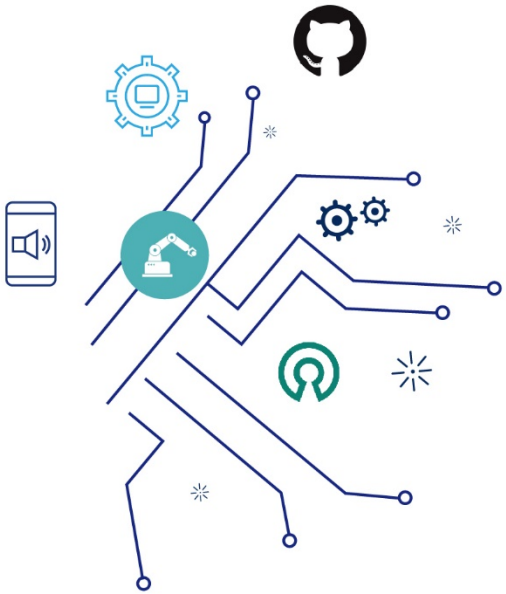
2019

공개SW 컨트리뷰톤



uftrace

주최 과학기술정보통신부 주관 nipa 정보통신산업진흥원

▪ 프로젝트 소개

- 프로젝트명

  uftrace

- 프로젝트 분야

  Tracing, Debugging, Profiling 및 프로그램 분석 도구

- 프로젝트 저장소

  https://github.com/namhyung/uftrace

- 활용 언어

  C / Python

- 프로젝트 난이도

  초/중급

- 참가자 모집 유형

  - 오픈소스 프로젝트 개발에 참여하고 싶으나 어려움을 겪고 계신 분
  - 리눅스 기반 C/C++ 프로그램 Tracing 도구 개발에 관심 있는 분
  - 컴파일러 및 바이너리 구조 분석에 관심 있는 분
  - 다양한 프로젝트 내부 분석에 관심 있는 분 (ex. C++ STL)

▪ 프로젝트 개요

- uftrace 는 코드 수정 없이 C/C++ 프로그램의 성능 측정과 실행 흐름을 추적(trace)하는 분석 도구로 소규모 개인 프로젝트부터 큰 규모의 오픈소스 프로젝트를 분석 및 디버깅 하는데 활용되고 있습니다.

  2016년에 github 에 공개된 이후로 리눅스 tracing summit 과 CppCon 같은 해외 컨퍼런스에 소개되며 사용자 층을 넓혀가고 있는 오픈소스 프로젝트입니다.

컨트리뷰톤 가이드

- **<STAGE 1> Git/Github 협업방법 훈련**
  - Git/Github 의 기본 실습
  - 협업 시 사용하는 명령 및 이슈 해결법 이해 (rebase 등)
  - 오픈소스 개발방식 이해

- **<STAGE 2> uftrace 프로젝트 소개**
  - 프로젝트에 대한 이해 (사용 분야 및 프로젝트 정의)
  - 기본적인 function tracing 실습 (record, replay)
  - 주요 오픈소스 프로젝트 분석 사례 공유 (node.js, clang, chrome 등)
  - 참고자료 1: https://uftrace.github.io/
  - 참고자료 2: https://github.com/namhyung/uftrace/wiki

▪ 컨트리뷰톤 가이드

- **<STAGE 3> 프로젝트 개발 환경 구성**
  - **개발** 및 테스트 환경 구성: IDE 및 편집기와 Debugging 툴 세팅
  - Git 개발 환경 구성: local, remote repo, upstream 구조 잡기
  - 소스 컴파일, 실행: 기본적인 example 실행
  - 개인적으로 분석하고 싶은 프로젝트를 uftrace 로 분석 시도

