



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

ANALYSE STATIQUE POUR LA CLASSIFICATION DES PROCÉDURES CANDIDATE A LA TASKIFICATION

Encadré par: Jean-Baptiste BESNARD

Karim SMAIL Sofiane BOUZAHER Asma KREDDIA

Atef DORAI

Master CHPS

<https://github.com/Taskification/Taskification>

May 12, 2020



PLAN

INTRODUCTION

Compilateur?
Plugin

FLASH BACK

Objectif de cette Application
Fonction pure-Fonction non pure
Infrastructure Clang - LLVM

Clang PLUGIN

Description du code
Exécution du Plugin
Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call
Relation entre Plugin - Remote functions
Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?
Plugin

2 FLASH BACK

Objectif de cette Application
Fonction pure-Fonction non pure
Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code
Exécution du Plugin
Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call
Relation entre Plugin - Remote functions
Utilité de Plugin

5 CONCLUSION



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?

Plugin

2 FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

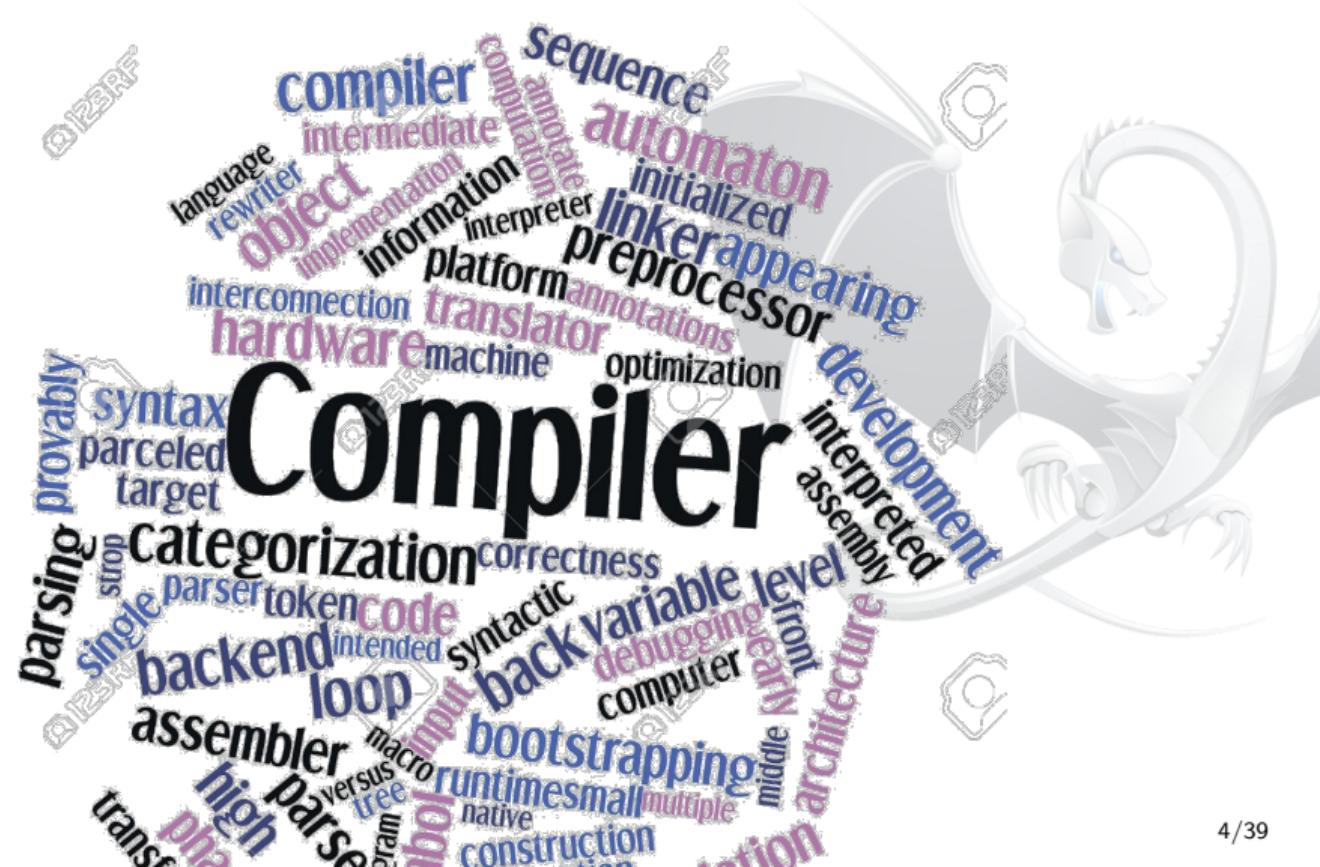
Utilité de Plugin

5 CONCLUSION



Compilateur?

