

```
void bar() {  
    getpid();  
}  
void foo() {  
    bar();  
}  
int main() {  
    foo();  
}
```

# Library Function Tracing

- Library Function Tracing works via PLT hooking

```
$ gcc -pg test.c
```

```
void bar() {  
    getpid();  
}  
void foo() {  
    bar();  
}  
int main() {  
    foo();  
}
```

# Library Function Tracing

- Library Function Tracing works via **PLT** hooking

```
$ gcc -pg test.c
```

```
void bar() {  
    getpid();  
}  
void foo() {  
    bar();  
}  
int main() {  
    foo();  
}
```

```
<bar>:  
    call <mcount@plt>  
    call <getpid@plt> # indirect call in PLT  
    ret  
  
<foo>:  
    call <mcount@plt>  
    call <bar>  
    ret  
  
<main>:  
    call <mcount@plt>  
    call <foo>  
    ret
```

# Library Function Tracing

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

```
Hello
```

#	DURATION	TID	FUNCTION
	1.087 us	[12411]	__monstartup();
	0.790 us	[12411]	__cxa_atexit();
		[12411]	main() {
		[12411]	foo() {
		[12411]	bar() {
	6.263 us	[12411]	getpid();
	7.016 us	[12411]	} /* bar */
	7.443 us	[12411]	} /* foo */
	7.826 us	[12411]	} /* main */

# Library Function Tracing

```
$ ufttrace tests/t-fork
```

```
# DURATION      TID      FUNCTION
      [14528] | main() {
127.033 us [14528] |   fork();
      [14528] |   wait() {
      [14540] |   } /* fork */
      [14540] |   a() {
      [14540] |       b() {
      [14540] |           c() {
1.507 us [14540] |               getpid();
2.987 us [14540] |           } /* c */
3.464 us [14540] |       } /* b */
3.854 us [14540] |   } /* a */
13.394 us [14540] | } /* main */
799.270 us [14528] | } /* wait */
      [14528] |   a() {
      [14528] |       b() {
      [14528] |           c() {
2.410 us [14528] |               getpid();
3.470 us [14528] |           } /* c */
3.833 us [14528] |       } /* b */
4.144 us [14528] |   } /* a */
952.797 us [14528] | } /* main */
```

**tracing  
multi-processes**

# Linux Kernel Function Tracing

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

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

```
Hello CppCon!
```



# Linux Kernel Function Tracing

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

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

```
Hello CppCon!
```

#	DURATION	TID	FUNCTION
	0.395 us	[ 8926]	__monstartup();
	0.354 us	[ 8926]	__cxa_atexit();
		[ 8926]	main() {
		[ 8926]	printf() {
	0.572 us	[ 8926]	sys_newfstat();
	1.316 us	[ 8926]	__do_page_fault();
	4.123 us	[ 8926]	} /* puts */
		[ 8926]	fflush() {
	5.229 us	[ 8926]	sys_write();
	6.454 us	[ 8926]	} /* fflush */
	11.171 us	[ 8926]	} /* main */

# Event Tracing (sched event)

```
$ uftrace t-fork
```

#	DURATION	TID	FUNCTION
		[14983]	main() {
225.620	us	[14983]	fork();
		[14983]	wait() {
		[14995]	} /* fork */
		[14995]	a() {
		[14995]	b() {
		[14995]	c() {
1.033	us	[14995]	getpid();
2.280	us	[14995]	} /* c */
2.677	us	[14995]	} /* b */
3.020	us	[14995]	} /* a */
11.131	us	[14995]	} /* main */
695.312	us	[14983]	} /* wait */
		[14983]	a() {
		[14983]	b() {
		[14983]	c() {
2.067	us	[14983]	getpid();
3.067	us	[14983]	} /* c */
3.444	us	[14983]	} /* b */
3.841	us	[14983]	} /* a */
950.334	us	[14983]	} /* main */

# Event Tracing (sched event)

```
$ uftrace -E linux:schedule t-fork
```

#	DURATION	TID	FUNCTION
		[14983]	main() {
225.620	us	[14983]	fork();
		[14983]	wait() {
		[14983]	/* linux:sched-out */
		[14995]	} /* fork */
		[14995]	a() {
		[14995]	b() {
		[14995]	c() {
1.033	us	[14995]	getpid();
2.280	us	[14995]	} /* c */
2.677	us	[14995]	} /* b */
3.020	us	[14995]	} /* a */
11.131	us	[14995]	} /* main */
676.988	us	[14983]	/* linux:sched-in */
695.312	us	[14983]	} /* wait */
		[14983]	a() {
		[14983]	b() {
		[14983]	c() {
2.067	us	[14983]	getpid();
3.067	us	[14983]	} /* c */
3.444	us	[14983]	} /* b */
3.841	us	[14983]	} /* a */
950.334	us	[14983]	} /* main */

# Filters

```
$ gcc -pg test.c
```

```
$ uftrace a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
$ gcc -pg test.c
```

```
$ uftrace -D 2 a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
$ gcc -pg test.c
```

```
$ uftrace -D 2 a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

**-D DEPTH**, --depth=DEPTH

Set global trace limit in nesting level.

```
$ gcc -pg test.c
```

```
$ uftrace -D 2 a.out
```

#	DURATION	TID	FUNCTION
	0.648 us	[32431]	__monstartup();
	0.480 us	[32431]	__cxa_atexit();
		[32431]	main() {
	0.215 us	[32431]	foo();
	0.717 us	[32431]	} /* main */

**-D DEPTH**, --depth=DEPTH

Set global trace limit in nesting level.



```
$ gcc -pg test.c
```

```
$ uftrace -F foo a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
$ gcc -pg test.c
```

```
$ uftrace -F foo a.out
```

```
# DURATION      TID      FUNCTION
0.531 us [21315] | __monstartup();
0.435 us [21315] | __cxa_atexit();
           [21315] | main() {
           [21315] |     foo() {
0.134 us [21315] |         bar();
0.564 us [21315] |     } /* foo */
0.890 us [21315] | } /* main */
```

**-F FUNC**, --filter=FUNC

Set filter to trace selected functions only.

```
$ gcc -pg test.c
```

```
$ uftrace -F foo a.out
```

#	DURATION	TID	FUNCTION
		[32432]	foo() {
	0.175 us	[32432]	bar();
	1.137 us	[32432]	} /* foo */

**-F FUNC**, --filter=FUNC

Set filter to trace selected functions only.

```
$ gcc -pg test.c
```

```
$ uftrace -N foo a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
$ gcc -pg test.c
```

```
$ uftrace -N foo a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

**-N FUNC**, --notrace=FUNC

Set filter not to trace selected functions  
(and children)

```
$ gcc -pg test.c
```

```
$ uftrace -N foo a.out
```

#	DURATION	TID	FUNCTION
	0.728 us	[32436]	__monstartup();
	0.505 us	[32436]	__cxa_atexit();
	0.741 us	[32436]	main();

**-N FUNC**, --notrace=FUNC

Set filter not to trace selected functions  
(and children)

```
$ gcc -pg test.c
```

```
$ uftrace -t 200ns a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
$ gcc -pg test.c
```

```
$ uftrace -t 200ns a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
		[21315]	foo() {
	0.134 us	[21315]	bar();
	0.564 us	[21315]	} /* foo */
	0.890 us	[21315]	} /* main */

```
-t TIME, --time-filter=TIME
```

Do not show small functions under the  
time threshold.



```
$ gcc -pg test.c
```

```
$ uftrace -t 200ns a.out
```

#	DURATION	TID	FUNCTION
	0.531 us	[21315]	__monstartup();
	0.435 us	[21315]	__cxa_atexit();
		[21315]	main() {
	0.564 us	[21315]	foo();
	0.890 us	[21315]	} /* main */

**-t TIME**, --time-filter=TIME

Do not show small functions under the  
**time threshold**.

# Report

```
$ gcc -pg test.c
```

```
$ gcc -pg test.c
```

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

```
$ gcc -pg test.c  
$ uftrace record a.out  
$ uftrace report
```

**uftrace report**

- Print statistics and summary for trace data

```
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
0.890 us	0.326 us	1	main
0.564 us	0.430 us	1	foo
0.531 us	0.531 us	1	__monstartup
0.435 us	0.435 us	1	__cxa_atexit
0.134 us	0.134 us	1	bar

```
uftrace report
```

- Print statistics and summary for trace data

```
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report -s total
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
0.890 us	0.326 us	1	main
0.564 us	0.430 us	1	foo
0.531 us	0.531 us	1	__monstartup
0.435 us	0.435 us	1	__cxa_atexit
0.134 us	0.134 us	1	bar

uftrace report

- Print statistics and summary for trace data

```
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report -s self
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
0.531 us	0.531 us	1	__monstartup
0.435 us	0.435 us	1	__cxa_atexit
0.564 us	0.430 us	1	foo
0.890 us	0.326 us	1	main
0.134 us	0.134 us	1	bar

uftrace report

- Print statistics and summary for trace data



```
$ gcc -pg test.c
$ uftrace record a.out
$ uftrace report -s call
```

Total time	Self time	Calls	Function
=====	=====	=====	=====
0.890 us	0.326 us	1	main
0.564 us	0.430 us	1	foo
0.531 us	0.531 us	1	__monstartup
0.435 us	0.435 us	1	__cxa_atexit
0.134 us	0.134 us	1	bar

