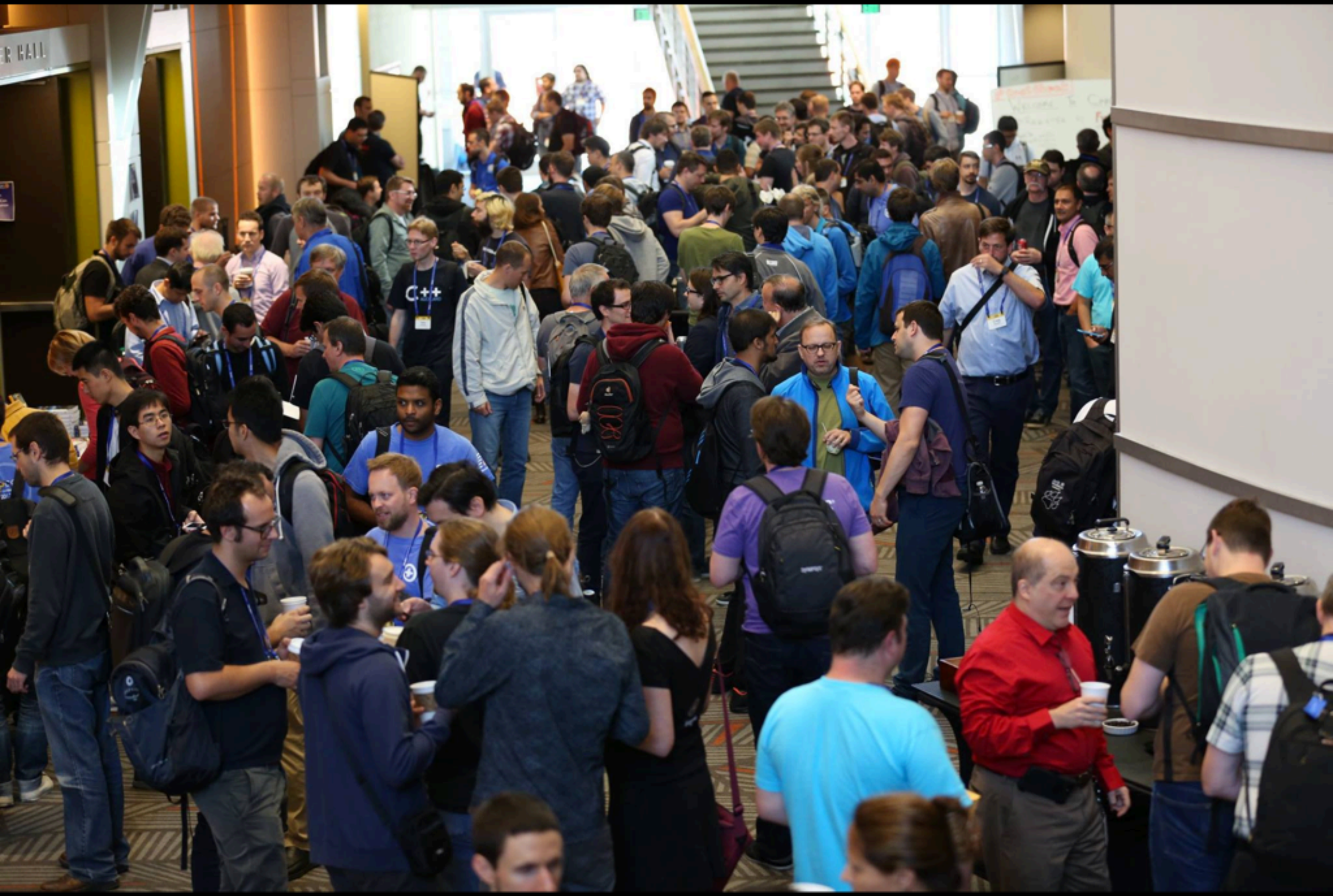
Trip reports

Piotr Padlewski

CppCon 2017

- Bellevue (WA), USA, koniec września 2017
- Tygodniowa konferencja
- Miałem okazje być bo:
  - prezentowałem Undefined Behavior is Awesome
  - byłem na praktykach w MS w Redmond w tym czasie


C++

gsl::owner<>

```
template <class T, class =
std::enable_if_t<std::is_pointer<T>::value>>
using owner = T;
```


Fact of Life: Abstractions are hiders
It's not a problem, it's the point.
⇒ Abstractions need tool support.
(But also: Good abstractions do need to be toolable.)

