

Traçabilité du processus de compilation d'un compilateur llvm, dans le cadre d'une compilation obfusquante



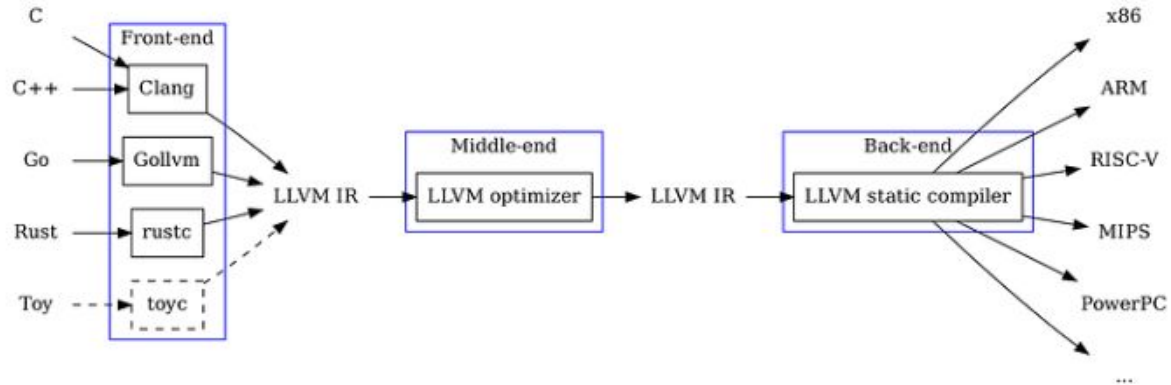
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Paul ENJALBERT

Tuteur:
Jean-Christophe BACH
Fabien DAGNAT



Contexte du stage



Chaîne de compilation d'un code compilé avec llvm

Travaux effectués

1. Reproductibilité

- Compiler le code llvm modifié (qomptrace) -> LLVM 14

- Compilation d'un code et l'obtention d'une trace

```
1 |[
2  {
3    "type": "event",
4    "event-type": "creation",
5    "name": "declare i32 @BlockSort(i32*, i8*, i32)\n",
6    "value-id": "4",
7    "operands": []
8  },
9  {
10   "type": "event",
11   "event-type": "creation",
12   "name": "\n",
13   "value-id": "8"
14  },
15  {
16   "type": "event",
17   "event-type": "creation",
18   "name": "i32 undef",
19   "value-id": "12",
20   "operands": []
21  },
22  {
23   "type": "event",
24   "event-type": "creation",
25   "name": "argument",
26   "value-id": "18"
27  },
28  {
```

- Utilisation d'un plugin gdb afin d'afficher les informations de debug

<pre> B+>0x401140 <fact> 855 push %rbp 0x401141 <fact+1> 855 mov %rsp,%rbp 0x401144 <fact+4> 855 sub \$0x98,%rsp 0x40114b <fact+11> 855 mov %rdi,-0x10(%rbp) 0x40114f <fact+15> 855 mov \$0x1,%eax 0x401154 <fact+20> 855 cmp \$0x1,%rdi 0x401158 <fact+24> 855 mov %rax,-0x8(%rbp) 0x40115c <fact+28> 859 jbe 0x401350 <fact+528> 0x401162 <fact+34> 859 mov -0x10(%rbp),%rax 0x401166 <fact+38> 1526 mov %rax,%rcx 0x401169 <fact+41> 1526 add \$0xfffffffffffffff,%rcx 0x40116d <fact+45> 1526 mov %rcx,-0x30(%rbp) 0x401171 <fact+49> 1536 mov %rax,%rdx 0x401174 <fact+52> 1536 add \$0xfffffffffffffff,%rdx 0x401178 <fact+56> 1550 and \$0x7,%rcx 0x40117c <fact+60> 1550 mov %rcx,-0x28(%rbp) 0x401180 <fact+64> 1550 mov \$0x1,%ecx 0x401185 <fact+69> 1554 cmp \$0x7,%rdx 0x401189 <fact+73> 1554 mov %rcx,-0x20(%rbp) 0x40118d <fact+77> 1554 mov %rax,-0x18(%rbp) 0x401191 <fact+81> 1558 jb 0x4012bc <fact+380> 0x401197 <fact+87> 1558 mov -0x10(%rbp),%rcx 0x40119b <fact+91> 1558 mov -0x30(%rbp),%rax 0x40119f <fact+95> 3477 and \$0xfffffffffffffff8,%rax 0x4011a3 <fact+99> 3477 mov %rax,-0x50(%rbp) 0x4011a7 <fact+103> 3477 mov \$0x1,%edx 0x4011ac <fact+108> 3477 xor %eax,%eax </pre>	<pre> -Trace- 855 appears in operation: creation of `855` <badref> = icmp ugt i64 %05, 1 icmp `495` `38` happening in Application of pass ModuleInlinerWrapperPass happening in Application of pass ModuleToPostOrderCGSCCPassAdaptor happening in Application of pass FunctionToLoopPassAdaptor on function `10` 855 appears in operation: Expire reference `855` </pre>
--	--

