

clang

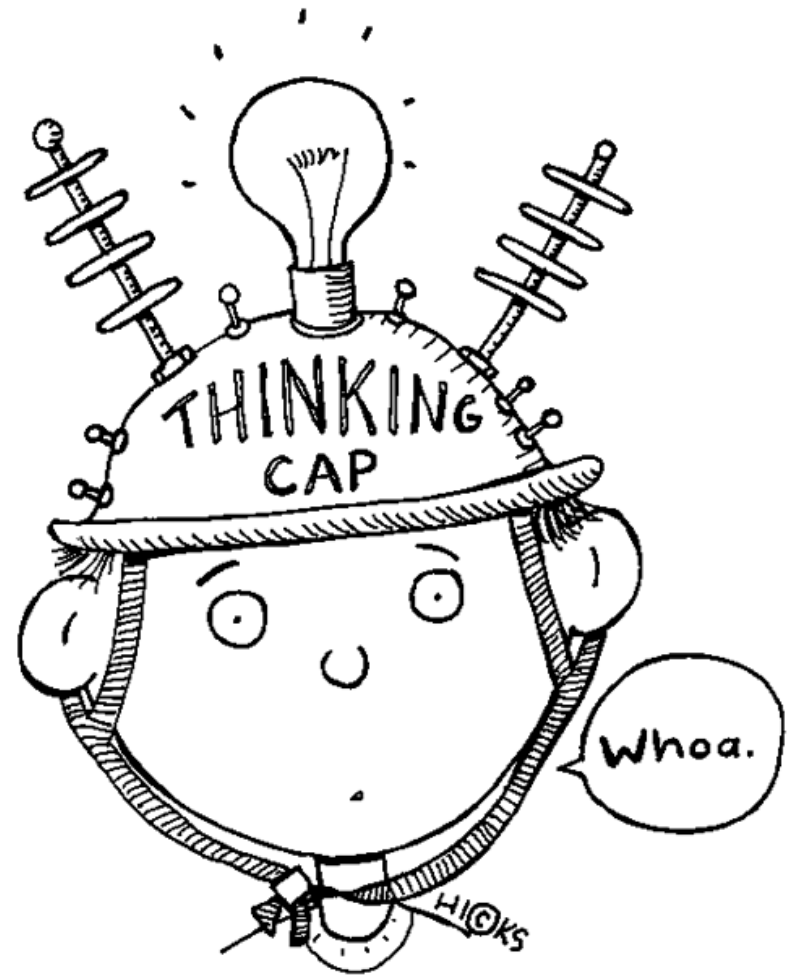
the short story about compiler...



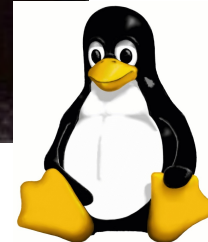
Mariusz Zaborski
<zaborskm@gmail.com>

Agenda

- Starter
- What is clang?
- Why clang?



Starter - imagine...



What is clang?



What is clang?

Compiler for:

C, C++ (C++11 feature complete),
Objective-C, Objective-C++

Project: LLVM

License: BSD-like

Developers: Apple, Google and ...



License fight

The GPLv3 forbids **tivoisation** loophole of code.

Tivoisation == forbidden editing code using hardware restrictions



License fight



Why clang?



Strong points of clang

- Fast compiles and low memory use
- Modular library based architecture
- GCC compatibility
- A real-world, production quality compiler
- A simple and hackable code base
- Expressive diagnostics



Strong points of clang

- **Fast compiles and low memory use**
- Modular library based architecture
- GCC compatibility
- A real-world, production quality compiler
- A simple and hackable code base
- **Expressive diagnostics**