When a Microsecond Is an Eternity: High Performance Trading Systems in C++

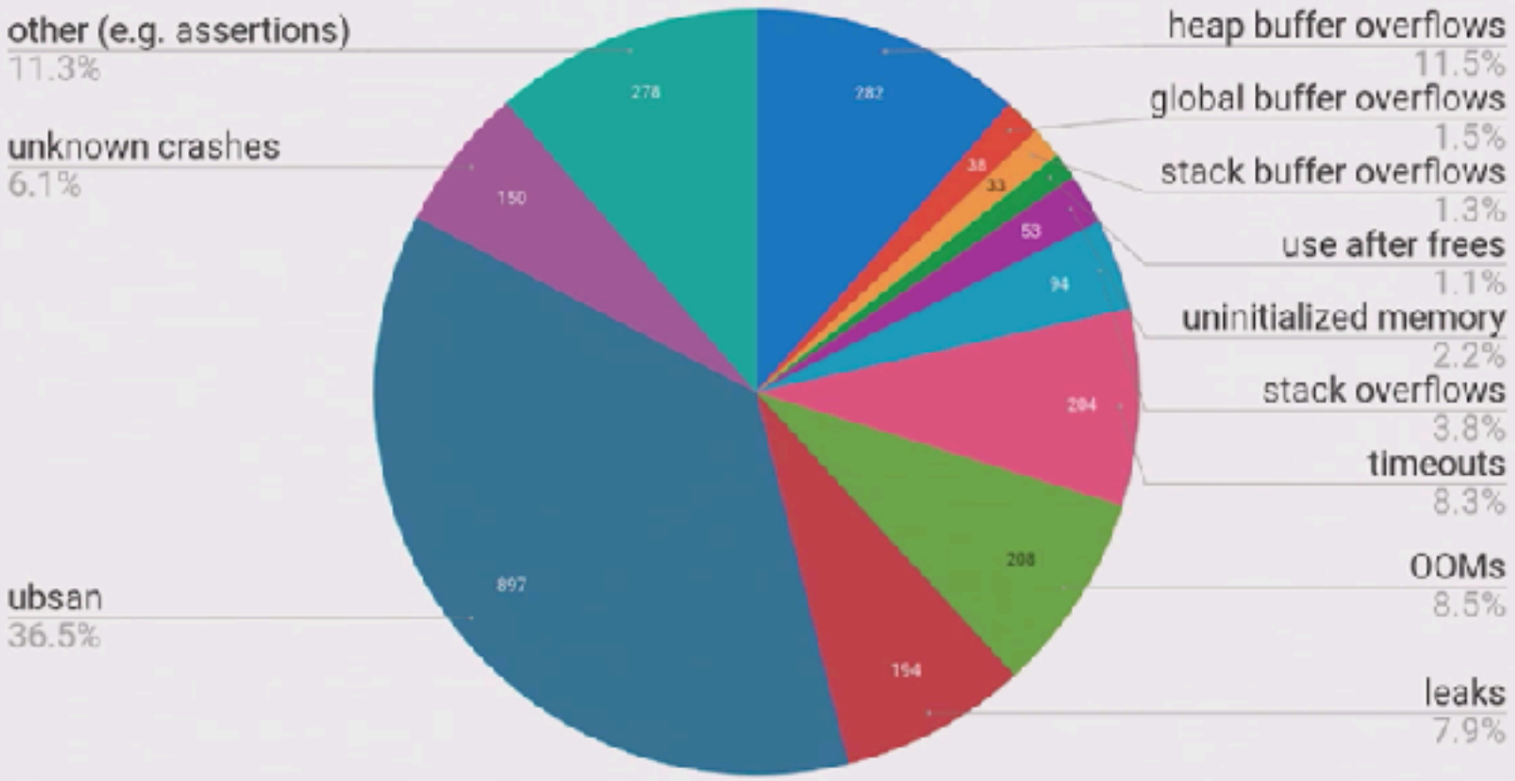
Carl Cook, Ph.D. Optiver

https://www.youtube.com/watch?v=NH1Tta7purM

- High frequency trading nie używa Profile Guided Optimizations - serwisy zwykle nie tradują ciągle więc profile mówi że kod tradujący (na szybkości którego najbardziej im zależy) jest zimny

- Używają likely/unlikely dużo

OSS-Fuzz: 2000+ bugs in 60+ OSS projects



Fuzz or lose! Why and how to make fuzzing a standard practice for C++

https://www.youtube.com/watch?v=k-Cv8Q3zWNQ

- Automated fuzzing tak jak unit testy
- prezentacje Kostya zawsze spoko

https://www.youtube.com/watch?v=JYG5LFHkUuE

- Z panelu można między innymi dowiedzieć się co to monada (monoid w kategorii endo funktorów)
- oraz wiele innych

cppcon | 2017
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

C++ as a
"Live at Head"
Language

Live At Head

https://www.youtube.com/watch?v=tISy7EJQPzI

- Google otworzyło bibliotekę używaną wewnętrznie - Abseil
- Główne założenie - brak kompatybilności wstecznej + dostarczenie narzędzia które przekonwertowuje kod ze starej wersji do nowej
- Live at head - pracuj na najnowszej wersji kompilując ją samemu

cppcon | 2017
THE C++ CONFERENCE • BELLEVUE, WASHINGTON

Meta: Thoughts on generative C++

interface (user code)

C++17

```
class Shape {
public:
    virtual int area() const =0;
    virtual void scale_by(double factor) =0;
    virtual ~Shape() noexcept { };

    // careful not to write a nonpublic or
    // nonvirtual function, or a copy/move
    // operation, or a data member; no
    // enforcement under maintenance
};
```

Proposed

```
interface Shape {
    int area() const;
    void scale_by(double factor);
};
```

