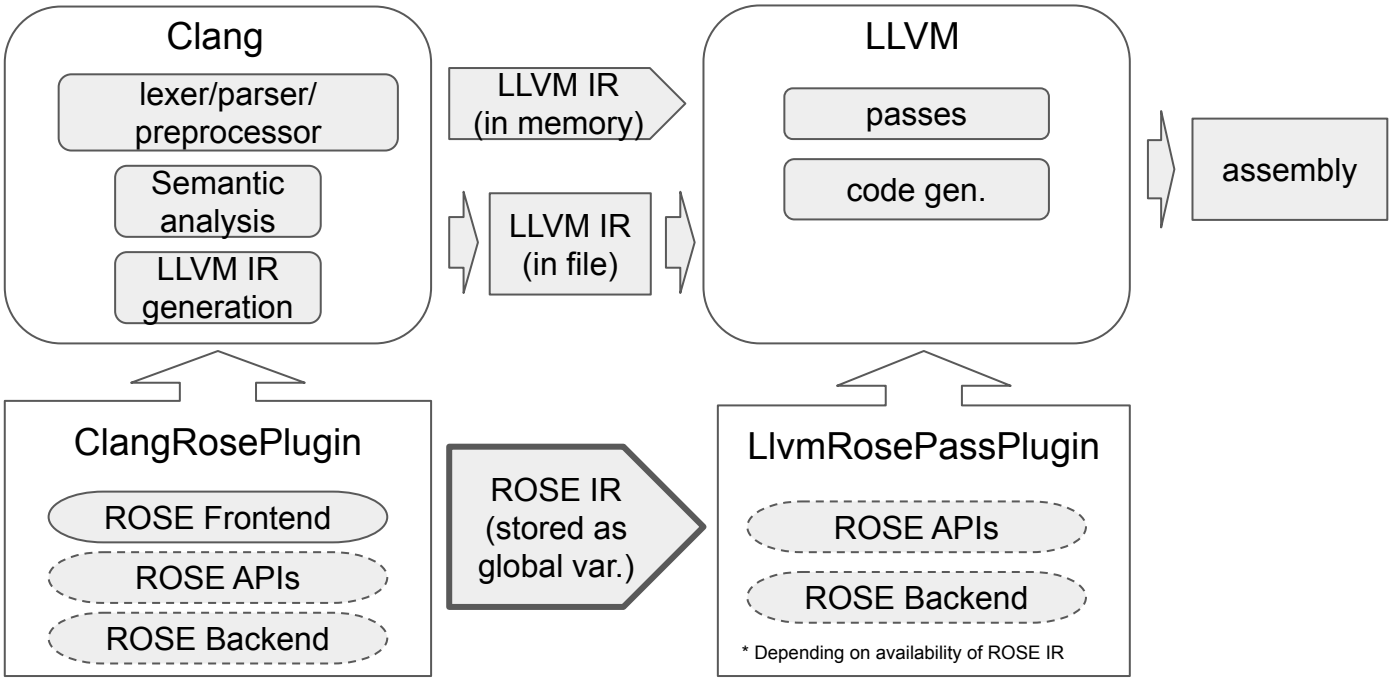
Clang
lexer/parser/
preprocessor
Semantic
analysis
LLVM IR
generation
LLVM IR
(in memory)
LLVM IR
(in file)
LLVM
passes
code gen.
assembly
ClangRosePlugin
ROSE Frontend
ROSE APIs
ROSE Backend
ROSE IR
(stored as
global var.)
LlvmRosePassPlugin
ROSE APIs
ROSE Backend
* Depending on availability of ROSE IR

Software requirement

Software:

- GCC 11.2. (to build LLVM)
- Clang/LLVM 14.0
- Boost 1.73 (built with LLVM)
- CMake 3.17.4 (to build LLVM-ROSE-Plugin)