multi-thre Thread 0x7ffff7f8a7 (asm) In: fact

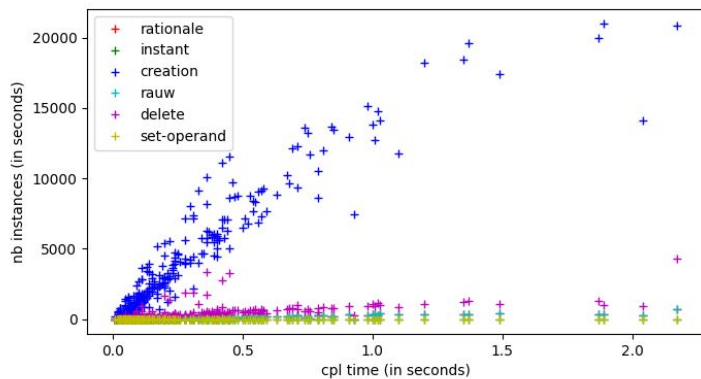
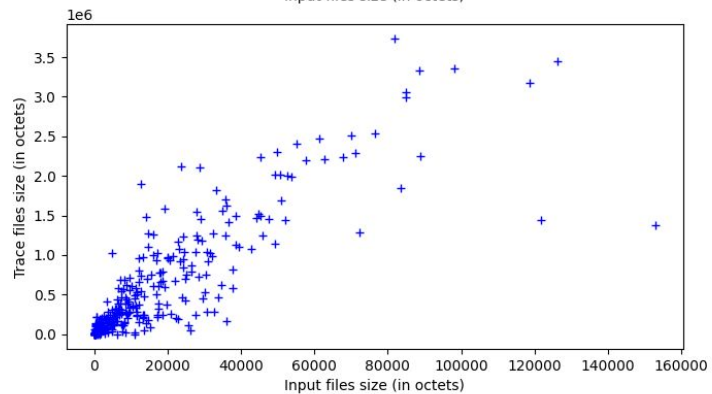
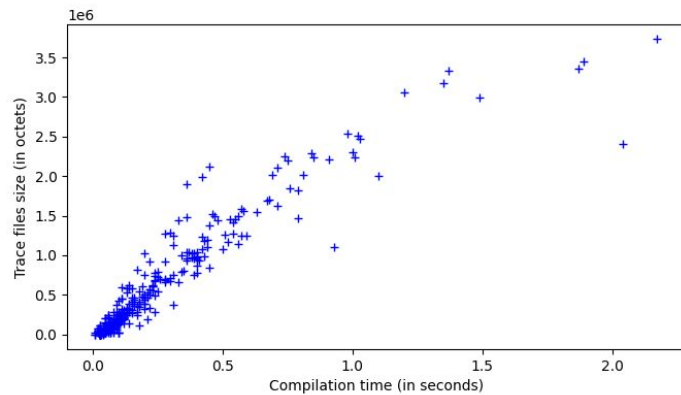
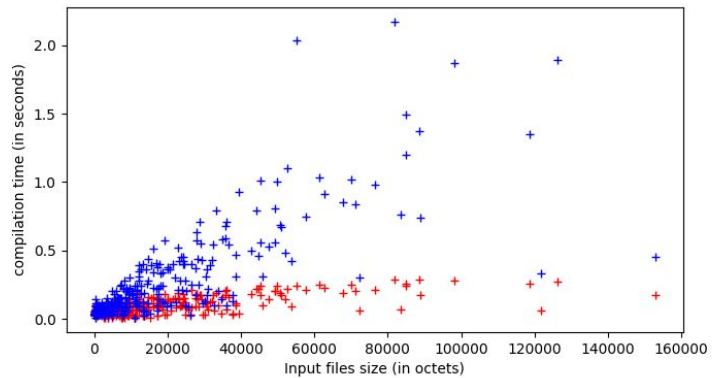
L10 PC: 0x401140