- Outils d'optimisation et d'amélioration des performances



Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

1940-1944

Remote Procedure Call

Relation entre PI

CONCLUSION



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

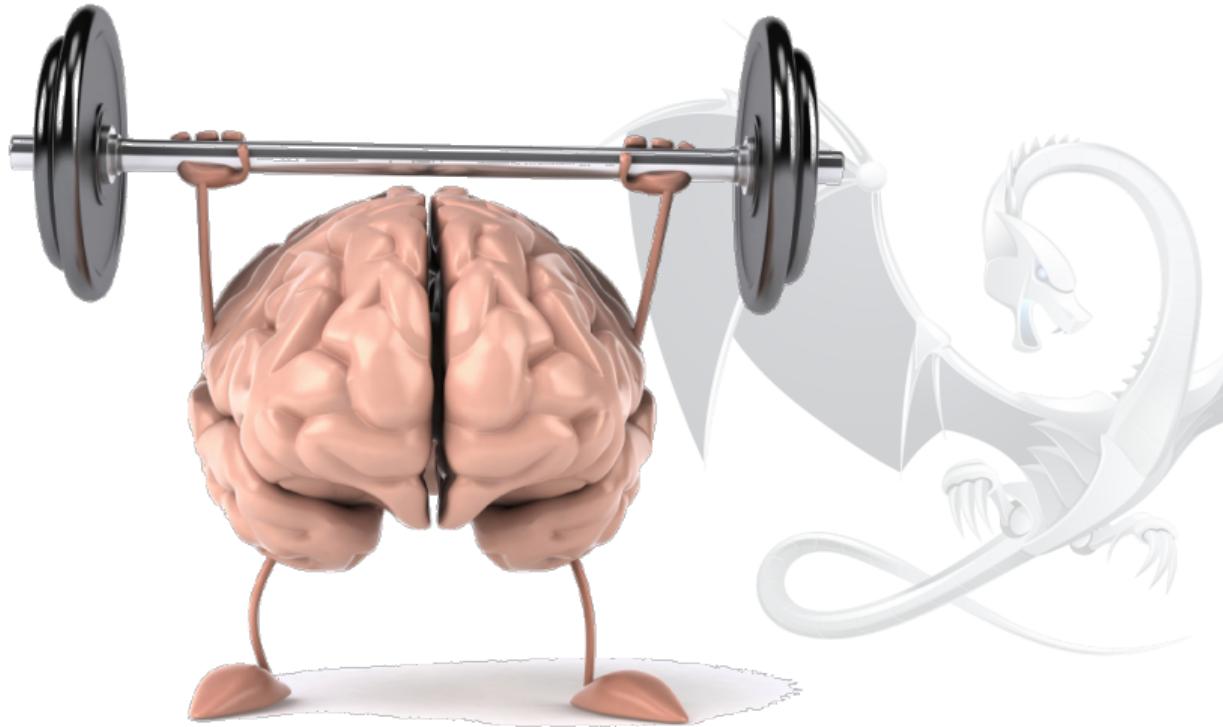
PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION





Comment?

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



Plugin

- Paquet qui Complète et apporte de nouvelles fonctionnalités pour le fameux "CLANG".



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?

Plugin

2 FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

5 CONCLUSION





Objectif

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

- **la mise en place d'un plugin dans le compilateur CLANG (LLVM) pour identifier les appels de fonctions candidats à la Taskification "fonction pures"**



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

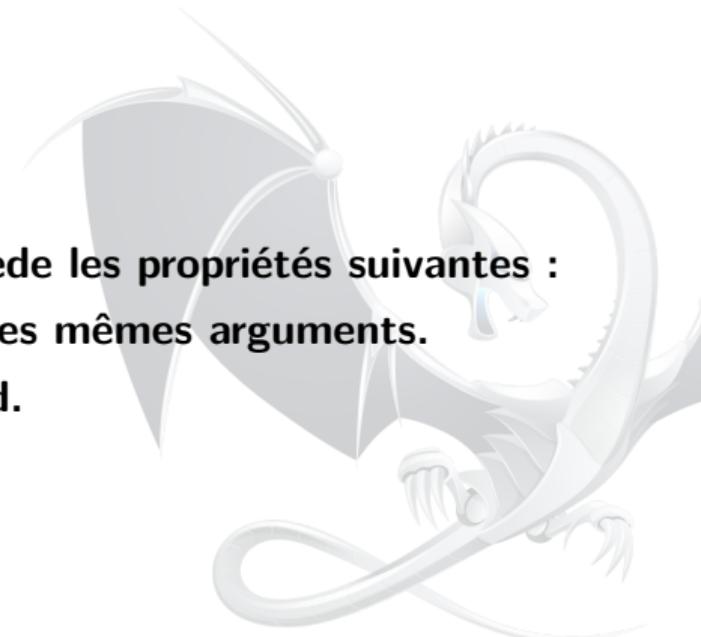
PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



Une fonction est dite pure lorsqu'elle possède les propriétés suivantes :

- **Sa valeur de retour ne varie pas avec les mêmes arguments.**
- **Son évaluation n'a pas d'effets de bord.**

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

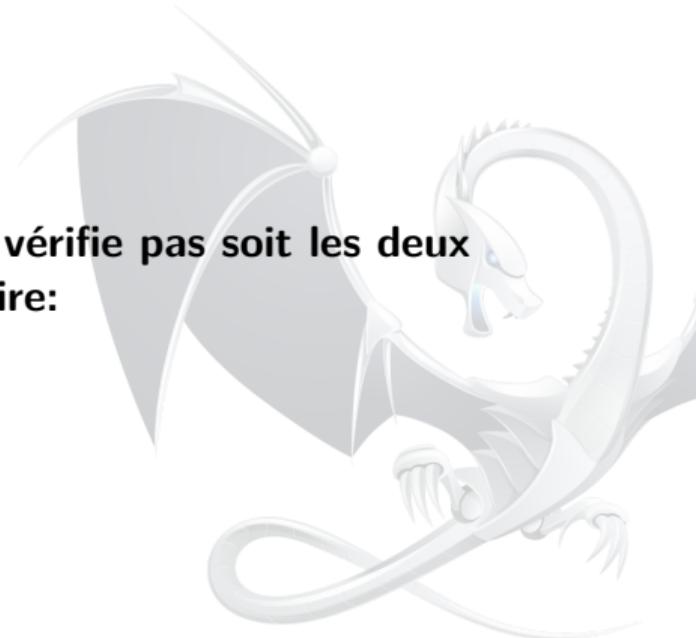
Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

