<Paperclip>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <06/09/2018> | <1.00> | <initial初稿> | <陈诺、胡雨奇> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 5

1.5 概述 5

2. 测试概要 5

3. 测试环境 5

4. 测试结果及分析 6

4.1 需求覆盖率及缺陷分布 13

4.2 缺陷严重程度 14

5. 缺陷清单 14

5.1 功能性缺陷 14

5.2 非功能性缺陷 15

6. 测试结论与建议 16

测试报告

# 简介

## 目的

<Paperclip> 的这一“测试计划”文档有助于实现以下目标:

* + - * 1. 为软件可靠性与安全性的评估提供依据
        2. 提供系统化、规范化、工程化、实用化的测试技术规范，尽早发现故障。
        3. 检查软件对误操作的处理能力
        4. 此计划适用于系统中各组成部分的软件测试工作。
        5. 验证软件是否满足软件设计所规定的技术要求

## 范围

测试工作包括：

* + 1. 功能测试
    2. 性能测试
    3. 易用性测试
    4. 安全性测试
    5. 兼容性测试
    6. 界面测试
    7. 压力测试
    8. 单元测试

## 定义、首字母缩写词和缩略语

定义：

1. 文档：用户自己的创作，为可编辑文档，默认仅自己可见，成功邀请协作者后，协作者也可见；
2. 发布：用户可以发布自己的文档，发布后会产生与该文档当前可编辑版本内容完全一样的一个PDF版本，包含版本号、发布时间等信息。发布后的PDF版本支持作者、协作者写批注、评论批注；
3. 邀请协作：邀请其他用户查看、批注自己已发布的文档（被邀请协作的用户可以看见该文档的所有版本）；
4. 笔记：用户看过一篇论文后对于此论文整体的探讨、感想、引申等等，可以写一篇关于此论文的笔记，所有用户都可以对该笔记进行：浏览、收藏、点赞、评论等操作；
5. 批注：用户对于一篇论文局部的想法、解释、引申等等，在论文中与论文的任意长度内容耦合，所有用户可以对所有批注进行评论；
6. 标记批注：用户可以选择标记某些论文的批注，用户可以选择显示所有已标记的批注，也可以选择隐藏所有批注。用户自己写的批注默认被自己标记；
7. 导出：用户可以选择某一篇论文导出PDF，导出时有两种选项：导出论文、导出论文及标记的批注。第一种仅仅产生论文的PDF，第二种产生的PDF会在文中把已标记批注所耦合的论文内容画上线，同时在文末按顺序列出标记的批注；

## 参考资料

《软件工程原理》 2013.2 沈备军，陈昊鹏，陈雨亭

《paperclip软件需求归约》

《paperclip测试用例》

## 概述

本测试文档以测试为核心，从测试概要开始，接着列出测试的环境，然后较详细的列出了测试的结果以及对于结果的分析，之后罗列出测试中发现的缺陷，最后得出了本次测试的结论与建议。

# 测试概要

测试时间为2018/09/04-2018/09/07，

测试地点为上海交通大学闵行校区，

主要测试人员有胡雨奇、陈诺。

测试中，涉及用户体验的均为手动访问系统测试。单元测试中，；前端的测试主要使用Blazmeter录制.jmx脚本，使用JMeter进行测试。

# 测试环境

软件：

Chrome 68.0.3440 64位版本

Blazmeter 4.1.0版本

Apache JMeter 4.0版本

Intellij IDEA（IDE）2017.3.4 64位版本

Junit 4.12版本

硬件：

Intel Core i5

内存 8G

268G固态硬盘 512G机械硬盘

# 测试结果及分析

测试的种类及通过率如下表所示

**表1. 测试种类及通过率**

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **用例数** | **通过数** | **通过率** |
| **功能测试** | **30** | **30** | **100.00%** |
| **性能测试** | **32** | **25** | **78.13%** |
| **易用性测试** | **5** | **5** | **100.00%** |
| **可靠性测试** | **0** | **0** | **0.00%** |
| **安全性测试** | **2** | **2** | **100.00%** |
| **兼容性测试** | **1** | **1** | **100.00%** |
| **界面测试** | **4** | **4** | **100.00%** |
| **压力测试** | **16** | **12** | **75.00%** |
| **部署测试** | **0** | **0** | **0.00%** |
| **单元测试** | **15** | **15** | **100.00%** |
| **合计** | **105** | **95** | **90.48%** |