Installation

1. Setup environment:
   a. source /nfs/casc/overture/ROSE/opt/rhel8/x86_64/boost/1_73_0/llvm/14.0.0/gcc/11.2.0/setup.sh
      - This will setup GCC11.2, Clang/LLVM 14.0 and Boost 1.73 altogether.
   b. source /nfs/casc/overture/ROSE/opt/rhel8/x86_64/cmake/3.17.4/setup.sh
2. Build ROSE
   a. Config with Clang and Clang++: /home/lin32/Development/projects/ROSE/rose/configure
   --prefix=/home/lin32/Development/projects/ROSE/build/autotool-default-EDG5-withClang/install_tree --enable-languages=c,c++
   --with-boost=/nfs/casc/overture/ROSE/opt/rhel8/x86_64/boost/1_73_0/llvm/14.0.0/gcc/11.2.0 --enable-boost-version-check=false
   --with-CXX_DEBUG=-g --with-C_OPTIMIZE=-O0 --with-CXX_OPTIMIZE=-O0
   CC=/nfs/casc/overture/ROSE/opt/rhel8/x86_64/llvm/14.0.0/gcc/11.2.0/bin/clang
   CXX=/nfs/casc/overture/ROSE/opt/rhel8/x86_64/llvm/14.0.0/gcc/11.2.0/bin/clang++
3. Build LLVM-ROSE-Plugin

```
1. git clone git@github.com:peihunglin/LLVM-ROSE-Plugin.git
2. cd LLVM-ROSE-Plugin
3. mkdir build
4. cd build
5. cmake -DCMAKE_C_COMPILER=clang -DCMAKE_CXX_COMPILER=clang++
-DROSE_ROOT=/home/lin32/Development/projects/ROSE/build/autotool-default-EDG5-withClang/install_tree
-DBoost_ROOT=/nfs/casc/overture/ROSE/opt/rhel8/x86_64/boost/1_73_0/llvm/14.0.0/gcc/11.2.0 ../
6. make
```

Example

- Currently the alias analysis is executed to compare alias information among all the operands of instructions.
- The report has 3 parts:
  - list of ROSE SgNode and its source line/column info.
  - the LLVM IR and source line info (only when -g is used) and
  - the alias analysis report and source line info (only when -g is used).
- For the alias analysis report, ROSE source code information is printed if
  - -g option is given to LLVM to provide source line/column info for the operands and
  - the source line/column info from LLVM matches the line/column info from ROSE.
- Example code:

```
#include <stdlib.h>

int main() {
   int a;
   int b;
   int *p=&a;
   int *q=&a; // q is now an alias of p
   int* r=0;
   if(rand()) // make sure both branches are analyzed
        r=&a;
    else
        r=&b;
    *r=100; // r points to a or b
    q = r;
}
```

- Command without debugging option: clang -O0 -Xclang -load -Xclang libClangRosePlugin.so -fpass-plugin=libLLVMRosePass.so ../test/test1.c -c

```
MayAlias:
        op1:            LLVM operand info: (i32* %2)

        op2:            LLVM operand info: (i32* %13)

MayAlias:
        op1:            LLVM operand info: (i32* %2)

        op2:            LLVM operand info: (i32* %12)

MayAlias:
        op1:            LLVM operand info: (i32 ()* @rand)

        op2:            LLVM operand info: (i32* %13)

MayAlias:
        op1:            LLVM operand info: (i32 ()* @rand)

        op2:            LLVM operand info: (i32* %12)
```

- Command with debugging information: clang -O0 -Xclang -load -Xclang libClangRosePlugin.so -fpass-plugin=libLLVMRosePass.so ../test/test1.c -c -g

```
MayAlias:
        op1:    src info: [9:7]
                LLVM operand info: (i32 ()* @rand)
                ROSE node Info: rand

        op2:    src info: [13:7]
                LLVM operand info: (i32* %12)
                ROSE node Info: *r = 100

MayAlias:
        op1:    src info: [9:7]
                LLVM operand info: (i32 ()* @rand)
                ROSE node Info: rand

        op2:    src info: [14:7]
                LLVM operand info: (i32* %13)
                ROSE node Info: q = r

MayAlias:
        op1:    src info: [6:9]
                LLVM operand info: (i32* %2)

        op2:    src info: [13:7]
                LLVM operand info: (i32* %12)
                ROSE node Info: *r = 100
```