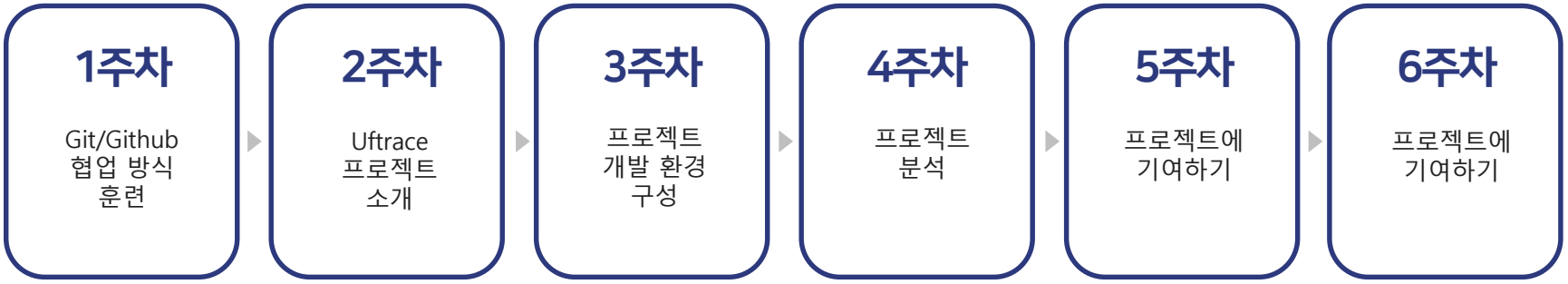
    Ex) node.js **엔진 분석**, cpython internal, MYSQL InnoDB **동작 과정 등**

- **<STAGE 4> 프로젝트 분석**
  - 바이너리 구조 분석을 위한 ELF, PLT/GOT, calling convention 등 배경 지식 이해
  - uftrace 주요 기능에 대한 소스 분석
  - 참가자별 (조별) 하나의 기능을 선정해서 그것이 실행되는 과정 분석

▪ 컨트리뷰톤 가이드

- **<STAGE 5> 프로젝트에 기여하기 (Contribution)**
  - 버그/리팩토링/Minor feature 패치 거리 찾기
  - 분석한 기능을 바탕으로 기능의 개선점, 리팩토링 여부 등에 대해 고민 및 구현
  - Pull Request 테스트 및 전송
  - 다양한 수정 사항(commit)을 해당 프로젝트에 Pull-Request 를 통해 기여
  - 다양한 프로젝트 usecase 에 대해 문서화 (wiki)
  - 참고자료: https://github.com/namhyung/uftrace/wiki/uftrace-for-node

▪ 운영 방안



- 팀 내 3~4 조 구성 (한 조당 3 명)
- 개인별, 조별 온라인 미션 진행 (문서, 코드, test 등 기여)
- 조별 온/오프라인 모임 주 1 회 진행
- 전체 오프라인 모임 격주 1 회 (세미나, 실습)
- 오프라인 모임: 선릉 Kosslab 센터, 주 1회 저녁 7시 반 (추후 멘티들과 협의)
- 온라인 모임, 질의응답 - github repo, https://gitter.im/uftrace/uftrace

멘토 소개

약 력
(현) Kosslab 6기 전담 개발자 - Linux Kernel BPF/XDP
(전) 디브레인 사이언스 - 공동 창업
Usenix Vault 19` 튜토리얼 세션 발표 - Performance Analysis in Linux Storage Stack with BPF
Kong API Gateway 기술 서적 저술 - Kong: Becoming a King of API Gateways
SW 마에스트로 7기 기술인증

- 성명 : 이호연
- 소속/직급 : Kosslab 전담 개발자

▪ 멘토 소개

약 력
(현) LG전자 선임연구원
서울대 컴퓨터공학 박사 수료
국민대 컴퓨터공학 학사 졸업
CppCon 2017 main program 발표 "Understanding the runtime behaviors of C++ programs using uftrace tool"
CppCon 2016 포스터 세션 People's choice award 수상 CppCon 2016 lightning talk 발표 "uftrace: A function graph tracer for C/C++ userspace programs"

