Httpcore需求和主要功能分析

2016K8009908053

1603班 孙越

1. **HTTP是什么**

根据RFC2616的定义，超文本传输协议（HTTP）是一种为共有的超文本信息传输系统定制的应用层协议。HTTP是一个客户端和服务器端请求和应答的标准（Transmission Control Protocol, TCP）。客户端通过使用Web浏览器、网络爬虫或者其它的工具发起一个到服务器上指定端口的HTTP请求。HTTP协议采用了请求/响应模型。整个基本的工作流程是客户端发送一个HTTP请求，说明客户端想要访问的资源和请求的动作，服务端收到请求之后开始处理，并根据请求做出相应的动作访问服务器资源，并以一个状态行作为响应，响应的内容包括消息协议的版本，成功或者错误编码加上包含服务器信息、实体元信息以及可能的实体内容。最后通过发送HTTP响应把结果返回给客户端。

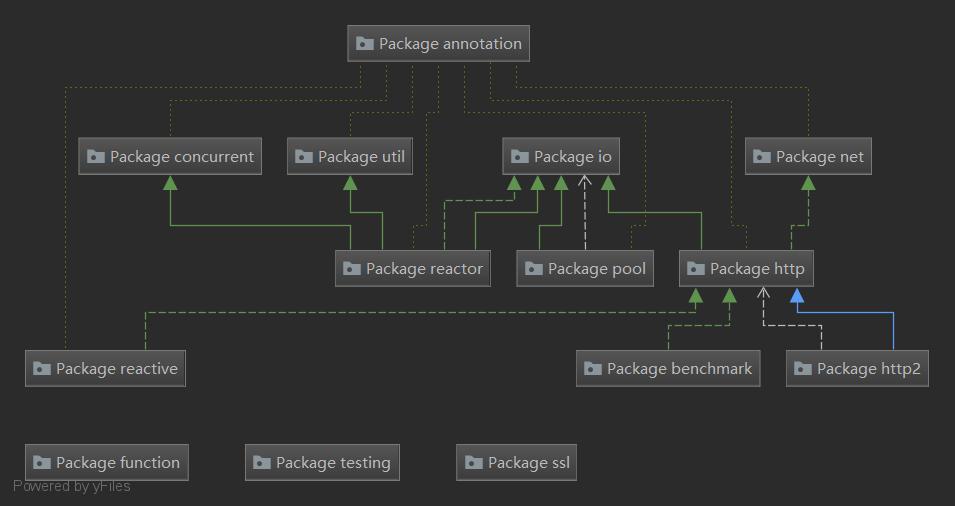
HTTP经历了多个版本，目前最常用的版本为HTTP1.1。版本号在每次HTTP请求的消息中是被要求的，一般处于消息头中。

1. **HttpCore的内容和主要功能**

HttpCore是基于超文本传输协议的一种实现方式，是一套实现了HTTP协议最基础功能的组件。HttpCore可以理解为一种人为编写的应用接口，这种接口实现了用户（应用端）与服务供应商（服务器端）还有中间的代理端之间信息交互的一致性，有效确保了信息传递的效率和安全性。

