Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

GRID & Cloud Computing Group

POP-C++

Status on application termination

Author:
Valentin Clément

Date: October 11, 2011
Revision: 1.0

Contents

1 Introduction

In a distributed application, the termination is a non trivial problem. In fact, when the main of the application is over, many parallel object can still be running.
The main of the application should be able to determine when the whole application is ended. This document aims to explains the current status of the application termination in POP-C++ 2.0 and find a way to make it better.

2 Current status

This chapter aims to explain the current situation of the application termination in POP-C++.

2.1 Parallel object destructor

As a parallel object is divided into two parts (interface-side and broker-side), so the termination of a parallel object is also divided into two parts.

Interface-side
The termination on the interface side is the easiest one. Indeed, the interface acts as a normal object and its destructor is called. When this destructor is called, the combox and the buffer linked with this interface will be destroyed as well. This action will drag along the destruction of the broker-side of the parallel object.

Broker-side
The borker-side will notice the connection lose and make a DecRef() on its related object. Once the instance counter reaches 0, the threads receiving the request will run until all invocations on the object have been terminated. Once this point is reached, the object itself will be destroyed.

2.2 Why the termination is slow ?

3 New termination

4 Table of figures

5 References

[1] Adrian Wyssen, *VirtualPOPC-1 : Project Report*. EIA-FR, Switzerland, June-August 2010.

[2] Valentin Clément, *POP-C++ over SSH Tunnel*. EIA-FR, Fribourg, Switzerland, September-November 2010.

[3] Valentin Clément, *POP-C++ Virtual-Secure : POP-C++ User and Installation manual add-on*. EIA-FR, Fribourg, Switzerland, January 2011.