

АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІСТЬ СИСТЕМИ ДЛЯ ПОШУКУ КЛОНІВ КОДУ ПРОГРАМ

ARCHITECTURE AND FUNCTIONALITY OF THE SYSTEM FOR SEARCHING PROGRAM CODE CLONES

Для побудови інструменту пошуку клонів коду програм в якості бази обраний LLVM. LLVM є компіляторною інфраструктурою з відкритим вихідним кодом на мові C++. В рамках цього проекту представлений інструментарій для розробки ПЗ. LLVM (рисунок 1) містить статичний компілятор, компоувальник, віртуальну машину і JIT-компілятор, які працюють над єдиним внутрішнім представлення програми (біткод). Біткод може бути представлений трьома різними формами: у текстовому вигляді; у вигляді структур даних в оперативній пам'яті; в подвійному вигляді. Інфраструктура LLVM дає можливість зберегти біт код в проміжних об'єктних файлах для подальшої оптимізації, в тому числі динамічної. Усі надані LLVM можливості по обробці внутрішнього представлення (в тому числі різні аналізи, перетворення і т.д.) можуть бути застосовні до біткоду.

Тому компіляторна інфраструктура LLVM є зручною платформою для семантичного аналізу програми. Використовуючи аналізи LLVM з біткода, можна отримати інформацію про потік даних і про потік управління. З біткода також можна отримати інформацію про номери рядків вихідного коду, з якого він був побудований, якщо є налагоджувальна інформація (при компіляції дана опція «-gddb»). Основними особливостями LLVM є: реалізація на C++; модульна і розширювана архітектура; статичний компілятор з можливістю динамічної компіляції біткода.

Для LLVM є кілька компіляторів переднього плану: C, C++, Objective-C (Clang, GCC / dragonegg).

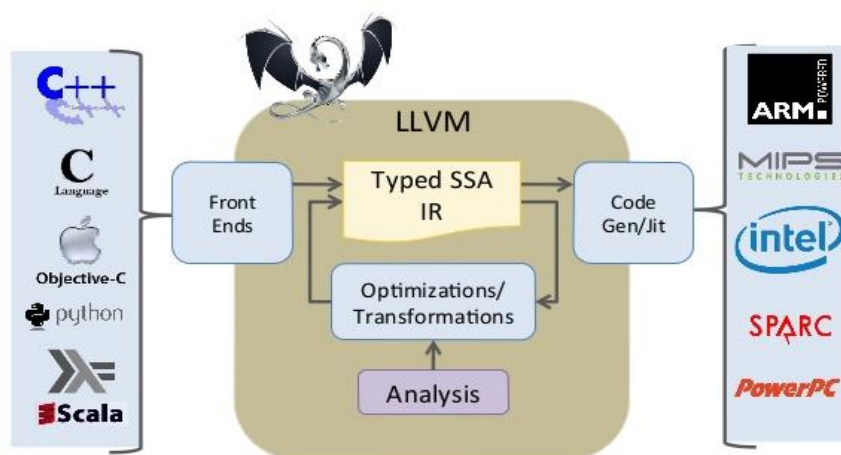


Рисунок 3.1. Структура LLVM (Low Level Virtual Machine)

Генерація ГЗП проекту проводиться на основі біткода під час компіляції проекту. Пошук клонів коду проводиться окремо. Для цього реалізований окремий інструмент, який на вхід отримує директорію з ГЗП графами і здійснює пошук клонів. Він також містить систему автоматичної генерації клонів коду, для аналізу і поліпшення розроблених алгоритмів. Реалізований інструмент підтримує паралельну обробку для багатоядерних систем. Одночасно можуть бути оброблені кілька параграфів для визначення клонів коду.