- **성명** : 김홍규
- **소속/직급** : LG전자 / 선임연구원

감사합니다

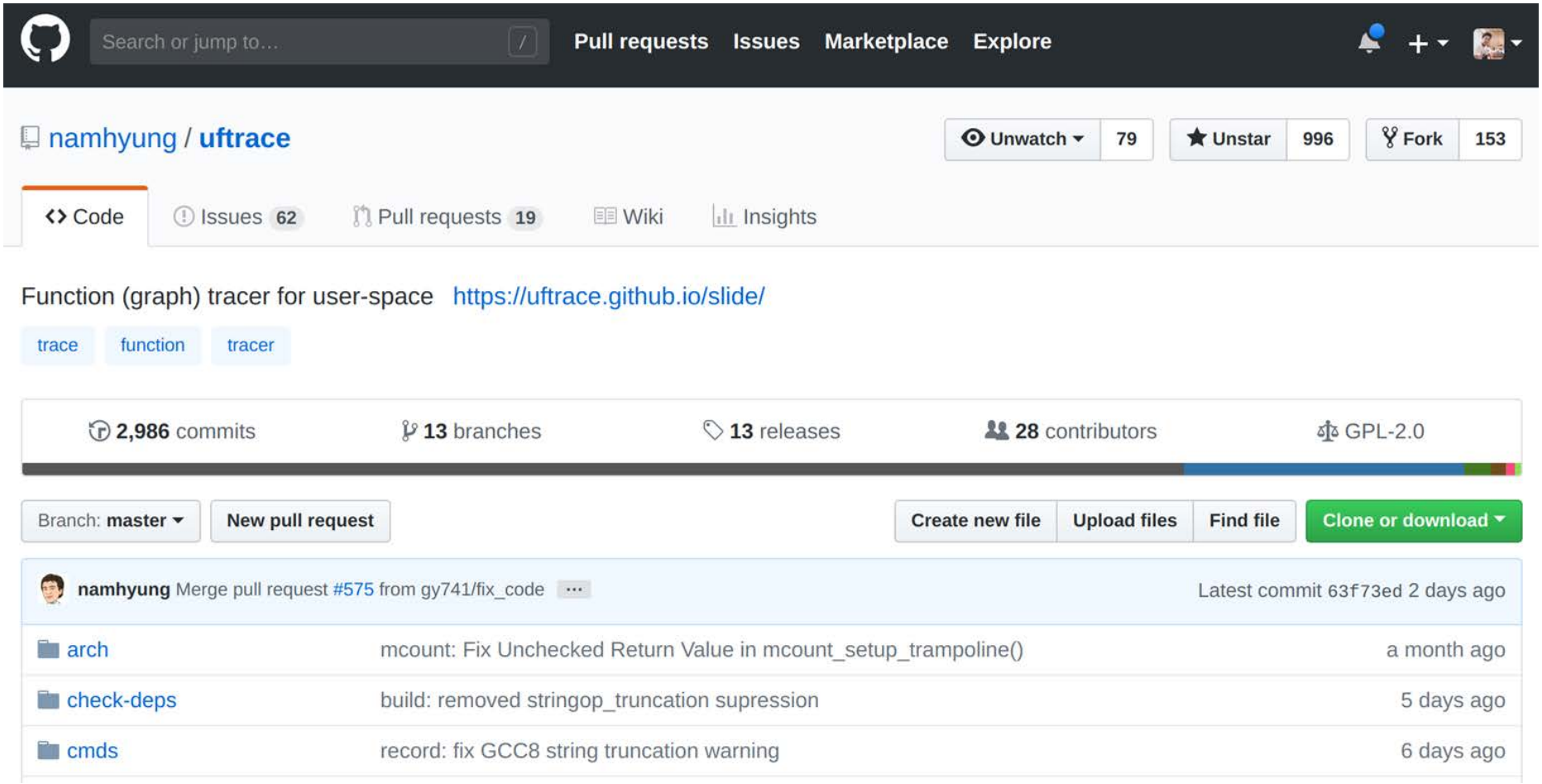
Introduction to uftrace

uftrace

Function tracer for C/C++ programs

- created by Namhyung Kim
  - LG Electronics open-source contribution team
  - Linux kernel developer (since 2010)
    - perf, ftrace, …