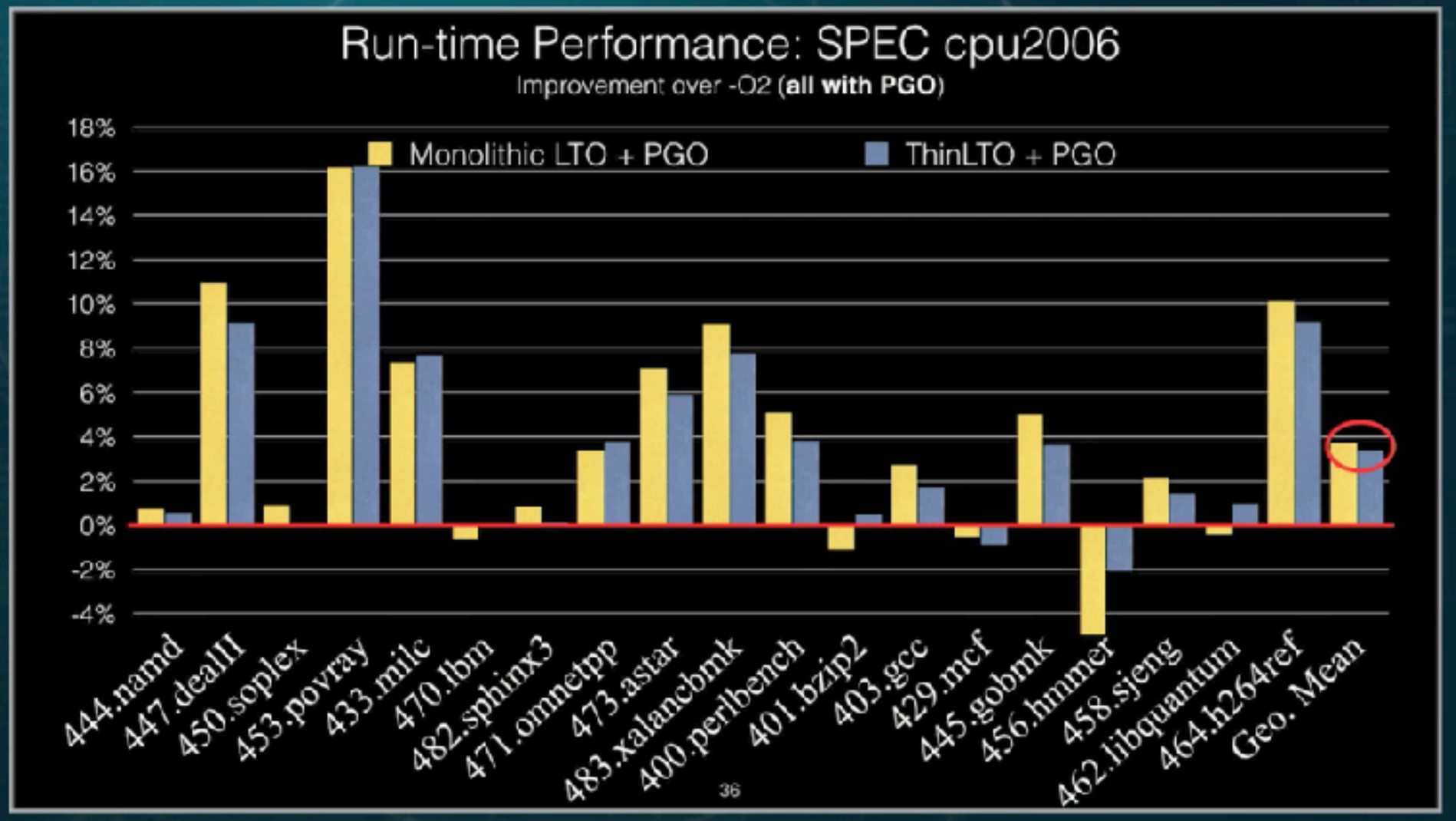
default + enforce: all public pure virtual functions
enforce: no data members, no copy/move

46

https://www.youtube.com/watch?v=4AfRAVcThyA

- Metaklasy - sposób tworzenia nowego “typu” klas
- metaklasa tak na prawdę generuje klasę w której kompilator generuje odpowiednie pola i funkcje
- przykładowo możemy wygenerować funkcje która przeiteruje się po wszystkich polach i zserializuje je
- Metaklasy mogą powodować powstawanie dialektów ;/

ThinLTO: Scalable and Incremental Link-Time Optimization



https://www.youtube.com/watch?v=p9nH2vZ2mNo

- Jak działa ThinLTO, dlaczego dobrze się skaluje i dlaczego nie powinno się używać LTO

```
#in

int
   i
   i
   s
   r
}
```

This is like saying:

"Somebody once told me that in basketball you can't hold the ball and run. I got a basketball and tried it and it worked just fine. He obviously didn't understand basketball."

http://www.eskimo.com/~scs/readings/undef.950311.html

You

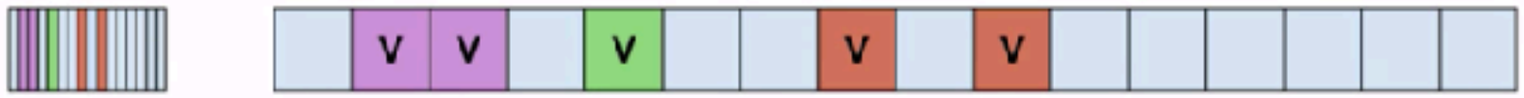
"This

support unaligned data accesses

https://www.youtube.com/watch?v=v1COuU2vU_w

- Dużo fajnych przykładów UB
- Przykłady narzędzi które znajdują UB

flat_hash_set฿




```
void erase(iterator it) {
  --size_;
  Group g{(it.ctrl_ - ctrl_) / 16 * 16 + ctrl_};
  *it.ctrl_ = g.MatchEmpty() ? kEmpty : kDeleted;
  it.slot_.~K()
}
```