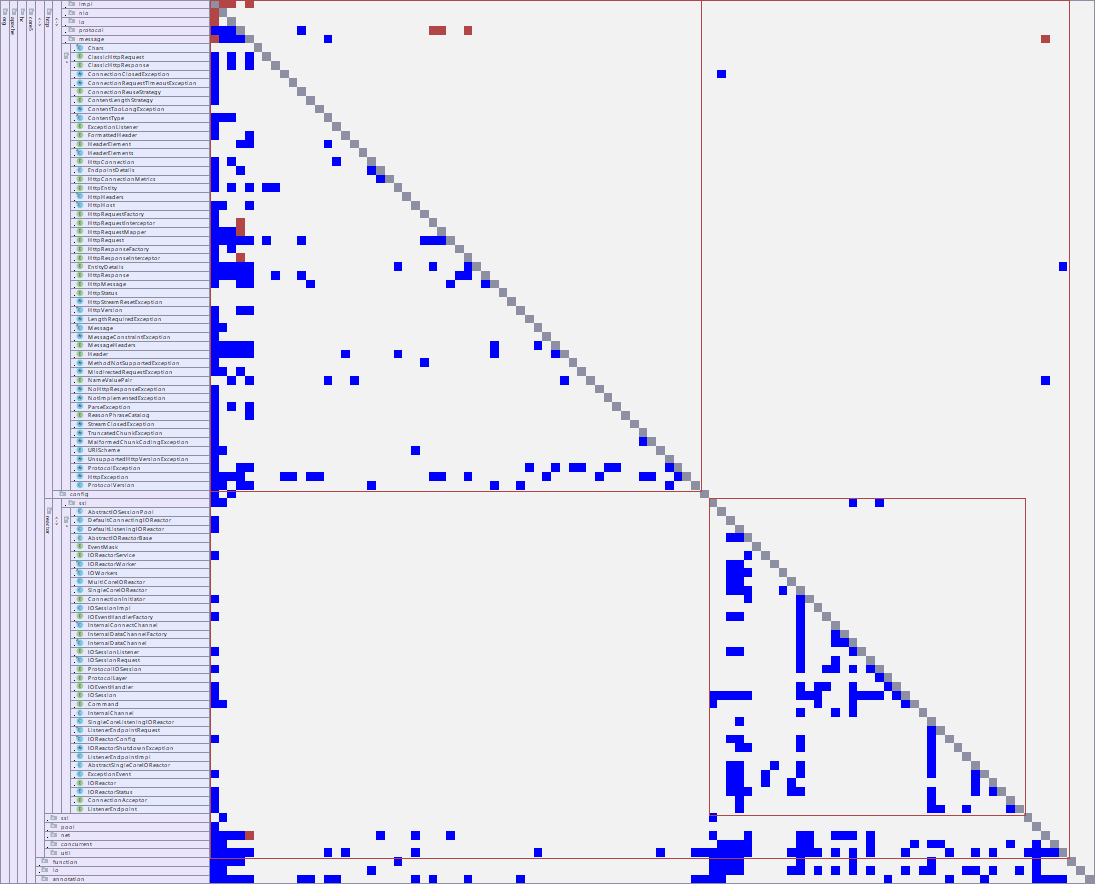
HttpCore需要实现最基本的HTTP传输，包括HttpRequest、HttpResponse等，以及具体方法比如GET和POST。这是最简单的模型，同时还要实现这些简单模型简单操作的复杂要求，包括阻塞与非阻塞的I/O模型等。

可以从代码的角度对整个HttpCore项目进行分析。本文只针对httpcomponents-core/src/main/java/org.apache.hc.core5/ 文件夹的代码进行分析，这些代码足够体现HttpCore的主体设计思想、主要需求实现和代码规范，其他文件夹中的代码与分析代码的相关性有限并且有较多重复，将不在本文报告中进行进一步阐述。