uftrace report

- Print statistics and summary for trace data

#	#												
#	uftrace diff												
#	[0] base: uftrace.data.pg.bug						(from uftrace record -d uftrace.data.pg.bug ./clang -pg -O2 mcount-test.c )						#
#	[1] diff: uftrace.data.pg.noinline/						(from uftrace record -d uftrace.data.pg.noinline ./clang -pg -O2 -fno-inline-functions mcount-test.c )						#
#	#												
Total time (diff)			Self time (diff)			Calls (diff)			Function				
=====													
9.892 ms	11.085 ms	+1.192 ms	9.892 ms	11.085 ms	+1.192 ms	107747	122650	+14903	llvm::opt::OptSpecifier::getID				
7.256 ms	8.127 ms	+871.370 us	7.256 ms	8.127 ms	+871.370 us	79188	90070	+10882	llvm::opt::OptSpecifier::OptSpecifier				
8.009 ms	9.328 ms	+1.318 ms	7.091 ms	8.161 ms	+1.069 ms	61527	69660	+8133	llvm::opt::Option::Option				
46.861 ms	54.249 ms	+7.388 ms	26.834 ms	30.591 ms	+3.757 ms	61485	69615	+8130	llvm::opt::OptTable::getOption				
5.670 ms	6.353 ms	+683.206 us	5.670 ms	6.353 ms	+683.206 us	61478	69607	+8129	llvm::opt::Option::isValid				
2.849 ms	3.200 ms	+350.685 us	2.849 ms	3.200 ms	+350.685 us	30833	34903	+4070	llvm::opt::Option::getID				
32.662 ms	36.915 ms	+4.253 ms	12.192 ms	13.928 ms	+1.736 ms	30793	34860	+4067	llvm::opt::Option::getAlias				
120.788 ms	138.345 ms	+17.557 ms	32.461 ms	37.508 ms	+5.047 ms	30748	34812	+4064	llvm::opt::Option::matches				
44.424 ms	51.316 ms	+6.892 ms	12.415 ms	13.739 ms	+1.324 ms	30685	34747	+4062	llvm::opt::Option::getGroup				
67.375 ms	71.041 ms	+3.665 ms	5.956 ms	6.130 ms	+173.495 us	33179	30010	-3169	llvm::isa_impl_wrap::doit				
6.197 ms	7.057 ms	+860.541 us	4.562 ms	5.471 ms	+909.093 us	32468	29379	-3089	llvm::simplify_type::getSimplifiedValue				
919.646 us	1.169 ms	+249.434 us	919.646 us	1.169 ms	+249.434 us	9547	12262	+2715	llvm::opt::Option::getAliasArgs				
9.373 ms	10.446 ms	+1.072 ms	6.450 ms	7.178 ms	+728.042 us	15514	18223	+2709	llvm::opt::OptTable::getInfo				
1.886 ms	2.075 ms	+188.051 us	1.886 ms	2.075 ms	+188.051 us	20037	22683	+2646	llvm::ArrayRef::operator []				
41.149 ms	41.882 ms	+732.190 us	40.095 ms	40.829 ms	+733.878 us	683520	681481	-2039	llvm::DenseMapInfo::isEqual				
52.739 ms	53.841 ms	+1.101 ms	52.739 ms	53.841 ms	+1.101 ms	897586	895603	-1983	llvm::detail::DenseMapPair::getFirst				
42.066 ms	47.425 ms	+5.359 ms	29.566 ms	34.559 ms	+4.993 ms	214070	212342	-1728	llvm::DenseMapBase::getNumBuckets				
45.235 ms	54.162 ms	+8.926 ms	33.253 ms	41.688 ms	+8.434 ms	199077	197354	-1723	llvm::DenseMapBase::getBuckets				
11.796 ms	12.071 ms	+274.442 us	11.796 ms	12.071 ms	+274.442 us	210557	208862	-1695	llvm::DenseMap::getNumBuckets				
10.950 ms	11.191 ms	+241.103 us	10.950 ms	11.191 ms	+241.103 us	195454	193768	-1686	llvm::DenseMap::getBuckets				
1.983 ms	2.039 ms	+56.173 us	1.215 ms	1.184 ms	-31.155 us	14003	12383	-1620	llvm::cast_convert_val::doit				
27.815 ms	29.227 ms	+1.412 ms	2.043 ms	2.148 ms	+104.995 us	16789	15188	-1601	llvm::isa_impl_cl::doit				
25.771 ms	27.078 ms	+1.307 ms	2.568 ms	2.843 ms	+274.990 us	16789	15188	-1601	llvm::isa_impl::doit				
40.415 ms	42.873 ms	+2.457 ms	2.067 ms	2.161 ms	+93.863 us	16789	15188	-1601	llvm::isa				
11.622 ms	11.677 ms	+55.663 us	11.234 ms	11.325 ms	+91.026 us	191518	189983	-1535	llvm::DenseMapInfo::getEmptyKey				
617.921 us	566.714 us	-51.207 us	617.921 us	566.714 us	-51.207 us	12016	10587	-1429	llvm::Value::getValueID				
1.969 ms	2.081 ms	+112.270 us	1.969 ms	2.081 ms	+112.270 us	20750	22084	+1334	llvm::opt::Arg::getOption				
5.973 ms	5.770 ms	-202.954 us	5.973 ms	5.770 ms	-202.954 us	70018	68708	-1310	std::__iterator_category				
3.636 ms	3.818 ms	+181.413 us	1.785 ms	1.896 ms	+111.021 us	11361	10102	-1259	llvm::cast				
381.719 ms	429.060 ms	+47.340 ms	92.869 ms	104.834 ms	+11.964 ms	145336	144142	-1194	llvm::DenseMapBase::LookupBucketFor				
30.100 ms	33.063 ms	+2.962 ms	18.311 ms	20.381 ms	+2.070 ms	211383	210315	-1068	llvm::DenseMapIterator::DenseMapIterator				
73.687 ms	84.183 ms	+10.495 ms	25.792 ms	28.032 ms	+2.239 ms	121355	120289	-1066	llvm::DenseMapBase::getBucketsEnd				
1.253 ms	1.268 ms	+14.705 us	410.400 us	401.142 us	-9.258 us	3691	2700	-991	__gnu_cxx::__ops::_Iter_comp_val::operator ()				
262.593 us	216.129 us	-46.464 us	262.593 us	216.129 us	-46.464 us	3651	2661	-990	strncmp				
835.815 us	855.955 us	+20.140 us	575.069 us	641.669 us	+66.600 us	3638	2648	-990	llvm::TargetLibraryInfoImpl::getLibFunc::\$_0::				
1.978 ms	1.956 ms	-22.558 us	1.111 ms	1.100 ms	-11.085 us	4949	3973	-976	std::advance				
599.225 us	631.064 us	+31.839 us	256.472 us	222.474 us	-33.998 us	4949	3973	-976	std::__advance				
25.091 ms	26.948 ms	+1.857 ms	2.061 ms	2.126 ms	+64.441 us	9938	9018	-920	llvm::dyn_cast				
10.254 ms	10.329 ms	+75.034 us	10.057 ms	10.150 ms	+93.643 us	172794	171904	-890	llvm::DenseMapInfo::getTombstoneKey				
856.735 us	837.266 us	-19.469 us	625.475 us	640.702 us	+15.227 us	3951	3156	-795	llvm::ValueHandleBase::isValid				
3.533 ms	3.590 ms	+56.500 us	3.533 ms	3.590 ms	+56.500 us	66196	65477	-719	llvm::DebugEpochBase::HandleBase::HandleBase				
265.633 us	231.075 us	-34.558 us	265.633 us	231.075 us	-34.558 us	5251	4533	-718	llvm::Use::get				
22.312 ms	22.428 ms	+115.268 us	22.312 ms	22.428 ms	+115.268 us	279455	278757	-698	std::forward				
150.565 us	140.122 us	-10.443 us	150.565 us	140.122 us	-10.443 us	2962	2265	-697	llvm::ValueHandleBase::getValPtr				
556.806 us	589.755 us	+32.949 us	556.806 us	589.755 us	+32.949 us	10936	10243	-693	llvm::AttributeImpl::isStringAttribute				
18.779 ms	21.301 ms	+2.522 ms	12.746 ms	15.123 ms	+2.376 ms	90683	90016	-667	llvm::DenseMapBase::getEmptyKey				
704.818 us	732.707 us	+27.889 us	614.304 us	631.835 us	+17.531 us	8850	8189	-661	llvm::PointerLikeTypeTraits::getAsVoidPointer				
15.487 ms	17.542 ms	+2.054 ms	10.585 ms	12.530 ms	+1.944 ms	75728	75074	-654	llvm::DenseMapBase::getTombstoneKey				
1.656 ms	1.619 ms	-37.270 us	1.656 ms	1.619 ms	-37.270 us	11892	11244	-648	memcmp				
16.442 ms	18.592 ms	+2.149 ms	10.279 ms	12.247 ms	+1.967 ms	73630	72988	-642	llvm::DenseMapBase::getHashValue				
4.089 ms	4.118 ms	+29.803 us	655.472 us	642.650 us	-12.822 us	3174	2537	-637	llvm::ValueHandleBase::ValueHandleBase				



```
$ gcc -pg fibonacci.c
```

```
$ gcc -pg fibonacci.c  
$ uftrace fibonacci 5  
fib(5) = 5
```

```
$ gcc -pg fibonacci.c
```

```
$ uftrace fibonacci 5
```

```
fib(5) = 5
```

#	DURATION	TID	FUNCTION
	0.620 us	[31321]	__monstartup();
	0.456 us	[31321]	__cxa_atexit();
		[31321]	main() {
	1.478 us	[31321]	atoi();
		[31321]	fib() {
		[31321]	fib() {
		[31321]	fib() {
	0.155 us	[31321]	fib();
	0.123 us	[31321]	fib();
	0.883 us	[31321]	} /* fib */
	0.125 us	[31321]	fib();
	1.483 us	[31321]	} /* fib */
		[31321]	fib() {
	0.125 us	[31321]	fib();
	0.125 us	[31321]	fib();
	0.774 us	[31321]	} /* fib */
	2.716 us	[31321]	} /* fib */
	4.382 us	[31321]	printf();
	9.456 us	[31321]	} /* main */

```
$ gcc -pg fibonacci.c
```

```
$ uftrace -A fib@arg1 fibonacci 5
```

```
fib(5) = 5
```

#	DURATION	TID	FUNCTION
	0.770 us	[31365]	__monstartup();
	0.492 us	[31365]	__cxa_atexit();
		[31365]	main() {
	1.507 us	[31365]	atoi();
		[31365]	fib(5) {
		[31365]	fib(4) {
		[31365]	fib(3) {
	1.293 us	[31365]	fib(2);
	0.172 us	[31365]	fib(1);
	2.295 us	[31365]	} /* fib */
	0.157 us	[31365]	fib(2);
	3.025 us	[31365]	} /* fib */
		[31365]	fib(3) {
	0.150 us	[31365]	fib(2);
	0.155 us	[31365]	fib(1);
	0.917 us	[31365]	} /* fib */
	5.232 us	[31365]	} /* fib */
	4.856 us	[31365]	printf();
	12.697 us	[31365]	} /* main */

```
$ gcc -pg fibonacci.c
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

```
fib(5) = 5
```

#	DURATION	TID	FUNCTION
	0.718 us	[31379]	__monstartup();
	0.464 us	[31379]	__cxa_atexit();
		[31379]	main() {
	1.442 us	[31379]	atoi();
		[31379]	fib(5) {
		[31379]	fib(4) {
		[31379]	fib(3) {
	1.395 us	[31379]	fib(2) = 1;
	0.174 us	[31379]	fib(1) = 1;
	2.562 us	[31379]	} = 2; /* fib */
	0.157 us	[31379]	fib(2) = 1;
	3.330 us	[31379]	} = 3; /* fib */
		[31379]	fib(3) {
	0.152 us	[31379]	fib(2) = 1;
	0.154 us	[31379]	fib(1) = 1;
	0.959 us	[31379]	} = 2; /* fib */
	5.351 us	[31379]	} = 5; /* fib */
	5.729 us	[31379]	printf();
	13.627 us	[31379]	} /* main */



```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

## ARGUMENTS

```
<argument>      := <symbol> "@" <specs>
<specs>         := <spec> | <spec> "," <spec>
<spec>          := ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>      := "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>    := "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>      := "retval" [ "/" <format> [ <size> ] ]
<format>        := "i" | "u" | "x" | "s" | "c" | "f" | "S"
<size>          := "8" | "16" | "32" | "64"
<reg>           := <arch-specific register name> # "rdi", "xmm0", "r0", ...
<stack>        := "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

## ARGUMENTS

```
<argument>      := <symbol> "@" <specs>
<specs>          := <spec> | <spec> "," <spec>
<spec>          := ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>       := "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>     := "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>       := "retval" [ "/" <format> [ <size> ] ]
<format>         := "i" | "u" | "x" | "s" | "c" | "f" | "S"
<size>           := "8" | "16" | "32" | "64"
<reg>            := <arch-specific register name> # "rdi", "xmm0", "r0", ...
<stack>         := "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

## ARGUMENTS

```
<argument>      := <symbol> "@" <specs>
<specs>          := <spec> | <spec> "," <spec>
<spec>          := ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>       := "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>     := "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>       := "retval" [ "/" <format> [ <size> ] ]
<format>         := "i" | "u" | "x" | "s" | "c" | "f" | "S"
<size>           := "8" | "16" | "32" | "64"
<reg>            := <arch-specific register name> # "rdi", "xmm0", "r0", ...
<stack>         := "stack" [ "+" ] <offset>
```

```
$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
```

## ARGUMENTS

```
<argument>      := <symbol> "@" <specs>
<specs>          := <spec> | <spec> "," <spec>
<spec>          := ( <int_spec> | <float_spec> | <ret_spec> )
<int_spec>       := "arg" N [ "/" <format> [ <size> ] ] [ "%" ( <reg> | <stack> ) ]
<float_spec>     := "fparg" N [ "/" ( <size> | "80" ) ] [ "%" ( <reg> | <stack> ) ]
<ret_spec>       := "retval" [ "/" <format> [ <size> ] ]
<format>         := "i" | "u" | "x" | "s" | "c" | "f" | "S"
<size>           := "8" | "16" | "32" | "64"
<reg>            := <arch-specific register name> # "rdi", "xmm0", "r0", ...
<stack>         := "stack" [ "+" ] <offset>
```

```
$ gcc -pg fibonacci.c
```

```
$ gcc -pg fibonacci.c  
$ uftrace record fibonacci 5  
fib(5) = 5
```

```
$ gcc -pg fibonacci.c  
$ uftrace record fibonacci 5  
fib(5) = 5  
$ uftrace dump
```

uftrace dump

- Print raw tracing data in the data files

```
$ gcc -pg fibonacci.c  
$ uftrace record fibonacci 5  
fib(5) = 5  
$ uftrace dump --chrome
```

**--chrome**

Show JSON style output as used by the Google Chrome tracing facility.

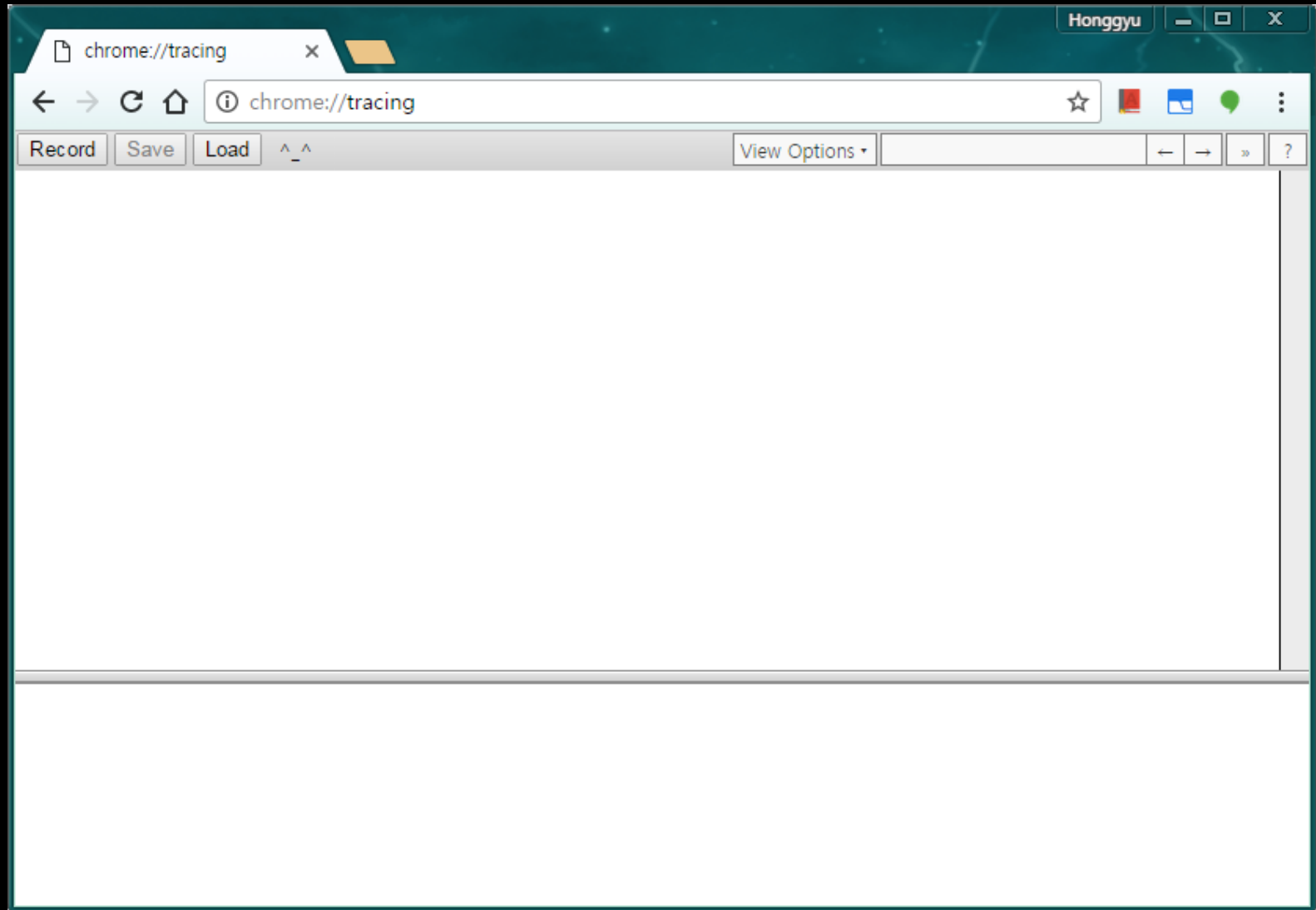


```
$ gcc -pg fibonacci.c
$ uftrace record fibonacci 5
fib(5) = 5
$ uftrace dump --chrome
```