฿90% true

Designing a Fast, Efficient, Cache-friendly Hash Table, Step by Step

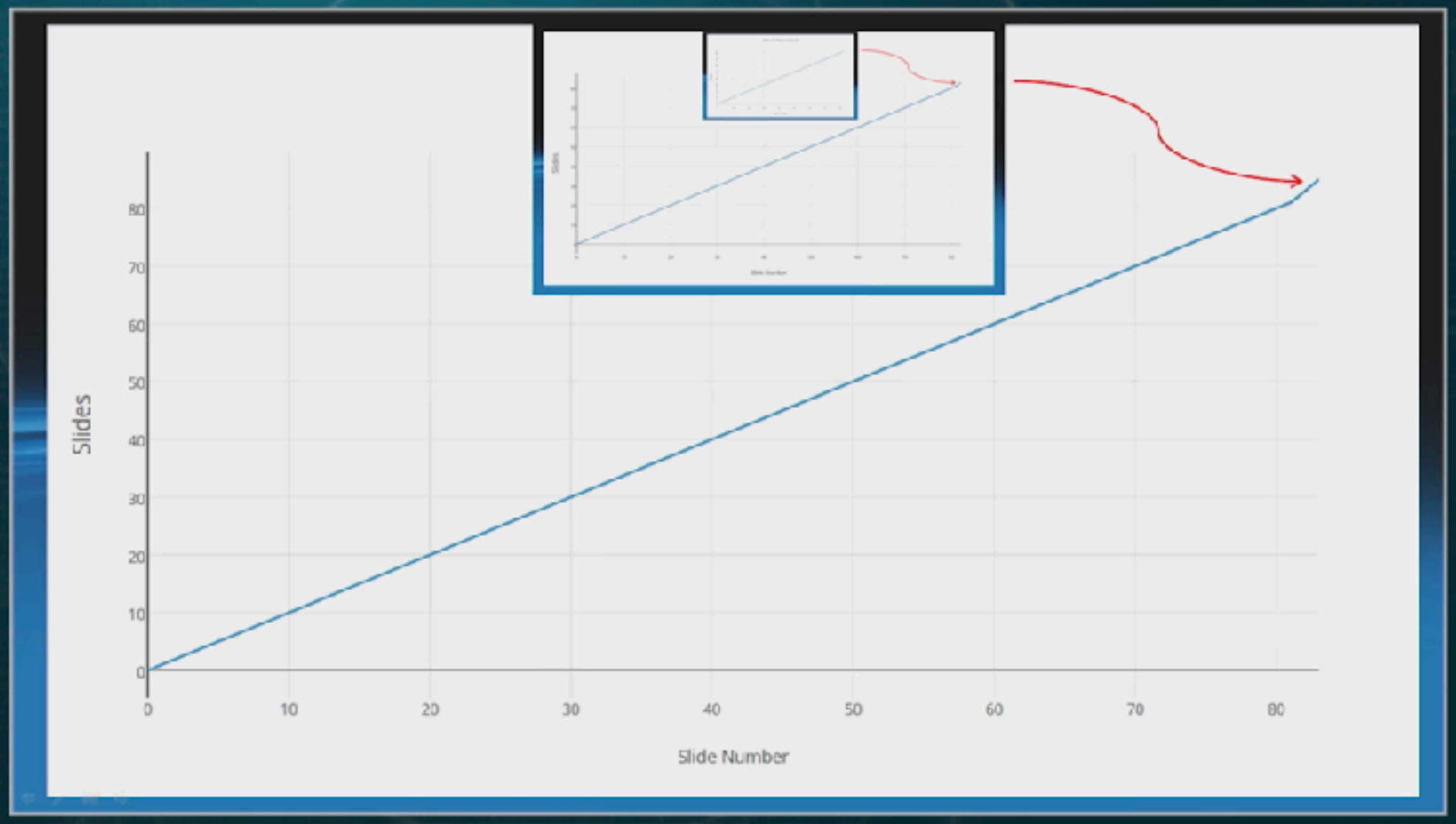
https://www.youtube.com/watch?v=ncHmEUmJZf4

- Jak działa swiss table - hashmapa od googla
- ciekawe tricki aby unikać undirection