Une fonction est dite impure lorsqu'elle ne vérifie pas soit les deux propriétés, ou bien l'une des deux c'est à dire:

- **Sa valeur de retour varie.**
- **Son évaluation a un effet de bord.**





INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

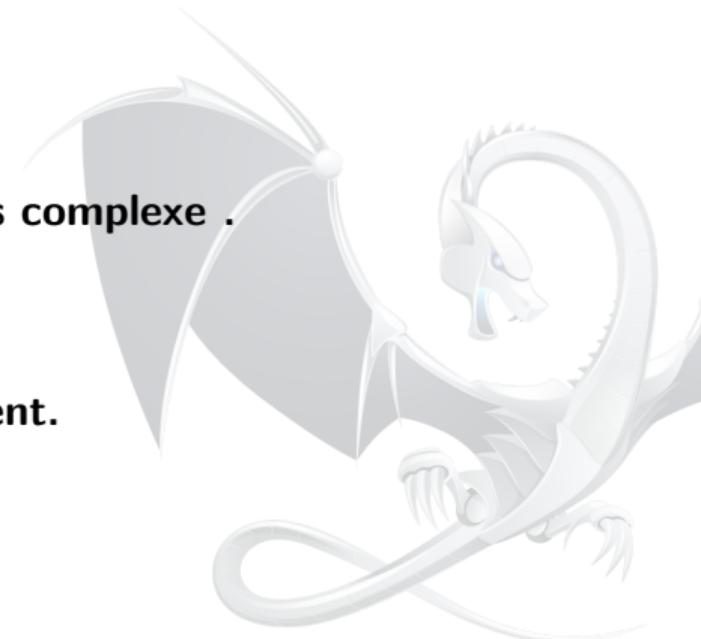
Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

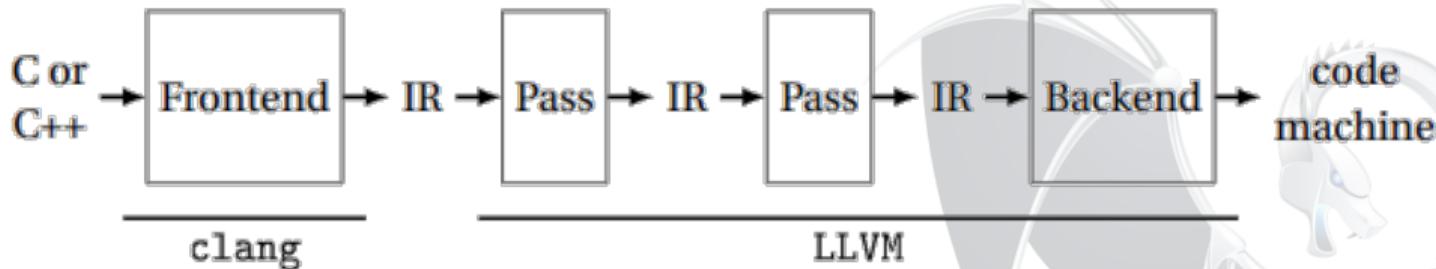
CONCLUSION

- **Fiable pour construire des programmes complexe .**
- **Prédictibles et donc facile a tester .**
- **Facile a lire et aussi a déboguer.**
- **Réutilisable dans d'autres environnement.**
- **Efficace pour le parallélisme .**



Infrastructure CLANG LLVM

- le Frontend, les Passes, le backend



Plusieurs outils vont venir intervenir tels que :

- CLANG
- OPT
- LLC
- LLVM-AS et LLVM-DIS
- LLI



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION





INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?

Plugin

2 FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

5 CONCLUSION





Description du code

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

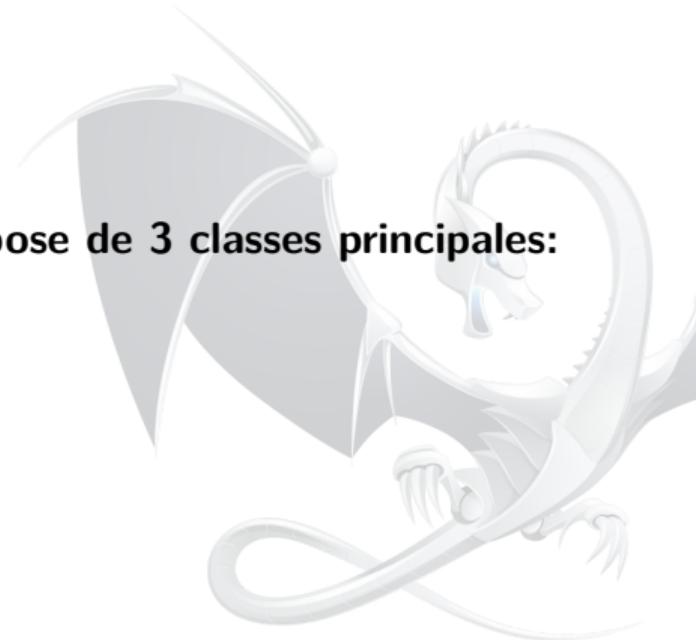
Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

Notre Plugin implémenté en C++ se compose de 3 classes principales:

