1. 问题描述Problem Description

In order to figure out the mapping relationship between the test cases and the functions in the source code of the target project, an approach is needed to statically or dynamically trace the execution of every function and record it.

1. 可用方案Candidate Approaches
   1. AspectC++ (Aspect-Oriented Programming)

AspectC++ ([http://www.aspectc.org](http://www.aspectc.org/)) is an aspect-oriented extension of C and C++ languages. It has a source-to-source compiler, which translates AspectC++ source code into compilable C++.The compiler is available under the GNU GPL.

Aspect-oriented programming (AOP) allows modularizing cross-cutting concerns in a single module, an aspect. Aspects can modify existing classes, but most commonly they provide 'advice' that runs before, after, or around existing functionality.

For the IUT project, AspectC++ enables us to log the “signature” (type, name and scope) of every function before, after, or around its execution WITHOUT revising the source code of the target project.

* + 1. 基本做法 Basic Implementation
       1. Coding

There are several concepts in AspectC++ which are essential for writing AspectC++ codes(refer to <http://www.aspectc.org/doc/ac-quickref.pdf>).

*aspect*

Aspects in AspectC++ implement in a modular way crosscutting concerns and are an extension to the class concept of C++. Additionally to attributes and methods, aspects may also contain *advice* *declarations*.

*advice*

An advice declaration is used either to specify code that should run when the *join* *points* specified by a *pointcut* *expression* are reached or to introduce a new method, attribute, or type to all *join* *points* specified by a *pointcut* *expression*.

*slice*

A slice is a fragment of a C++ element like a class. It may be used by

introduction *advice* to implemented static extensions of the program.

*join* *point*

In AspectC++ join points are defined as points in the component code where aspects can interfere. A join point refers to a method, an attribute, a type (class, struct, or union), an object, or a point from which a join point is accessed.

*pointcut*

A pointcut is a set of join points described by a *pointcut* *expression*.

*pointcut* *expression*

Pointcut expressions are composed from *match expressions* used to find a set of join points, from pointcut functions used to filter or map specific join points from a pointcut, and from algebraic operators used to combine pointcuts.

*match expression*

Match expressions are strings containing a search pattern.

*order declaration*

If more than one *aspect* affects the same *join* *point* an *order declaration* can be used to define the order of advice code execution.

In AspectC++, *aspect*s are defined in the header file with the extension name “.ah”. And in the definition of an *aspect*, we can define several *pointcut*s and *advice*s that execute additional functions when the target program executes to the *join point*s of the *pointcut*.

* + - 1. Compiling (Weaving)

AspectC++ requires an specific compiler to “weave” the code we write in the “.ah” file to the target program – ac++/ag++.

The program ac++ is a compiler for the AspectC++ programming language. It is implemented as a preprocessor that transforms AspectC++ code into ordinary C++ code. During this transformation aspect code, which is defined by aspects, is woven statically into the component code. Aspects are a special AspectC++ language element, which can be used to implement crosscutting concerns in separate modules. Aspect definitions have to be implemented in special “aspect header files”, which normally have the filename extension “.ah”. After the code transformation the output of ac++ can be compiled to executable code with ordinary C++ compilers like GNU g++, or Microsoft VisualC++.

The ag++ program provides a more intuitive frontend to the AspectC++ weaver (ac++) in a GNU environment. The only preliminaries are a working installation of GNU C++ compiler, which also can run within a cygwin environment. It basically wraps the functionality of the aspect weaver and the c++ compiler into one single program.

The usage of ac++/ag++ is similar to g++, a simple example (assume “.ah” files share the same directory of the “main.cc” file) is

“ag++ main.cc –o main”

* + 1. 优缺点 Pros and Cons
       1. Pros

1. Based on AspectC++ APIs, simple to implement.
2. No change of source code the target project.
3. Few LoC, reduce the possibility of error.
   * + 1. Cons
4. Require a specific compiler to compile.
5. Need to learn AspectC++.
6. Might have performance problem.
7. 决定