cppcon | 2017

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

Postmodern C++



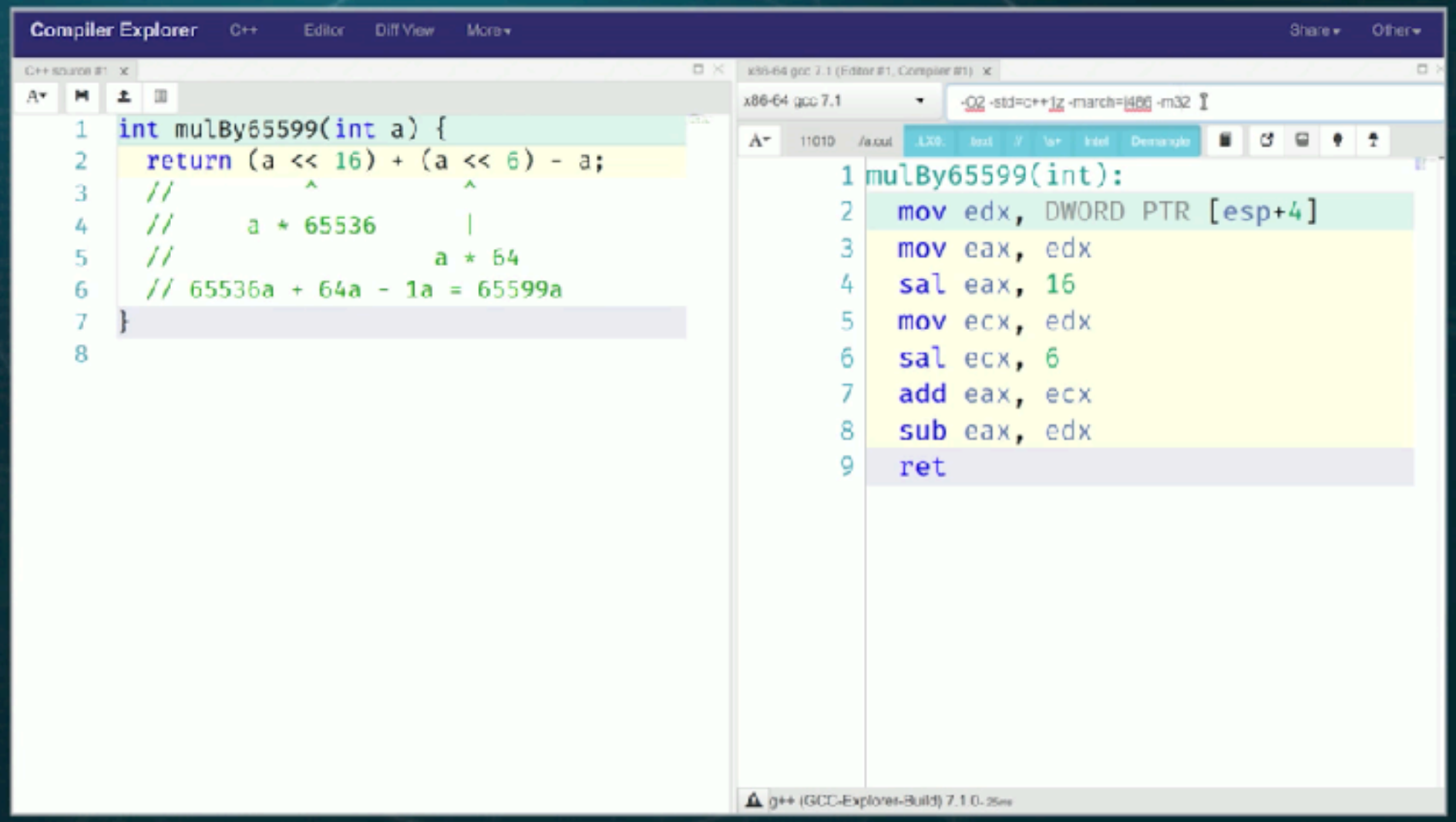
https://www.youtube.com/watch?v=QTLn3goa3A8

- 1. jest śmieszna

cppcon | 2017

THE C++ CONFERENCE • BELLEVUE, WASHINGTON

What Has My Compiler Done for Me Lately? Unbolting the Compiler's Lid



https://www.youtube.com/watch?v=bSkpMdDe4g4

```
I  chandlerc@balerion ~/s/going-nowhere-faster  master ↑1…                Fri 29 Sep 2017 06:05:[0/4195]
❯ ninja                                                                  Fri 29 Sep 2017 06:21:35 AM PDT
[2/2] LINK clamp_bench
I  chandlerc@balerion ~/s/going-nowhere-faster  master ↑1…
❯ ./cache_bench                                                  559ms 〈 Fri 29 Sep 2017 06:40:31 AM PDT
Run on (128 X 2200 MHz CPU s)
2017-09-29 06:40:41
----------------------------------------------------------
Benchmark                Time           CPU Iterations
----------------------------------------------------------
cacheBench/13          422 ns        422 ns    1658922   18.0954GB/s 8kb
cacheBench/14          834 ns        834 ns     840089   18.3032GB/s 16kb
cacheBench/15         1672 ns       1672 ns     419090   18.2524GB/s 32kb
cacheBench/16         3548 ns       3548 ns     197775   17.2037GB/s 64kb
cacheBench/17         7901 ns       7901 ns      88498   15.4496GB/s 128kb
cacheBench/18        17909 ns      17908 ns      39075   13.6328GB/s 256kb
cacheBench/19        39739 ns      39739 ns      17507   12.2873GB/s 512kb
cacheBench/20        91956 ns      91955 ns       7586    10.62GB/s 1024kb
cacheBench/21       207311 ns     207308 ns       3386   9.42139GB/s 2048kb
cacheBench/22       439510 ns     439498 ns       1597   8.88798GB/s 4096kb
cacheBench/23      1119992 ns    1119957 ns        629   6.97571GB/s 8192kb
cacheBench/24      5736579 ns    5736398 ns        121   2.72383GB/s 16384kb
cacheBench/25     16324341 ns   16323650 ns         43    1.9144GB/s 32768kb
cacheBench/26     39511164 ns   39509812 ns         18   1.58189GB/s 65536kb
I  chandlerc@balerion ~/s/going-nowhere-faster  master ↑1…
❯                                                         14.6s 〈 Fri 29 Sep 2017 06:40:56 AM PDT

[ balerion                0:fish- 1:fish* 2:fish#                [ 2017-09-29 06:41:20 (PDT) ]
```

Going Nowhere Faster

https://www.youtube.com/watch?v=2EWejmkKIxs

- Optymalizacje i architektura procesora

LLVM dev meeting

San Jose, 18-19.10.2017

BONUS - How to deprecate something in a short time!