- **Class PluginASTAction**
- **Class ASTConsumer**
- **Class RecursiveASTVisitor**





Description du code

- En-têtes

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

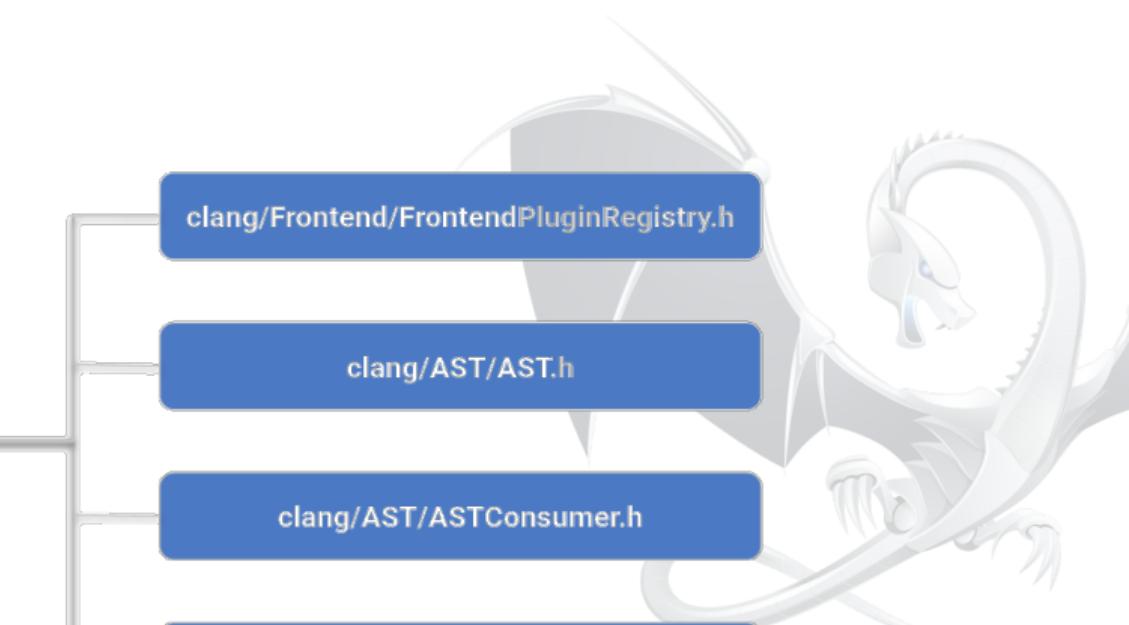
Documentation de clang

clang/Frontend/FrontendPluginRegistry.h

clang/AST/AST.h

clang/AST/ASTConsumer.h

clang/AST/RecursiveASTVisitor.h





Description du code

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

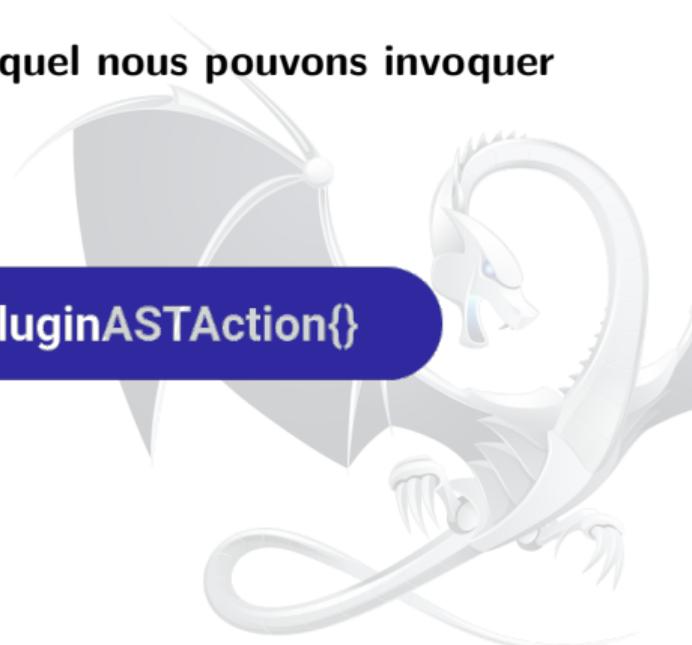
Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- **ASTAction** : Point d'entrée à partir duquel nous pouvons invoquer notre **ASTConsumer**

```
class ImpureAction : public PluginASTAction{}
```





Description du code

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

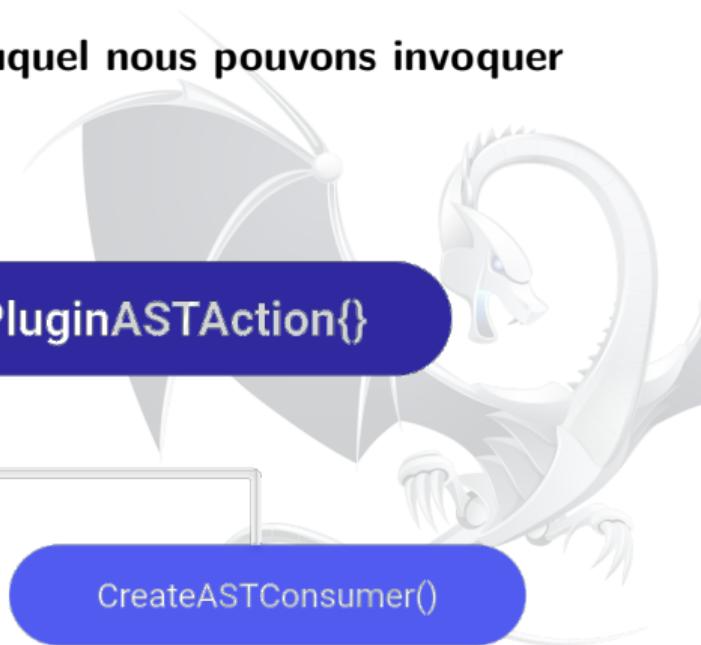
CONCLUSION

- **ASTAction** : Point d'entrée à partir duquel nous pouvons invoquer notre ASTConsumer

```
class ImpureAction : public PluginASTAction{}
```

```
ParseArgs()
```

```
CreateASTConsumer()
```



