



Collabora Productivity



peralex

Brute force clang plugins

By Noel Grandin

Some kind of engineer at Collabora
Productivity and Peralex Electronics



Hardware for analysis

Intel Core i7 with 4 cores, running at 4GHz

16G ram

256G SSD

Nice CPU cooler and fan (I forgot which, but very important for long running builds)

Prior art

- Complex
- Many man years of work
- Assumptions no longer valid

Path to freedom

- Database?
- No.
- Just dump a log file.

First brute force plugin

commit b44cbb26efe1d0b0950b1e1613e131b506dc3876

Author: Noel Grandin <noel@peralex.com>

Date: Tue Jan 20 12:38:10 2015 +0200

new loplugin: change virtual methods to non-virtual

Where we can prove that the virtual method is never overridden.

First plugin: bash analysis

```
$ grep 'definition' log.txt | cut -f 2 | sort -u > definition.txt
```

```
$ grep 'overriding' log.txt | cut -f 2 | sort -u > overriding.txt
```

```
$ cat definition.txt overriding.txt | sort | uniq -u > result.txt
```

Problem 1: Mixed output

```
std::string output;
```

```
for (const MyFuncInfo & s : definitionSet)
```

```
    output += "definition:\t" + s.name + "\t" + s.sourceLocation + "\n";
```

```
std::ofstream myfile;
```

```
myfile.open( "unnecessaryvirtual.log", std::ios::app | std::ios::out);
```

```
myfile << output;
```

```
myfile.close();
```

Problem 2: Bash is horrible

(for anything maintainable)

==> python

Problem 3: Python veerrryyyy slooowww

```
with io.open("loplugin.unnecessaryvirtual.log", "rb",  
buffering=1024*1024) as txt:
```

```
    for line in txt:
```

```
        tokens = line.strip().split("\t")
```

```
        if tokens[0] == "definition:":
```

```
            fullMethodName = tokens[1]
```

```
            sourceLocation = tokens[2]
```

```
            definitionSet.add(fullMethodName)
```

Result happiness!

Current set of plugins (1)

unusedfields

unusedmethods

constantparam

Current set of plugins (2)

unnecessaryvirtual

unusedenumconstants

singlevalfields

Current set of plugins (3)

countusersofdefaultparams

expandablemethods

inlinefields

mergeclasses

Ideas wanted!