- STLPort (a C++ runtime library) was a blocker for switching to Clang (and libc++).
- "Unbundled" Android 1st party apps didn't want to switch to libc++/Clang.
- It's hard to incentivize good behavior.
  - "Nothing really changes", maintenance is viewed as "unnecessary churn", ...
  - But we **want/need** to remove deprecated components in a reasonable timeframe.
  - Sound familiar yet? This story probably resonates with many of us here.
- Enter the "Sleep Constructor".

Google

LLVM DEVELOPERS' MEETING
2017 · SAN JOSE, CA

Compiling Android userspace and Linux kernel with LLVM

https://www.youtube.com/watch?v=6l4DtR5exwo

The Sleep Constructor

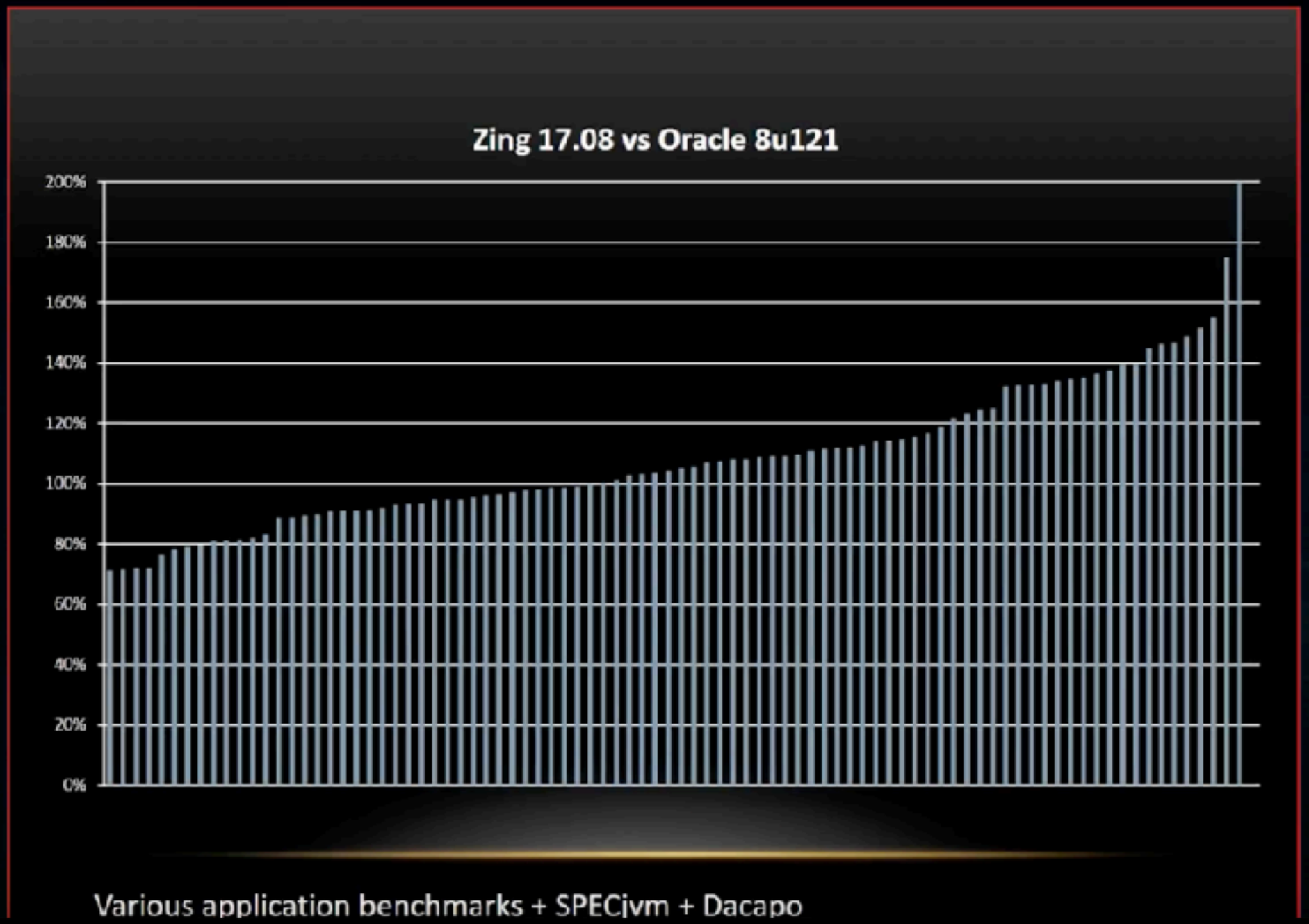
```
__attribute__((constructor))
void incentivize_stlport_users() {
  ALOGE("Hi! I see you're still using stlport. Please stop doing that.\n");
  ALOGE("All you have to do is delete the stlport lines from your makefile\n");
  ALOGE("and then you'll get the shiny new libc++\n");
  sleep(8);
}
```