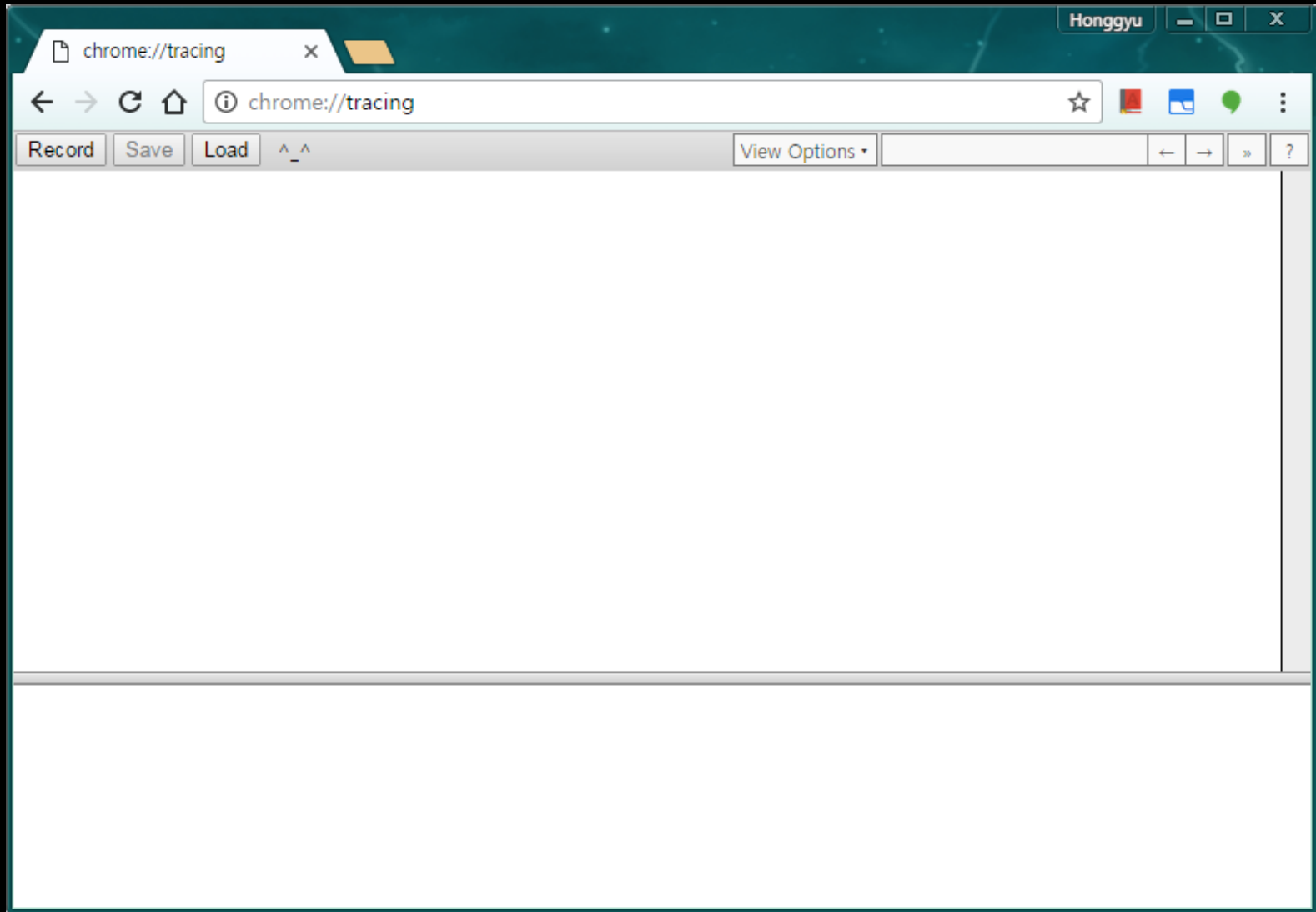
```
{ "traceEvents": [
  { "ts": 5913706403443, "ph": "B", "pid": 32256, "name": "__monstartup" },
  { "ts": 5913706403444, "ph": "E", "pid": 32256, "name": "__monstartup" },
  { "ts": 5913706403447, "ph": "B", "pid": 32256, "name": "__cxa_atexit" },
  { "ts": 5913706403447, "ph": "E", "pid": 32256, "name": "__cxa_atexit" },
  { "ts": 5913706403448, "ph": "B", "pid": 32256, "name": "main" },
  { "ts": 5913706403448, "ph": "B", "pid": 32256, "name": "atoi" },
  { "ts": 5913706403450, "ph": "E", "pid": 32256, "name": "atoi" },
  { "ts": 5913706403450, "ph": "B", "pid": 32256, "name": "fib" },
  { "ts": 5913706403450, "ph": "B", "pid": 32256, "name": "fib" },
    ...
  { "ts": 5913706403452, "ph": "E", "pid": 32256, "name": "fib" },
  { "ts": 5913706403453, "ph": "E", "pid": 32256, "name": "fib" },
  { "ts": 5913706403453, "ph": "E", "pid": 32256, "name": "fib" },
  { "ts": 5913706403453, "ph": "B", "pid": 32256, "name": "printf" },
  { "ts": 5913706403457, "ph": "E", "pid": 32256, "name": "printf" },
  { "ts": 5913706403458, "ph": "E", "pid": 32256, "name": "main" }
], "metadata": {
  "command_line": "uftrace record fibonacci 5 ",
  "recorded_time": "Thu Sep 22 22:31:17 2016"
} }
```

```
$ gcc -pg fibonacci.c  
$ uftrace record fibonacci 5  
fib(5) = 5  
$ uftrace dump --chrome > fib.json
```

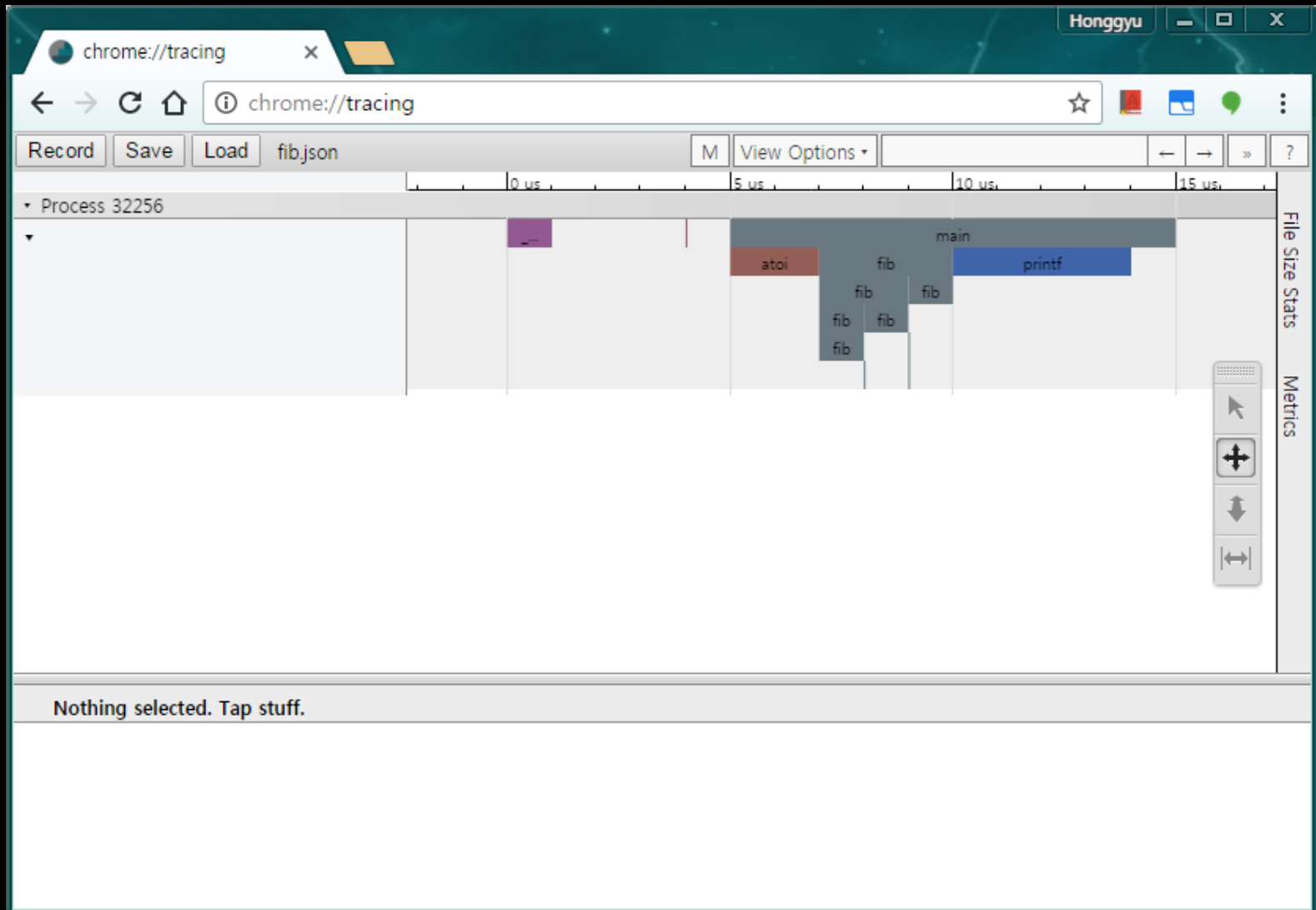
# 1. Open Chrome Browser



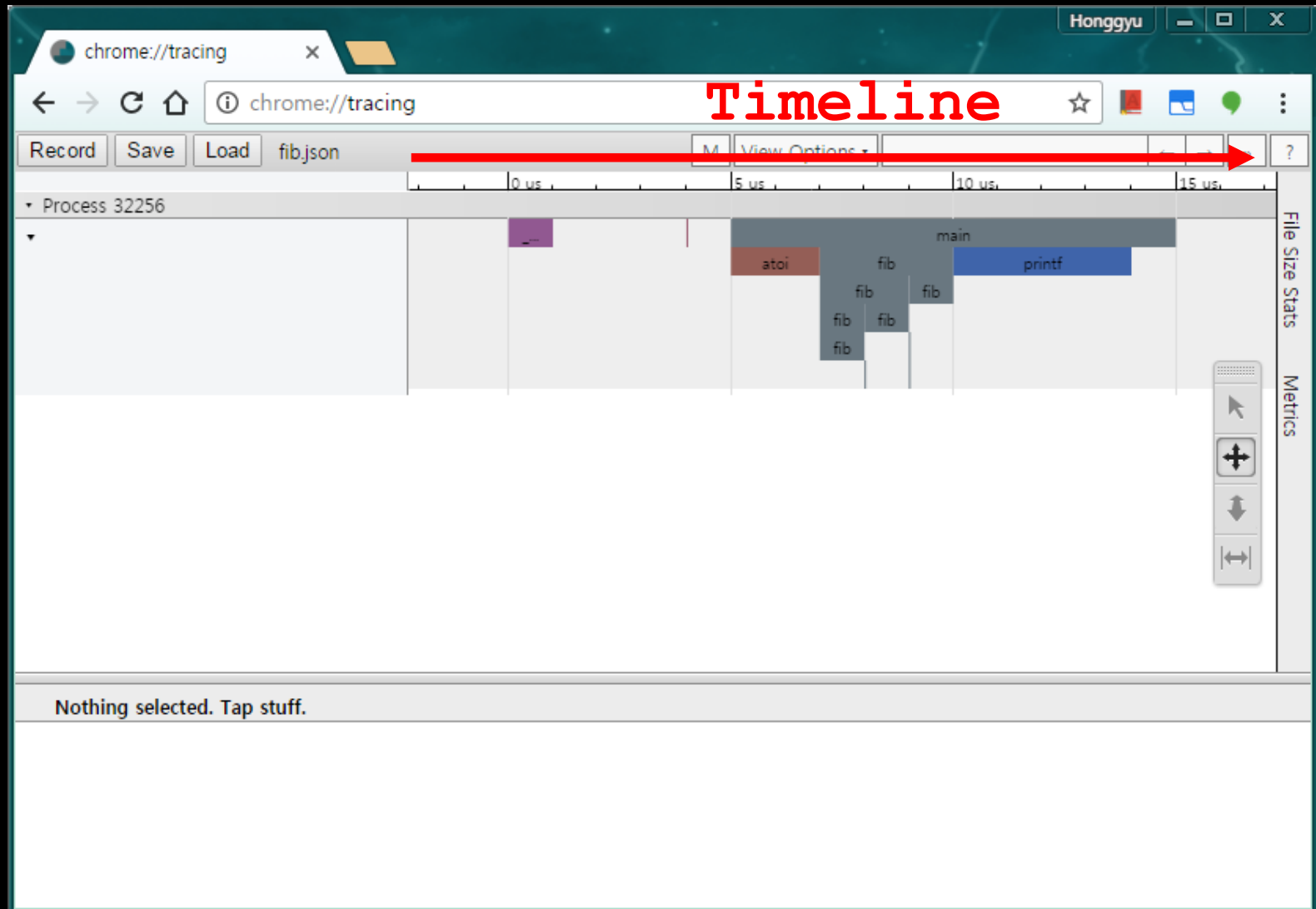
1. Open Chrome Browser
2. Load JSON file in **chrome://tracing**



