# 课题项目设计文档：使用 ACE 实现迪备客户端更新命令行工具（HF 项目）

# 概要设计

# 1、引言

## 1.1 项目背景

更新下载器是软件支持中的重要工具。它能够从特定地址检测更新、下载更新、安装更新。

更新下载器可以通过HTTP、FTP或其他协议，实现程序的更新管理与下载安装。

总之，更新下载器有着实用的用途。它对软件的更新至关重要。

## 1.2 项目目的

项目旨在实现DBackup客户端更新命令行工具，为DBackup下载更新，并根据输入参数下载最新版本客户端程序。

# 2、综合描述

## 2.1 用户类和特性

迪备客户端更新命令行工具的用户主体是系统管理员。他们能够用此工具更新本机的迪备客户端。

## 2.2 运行环境

更新工具依托一定的操作系统、数据库系统与Web服务器条件；同时，它需要在应用框架与语言Runtime支持下运行：

* 操作系统：更新工具可以在CentOS 7环境进行打包部署。
* 应用程序框架：更新工具在应用程序框架下进行开发、部署。需要ACE框架作为依赖。
* 语言Runtime：更新工具运行于glibc的语言Runtime上。

# 3、功能性需求分析

在更新工具的功能性需求分析中，需要对其基本功能进行定义和规划。

基本要求有：

1. 使用 C++ 实现一个控制台执行程序

2. 基于开源网络库 ACE(ADAPTIVE Communication Environment)搭建程序框架

3. 根据输入参数，比如代码分支名称，资源类型（比如指定 DB2 和 Informix)，到ftp://scutech@ftp.scutech.com/ftp\_product\_installer/dbackup3/rpm 下载最新版本的客户端程序。注意下级目录的命名方式：分支类型、分支名称、debug 或 release、CPU 架构、当前版本号，比如hotfix/bug-60697/debug/x86\_64/8.0.35276

4. 要求可以指定并发下载的线程数，多线程下载

5. 每个功能模块需要有 gtest 单元测试、Doxygen 对代码注释

6. 执行新安装或升级操作

7. 给出完善的报错信息

# 4、非功能需求分析

## 4.1 性能需求

更新工具对性能需求高。由于服务器限速，或者操作系统对TCP窗口的额外限制，最好支持多线程的文件下载。

## 4.4 可维护性需求

更新工具需要满足可维护性，来应对未来的代码更改。只有保证代码的良好质量、维护良好的文档、有良好的运行日志记录，才能确保可维护性达到要求。迪备更新工具的可维护性需求有：

* 代码质量：更新工具需要具备良好的代码质量，遵循统一的编码规范和设计原则。可以采用代码审查、测试管理等技术手段来提高代码质量。
* 文档化：更新工具需要提供详尽的文档和注释，文档可以是Doxygen自动化生成的。这样能够方便后续开发人员进行修改和维护。
* 日志记录：更新工具需要对系统的运行情况进行记录和分析，以便使用日志分析工具来对错误信息和系统异常信息进行分析和处理。

# 5、项目进度安排

对于更新工具，需要进行需求分析、概要设计、详细设计、编码、测试与部署维护。经过考虑，更新工具的合理的项目进度规划安排如下表所示：

|  |  |  |
| --- | --- | --- |
| **日期** | **阶段** | **工作内容** |
| **2023.07.10 2023.05.15** | 需求分析阶段 | 详细调研更新工具的功能需求与非功能需求。对比现存系统的优劣与更新工具的实际需要。 |
| **2023.05.15 2023.05.28** | 概要设计阶段 | 根据需求分析的结果，设计更新工具的整体架构、模块，以及其之间的联系。 进行技术选型，选取了更新工具的运行环境以及采用的技术方案。 |
| **2023.05.20 2023.05.21** | 详细设计阶段 | 为更新工具设计详细具体的实现方案。进行模块划分、控制流、数据结构的设计。 |
| **2023.05.21 2023.06.01** | 编码阶段 | 对更新工具进行编程开发，在此过程中采取测试驱动开发的实践，并且采用模块化开发的方法，最后组装模块。得到完整的更新工具。 |
| **2023.06.01 2023.06.07** | 测试阶段 | 对更新工具进行了完整的测试，确保功能完善、符合需求。 |
| **2023.06.07 2023.06.28** | 部署和维护阶段 | 将更新工具部署运行，进入维护阶段。 |