- inspired by ftrace framework in the kernel
- **record** and **replay** model


Search or jump to…
Pull requests
Issues
Marketplace
Explore
namhyung / uftrace
Unwatch 79
Unstar 996
Fork 153
Code
Issues 62
Pull requests 19
Wiki
Insights
Function (graph) tracer for user-space https://uftrace.github.io/slide/
trace
function
tracer
2,986 commits
13 branches
13 releases
28 contributors
GPL-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
namhyung Merge pull request #575 from gy741/fix_code
Latest commit 63f73ed 2 days ago
arch
mcount: Fix Unchecked Return Value in mcount_setup_trampoline()
a month ago
check-deps
build: removed stringop_truncation supression
5 days ago
cmds
record: fix GCC8 string truncation warning
6 days ago

README.md

build passing coverity passed

uftrace

The uftrace tool is to trace and analyze execution of a program written in C/C++. It was heavily inspired by the ftrace framework of the Linux kernel (especially function graph tracer) and supports userspace programs. It supports various kind of commands and filters to help analysis of the program execution and performance.

```
$ sudo uftrace -k a.out
[sudo] password for honggyu:
Hello World!
# DURATION    TID     FUNCTION
   1.116 us [ 6267] | __monstartup();
   0.603 us [ 6267] | __cxa_atexit();
            [ 6267] | main() {
            [ 6267] |   printf() {
            [ 6267] |     sys_newfstat() {
```

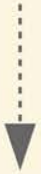
```
$ gcc -pg -o fibonacci tests/s-fibonacci.c

$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
# DURATION    TID     FUNCTION
   0.633 us [ 2851] | __monstartup();
   0.480 us [ 2851] | __cxa_atexit();
            [ 2851] | main() {
   0.546 us [ 2851] |   atoi();
            [ 2851] |   fib(5) {
            [ 2851] |     fib(4) {
            [ 2851] |       fib(3) {
   1.146 us [ 2851] |         fib(2) = 1;
   0.077 us [ 2851] |         fib(1) = 1;
   1.823 us [ 2851] |       } = 2; /* fib */
   0.062 us [ 2851] |       fib(2) = 1;
   2.199 us [ 2851] |     } = 3; /* fib */
            [ 2851] |     fib(3) {
   0.061 us [ 2851] |       fib(2) = 1;
   0.067 us [ 2851] |       fib(1) = 1;
   0.474 us [ 2851] |     } = 2; /* fib */
   3.317 us [ 2851] |   } = 5; /* fib */
   4.343 us [ 2851] | } /* main */
```

uftrace

“C/C++ execution flow”

```
# Default == record + replay

$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
# DURATION     TID     FUNCTION
   0.633 us [ 2851] | __monstartup();
   0.480 us [ 2851] | __cxa_atexit();
            [ 2851] | main() {
   0.546 us [ 2851] |   atoi();
            [ 2851] |   fib(5) {
            [ 2851] |     fib(4) {
            [ 2851] |       fib(3) {
   1.146 us [ 2851] |         fib(2) = 1;
   0.077 us [ 2851] |         fib(1) = 1;
   1.823 us [ 2851] |       } = 2; /* fib */
   0.062 us [ 2851] |       fib(2) = 1;
   2.199 us [ 2851] |     } = 3; /* fib */
            [ 2851] |     fib(3) {
   0.061 us [ 2851] |       fib(2) = 1;
   0.067 us [ 2851] |       fib(1) = 1;
   0.474 us [ 2851] |     } = 2; /* fib */
   3.317 us [ 2851] |   } = 5; /* fib */
   4.343 us [ 2851] | } /* main */
```

uftrace



“C/C++ execution flow”