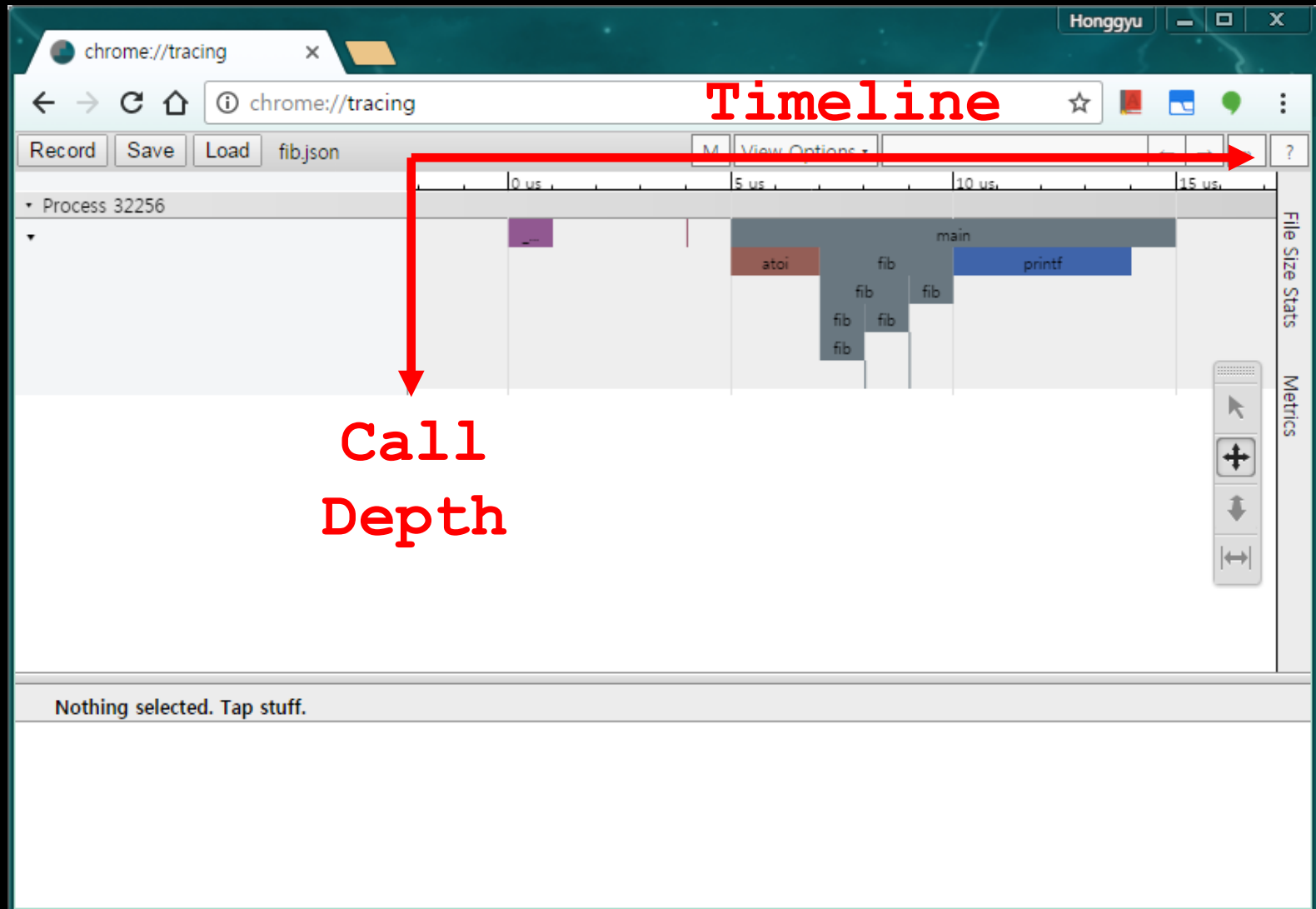
1. Open Chrome Browser
2. Load JSON file in **chrome://tracing**



1. Open Chrome Browser
2. Load JSON file in **chrome://tracing**



1. Open Chrome Browser
2. Load JSON file in **chrome://tracing**



# HTML File Generation

- "trace2html" file is to translate .json into .html file.
  - <https://github.com/catapult-project/catapult/blob/master/tracing/bin/trace2html>

```
$ trace2html trace-fib.json
```



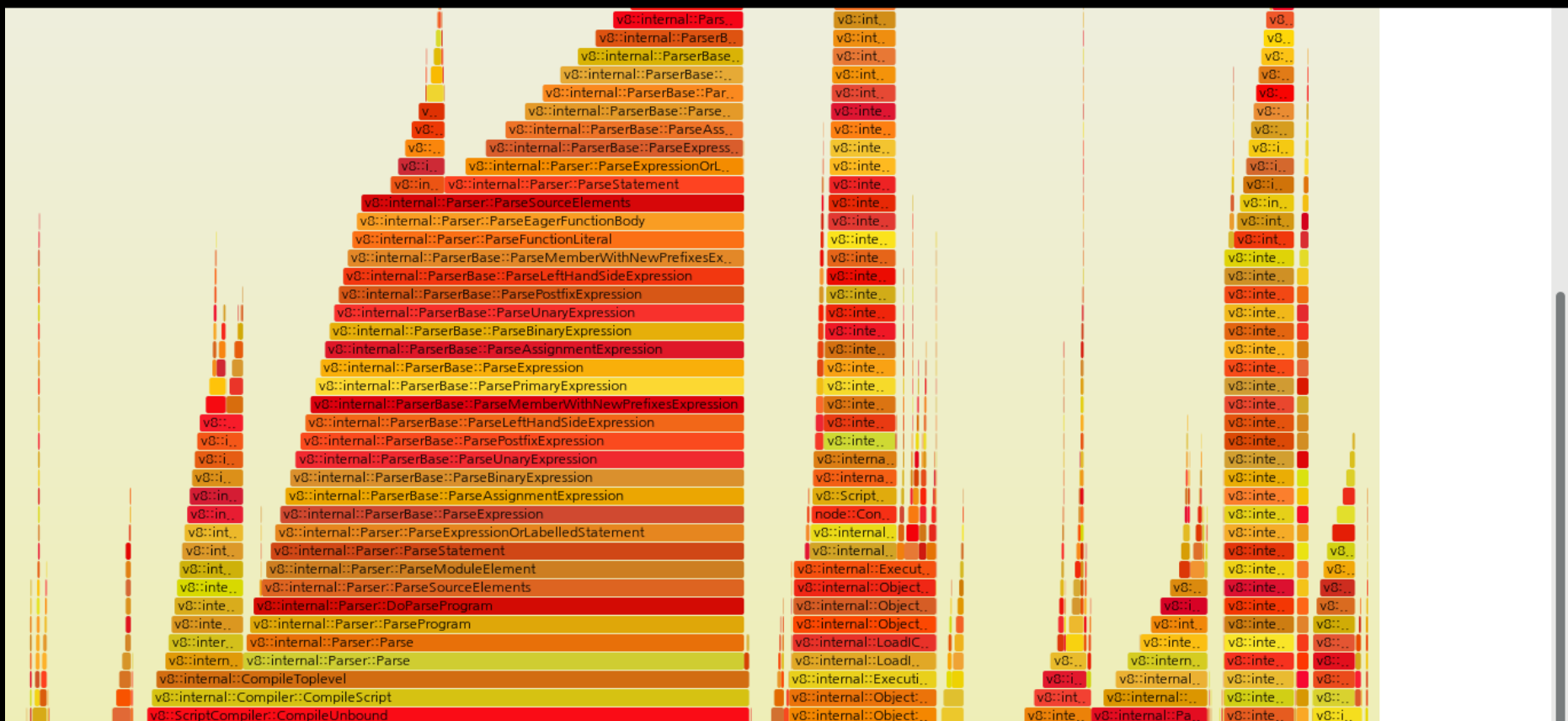
# HTML File Generation

- "trace2html" file is to translate .json into .html file.
  - <https://github.com/catapult-project/catapult/blob/master/tracing/bin/trace2html>

```
$ trace2html trace-fib.json  
trace-fib.html
```

# Frame Graph Output

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



# **Tracing Pre-Built Binaries**

**(without -pg compilation)**

**Examining  
Compilation Procedures**

# Tracing Pre-Built Binaries

```
$ /usr/bin/gcc
```

# Tracing Pre-Built Binaries

```
$ /usr/bin/gcc hello.c
```

# Tracing Pre-Built Binaries

```
$ uftrace /usr/bin/gcc hello.c
```

# Tracing Pre-Built Binaries

```
$ uftrace /usr/bin/gcc hello.c
```

```
uftrace: /home/honggyu/work/uftrace/cmd-record.c:1477:check_binary
```

```
ERROR: Can't find 'mcount' symbol in the '/usr/bin/gcc'.
```

```
It seems not to be compiled with -pg or -finstrument-functions flag  
which generates traceable code. Please check your binary file.
```

# Tracing Pre-Built Binaries

```
$ uftrace /usr/bin/gcc hello.c
```

```
uftrace: /home/honggyu/work/uftrace/cmd-record.c:1477:check_binary
```

```
ERROR: Can't find 'mcount' symbol in the '/usr/bin/gcc'.
```

```
It seems not to be compiled with -pg or -finstrument-functions flag  
which generates traceable code. Please check your binary file.
```

```
$ uftrace --force /usr/bin/gcc hello.c
```

**uftrace can still trace  
library function calls**



# Tracing Pre-Built Binaries

```
$ uftrace --force /usr/bin/gcc hello.c
```

```
# DURATION      TID      FUNCTION
  7.030 us [87482] | malloc();
  1.007 us [87482] | sbrk();
  0.530 us [87482] | malloc();
  8.672 us [87482] | memcpy();
                        [87482] | _obstack_begin() {
  3.050 us [87482] |     malloc();
  4.613 us [87482] | } /* _obstack_begin */
  0.423 us [87482] | malloc();
  1.763 us [87482] | calloc();
  0.280 us [87482] | malloc();
  0.180 us [87482] | malloc();
  0.400 us [87482] | free();
  1.053 us [87482] | realloc();
  0.990 us [87482] | __fsetlocking();
  0.167 us [87482] | __fsetlocking();
  0.097 us [87482] | __fsetlocking();
                        [87482] | setlocale() {
  0.236 us [87482] |     free();
  0.177 us [87482] |     free();
 56.100 us [87482] | } /* setlocale */
                        [87482] | setlocale() {
  0.240 us [87482] |     free();
  0.130 us [87482] |     free();
  4.047 us [87482] | } /* setlocale */
  7.720 us [87482] | bindtextdomain();
  1.477 us [87482] | textdomain();
                        [87482] | gettext() {
  0.200 us [87482] |     free();
  0.370 us [87482] |     free();
  0.293 us [87482] |     free();
  0.147 us [87482] |     free();
```

```
...
```

**--force**

**Trace even if executable is  
not instrumented**

# Tracing Pre-Built Binaries

```
$ uftrace --force -t 200us /usr/bin/gcc hello.c
```

#	DURATION	TID	FUNCTION
		[88132]	} /* vfork */
		[88132]	execv() {
		[88133]	} /* vfork */
		[88133]	execvp() {
361.230	us	[88133]	memset();
366.410	us	[88133]	memset();
365.573	us	[88133]	memset();
354.307	us	[88133]	memset();
357.133	us	[88133]	memset();
360.316	us	[88133]	memset();
364.249	us	[88133]	memset();
		[88134]	} /* vfork */
		[88134]	} /* vfork */
205.504	us	[88134]	gettext();
		[88134]	vfork() {
		[88135]	} /* vfork */
		[88135]	execvp() {
359.919	us	[88134]	} /* vfork */
		[88134]	waitpid() {
314.197	us	[88135]	bfd_link_hash_traverse();
2.746	ms	[88135]	bfd_elf_size_dynamic_sections();
1.492	ms	[88135]	bfd_elf_size_dynsym_hash_dynstr();
373.356	us	[88135]	_bfd_fix_excluded_sec_syms();
340.170	us	[88135]	bfd_close();
49.408	ms	[88134]	} /* waitpid */

**-t TIME, --time-filter=TIME**

Do not show functions which  
run under the time threshold.  
If some functions explicitly  
have the 'trace' trigger applied,  
those are always traced  
regardless of execution time.