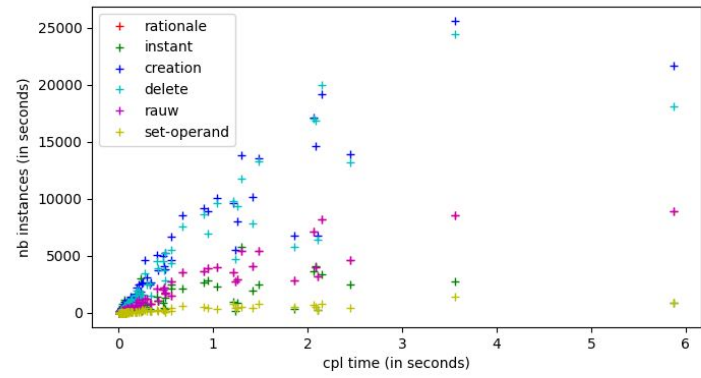
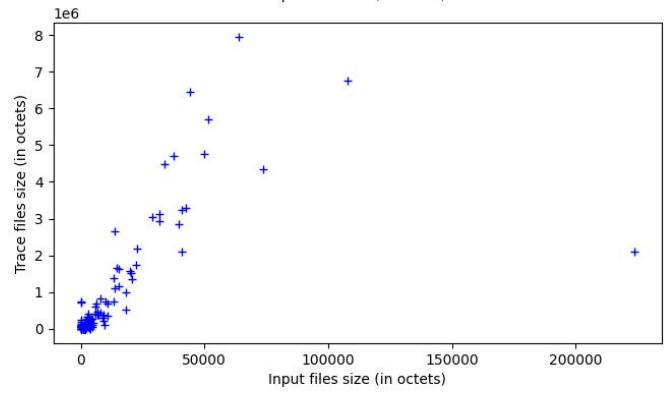
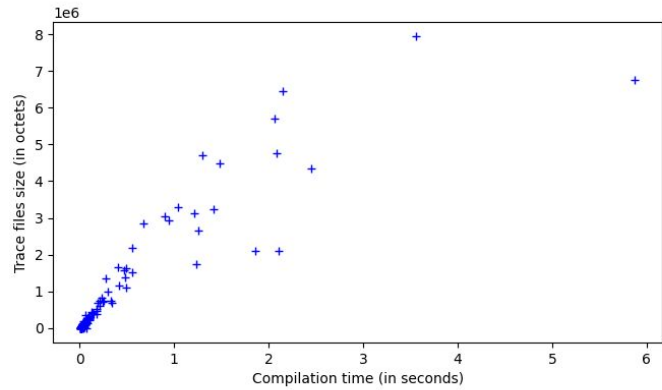
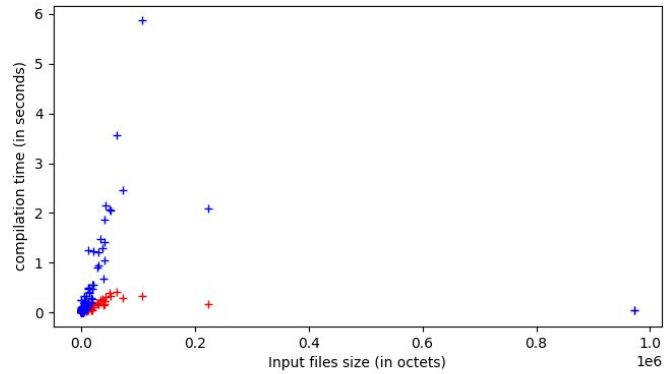
Travaux effectués

2. Evaluation

- Evaluation du temps de compilation avec la llvm test suite

- Mise en place d'une campagne de test qui compare:
 1. Temps de compilation
 2. Tailles des fichiers de trace
 3. Contenus des fichiers de trace





Travaux effectués

3. Portabilité

- Portabilité des modifications de qomptrace sur LLVM 20
- Changement important entre ces deux versions : opaque pointers

Travaux effectués

4. Autres

- Image Docker
- Correction d'un bug lors de l'utilisation de passes d'optimisations
- Nommage des fichiers de trace en fonction du fichier d'entrée + répertoire de stockage

Suite du stage

- Faire marcher la campagne d'évaluation sur la llvm test suite
- Portabilité des passes d'obfuscation de ollvm sur LLVM 20
- Ajouter de nouvelles informations de trace
- Intégrer la mesure de la mémoire dans la campagne d'évaluation