表1 项目进度安排

# 详细设计

# 1、引言

## 1.1 编写目的

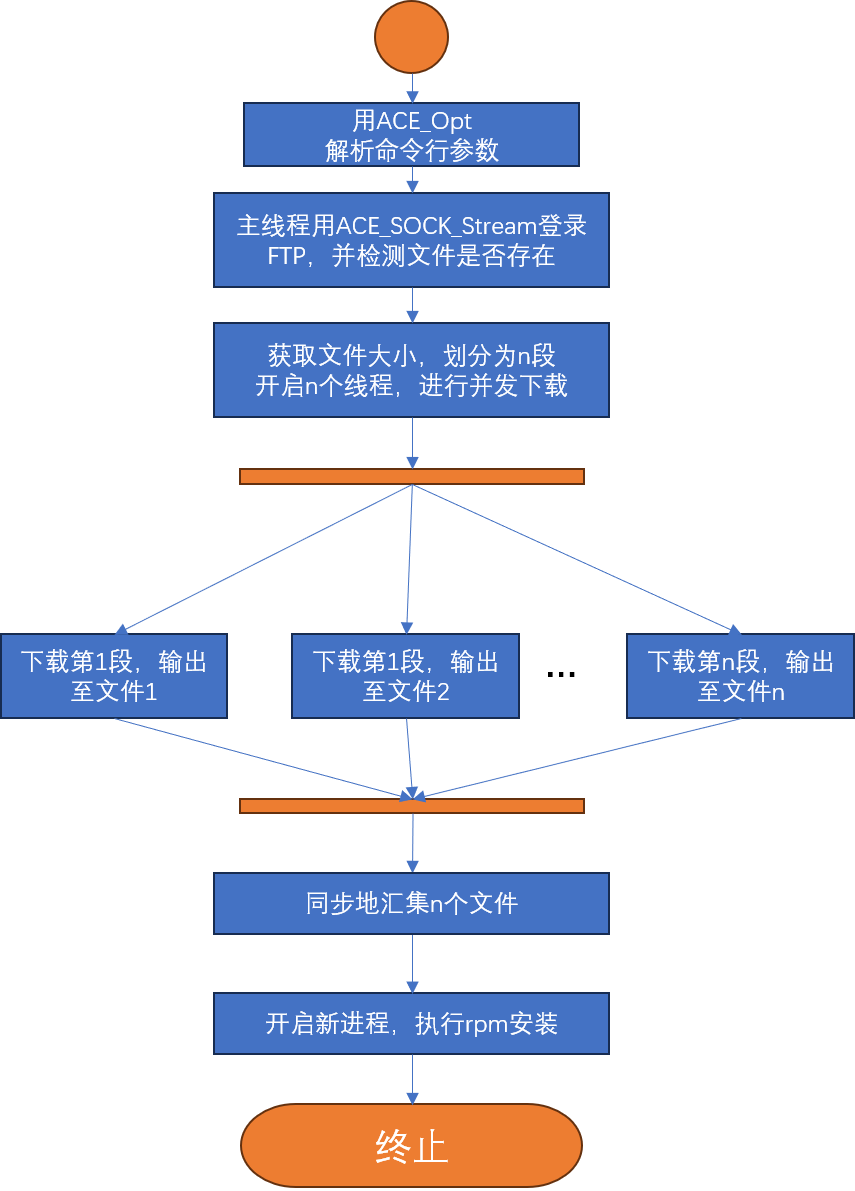
为了实现更新工具，需要对更新工具进行概要设计、详细设计。

需要根据需求分析的结果，设计更新工具的整体架构、模块，以及其之间的联系、进行技术选型，选取更新工具的运行环境以及采用的技术方案，得到概要设计。

需要为更新工具设计详细具体的实现方案。进行模块划分、控制流设计，得到详细设计。

## 1.2 概要设计

## 1.3 业务流程图



**图2 下载器流程图**

**工具设计**

工具负责读取用户指定的参数，并到FTP服务器寻找相应的文件，并发地进行下载，最后新建进程，对包进行安装。