# Tracing Pre-Built Binaries

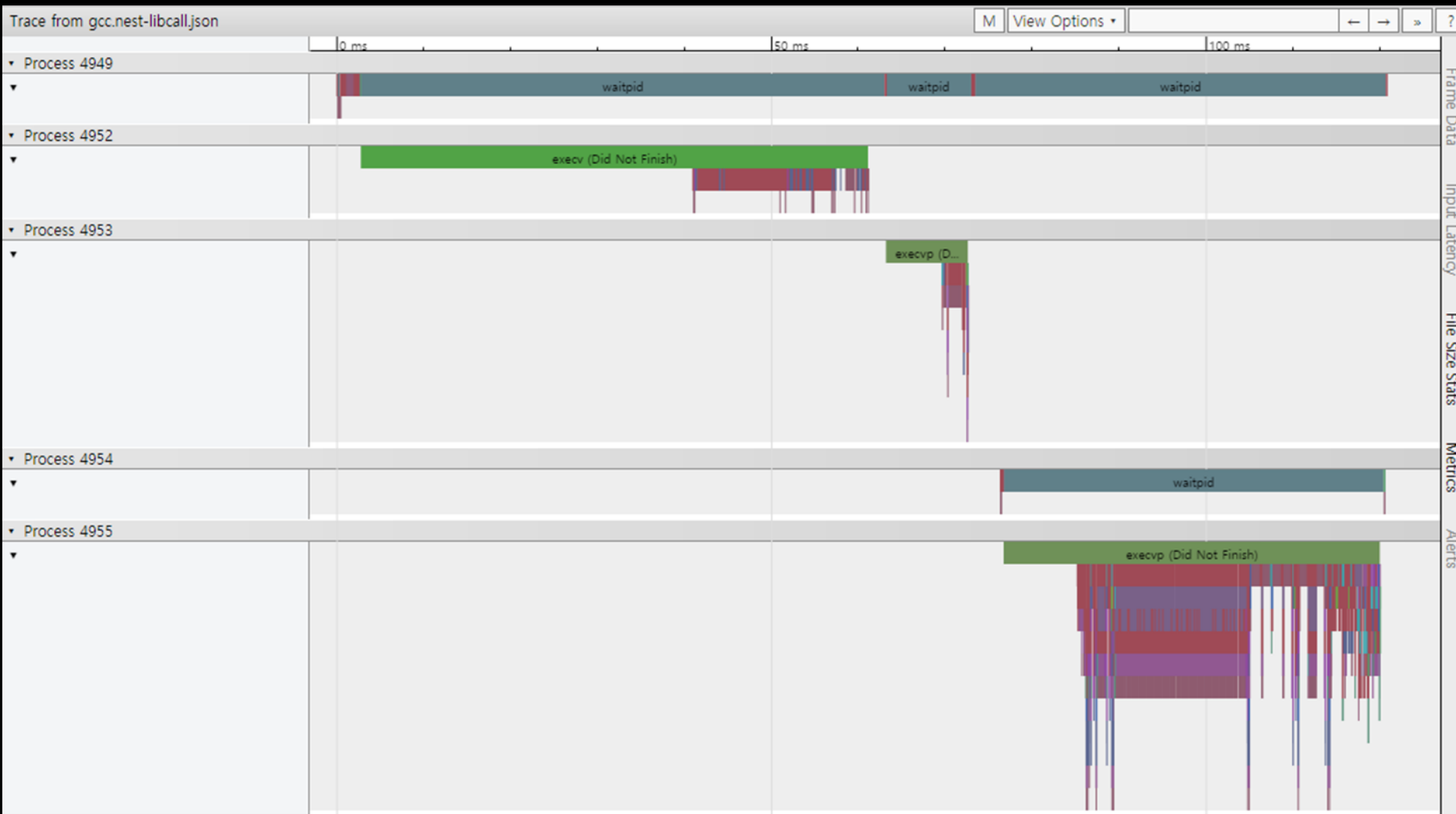
```
$ uftrace --force -t 200us --auto-args /usr/bin/gcc hello.c
```

#	DURATION	TID	FUNCTION
		[88132]	} = 0; /* vfork */
		[88132]	<b>execv("/usr/lib/gcc/x86_64-linux-gnu/5/cc1")</b> {
		[88133]	} = 0; /* vfork */
		[88133]	<b>execvp("as")</b> {
361.230	us	[88133]	memset(0x7fce671b6020, 0, 524296) = 0x7fce671b6020;
366.410	us	[88133]	memset(0x7fce64e3a020, 0, 524296) = 0x7fce64e3a020;
365.573	us	[88133]	memset(0x7fce64db9020, 0, 524296) = 0x7fce64db9020;
354.307	us	[88133]	memset(0x7fce64d38020, 0, 524296) = 0x7fce64d38020;
357.133	us	[88133]	memset(0x7fce64cb7020, 0, 524296) = 0x7fce64cb7020;
360.316	us	[88133]	memset(0x7fce64c36020, 0, 524296) = 0x7fce64c36020;
364.249	us	[88133]	memset(0x7fce64bb5020, 0, 524296) = 0x7fce64bb5020;
		[88134]	} = 0; /* vfork */
		[88134]	} = 88134; /* vfork */
205.504	us	[88134]	gettext();
		[88134]	vfork() {
		[88135]	} = 0; /* vfork */
		[88135]	<b>execvp("/usr/bin/ld")</b> {
359.919	us	[88134]	} = 88135; /* vfork */
		[88134]	waitpid(88135, 0x1560650, 0) {
314.197	us	[88135]	bfd_link_hash_traverse();
2.746	ms	[88135]	bfd_elf_size_dynamic_sections();
1.492	ms	[88135]	bfd_elf_size_dynsym_hash_dynstr();
373.356	us	[88135]	_bfd_fix_excluded_sec_syms();
340.170	us	[88135]	bfd_close();
49.408	ms	[88134]	} = 88135; /* waitpid */

**--auto-args**  
Automatically record arguments  
and return values of well-known  
library functions.  
(work-in-progress)

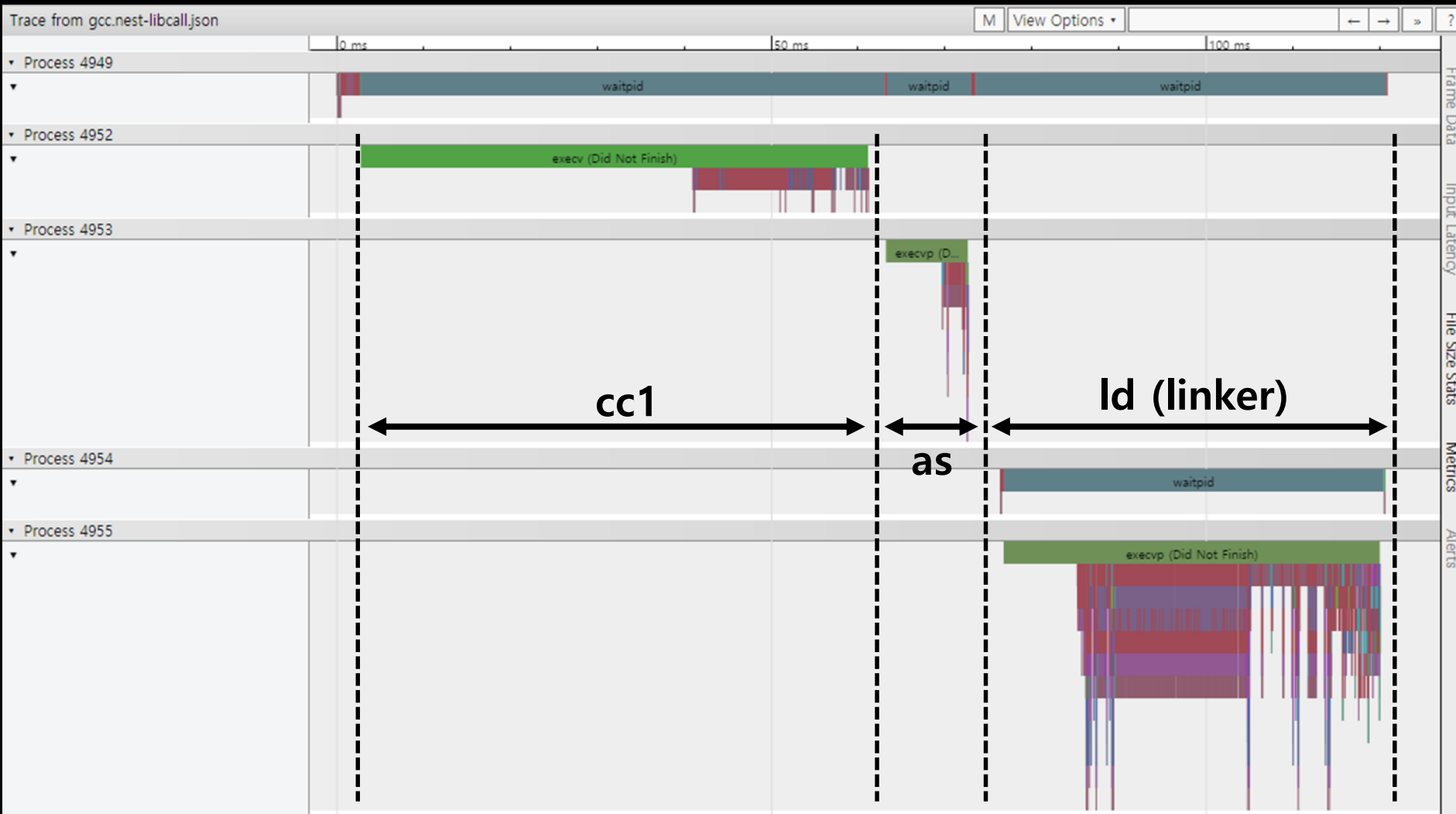
# Tracing Pre-Built Binaries

```
$ uftrace dump --chrome
```



# Tracing Pre-Built Binaries

```
$ uftrace dump --chrome
```



# Nested Library Tracing

```
$ uftrace --nest-libcall --auto-args \  
    /usr/bin/clang hello.c
```

--nest-libcall

Trace function calls between libraries.

By default, uftrace only record library call from the main executable.

# Nested Library Tracing

```
0.284 us [175968] | strlen("/usr/bin/ld") = 11;
               [175968] | llvm::sys::commandLineFitsWithinSystemLimits() {
...
21.584 us [175968] | } /* llvm::sys::commandLineFitsWithinSystemLimits */
0.197 us [175968] | llvm::opt::ArgList::getLastArg();
0.420 us [175968] | memcpy(0x7ffc7ba7a020, 0x28a07d0, 384) = 0x7ffc7ba7a020;
0.323 us [175968] | strlen("/usr/lib/llvm-3.8/bin/clang") = 27;
               [175968] | llvm::sys::ExecuteAndWait() {
0.360 us [175968] |     memcpy(0x7ffc7ba79b18, 0x2883dc0, 27) = 0x7ffc7ba79b18;
3.093 us [175968] |     access();
0.153 us [175968] |     std::_V2::system_category();
               [175968] |     std::__cxx11::basic_string::_M_create() {
               [175968] |         operator new() {
0.490 us [175968] |             malloc(28) = 0x28a1150;
1.053 us [175968] |             } /* operator new */
1.566 us [175968] |         } /* std::__cxx11::basic_string::_M_create */
0.253 us [175968] |     memcpy(0x28a1150, 0x2883dc0, 27) = 0x28a1150;
247.286 us [175968] |     posix_spawn();
               [175968] |     operator delete() {
0.590 us [175968] |         free(0x28a1150);
1.500 us [175968] |     } /* operator delete */
               [175968] |     waitpid(175980, 0x7ffc7ba79bfc, 0) {
...