Strong points of clang

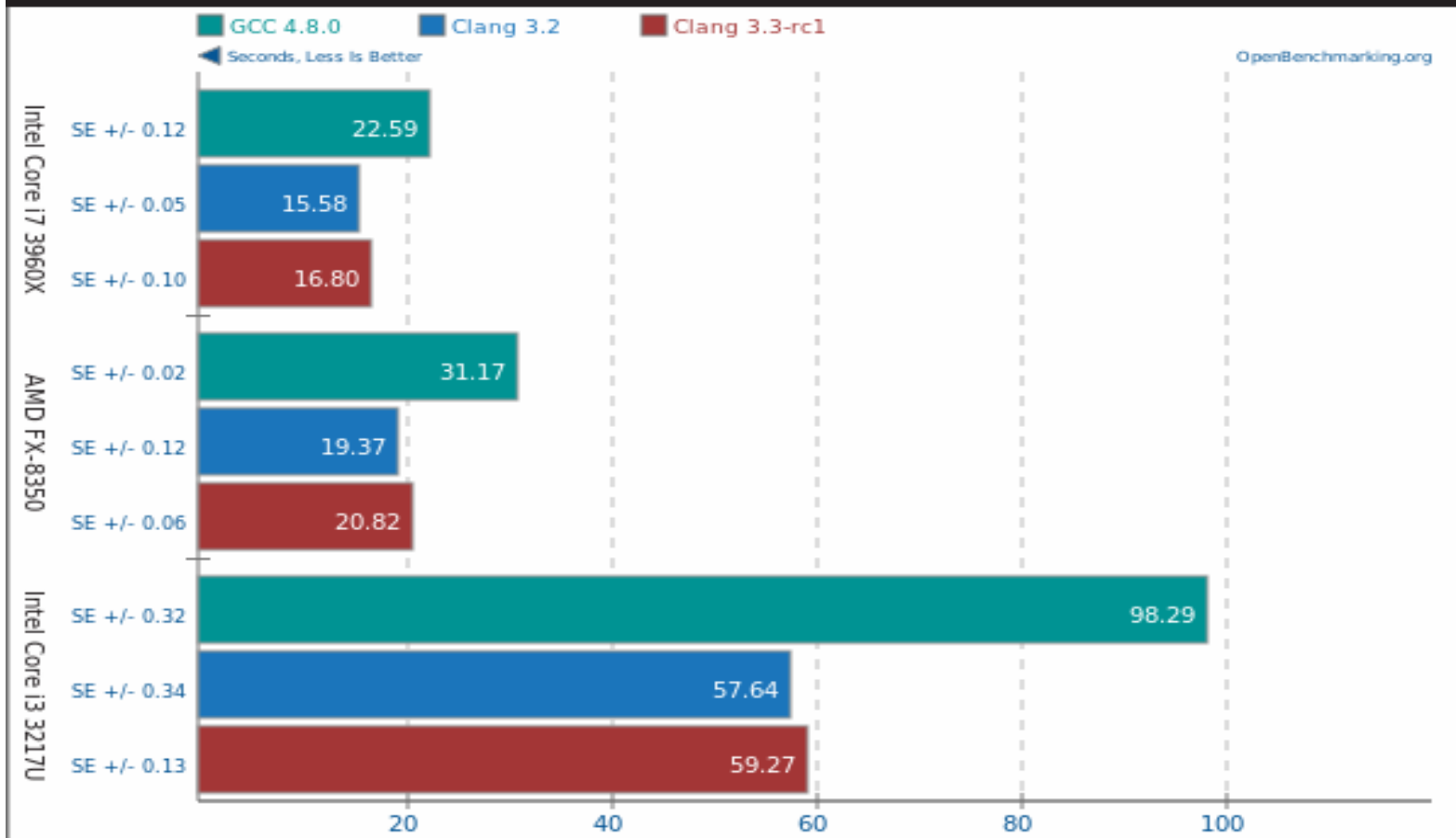
- **Fast compiles and low memory use**
- Modular library based architecture
- GCC compatibility
- A real-world, production quality compiler
- A simple and hackable code base
- Expressive diagnostics