基于上述组成部分的要素，我们可以设计出组件的结构：

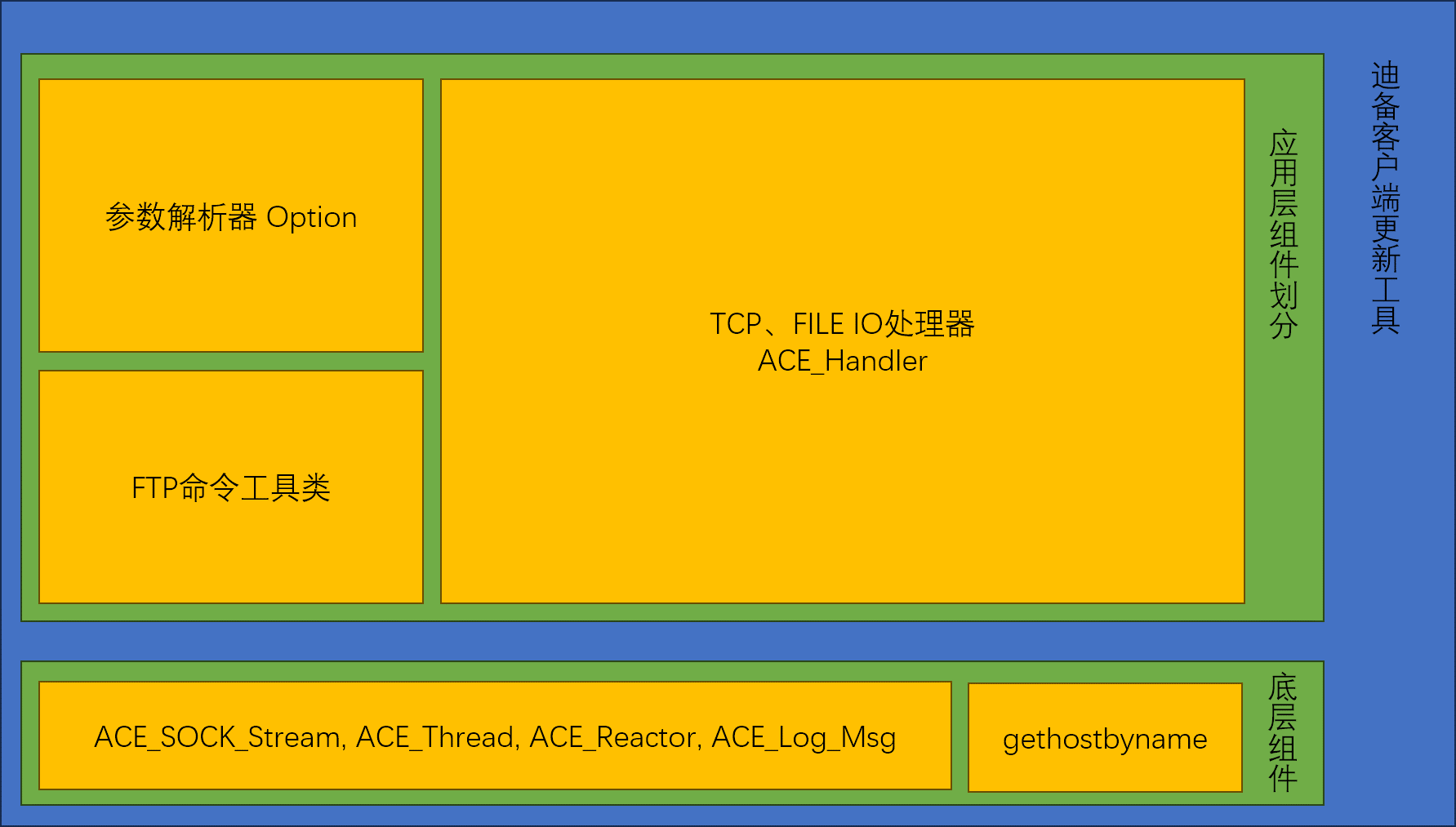


图3 下载器总体架构设计

其中，参数解析器负责解析argv数组传入的用户指定参数。它将argv数组解析为Option结构体中的机器可读数据，便于程序进一步处理。

FTP命令工具类负责FTP交互命令的生成。通过指定一些参数，FTP命令工具类可以输出FTP的命令报文，便于与FTP服务器做交互。

处理器都实现了ACE\_Handler接口，能够与ACE\_Connector做集成。TCP处理器分为FTP控制连接处理器和FTP数据连接处理器，负责与FTP服务器的控制和数据交互。FILE IO处理器则负责分块下载文件的读写。