Description du code

- **ASTConsumer : Parcourir l'AST généré par l'analyseur du Clang**

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



```
class ImpureExprConsumer : public clang::ASTConsumer
```

Description du code

- **ASTConsumer : Parcourir l'AST généré par l'analyseur du Clang**

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

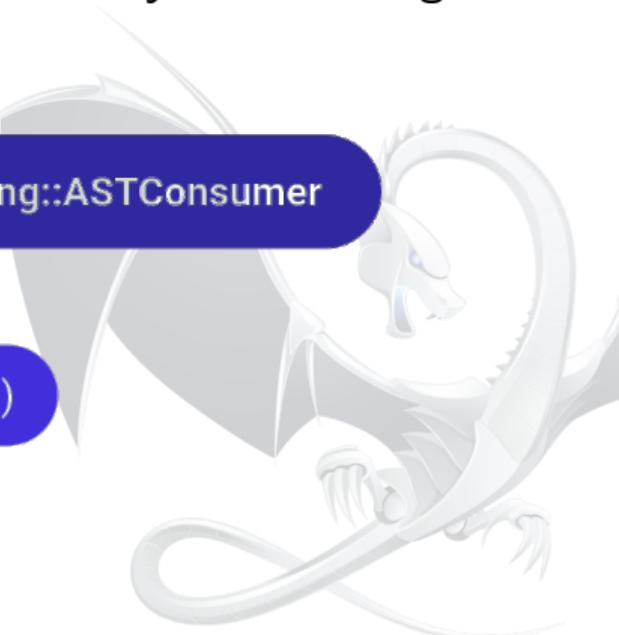
Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

```
class ImpureExprConsumer : public clang::ASTConsumer
```

```
HandleTranslationUnit()
```



Description du code

- **ASTConsumer** : Parcourir l'AST généré par l'analyseur du Clang

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

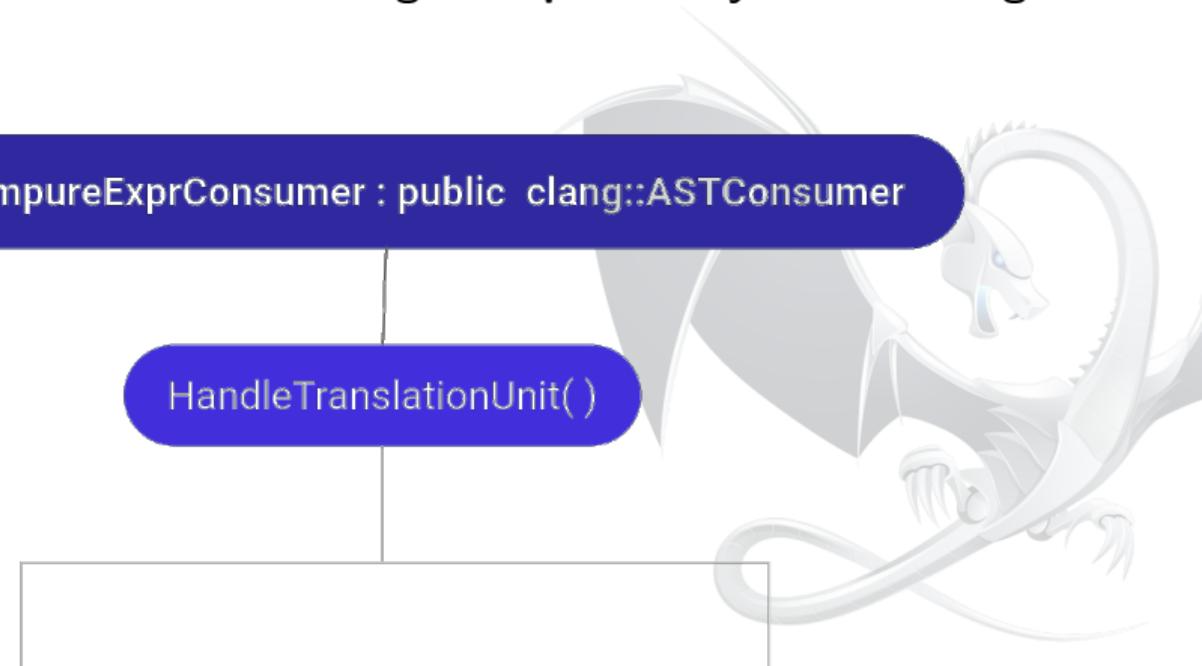
CONCLUSION

```
class ImpureExprConsumer : public clang::ASTConsumer
```

```
HandleTranslationUnit()
```

```
check_ impure_ function()
```

```
check_ ref_ expr()
```



Description du code

- **RecursiveASTVisitor** : Visiter tout les types de nœud dans l' AST

```
class ImpureExprVisitor  
: public RecursiveASTVisitor<ImpureExprVisitor> {...}
```