Function Call Trace

```
# Default == record + replay

$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
# DURATION     TID     FUNCTION
   0.633 us [ 2851] | __monstartup();
   0.480 us [ 2851] | __cxa_atexit();
            [ 2851] | main() {
   0.546 us [ 2851] |   atoi();
            [ 2851] |   fib(5) {
            [ 2851] |     fib(4) {
            [ 2851] |       fib(3) {
   1.146 us [ 2851] |         fib(2) = 1;
   0.077 us [ 2851] |         fib(1) = 1;
   1.823 us [ 2851] |       } = 2; /* fib */
   0.062 us [ 2851] |       fib(2) = 1;
   2.199 us [ 2851] |     } = 3; /* fib */
            [ 2851] |     fib(3) {
   0.061 us [ 2851] |       fib(2) = 1;
   0.067 us [ 2851] |       fib(1) = 1;
   0.474 us [ 2851] |     } = 2; /* fib */
   3.317 us [ 2851] |   } = 5; /* fib */
   4.343 us [ 2851] | } /* main */
```

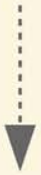
uftrace

“C/C++ each function

Execution time”

```
# Default == record + replay

$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
# DURATION     TID     FUNCTION
   0.633 us [ 2851] | __monstartup();
   0.480 us [ 2851] | __cxa_atexit();
            [ 2851] | main() {
   0.546 us [ 2851] |   atoi();
            [ 2851] |   fib(5) {
            [ 2851] |     fib(4) {
            [ 2851] |       fib(3) {
   1.146 us [ 2851] |         fib(2) = 1;
   0.077 us [ 2851] |         fib(1) = 1;
   1.823 us [ 2851] |       } = 2; /* fib */
   0.062 us [ 2851] |       fib(2) = 1;
   2.199 us [ 2851] |     } = 3; /* fib */
            [ 2851] |     fib(3) {
   0.061 us [ 2851] |       fib(2) = 1;
   0.067 us [ 2851] |       fib(1) = 1;
   0.474 us [ 2851] |     } = 2; /* fib */
   3.317 us [ 2851] |   } = 5; /* fib */
   4.343 us [ 2851] | } /* main */
```

uftrace



"Arguments"

based on
Function Call Trace

```
# Default == record + replay

$ uftrace -A fib@arg1 -R fib@retval fibonacci 5
# DURATION     TID     FUNCTION
   0.633 us [ 2851] | __monstartup();
   0.480 us [ 2851] | __cxa_atexit();
            [ 2851] | main() {
   0.546 us [ 2851] |   atoi();
            [ 2851] |   fib(5) {
            [ 2851] |     fib(4) {
            [ 2851] |       fib(3) {
   1.146 us [ 2851] |         fib(2) = 1;
   0.077 us [ 2851] |         fib(1) = 1;
   1.823 us [ 2851] |       } = 2; /* fib */
   0.062 us [ 2851] |       fib(2) = 1;
   2.199 us [ 2851] |     } = 3; /* fib */
            [ 2851] |     fib(3) {
   0.061 us [ 2851] |       fib(2) = 1;
   0.067 us [ 2851] |       fib(1) = 1;
   0.474 us [ 2851] |     } = 2; /* fib */
   3.317 us [ 2851] |   } = 5; /* fib */
   4.343 us [ 2851] | } /* main */
```

uftrace

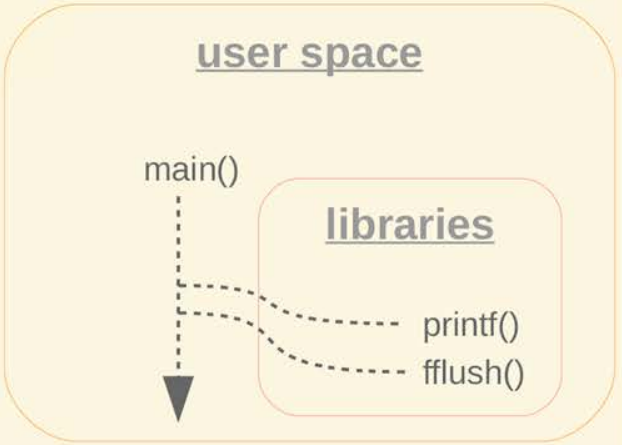
"Return values"