# 3、类图及接口设计

## 3.1 类图设计

在更新工具中，涉及到若干个类，说明如下：

* Option类：负责从argv中解析出机器可读的选项结构体。
* FtpCommandHelper类：能够生成与下载任务相关的FTP指令，并从选项构造FTP文件路径，聚焦FTP相关的内容。
* FtpControlHandler类：继承ACE\_Handler，负责处理Connector打开的FTP控制连接。具体行为是：确定文件存在，并且打开数据连接并下载。
* FtpDataHandler类：继承ACE\_Handler，负责处理Connector打开的FTP数据连接。具体行为是：下载文件，分块缓存，并flush入硬盘。
* Downloader类：负责实现并发下载相关的逻辑，创建Socket Handler并驱动FtpControlHandler和FtpDataHandler进行数据下载与汇集操作。
* Updater类：是应用程序的入口。驱动Downloader下载，并在下载完成后开启新的进程，进行rpm包的安装。

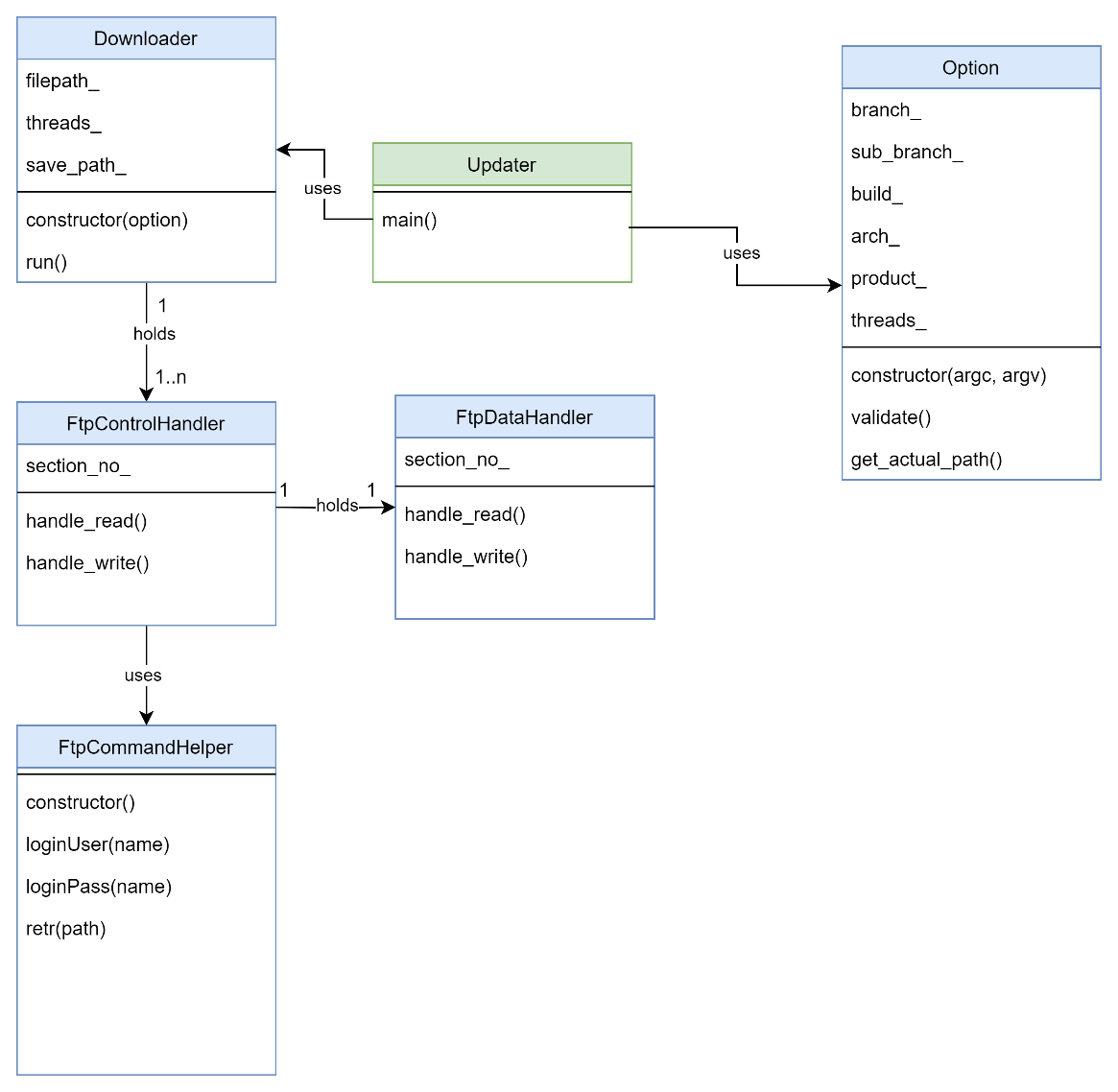


图4 更新工具类图