```

**What's good having  
function trace?**



# **Use Case**

**QT QML engine bug fix  
using uftrace**

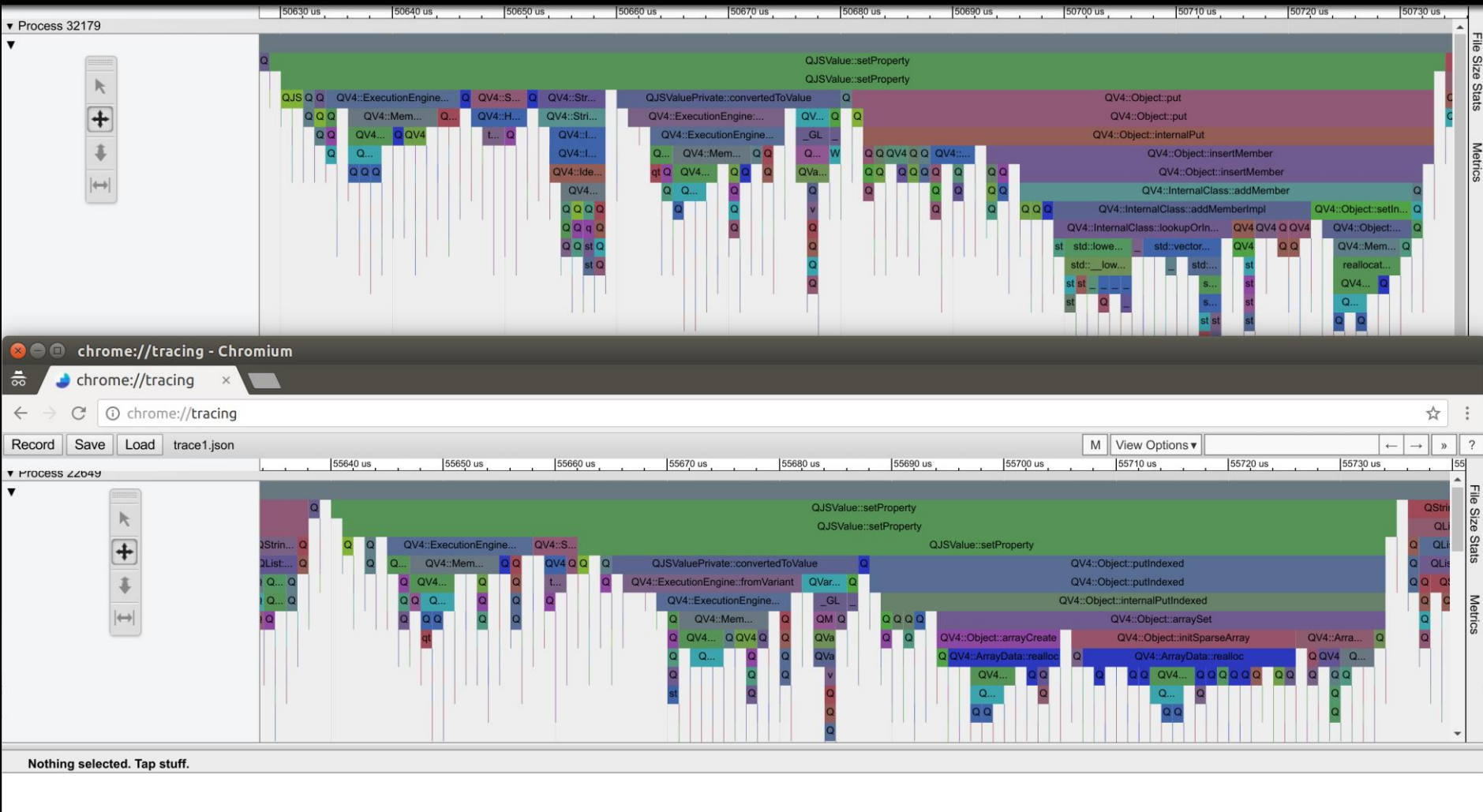
# Solving a bug via lateral thinking

*(or: How I solved a bug in 2 hours instead of 2 weeks)*

**Giuseppe D'Angelo**

Senior Software Engineer, KDAB (UK)  
CppCon 2017





working case

broken case

# **Solving a bug via lateral thinking**

*Lightning Talk at CppCon 2017*

**Giuseppe D'Angelo**

# **constexpr Function**

```
#include <stdio>
#include <stdlib>

constexpr int fib(const int n)
{
    if (n <= 2)
        return 1;
    return fib(n - 1) + fib(n - 2);
}

int main(int argc, char* argv[])
{
    constexpr int n = 7;
    const int result = fib(n);
    printf("%d\n", result);

    return fib(5);
}
```

```
$ g++ -pg -std=c++14 constexpr.cpp
$ uftrace -A fib@arg1/u -R fib@retval -A printf@arg1/s,arg2/i a.out
13
```

#	DURATION	TID	FUNCTION
	1.540 us	[160012]	__monstartup();
	0.900 us	[160012]	__cxa_atexit();
		[160012]	main() {
	8.884 us	[160012]	printf("%d\n", 13);
		[160012]	fib(5) {
		[160012]	fib(4) {
		[160012]	fib(3) {
	3.130 us	[160012]	fib(2) = 1;
	0.193 us	[160012]	fib(1) = 1;
	4.860 us	[160012]	} = 2; /* fib */
	0.210 us	[160012]	fib(2) = 1;
	5.873 us	[160012]	} = 3; /* fib */
		[160012]	fib(3) {
	0.140 us	[160012]	fib(2) = 1;
	0.180 us	[160012]	fib(1) = 1;
	1.137 us	[160012]	} = 2; /* fib */
	7.880 us	[160012]	} = 5; /* fib */
	18.283 us	[160012]	} /* main */



```
$ g++ -pg -std=c++14 constexpr.cpp
$ uftrace -A fib@arg1/u -R fib@retval -A printf@arg1/s,arg2/i a.out
13
```

#	DURATION	TID	FUNCTION
	1.540 us	[160012]	__monstartup();
	0.900 us	[160012]	__cxa_atexit();
		[160012]	main() {
	8.884 us	[160012]	printf("%d\n", 13);
		[160012]	fib(5) {
		[160012]	fib(4) {
		[160012]	fib(3) {
	3.130 us	[160012]	fib(2) = 1;
	0.193 us	[160012]	fib(1) = 1;
	4.860 us	[160012]	} = 2; /* fib */
	0.210 us	[160012]	fib(2) = 1;
	5.873 us	[160012]	} = 3; /* fib */
		[160012]	fib(3) {
	0.140 us	[160012]	fib(2) = 1;
	0.180 us	[160012]	fib(1) = 1;
	1.137 us	[160012]	} = 2; /* fib */
	7.880 us	[160012]	} = 5; /* fib */
	18.283 us	[160012]	} /* main */

```
$ g++ -pg -std=c++14 constexpr.cpp
$ uftrace -A fib@arg1/u -R fib@retval -A printf@arg1/s,arg2/i a.out
13
```

#	DURATION	TID	FUNCTION
	1.540 us	[160012]	__monstartup();
	0.900 us	[160012]	__cxa_atexit();
		[160012]	main() {
	8.884 us	[160012]	printf("%d\n", 13);
		[160012]	fib(5) {
		[160012]	fib(4) {
		[160012]	fib(3) {
	3.130 us	[160012]	fib(2) = 1;
	0.193 us	[160012]	fib(1) = 1;
	4.860 us	[160012]	} = 2; /* fib */
	0.210 us	[160012]	fib(2) = 1;
	5.873 us	[160012]	} = 3; /* fib */
		[160012]	fib(3) {
	0.140 us	[160012]	fib(2) = 1;
	0.180 us	[160012]	fib(1) = 1;
	1.137 us	[160012]	} = 2; /* fib */
	7.880 us	[160012]	} = 5; /* fib */
	18.283 us	[160012]	} /* main */

**But...**

```
#include <stdio>
#include <stdlib>

constexpr int fib(const int n)
{
    if (n <= 2)
        return 1;
    return fib(n - 1) + fib(n - 2);
}

int main()
{
    constexpr int n = 7;
    const int result = fib(n);
    printf("fib(%d) = %d\n", n, result);

    return fib(5);
}
```

```
#include <stdio>
#include <stdlib>

constexpr int fib(const int n)
{
    if (n <= 2)
        return 1;
    return fib(n - 1) + fib(n - 2);
}

int main()
{
    constexpr int n = 7;
    const int result = fib(n);
    printf("fib(%d) = %d\n", n, result);

    return fib(5);
}
```

```
$ clang++ -pg -std=c++14 constexpr.cpp
$ uftrace -A fib@arg1/u -R fib@retval -A printf@arg1/s,arg2/i a.out
13
```

```
# DURATION      TID      FUNCTION
    [20035] | main() {
    [20035] |   fib(7) {
    [20035] |     fib(6) {
    [20035] |       fib(5) {
    [20035] |         fib(4) {
    [20035] |           fib(3) {
0.380 us [20035] |             fib(2) = 1;
0.194 us [20035] |             fib(1) = 1;
3.117 us [20035] |             } = 2; /* fib */
0.167 us [20035] |             fib(2) = 1;
6.687 us [20035] |             } = 3; /* fib */
    [20035] |           fib(3) {
0.150 us [20035] |             fib(2) = 1;
0.156 us [20035] |             fib(1) = 1;
1.130 us [20035] |             } = 2; /* fib */
8.407 us [20035] |             } = 5; /* fib */
    ...
```

**clang generates a different code**

**std::string\_view**

```
$ cat string.cpp
#include <iostream>
```

```
int main()
{
    const char* msg = "std::string test!";
    print_string(msg);
}
```



```
$ cat string.cpp
```

```
#include <iostream>
```

```
#include <string>
```

```
void print_string(const std::string& s)
```

```
{
```

```
    std::cout << s << '\n';
```

```
}
```

```
int main()
```

```
{
```

```
    const char* msg = "std::string test!";
```

```
    print_string(msg);
```

```
}
```

```
$ g++ -pg string.cpp -o string
$ uftrace -F main -D 3 string
std::string test!
```

```
$ g++ -pg string.cpp -o string
```

```
$ uftrace -F main -D 3 string
```

```
std::string test!
```

#	DURATION	TID	FUNCTION
		[126472]	main() {
1.430	us	[126472]	std::allocator::allocator();
		[126472]	std::__cxx11::basic_string::basic_string() {
0.720	us	[126472]	std::__cxx11::basic_string::_M_local_data();
1.306	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::_Alloc_hider();
0.214	us	[126472]	std::char_traits::length();
6.544	us	[126472]	std::__cxx11::basic_string::_M_construct();
10.346	us	[126472]	} /* std::__cxx11::basic_string::basic_string */
		[126472]	print_string() {
19.283	us	[126472]	std::operator <<();
2.464	us	[126472]	std::operator <<();
22.713	us	[126472]	} /* print_string */
		[126472]	std::__cxx11::basic_string::~~basic_string() {
3.820	us	[126472]	std::__cxx11::basic_string::_M_dispose();
1.303	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::~~_Alloc_hider();
5.843	us	[126472]	} /* std::__cxx11::basic_string::~~basic_string */
0.850	us	[126472]	std::allocator::~~allocator();
44.803	us	[126472]	} /* main */

```
$ g++ -pg string.cpp -o string
```

```
$ uftrace -F main -D 3 string
```

```
std::string test!
```

#	DURATION	TID	FUNCTION
		[126472]	main() {
1.430	us	[126472]	std::allocator::allocator();
		[126472]	std::__cxx11::basic_string::basic_string() {
0.720	us	[126472]	std::__cxx11::basic_string::_M_local_data();
1.306	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::_Alloc_hider();
0.214	us	[126472]	std::char_traits::length();
6.544	us	[126472]	std::__cxx11::basic_string::_M_construct();
10.346	us	[126472]	} /* std::__cxx11::basic_string::basic_string */
		[126472]	print_string() {
19.283	us	[126472]	std::operator <<();
2.464	us	[126472]	std::operator <<();
22.713	us	[126472]	} /* print_string */
		[126472]	std::__cxx11::basic_string::~~basic_string() {
3.820	us	[126472]	std::__cxx11::basic_string::_M_dispose();
1.303	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::~~_Alloc_hider();
5.843	us	[126472]	} /* std::__cxx11::basic_string::~~basic_string */
0.850	us	[126472]	std::allocator::~~allocator();
44.803	us	[126472]	} /* main */

```
$ g++ -pg string.cpp -o string
```

```
$ uftrace -F main -D 3 string
```

```
std::string test!
```

#	DURATION	TID	FUNCTION
		[126472]	main() {
1.430	us	[126472]	std::allocator::allocator();
		[126472]	std::__cxx11::basic_string::basic_string() {
0.720	us	[126472]	std::__cxx11::basic_string::_M_local_data();
1.306	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::_Alloc_hider();
0.214	us	[126472]	std::char_traits::length();
6.544	us	[126472]	std::__cxx11::basic_string::_M_construct();
10.346	us	[126472]	} /* std::__cxx11::basic_string::basic_string */
		[126472]	print_string() {
19.283	us	[126472]	std::operator <<();
2.464	us	[126472]	std::operator <<();
22.713	us	[126472]	} /* print_string */
		[126472]	std::__cxx11::basic_string::~~basic_string() {
3.820	us	[126472]	std::__cxx11::basic_string::_M_dispose();
1.303	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::~~_Alloc_hider();
5.843	us	[126472]	} /* std::__cxx11::basic_string::~~basic_string */
0.850	us	[126472]	std::allocator::~~allocator();
44.803	us	[126472]	} /* main */

```
$ g++ -pg -O2 string.cpp -o string2
```

```
$ uftrace -F main -D 3 string2
```

```
std::string test!
```

```
$ g++ -pg string.cpp -o string
```

```
$ uftrace -F main -D 3 string
```

```
std::string test!
```

#	DURATION	TID	FUNCTION
		[126472]	main() {
1.430	us	[126472]	std::allocator::allocator();
		[126472]	std::__cxx11::basic_string::basic_string() {
0.720	us	[126472]	std::__cxx11::basic_string::_M_local_data();
1.306	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::_Alloc_hider();
0.214	us	[126472]	std::char_traits::length();
6.544	us	[126472]	std::__cxx11::basic_string::_M_construct();
10.346	us	[126472]	} /* std::__cxx11::basic_string::basic_string */
		[126472]	print_string() {
19.283	us	[126472]	std::operator <<();
2.464	us	[126472]	std::operator <<();
22.713	us	[126472]	} /* print_string */
		[126472]	std::__cxx11::basic_string::~~basic_string() {
3.820	us	[126472]	std::__cxx11::basic_string::_M_dispose();
1.303	us	[126472]	std::__cxx11::basic_string::_Alloc_hider::~~_Alloc_hider();
5.843	us	[126472]	} /* std::__cxx11::basic_string::~~basic_string */
0.850	us	[126472]	std::allocator::~~allocator();
44.803	us	[126472]	} /* main */

```
$ g++ -pg -O2 string.cpp -o string2
```

```
$ uftrace -F main -D 3 string2
```

```
std::string test!
```