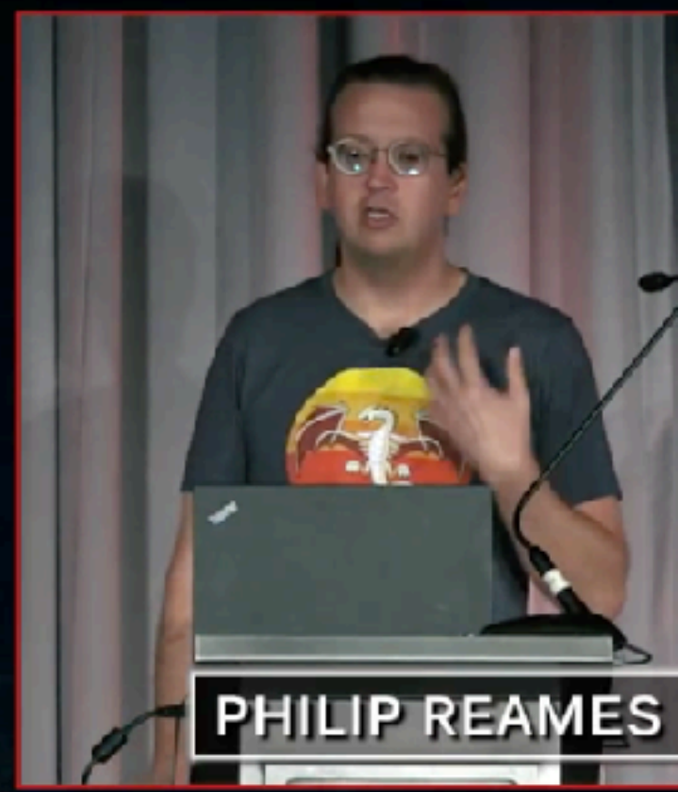
- Seriously, we added an 8 second sleep in May 2015! (AOSP)
- And then we doubled it to 16 seconds in June 2015!
- Deleted it in August 2015, because no one was left using STLPort!

Google

Compiling Android userspace and Linux kernel with LLVM

- w jaki sposób używają llvm na androidzie
- Jak skompilowano kernela clangiem





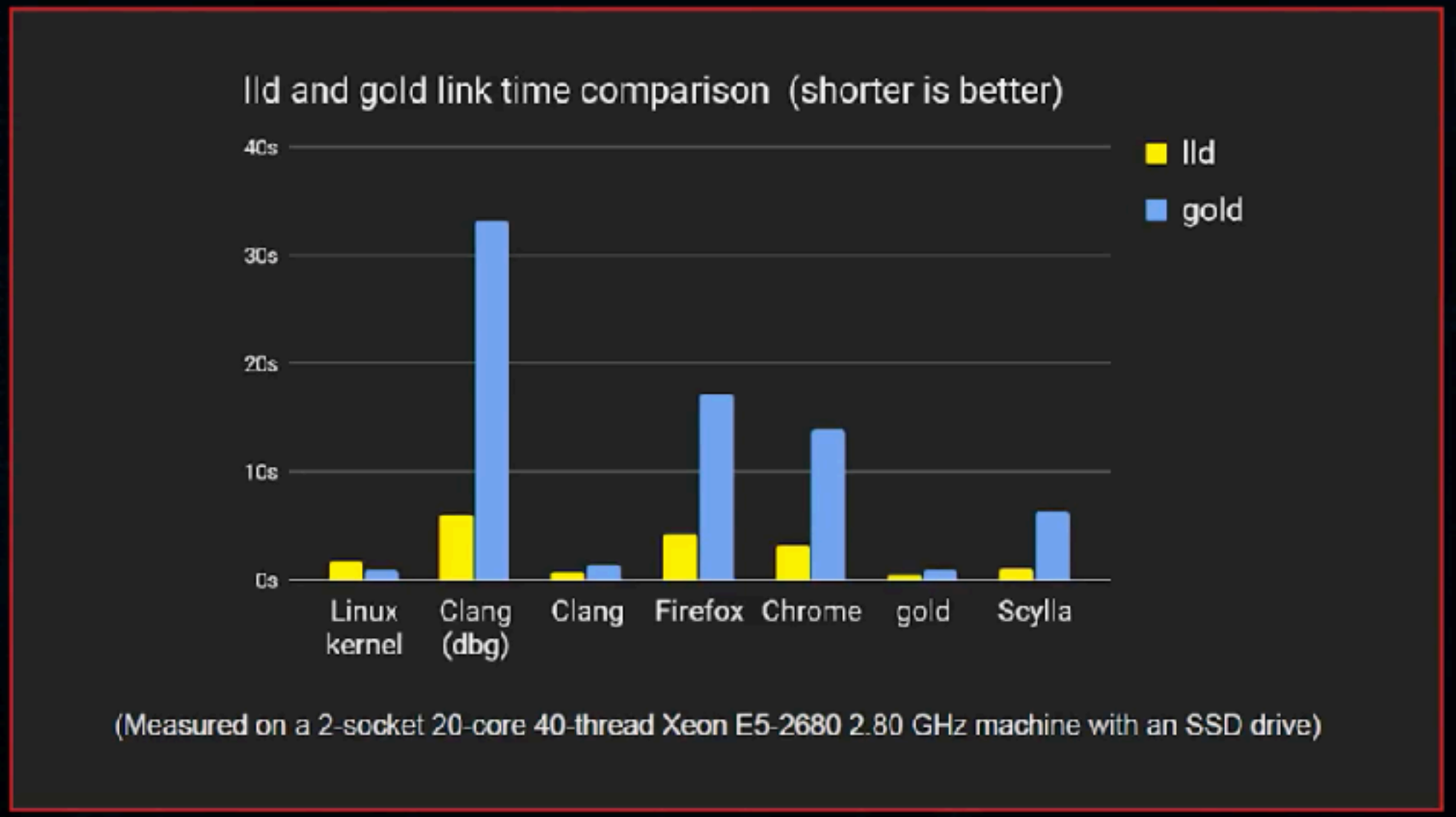
https://www.youtube.com/watch?v=Uqch1rjPls8