Fast compiles

Timed PHP Compilation v5.2.9

Time To Compile



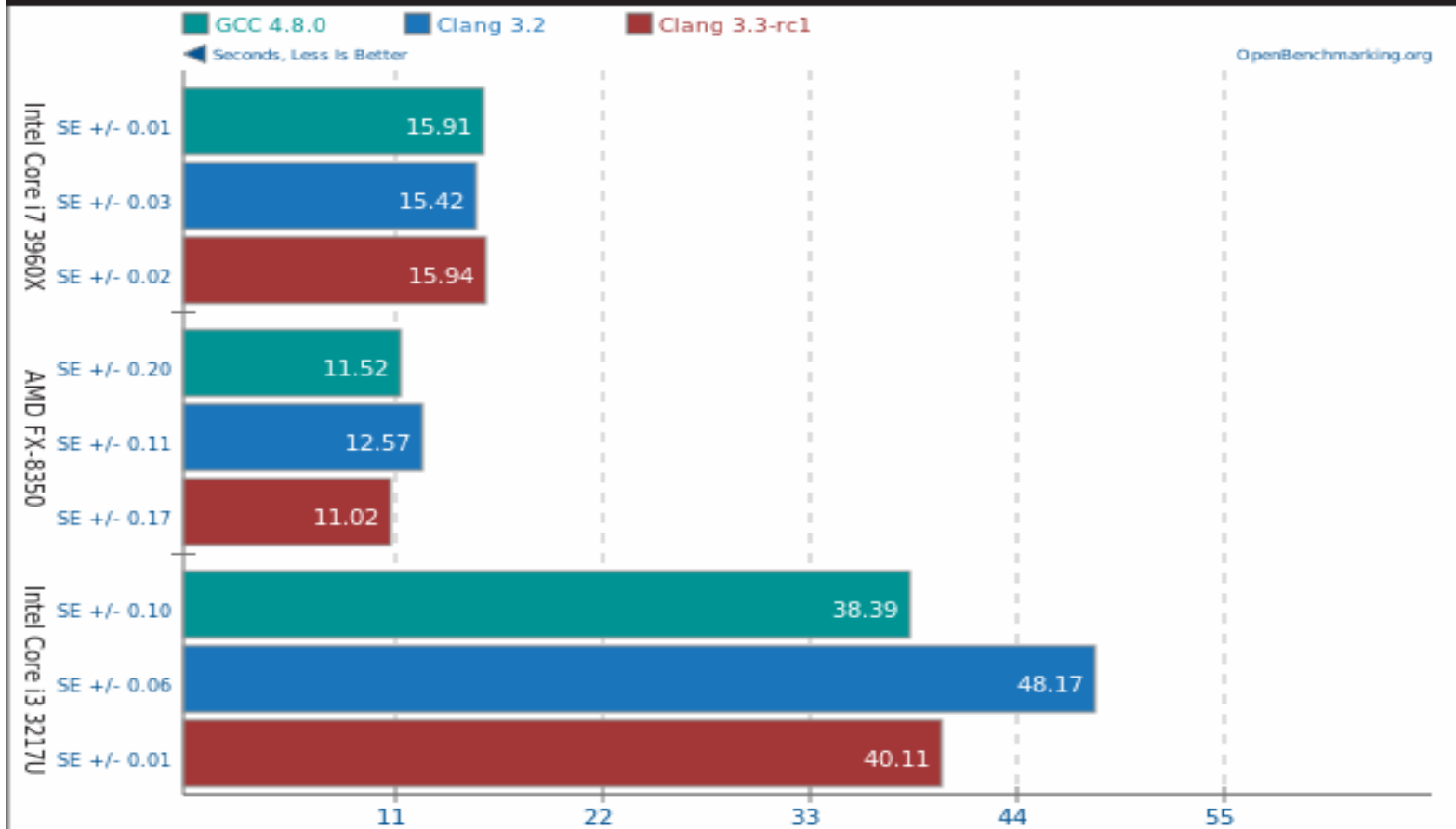
Powered By Phoronix Test Suite 4.6.0m1

1. (CC) gcc options: -O3 -march=native -pedantic -ldl -lz -lm

Article link http://www.phoronix.com/scan.php?page=article&item=llvm_clang33_3way&num=1

What about runtime?

Timed HMMer Search v2.3.2 Pfam Database Search



Powered By Phoronix Test Suite 4.6.0m1

1. (CC) gcc options: -O3 -march=native -pthread -lhmm -lsquid -lm

Article link http://www.phoronix.com/scan.php?page=article&item=llvm_clang33_3way&num=1

Strong points of clang

- Fast compiles and low memory use
- Modular library based architecture
- GCC compatibility
- A real-world, production quality compiler
- A simple and hackable code base
- **Expressive diagnostics**



Expressive diagnostics

```
float
min(float *x, int n)
{
    float *m;
    int i;

    for (m = x, i = 1; i < n; i++)
        if (x[i] < *m)
            m = *x;

    return m;
}
```



Error - gcc 4.7

```
mint.c: In function 'min':  
mint.c:9:6: error: incompatible types when assigning to type 'float *'  
from type 'float'  
mint.c:11:2: error: incompatible types when returning type 'float *' bu  
t 'float' was expected
```



Error - clang

```
mint.c:9:6: error: assigning to 'float *' from incompatible type  
      'float'; remove *  
      m = *x;
```

```
      ^  ~
```

```
mint.c:11:9: error: returning 'float *' from a function with  
      incompatible result type 'float'; dereference with *  
      return m;
```

```
      ^
```

```
      *
```

```
2 errors generated.
```



Error - gcc 4.8

```
main.c: In function 'min':
```

```
main.c:9:6: error: incompatible types when assigning to type 'float *'  
from type 'float'
```

```
    m = *x;  
      ^
```

```
main.c:11:2: error: incompatible types when returning type 'float *' but  
t 'float' was expected
```

```
    return m;  
    ^
```



Expressive diagnostics

```
#define abs(a) ((a) >= 0 : (a) ? -1.0*(a) )
```

```
double
```

```
normL1(double *x, int n)
```

```
{
```

```
    float s;
```

```
    int i;
```

```
    for (i = 0; i < n; i++)
```

```
        s += abs(x[i]);
```

```
    return s;
```



Error's

```
|l1.c: In function 'normL1':  
|l1.c:10:8: error: expected ')' before ':' token
```

```
|l1.c:10:8: error: expected ')'
           s += abs(x[i]);
```



```
|l1.c:1:26: note: expanded from macro 'abs'
#define abs(a) ((a) >= 0 : (a) ? -1.0*(a))
```



Error's

```
l1.c: In function 'normL1':  
l1.c:1:26: error: expected ')' before ':' token  
#define abs(a) ((a) >= 0 : (a) ? -1.0*(a))  
                                ^  
l1.c:10:8: note: in expansion of macro 'abs'  
    s += abs(x[i]);  
        ^
```

```
l1.c:10:8: error: expected ')'  
        s += abs(x[i]);  
            ^
```

```
l1.c:1:26: note: expanded from macro 'abs'  
#define abs(a) ((a) >= 0 : (a) ? -1.0*(a))  
                                ^
```

Story is end...

- Good alternative
- Compilers race
- Philosophy issue
- Just try it:

```
# alias cc='clang'
```



Thank you!



Mariusz Zaborski
<zaborskm@gmail.com>

Feel free to discuss...