#	DURATION	TID	FUNCTION
		[126639]	main() {
1.807	us	[126639]	operator new();
		[126639]	print_string() {
11.350	us	[126639]	std::__ostream_insert();
0.514	us	[126639]	std::__ostream_insert();
13.033	us	[126639]	} /* print_string */
2.440	us	[126639]	operator delete();
20.010	us	[126639]	} /* main */

```
$ cat string.cpp
```

```
#include <iostream>
```

```
#include <string>
```

```
void print_string(const std::string& s)
```

```
{
```

```
    std::cout << s << '\n';
```

```
}
```

```
int main()
```

```
{
```

```
    const char* msg = "std::string test!";
```

```
    print_string(msg);
```

```
}
```

```
$ cat string_view.cpp
#include <iostream>
#include <string_view>
```

```
void print_string_view(const std::string_view& sv)
{
    std::cout << sv << '\n';
}
```

```
int main()
{
    const char* msg = "std::string_view!";
    print_string_view(msg);
}
```



```
$ g++ -std=c++1z -pg string_view.cpp -o string_view
$ uftrace -F main -D 3 string_view
std::string_view!
```

```
$ g++ -std=c++1z -pg string_view.cpp -o string_view
```

```
$ uftrace -F main -D 3 string_view
```

```
std::string_view!
```

#	DURATION	TID	FUNCTION
		[126684]	main() {
		[126684]	std::basic_string_view::basic_string_view() {
0.380	us	[126684]	std::char_traits::length();
1.786	us	[126684]	} /* std::basic_string_view::basic_string_view */
		[126684]	print_string_view() {
7.970	us	[126684]	std::operator <<();
2.590	us	[126684]	std::operator <<();
11.767	us	[126684]	} /* print_string_view */
14.966	us	[126684]	} /* main */

```
$ g++ -std=c++1z -pg string_view.cpp -o string_view
```

```
$ uftrace -F main -D 3 string_view
```

```
std::string_view!
```

#	DURATION	TID	FUNCTION
		[126684]	main() {
		[126684]	std::basic_string_view::basic_string_view() {
0.380 us		[126684]	std::char_traits::length();
1.786 us		[126684]	} /* std::basic_string_view::basic_string_view */
		[126684]	print_string_view() {
7.970 us		[126684]	std::operator <<();
2.590 us		[126684]	std::operator <<();
11.767 us		[126684]	} /* print_string_view */
14.966 us		[126684]	} /* main */

```
$ g++ -std=c++1z -pg -O2 string_view.cpp -o string_view2
```

```
$ uftrace -F main -D 3 string_view2
```

```
std::string_view!
```

```
$ g++ -std=c++1z -pg string_view.cpp -o string_view
```

```
$ ufttrace -F main -D 3 string_view
```

```
std::string_view!
```

#	DURATION	TID	FUNCTION
		[126684]	main() {
		[126684]	std::basic_string_view::basic_string_view() {
0.380	us	[126684]	std::char_traits::length();
1.786	us	[126684]	} /* std::basic_string_view::basic_string_view */
		[126684]	print_string_view() {
7.970	us	[126684]	std::operator <<();
2.590	us	[126684]	std::operator <<();
11.767	us	[126684]	} /* print_string_view */
14.966	us	[126684]	} /* main */

```
$ g++ -std=c++1z -pg -O2 string_view.cpp -o string_view2
```

```
$ ufttrace -F main -D 3 string_view2
```

```
std::string_view!
```

#	DURATION	TID	FUNCTION
		[126698]	main() {
		[126698]	print_string_view() {
12.024	us	[126698]	std::__ostream_insert();
0.546	us	[126698]	std::__ostream_insert();
14.674	us	[126698]	} /* print_string_view */
15.613	us	[126698]	} /* main */

**No memory allocation and deallocation is required!**

# **STL Containers**

## **Performance Comparison**

**std::vector**

**std::deque**

**std::list**

# Benchmark

# Benchmark

```
std::vector<std::string> vec;
```

```
void bench_vector_push_back(int iter) {  
    std::string s("Hello");  
    while (iter--)  
        vec.push_back(s);  
}
```

```
int main()  
{  
    int iter = 3000000;  
    bench_vector_push_back(iter);  
}
```

# Benchmark

```
std::vector<std::string> vec;
```

```
void bench_vector_push_back(int iter) {  
    std::string s("Hello");  
    while (iter--)  
        vec.push_back(s);  
}
```

```
int main()  
{  
    int iter = 3000000;  
    bench_vector_push_back(iter);  
}
```



# Benchmark

```
std::vector<std::string> vec;  
std::deque<std::string> deq;
```

```
void bench_vector_push_back(int iter) {  
    std::string s("Hello");  
    while (iter--)  
        vec.push_back(s);  
}
```

```
void bench_deque_push_back(int iter) {  
    std::string s("Hello");  
    while (iter--)  
        deq.push_back(s);  
}
```

```
int main()  
{  
    int iter = 3000000;  
    bench_vector_push_back(iter);  
    bench_deque_push_back(iter);  
}
```

# Benchmark

```
std::vector<std::string> vec;
std::deque<std::string> deq;
std::list<std::string> lis;

void bench_vector_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        vec.push_back(s);
}

void bench_deque_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        deq.push_back(s);
}

void bench_list_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        lis.push_back(s);
}

int main()
{
    int iter = 3000000;
    bench_vector_push_back(iter);
    bench_deque_push_back(iter);
    bench_list_push_back(iter);
}
```

# Benchmark

```
std::vector<std::string> vec;
std::deque<std::string> deq;
std::list<std::string> lis;

void bench_vector_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        vec.push_back(s);
}

void bench_deque_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        deq.push_back(s);
}

void bench_list_push_back(int iter) {
    std::string s("Hello");
    while (iter--)
        lis.push_back(s);
}

int main()
{
    int iter = 3000000;
    bench_vector_push_back(iter);
    bench_deque_push_back(iter);
    bench_list_push_back(iter);
}
```

```
$ uftrace record \  
  -d uftrace.data.bench \  
  --nest-libcall \  
  -A malloc@arg1 -R malloc@retval -A free@arg1 \  
  -A memcpy@arg3 -A memmove@arg3 \  
  ./std-string
```

```
$ uftrace record \  
  -d uftrace.data.bench \  
  --nest-libcall \  
  -A malloc@arg1 -R malloc@retval -A free@arg1 \  
  -A memcpy@arg3 -A memmove@arg3 \  
  ./std-string
```

```
$ uftrace graph
```

**uftrace graph**

- Show function call graph

# \$ uftrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

# \$ uftrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

# \$ ufttrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```



## \$ ufttrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

## \$ uftrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

# \$ ufttrace graph

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
: | | |
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
: | |
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
: | |
14.972 ms : | +- (187500) memcpy
: | |
993.690 us : | +- (15) memmove
: | |
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
: |
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
: |
230.213 ms : +- (3000000) memcpy
: |
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

**\$ uftrace graph**

```
5.321 s : (1) main
2.176 s : +- (1) bench_vector_push_back
1.365 s : | +- (23) std::vector::_M_insert_aux
145.377 us : | | +- (23) operator new
122.596 us : | | | (23) malloc
```

**copying the original buffer  
to a new buffer**

```
685.339 ms : | | +- (4194326) memcpy
: | | |
2.888 ms : | | +- (22) operator delete
2.857 ms : | | (22) free
```

**push\_back an item**

```
336.277 ms : | +- (2999977) memcpy
: |
726.388 ms : +- (1) bench_deque_push_back
```

```
217.685 ms : | +- (2812500) memcpy
: | |
167.695 ms : | +- (187500) std::deque::_M_push_back_aux
101.126 ms : | +- (187515) operator new
60.892 ms : | | (187515) malloc
```

```
14.972 ms : | +- (187500) memcpy
```

```
993.690 us : | +- (15) memmove
```

```
12.357 us : | +- (15) operator delete
5.924 us : | (15) free
```

```
2.418 s : +- (1) bench_list_push_back
1.057 s : +- (3000000) operator new
423.438 ms : | (3000000) malloc
```

```
230.213 ms : +- (3000000) memcpy
```

```
199.812 ms : +- (3000000) std::__detail::_List_node_base::_M_hook
```

**Let's See Timeline based  
replay output**

**bench\_vector\_push\_back**

**std::vector<std::string>**

```
$ uftrace replay
```

```
    ...  
[121878] | main() {  
[121878] |     bench_vector_push_back() {  
    ...
```

**std::vector push\_back**

\$ uftrace replay

```
...
[121878] | main() {
[121878] |   bench_vector_push_back() {
[121878] |     std::vector::_M_insert_aux() {
[121878] |       operator new() {
3.533 us [121878] |         malloc(32) = 0xdc6550;
4.200 us [121878] |       } /* operator new */
2.006 us [121878] |       memcpy(5);
7.777 us [121878] |     } /* std::vector::_M_insert_aux */
...

```

std::vector push\_back



\$ uftrace replay

```
...
[121878] | main() {
[121878] |     bench_vector_push_back() {
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
3.533 us [121878] |                 malloc(32) = 0xdc6550;
4.200 us [121878] |             } /* operator new */
2.006 us [121878] |             memcpy(5);
7.777 us [121878] |         } /* std::vector::_M_insert_aux */
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
0.227 us [121878] |                 malloc(64) = 0xdc6580;
0.780 us [121878] |             } /* operator new */
0.250 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
[121878] |             operator delete() {
1.813 us [121878] |                 free(0xdc6550);
3.460 us [121878] |             } /* operator delete */
6.370 us [121878] |         } /* std::vector::_M_insert_aux */
...

```

std::vector push\_back

\$ uftrace replay

```
...
[121878] | main() {
[121878] |     bench_vector_push_back() {
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
3.533 us [121878] |                 malloc(32) = 0xdc6550;
4.200 us [121878] |             } /* operator new */
2.006 us [121878] |             memcpy(5);
7.777 us [121878] |         } /* std::vector::_M_insert_aux */
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
0.227 us [121878] |                 malloc(64) = 0xdc6580;
0.780 us [121878] |             } /* operator new */
0.250 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
[121878] |             operator delete() {
1.813 us [121878] |                 free(0xdc6550);
3.460 us [121878] |             } /* operator delete */
6.370 us [121878] |         } /* std::vector::_M_insert_aux */
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
0.244 us [121878] |                 malloc(128) = 0xdc65d0;
0.743 us [121878] |             } /* operator new */
0.186 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
[121878] |             operator delete() {
0.320 us [121878] |                 free(0xdc6580);
0.897 us [121878] |             } /* operator delete */
3.737 us [121878] |         } /* std::vector::_M_insert_aux */
...
```

**std::vector push\_back**

\$ uftrace replay

```
...
[121878] | main() {
[121878] |     bench_vector_push_back() {
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
3.533 us [121878] |                 malloc(32) = 0xdc6550;
4.200 us [121878] |             } /* operator new */
2.006 us [121878] |             memcpy(5);
7.777 us [121878] |         } /* std::vector::_M_insert_aux */
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
0.227 us [121878] |                 malloc(64) = 0xdc6580;
0.780 us [121878] |             } /* operator new */
0.250 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
[121878] |             operator delete() {
1.813 us [121878] |                 free(0xdc6550);
3.460 us [121878] |             } /* operator delete */
6.370 us [121878] |         } /* std::vector::_M_insert_aux */
[121878] |         std::vector::_M_insert_aux() {
[121878] |             operator new() {
0.244 us [121878] |                 malloc(128) = 0xdc65d0;
0.743 us [121878] |             } /* operator new */
0.186 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
0.160 us [121878] |             memcpy(5);
[121878] |             operator delete() {
0.320 us [121878] |                 free(0xdc6580);
0.897 us [121878] |             } /* operator delete */
3.737 us [121878] |         } /* std::vector::_M_insert_aux */
0.167 us [121878] |         memcpy(5);
...
```

copying the original buffer  
to a new buffer

normal push\_back  
within buffer

std::vector push\_back

```

...
[121878] | bench_vector_push_back() {
...
[121878] |     std::vector::_M_insert_aux() {
[121878] |         operator new() {
0.244 us [121878] |             malloc(128) = 0xdc65d0;
0.743 us [121878] |         } /* operator new */
0.186 us [121878] |         memcpy(5);
0.160 us [121878] |         memcpy(5);
0.160 us [121878] |         memcpy(5);
[121878] |         operator delete() {
0.320 us [121878] |             free(0xdc6580);
0.897 us [121878] |         } /* operator delete */
3.737 us [121878] |     } /* std::vector::_M_insert_aux */
0.167 us [121878] |     memcpy(5);
[121878] |     std::vector::_M_insert_aux() {
[121878] |         operator new() {
0.220 us [121878] |             malloc(256) = 0xdc6660;
0.693 us [121878] |         } /* operator new */
0.164 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.154 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
[121878] |         operator delete() {
0.407 us [121878] |             free(0xdc65d0);
0.854 us [121878] |         } /* operator delete */
4.256 us [121878] |     } /* std::vector::_M_insert_aux */
0.157 us [121878] |     memcpy(5);
0.157 us [121878] |     memcpy(5);
0.157 us [121878] |     memcpy(5);
[121878] |     std::vector::_M_insert_aux() {
[121878] |         operator new() {
0.520 us [121878] |             malloc(512) = 0xdc6770;
1.000 us [121878] |         } /* operator new */
0.154 us [121878] |         memcpy(5);
...