**Figure 1. Relation between components**

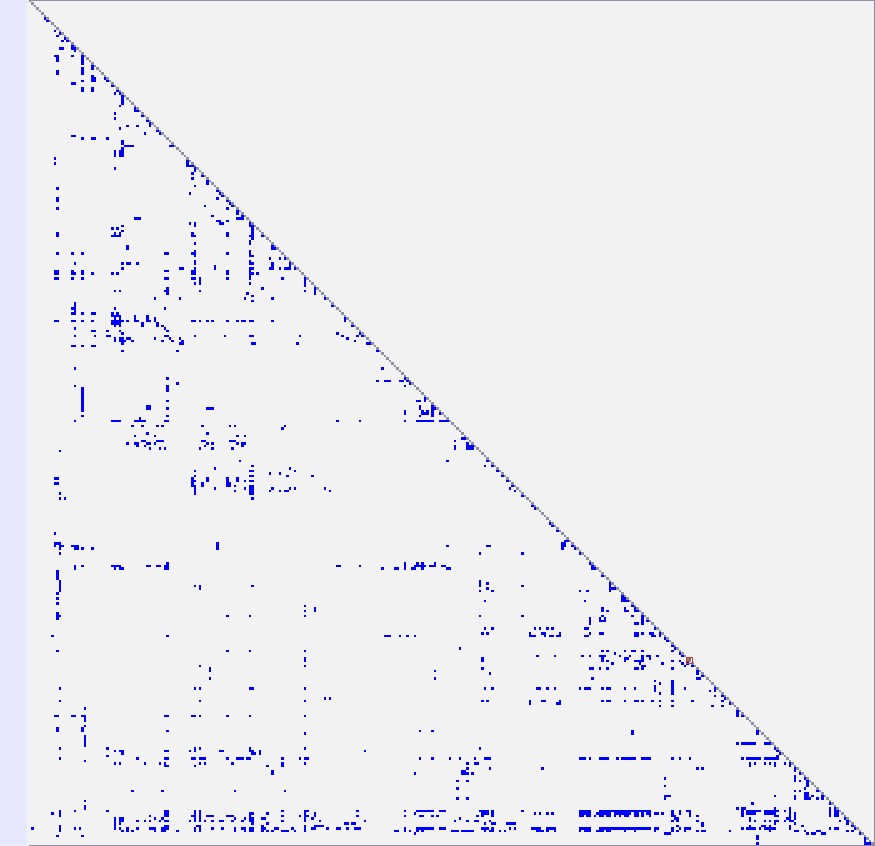
如图为本文分析的文件夹中包含的多个包之间的对应关系，不难发现其中的function、testing和ssl与代码主体的相关性有限。根据UML图对于包之间的描述，包reactor中有对包concurrent、util和io的继承，有对包io的实现；包pool中有对包io的继承和局部变量、方法或者静态方法的调用（依赖关系）；包http有对包io的继承和包net的实现。以上三个包构成了分析问题的核心。下面的三个包reactive、benchmark、http2是对包http的继承和依赖，所以包http是我们分析的重点。



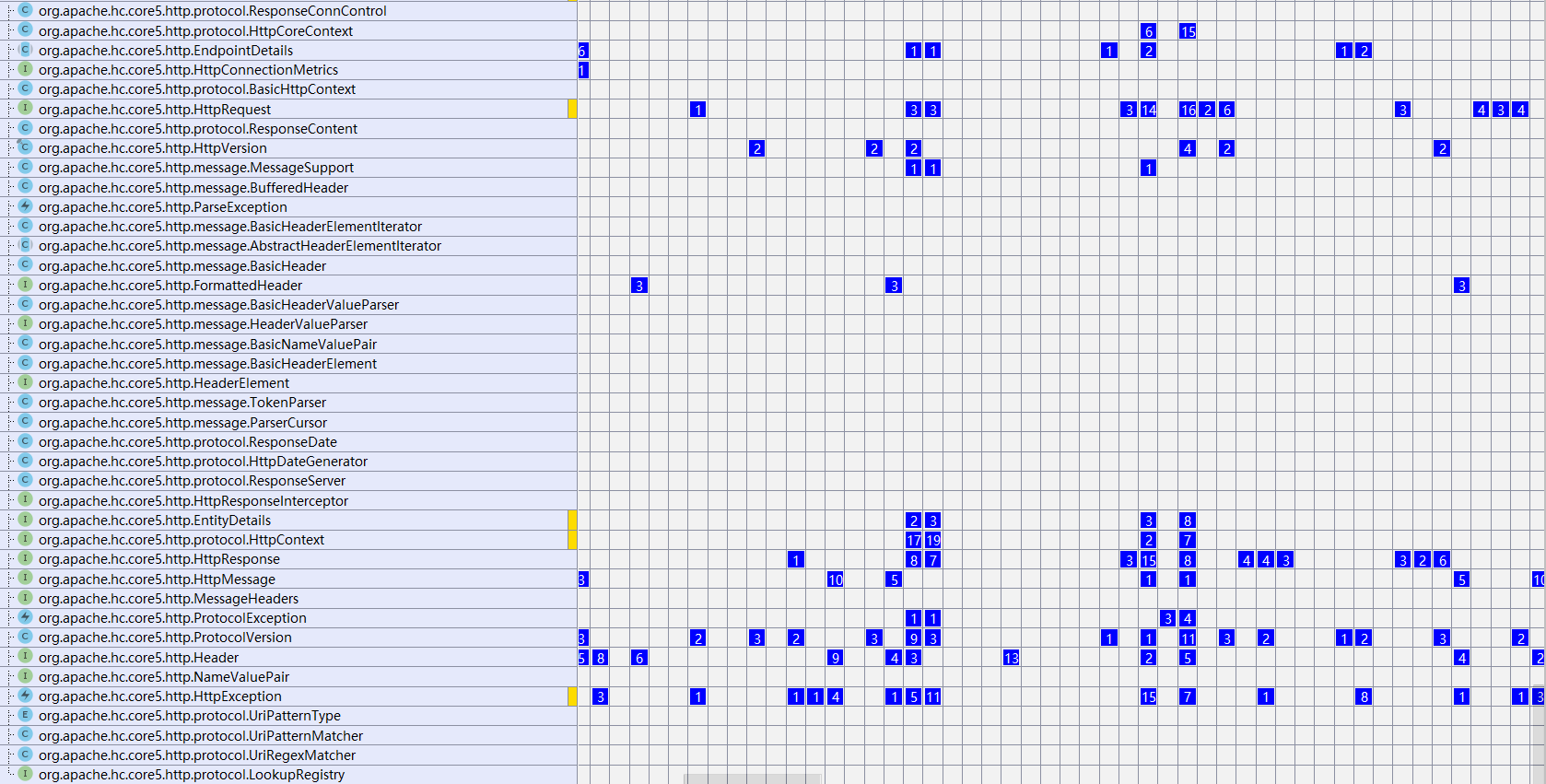
**Figure 2. DSM plot of the main frame----org.apache.hc.core5**