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



Description du code

- **RecursiveASTVisitor** : Visiter tout les types de nœud dans l' AST

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

```
class ImpureExprVisitor  
: public RecursiveASTVisitor<ImpureExprVisitor> {...}
```

```
bool VisitDecl()
```

```
bool VisitVarDecl()
```

```
bool VisitDeclRefExpr()
```



Exécution du Plugin

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- **load**
- **Xclang**



```
CC=clang
CFLAGS=`llvm-config --libs --cxxflags` -O3 -Wall -std=c++17

all: plugin.so

plugin.so: plugin.cpp
    $(CC) -fPIC -shared $(CFLAGS) -o $@ $^ -fPIC

run: t.c plugin.so
    $(CC) -Xclang -load ./plugin.so -Xclang -add-plugin -Xclang impurewarnings t.c

clean:
    rm -rf plugin.so ./a.out
```

Fiabilité du Plugin

- On prend le code source suivant comme exemple d'application de notre plugin :

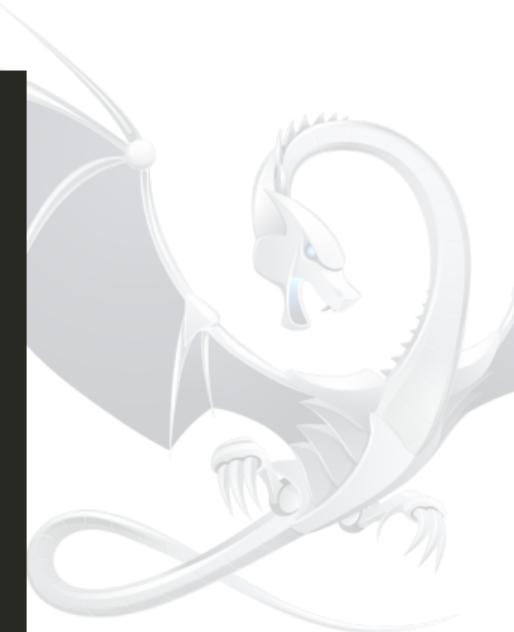
```
#include <stdio.h>

int global_variable = 99;

int pure(void)
{
    return 5;
}

int impure()
{
    global_variable = 9;
    printf("LL\n");
    return 0;
}

int main(int argc, char ** argv)
{
    pure();
    impure();
    return 0;
}
```



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



Résultat

CALCUL
HAUTE
PERFORMANCE
SIMULATION

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette
Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

```

Functions
=====
==== Exploring pure
ReturnStmt 0x558fca2b5490
|-IntegerLiteral 0x558fca2b5470 'int' 5
==== Exploring impure
BinaryOperator 0x558fca2b55f0 'int' '='
|-DeclRefExpr 0x558fca2b55b0 'int' lvalue Var 0x558fca2b2ba8 'global_variable' 'int'
|-IntegerLiteral 0x558fca2b55d0 'int' 9
CallExpr 0x558fca2b56b0 'int'
|-ImplicitCastExpr 0x558fca2b5698 'int (*)((const char *, ...))' <FunctionToPointerDecay>
|-DeclRefExpr 0x558fca2b5610 'int ((const char *, ...))' Function 0x558fca2a2680 'printf' 'int ((const char *, ...))'
|-ImplicitCastExpr 0x558fca2b56f0 'const char *' <NoOp>
|-ImplicitCastExpr 0x558fca2b5dd0 'char *' <ArrayToPointerDecay>
|`StringLiteral 0x558fca2b5630 'char [4]' lvalue "LL\n"
t.c10:1: warning: impure is an impure function
int impure()
^
ReturnStmt 0x558fca2b5728
|-IntegerLiteral 0x558fca2b5708 'int' 0
==== Exploring main
CallExpr 0x558fca2b59e0 'int'
|-ImplicitCastExpr 0x558fca2b59c8 'int (*)()<void>' <FunctionToPointerDecay>
|-DeclRefExpr 0x558fca2b5980 'Int (void)' Function 0x558fca2b53d0 'pure' 'int (void)'
t.c17:1: warning: main is an impure function
int main(int argc, char ** argv)
^
CallExpr 0x558fca2b5a60 'int'
|-ImplicitCastExpr 0x558fca2b5a48 'int (*)()' <FunctionToPointerDecay>
|-DeclRefExpr 0x558fca2b5a00 'Int ()' Function 0x558fca2b5510 'impure' 'int ()'
t.c17:1: warning: main is an impure function
ReturnStmt 0x558fca2b5aa0
|-IntegerLiteral 0x558fca2b5a80 'int' 0
=====
Global Variables
=====
VarDecl 0x558fca28dc20 </usr/include/x86_64-linux-gnu/bits/libio.h:319:1, col:29> _IO_2_1_stdin_ 'struct _IO_FILE_plus'; 'struct _IO_FILE_plus' extern
VarDecl 0x558fca28dcf0 </usr/include/x86_64-linux-gnu/bits/libio.h:320:1, col:29> _IO_2_1_stdout_ 'struct _IO_FILE_plus'; 'struct _IO_FILE_plus' extern
VarDecl 0x558fca28dd78 </usr/include/x86_64-linux-gnu/bits/libio.h:321:1, col:29> _IO_2_1_stderr_ 'struct _IO_FILE_plus'; 'struct _IO_FILE_plus' extern
VarDecl 0x558fca29d760 </usr/include/stdio.h:135:1, col:25> stdin_ 'struct _IO_FILE *' extern
VarDecl 0x558fca29d7f0 </usr/include/stdio.h:136:1, col:25> stdout_ 'struct _IO_FILE *' extern
VarDecl 0x558fca29d880 </usr/include/stdio.h:137:1, col:25> stderr_ 'struct _IO_FILE *' extern
VarDecl 0x558fca2b178 #</usr/include/x86_64-linux-gnu/bits/sys_errlist.h:261:1, col:12> sys_errlist_ 'int' extern
VarDecl 0x558fca2b1e0 </usr/include/x86_64-linux-gnu/bits/sys_errlist.h:271:1, col:38> col:26 sys_errlist_ 'const char *const []' extern
VarDecl 0x558fca2b2b8 <.c1:1, col:23> col:5 used global_variable_ 'int' cinit
|`IntegerLiteral 0x558fca2b2c10 <col:23> 'int' 99
^
DeclRefExpr
=====
DeclRefExpr 0x558fca2b5b0 'int' lvalue Var 0x558fca2b2ba8 'global_variable' 'int'
=====
t.c12:2: warning: global_variable is a global variable
global_variable = 9;
^
4 warnings generated.