```

**std::vector push\_back**

```

...
[121878] | bench_vector_push_back() {
...
[121878] |     std::vector::_M_insert_aux() {
[121878] |         operator new() {
1.040 us [121878] |             malloc(1024) = 0xdc6980;
1.510 us [121878] |         } /* operator new */
0.157 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.156 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.154 us [121878] |         memcpy(5);
0.153 us [121878] |         memcpy(5);
0.150 us [121878] |         memcpy(5);
0.150 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.153 us [121878] |         memcpy(5);
0.153 us [121878] |         memcpy(5);
0.153 us [121878] |         memcpy(5);
0.146 us [121878] |         memcpy(5);
0.150 us [121878] |         memcpy(5);
0.154 us [121878] |         memcpy(5);
0.157 us [121878] |         memcpy(5);
0.156 us [121878] |         memcpy(5);
[121878] |         operator delete() {
0.274 us [121878] |             free(0xdc6770);
0.700 us [121878] |         } /* operator delete */
9.369 us [121878] |     } /* std::vector::_M_insert_aux */
0.157 us [121878] |     memcpy(5);
0.170 us [121878] |     memcpy(5);
0.160 us [121878] |     memcpy(5);
...

```

copying the original buffer  
to a new buffer

normal push\_back  
within buffer

std::vector push\_back

**bench\_deque\_push\_back**

**std::deque<std::string>**

\$ uftrace replay

```
...
[121878] | bench_deque_push_back() {
0.083 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.074 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
[121878] |     std::deque::_M_push_back_aux() {
[121878] |         operator new() {
4.682 us [121878] |             malloc(512) = 0xdc6550;
6.328 us [121878] |         } /* operator new */
0.085 us [121878] |         memcpy(5);
7.552 us [121878] |     } /* std::deque::_M_push_back_aux */
0.074 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.083 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
```

...

**std::deque push\_back**

\$ uftrace replay

```
...
[121878] | bench_deque_push_back() {
0.083 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.076 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.074 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
[121878] |     std::deque::_M_push_back_aux() {
[121878] |         operator new() {
4.682 us [121878] |             malloc(512) = 0xdc6550;
6.328 us [121878] |         } /* operator new */
0.085 us [121878] |         memcpy(5);
7.552 us [121878] |     } /* std::deque::_M_push_back_aux */
0.074 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.083 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
...
std::deque push_back
```

**push\_back inside a chunk buffer**

**15 memcpy calls:  
32 bytes (size of std::string) \* 15  
= 480 bytes spent in buffer**

**additional  
chunk  
allocation**



\$ uftrace replay

```
...
[121878] | bench_deque_push_back() {
...
[121878] |     std::deque::_M_push_back_aux() {
[121878] |         operator new() {
4.682 us [121878] |             malloc(512) = 0xdc6550;
6.328 us [121878] |         } /* operator new */
0.085 us [121878] |         memcpy(5);
7.552 us [121878] |     } /* std::deque::_M_push_back_aux */
0.074 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.083 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.075 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
0.075 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.077 us [121878] |     memcpy(5);
0.079 us [121878] |     memcpy(5);
0.080 us [121878] |     memcpy(5);
0.078 us [121878] |     memcpy(5);
[121878] |     std::deque::_M_push_back_aux() {
[121878] |         operator new() {
0.305 us [121878] |             malloc(512) = 0xdc6760;
0.882 us [121878] |         } /* operator new */
0.079 us [121878] |         memcpy(5);
1.256 us [121878] |     } /* std::deque:: M push back aux */
```

**std::deque push\_back**

**bench\_list\_push\_back**

**std::list<std::string>**

\$ uftrace replay

```
...
[121878] | bench_list_push_back() {
[121878] |     operator new() {
0.450 us [121878] |         malloc(48) = 0xdc62f0;
0.677 us [121878] |     } /* operator new */
0.080 us [121878] |     memcpy(5);
5.139 us [121878] |     std::__detail::_List_node_base::_M_hook();
[121878] |     operator new() {
0.240 us [121878] |         malloc(48) = 0xfd2580;
0.480 us [121878] |     } /* operator new */
0.083 us [121878] |     memcpy(5);
0.080 us [121878] |     std::__detail::_List_node_base::_M_hook();
[121878] |     operator new() {
0.400 us [121878] |         malloc(48) = 0xdca0a0;
0.641 us [121878] |     } /* operator new */
0.085 us [121878] |     memcpy(5);
0.071 us [121878] |     std::__detail::_List_node_base::_M_hook();
[121878] |     operator new() {
0.251 us [121878] |         malloc(48) = 0x1cac6d0;
0.479 us [121878] |     } /* operator new */
0.075 us [121878] |     memcpy(5);
0.069 us [121878] |     std::__detail::_List_node_base::_M_hook();
[121878] |     operator new() {
0.304 us [121878] |         malloc(48) = 0xdc8200;
0.511 us [121878] |     } /* operator new */
0.076 us [121878] |     memcpy(5);
0.066 us [121878] |     std::__detail::_List_node_base::_M_hook();
...
```

32 bytes of std::string  
+ 8 bytes of pointer \* 2

size of "Hello"

std::list push\_back

```
$ uftrace graph bench_list_push_back
```

```
#
```

```
# function graph for 'bench_list_push_back' (session: 53a12394b0ce1367)
```

```
#
```

```
backtrace
```

```
=====
```

```
backtrace #0: hit 1, time 2.418 s
```

```
[0] main (0x400cc9)
```

```
[1] bench_list_push_back (0x401044)
```

```
calling functions
```

```
=====
```

```
2.418 s : (1) bench_list_push_back
```

```
1.057 s : +-(3000000) operator new
```

```
423.438 ms : | (3000000) malloc
```

```
: |
```

```
230.213 ms : +-(3000000) memcpy
```

```
: |
```

```
199.812 ms : +-(3000000) std::__detail::_List_node_base::_M_hook
```

**std::list push\_back**

# **C++17 Parallel Algorithm Tracing**

```
$ cat sort.cpp
#include <vector>
#include <parallel/algorithm>

int main()
{
    std::vector<int> v(100000);

    __gnu_parallel::sort(v.begin(), v.end());
    return 0;
}
```

```
$ cat sort.cpp
#include <vector>
#include <parallel/algorithm>

int main()
{
    std::vector<int> v(100000);

    __gnu_parallel::sort(v.begin(), v.end());
    return 0;
}
```

```
$ g++-7 -pg -std=c++1z -fopenmp sort.cpp
```

```
$ uftrace record -t 1us a.out
```





# **(Python) Scripting Support**

# uftrace script

- **uftrace is able to run (python) script**
  - for each C/C++ function
  - for some specific functions defined in the script
    - **UFTRACE\_FUNC** = [ "foo", "bar", ... ]
      - function filter

# uftrace script

- **uftrace is able to run (python) script**
  - for each C/C++ function
  - for some specific functions defined in the script
    - **UFTRACE\_FUNC** = [ "foo", "bar", ... ]
      - function filter
- **Script can be executed**
  - during record time
    - able to perform additional action, but slow
  - with recorded data (uftrace.data)
    - fast and reliable, recommended
    - using "uftrace **script**" command

# uftrace script

- **uftrace script APIs**
  - uftrace\_entry(context)
  - uftrace\_exit(context)
  - uftrace\_begin()
  - uftrace\_end()

# A simple python script

- Count the number of function entries.

```
$ cat scripts/count.py  
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
$ gcc -pg test.c
```

```
void bar() {  
}
```

```
void foo() {  
    bar();  
}
```

```
int main() {  
    foo();  
}
```

```
$ gcc -pg test.c
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```



```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```



```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
```

```
$ uftrace -S count.py a.out
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():  
    pass
```

```
def uftrace_entry(args):  
    global count  
    count += 1
```

```
def uftrace_exit(args):  
    pass
```

```
def uftrace_end():  
    print(count)
```

```
# FUNCTION
```

```
main() {  
    foo() {  
        bar() {  
            } /* bar */  
        } /* foo */  
    } /* main */
```

```
$ gcc -pg test.c
$ uftrace -S count.py a.out
```

3

```
# FUNCTION
```

```
main() {
    foo() {
        bar() {
            } /* bar */
        } /* foo */
    } /* main */
```

```
$ cat count.py
```

```
count = 0
```

```
def uftrace_begin():
    pass
```

```
def uftrace_entry(args):
    global count
    count += 1
```

```
def uftrace_exit(args):
    pass
```

```
def uftrace_end():
    print(count)
```

# Context Info to Script

```
/* context information passed to script */
script_context = {
    int      tid;
    int      depth;
    long     timestamp;
    long     duration;      # exit only
    long     address;
    string   name;
    list     args;          # entry only (if available)
    value    retval;        # exit only (if available)
};
```



# Example usage (w/ matplotlib)

```
$ uftrace script -S plot-alloc-size.py --record /usr/bin/node -e ''
```

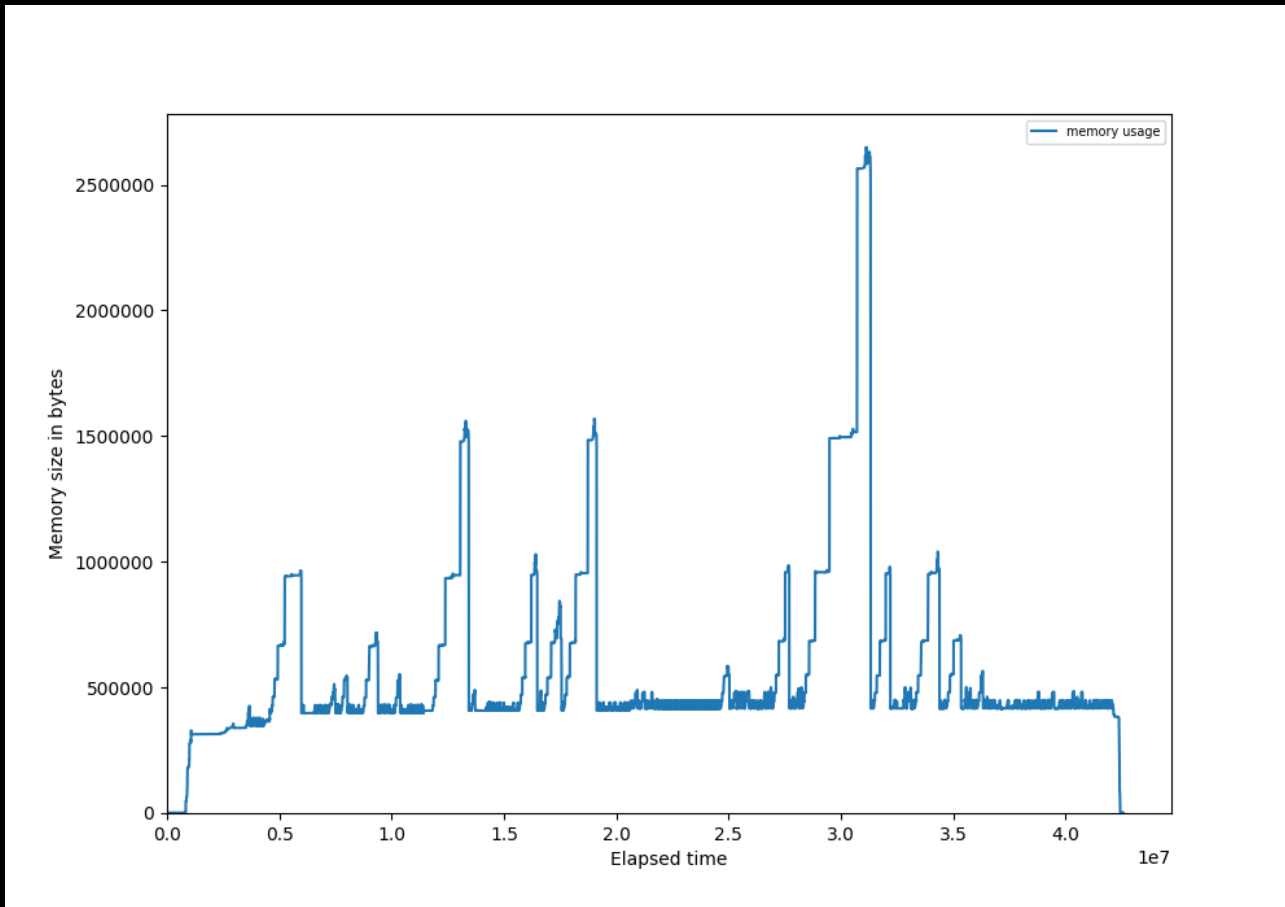
- script may contain the following comment for record options
  - # uftrace-option:

**--record** COMMAND [command-options]

Record a new trace before running a given script.

# Example usage (w/ matplotlib)

```
$ uftrace script -S plot-alloc-size.py --record /usr/bin/node -e ''
```



**--record** COMMAND [command-options]  
Record a new trace before running a given script.

**Thanks!**

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

# **Appendix**

# Tracing Issue

- **-pg**
  - inserts mcount() call at the entry of each function
  - when mcount is called, return address is replaced to ufttrace's mcount\_exit()
  - cannot see inlined functions
    - use -fno-inline-functions
    - or manually add `__attribute__((noinline))` to the function
- **-finstrument-functions**
  - inserts hooking stubs both at entry and exit
  - able to see inlined functions but slower
- **-fxray-instrument**
  - experimental support as of yet