based on
Function Call Trace

uftrace can trace User + Lib + Kernel

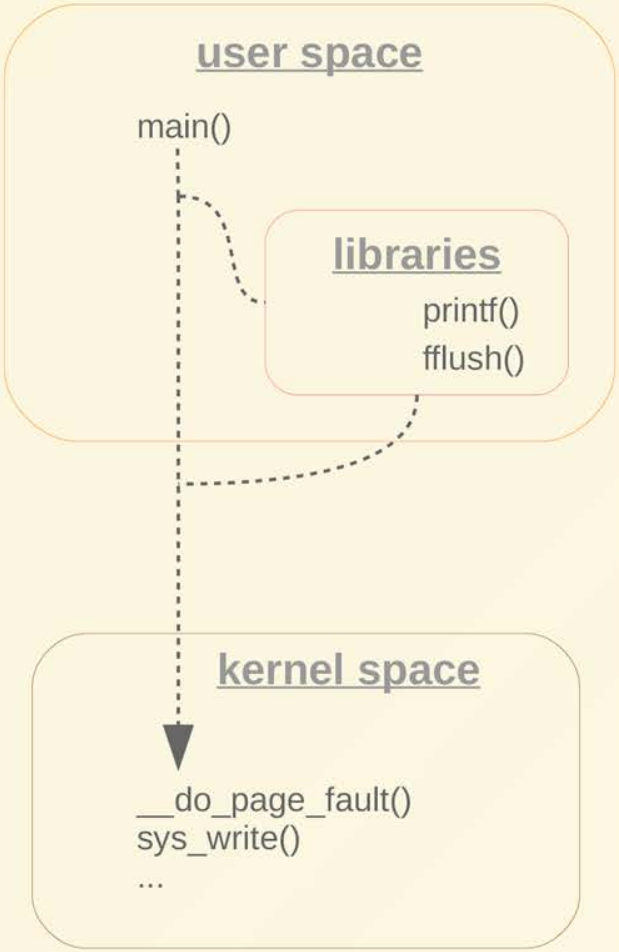
showing the execution flow

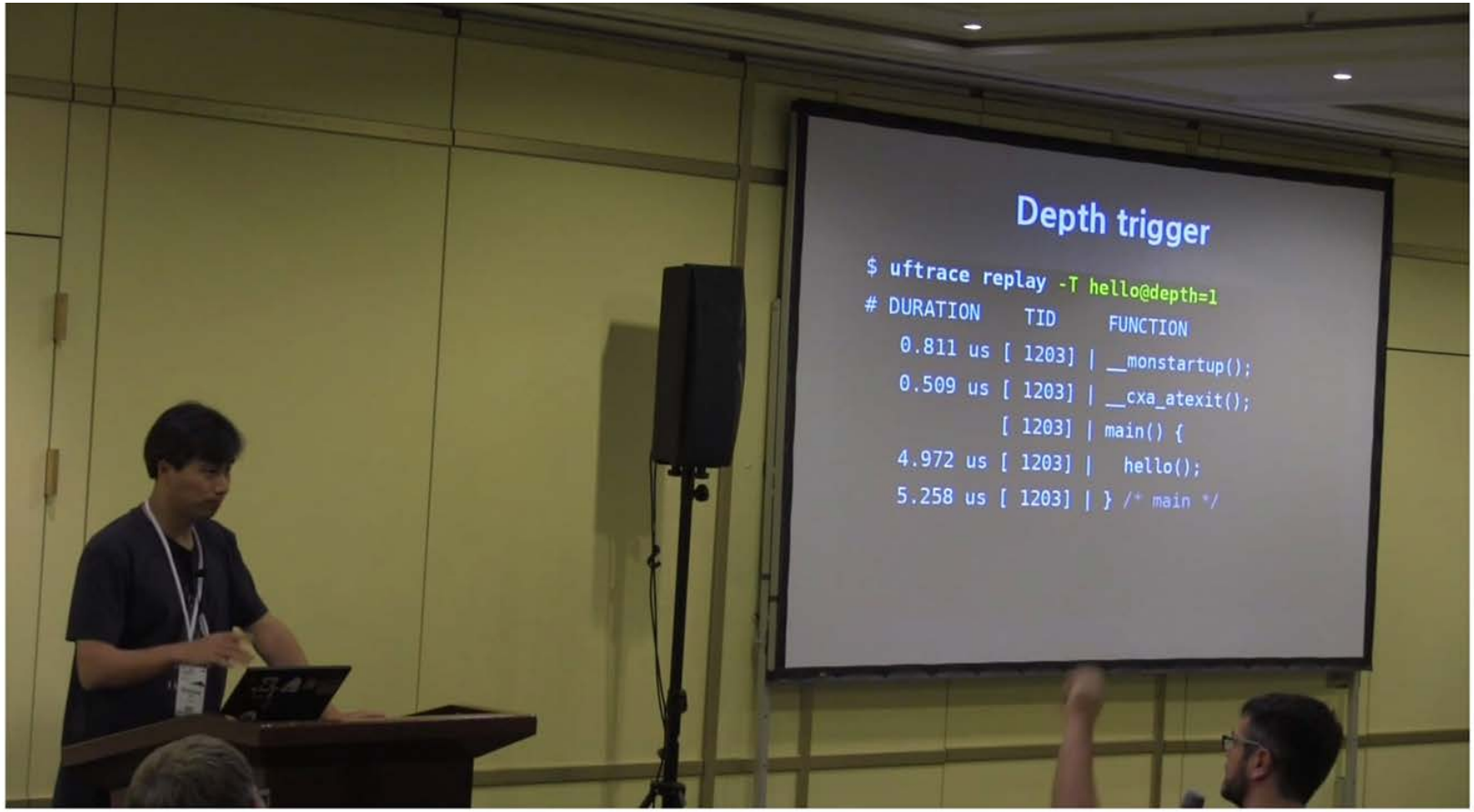
```
$ uftrace record hello
Hello OSSEU17 !!

$ uftrace replay
# DURATION    TID     FUNCTION
   0.710 us [13588] | __monstartup();
   0.713 us [13588] | __cxa_atexit();
            [13588] | main() {
   4.107 us [13588] |   printf();
   4.046 us [13588] |   fflush();
   8.815 us [13588] | } /* main */
```



```
$ uftrace record -k hello
Hello OSSEU17 !!

$ uftrace replay
# DURATION    TID     FUNCTION
   1.060 us [13565] | __monstartup();
   1.113 us [13565] | __cxa_atexit();
            [13565] | main() {
            [13565] |   printf() {
   3.173 us [13565] |     sys_newfstat();
   6.107 us [13565] |     __do_page_fault();
  17.713 us [13565] |   } /* printf */
            [13565] |   fflush() {
   7.198 us [13565] |     sys_write();
  12.270 us [13565] |   } /* fflush */
  30.661 us [13565] | } /* main */
```

Integrated tracer !





http://tracingsummit.org/wiki/TracingSummit2016uftrace

https://youtu.be/U3hTR6-pWu0

cppcon | 2016

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

HONGGYU KIM

uftrace: A function graph tracer for C/C++ userspace programs

CppCon.org

```
$ 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 */
```

https://cppcon.org/ https://youtu.be/LNav5qvyK7I

https://github.com/CppCon/CppCon2016/blob/master/Lightning%20Talks%20and%20Lunch%20Sessions/uftrace%20-%20A%20function%20graph%20tracer%20C%20C%2B%2B%20userspace%20programs/uftrace%20-%20A%20function%20graph%20tracer%20C%20C%2B%2B%20userspace%20programs%20-%20Namhyung%20Kim%20and%20Honggyu%20Kim%20-%20CppCon%202016.pdf