- Dużo przydatnej wiedzy o JITach

LLVM DEVELOPERS' MEETING

2017 · SAN JOSE, CA

lld: A Fast, Simple, and Portable Linker



https://www.youtube.com/watch?v=yTtWohFzS6s

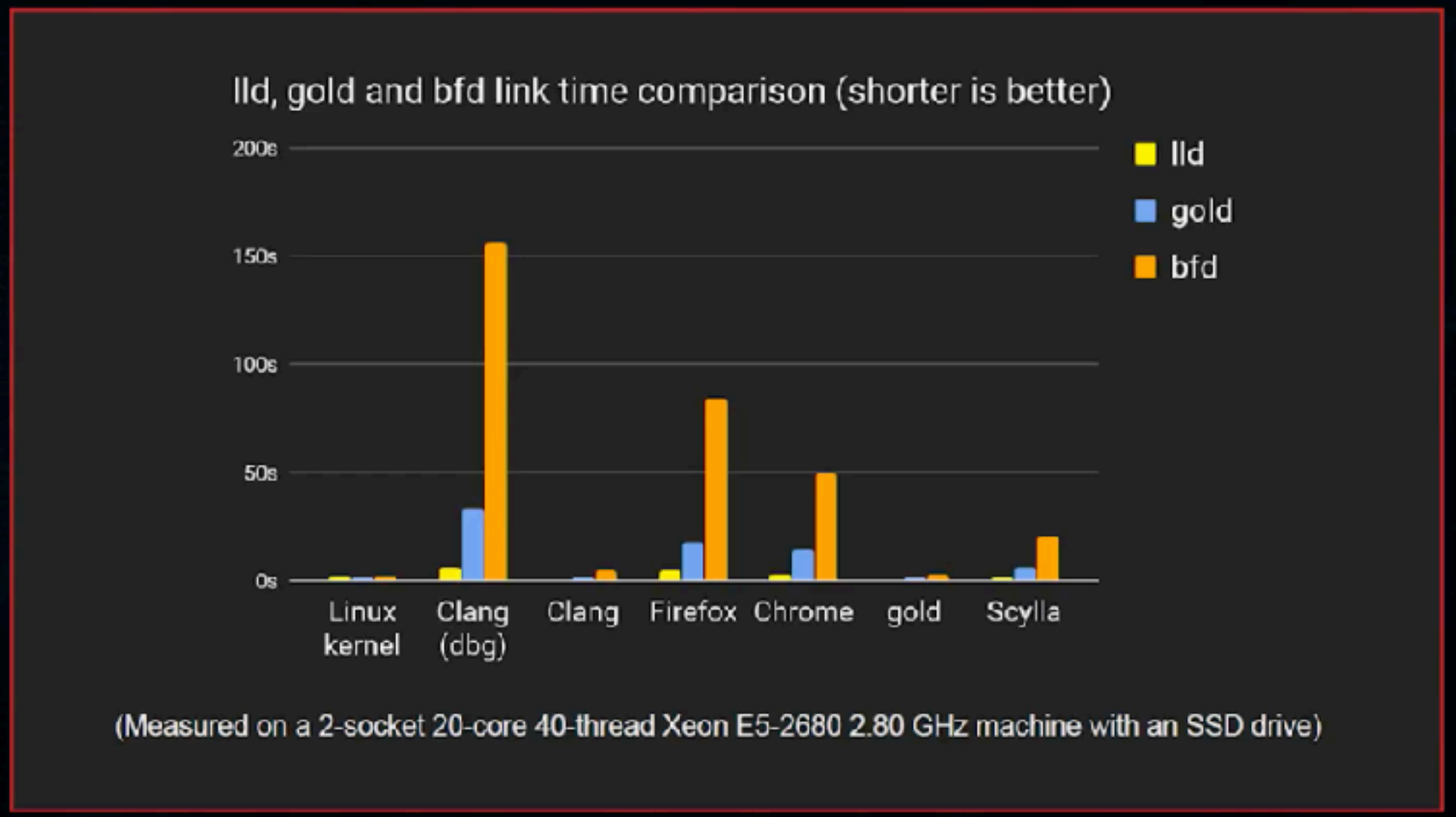
- LLD jest bardzo szybki i potrafi poprawnie zlinkować kernela freeBSD

LLVM DEVELOPERS' MEETING

2017 · SAN JOSE, CA

RUI UEYAMA

lld: A Fast, Simple, and Portable Linker



(Measured on a 2-socket 20-core 40-thread Xeon E5-2680 2.80 GHz machine with an SSD drive)


John St
San Pedro St
STOP
L L V M

Code::Dive

Piotr Kozłowski


Conclusion {
spatial mapping
spatial understanding
input gestures
voice commands
cursor and raycast
}
code::dive
NOKIA
fot. filipbasara.com

Ogłoszenia parafialne

- Środa zostaje
- Czy zaczynać o 18:00? - większością zostaje 18:35
- 6 Grudnia będziemy mieli gościa z Niemiec