上图是对httpcomponents-core/src/main/java/org.apache.hc.core5/进行依赖分析(dependency structure matrix, DSM)后得到的一张DSM图像。DSM图像可以形象的展现类与类之间、类与接口之间的关系。对于大型复杂系统来说，相比起UML图而言，DSM图对于任务之间的信息交互的展示更加直观，可以明显的体现出依赖关系。比如上图中是将core5核心代码中的两个最主要的package文件夹http和reactor展开之后的结果，可以发现文件的架构呈现出高聚合性和低相关性的特点，即包与包之间的联系较少，而包内联系较为紧密。

下图是对package http的所有接口与类进行依赖矩阵分析之后得到的图。整个结构看上去较为清晰，没有明显的过于依赖其他模块的结构，说明程序设计的基本思路较为清晰，没有冗杂。同时不难发现底层的几行模块被大量其他模块使用。这说明文件整体（或者说HttpCore这个API整体）对于底层几行模块的依赖性较大。我们一般称这类模块为工具模块或者基础模块。这些模块也将成为之后对于程序分析的切入点。



**Figure 3. DSM plot of the main package----org.apache.hc.core5/http**



**Figure 4. The bottom of DSM plot of the main package----org.apache.hc.core5/http**

从上图可以发现那些基础模块包括：HttpMessage, HttpResponse, HttpRequest, HttpHeader, HttpException等，而这些恰好是HttpCore实现的基本功能。

1. **具体功能分析**



以上是对于整个http包的UML分析图，其主要功能可以分为以下几块：HttpHeaders，即Http请求中的消息头部分，其最主要的实现为HttpMessage，包括HttpRequest和HttpResponse，是HttpCore最核心的内容，也是整个项目最需要实现的对象；HttpEntity，Http请求中的消息实体，包括其他附属内容；HttpConnection，包括网络服务器的连接状态；HttpException，其主要实现是协议例外。其他的包与整个功能主体的相互关联较小，本文就暂且不做过多的分析。将对每一个功能的具体实现和包之间的相互调用关系进行梳理。

**【参考文献】**

[1]<https://www.w3.org/Protocols/rfc2616/rfc2616.html> , last access 2018.10.31 原文如下：“The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.”