```





INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction

non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?

Plugin

2 FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

5 CONCLUSION





Remote Procedure Call

- Le principe est de lancer n processus qui peuvent s'exécuter en n cœur différents



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

• MPIX_Offload :

```
int MPIX_Offload(void ** buffers,  
                  MPI_Datatype * datatypes,  
                  int sendcount,  
                  void * recvbuf,  
                  MPI_Datatype recvtype,  
                  char * fname,  
                  int dest)  
{  
    MPI_Request req;
```



Relation entre Plugin - Remote functions

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

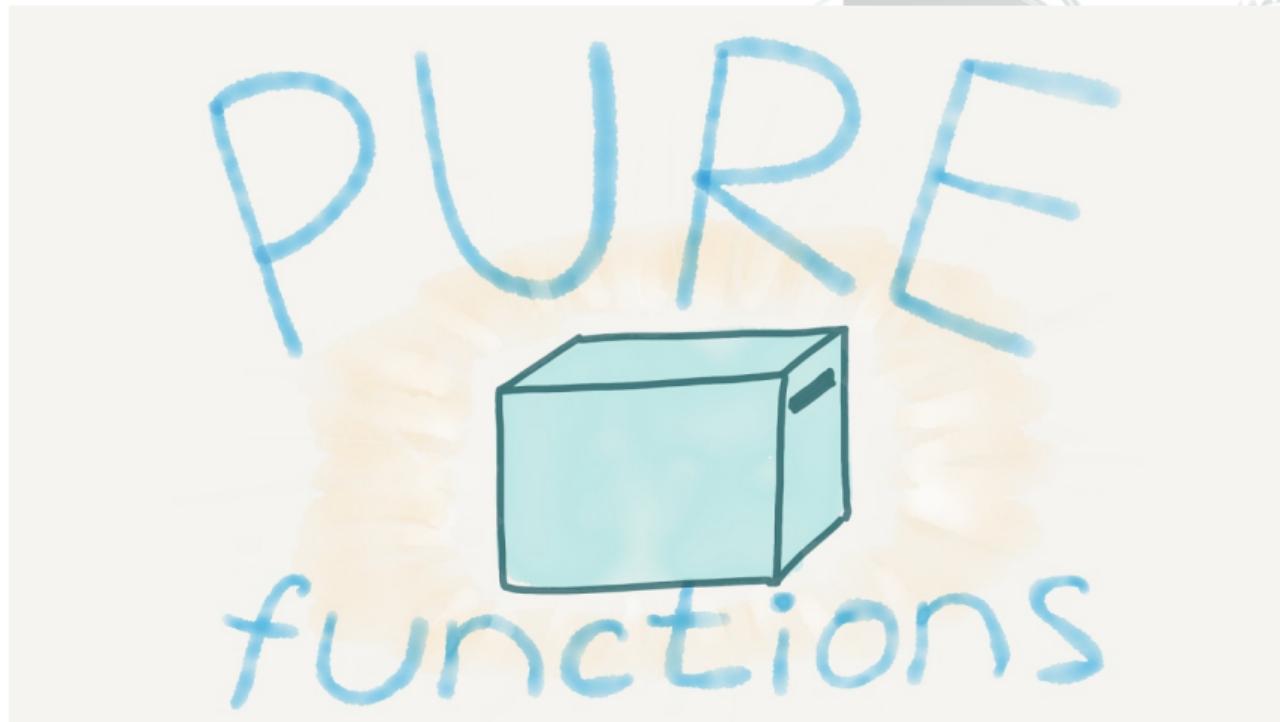
Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- **Plugin est l'outil d'identification des fonctions candidates à être appelé à distance.**



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- La fonction MPIX_Offload est le moyen d'appeler ces fonctions candidates à distance.





INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

• Tâches successives



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

- Viser la parallélisation automatique :



```
1 int pure_func(int a, int b)
2 {
3     return a*b;
4 }
5
6 main :
7 //traitement
8
9 int tab[128];
10 int i;
11 for(i=0; i<128; i++)
12 {
13     tab[i] = pure_func(i, i);
14 }
```

CONCLUSION

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- Possibilité de parallélisation en utilisant openMP :

```
#pragma omp parallel
for (i=0; i<128; i++)
{
    tab[i] = pure_func(i, i);
}
```

1
2
3
4
5



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

• En utilisant MPIX_Offload :

```
for (i=0; i<128; i++)
{
    void* buffers[2] = {&i, &i};
    MPI_Datatype type[2] = {MPI_INT, MPI_INT};

    MPIX_Offload(buffers,
                  type,
                  2,
                  &tab[i],
                  MPI_INT,
                  "pure_func",
                  i);
}
```

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

• RPC vs OpenMP :

```

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Request req[128];
    if (rank == 0)
    {
        for (i=0; i<128; i++)
        {
            void* buffers[2] = {&i, &i};
            MPI_Datatype type[2] = {MPI_INT, MPI_INT};

            MPIX_Ioffload(buffers,
                           type,
                           2,
                           &tab[i],
                           MPI_INT,
                           "pure_func",
                           (i)%size, /*pour faire appeler la function dans des processus deffirent ←
                                       autant qu'on a dans notre machine*/
                           &req[i]);
        }
        MPI_Waitall(req, 128, MPI_STATUSES_IGNORE);
    }

```



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION





INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

CONCLUSION

1 INTRODUCTION

Compilateur?

Plugin

2 FLASH BACK

Objectif de cette Application

Fonction pure-Fonction non pure

Infrastructure Clang - LLVM

3 Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

4 PARALLÉLISATION VIA RPC

Remote Procedure Call

Relation entre Plugin - Remote functions

Utilité de Plugin

5 CONCLUSION





Conclusion

INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

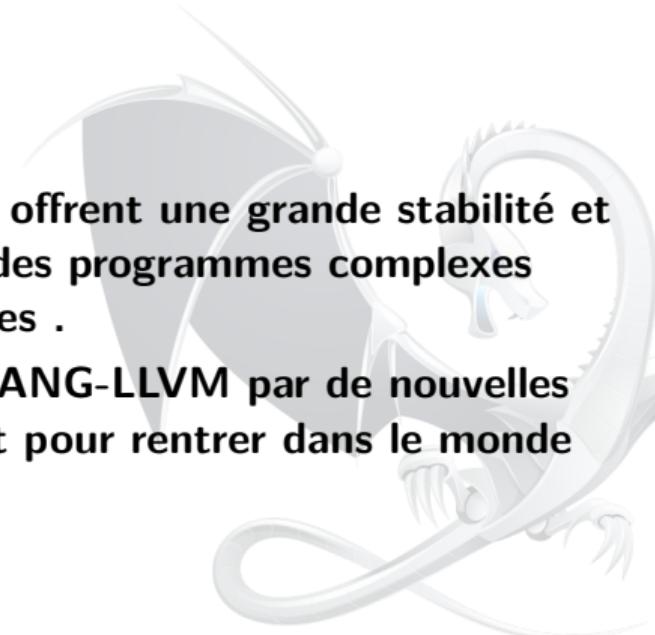
Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- **Fonctions pures sont des fonctions qui offrent une grande stabilité et performance lors de l'implementation des programmes complexes comparément avec les fonctions impures .**
- **Alimenter la chaîne de compilation CLANG-LLVM par de nouvelles fonctionnalités trace le point de départ pour rentrer dans le monde vaste des compilateurs.**



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang -
LLVM

Clang PLUGIN

Description du code

Exécution du Plugin

Fiabilité du Plugin

PARALLÉLISATION

VIA RPC

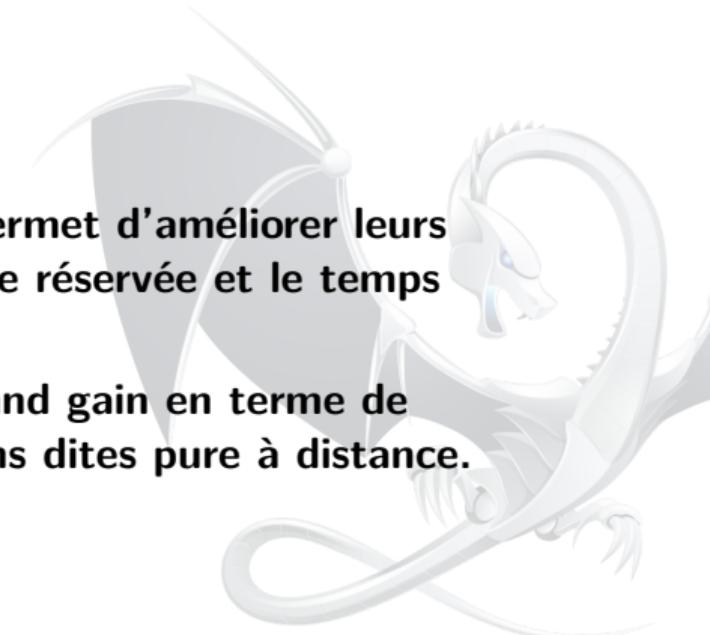
Remote Procedure Call

Relation entre Plugin -
Remote functions

Utilité de Plugin

CONCLUSION

- **Intégrer des plugins au compilateur permet d'améliorer leurs performances en jouant sur la mémoire réservée et le temps d'exécution.**
- **La parallélisation via RPC offre un grand gain en terme de performances, en appelant des fonctions dites pure à distance.**



INTRODUCTION

Compilateur?

Plugin

FLASH BACK

Objectif de cette

Application

Fonction pure-Fonction
non pure

Infrastructure Clang - LLVM

Chang F. LOOM

Description du code

Execution du Plugin

PARALLÉLISATION

VIA BDC

Remote Procedure

CONCLUSION