其中，除去性能测试和压力测试之外全部100%通过。性能测试和压力测试中，性能测试没有出现错误，但响应时间达不到本系统《软件需求归约》中2s的标准，故出现了7个未通过的用例。压力测试中，除去响应时间过长外，还出现了错误。压力测试中出现的错误（JMeter的GUI中查看结果树得知）全部来自于拒绝访问，没有出现严重影响系统或对数据库造成不可回滚的改变。

下面着重就性能测试和压力测试做出一些分析。

**性能测试：**

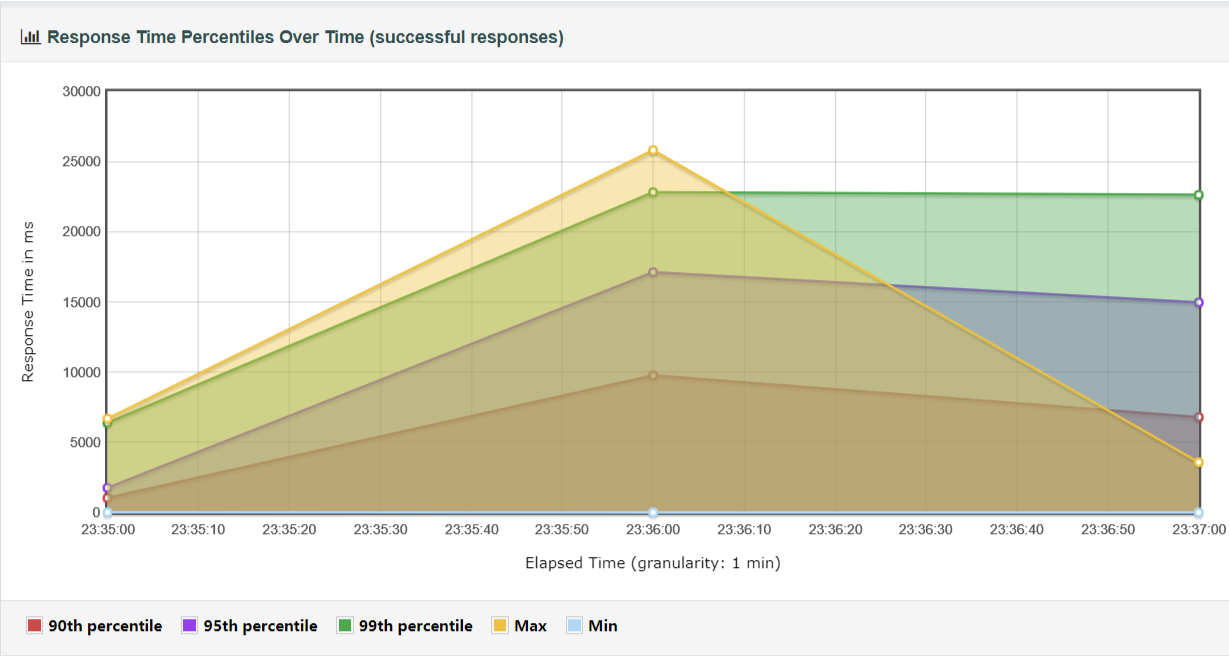
性能测试包括两组：1000在线用户且50并发用户不断请求系统能提供的各种操作、2000在线用户且100并发用户不断请求系统能够提供的各种操作。

**由于所有测试的请求连续不断的在一个测试中进行，故我们认为尽管并发量不大，但50或100的并发量对应的在线人数会比较可观。同时，由于绝大部分功能的请求都在短时间内并发进行，这样的测试数据应该更接近真实使用场景，并且测试出的性能会差于单独测某一项功能所测试出的性能。**

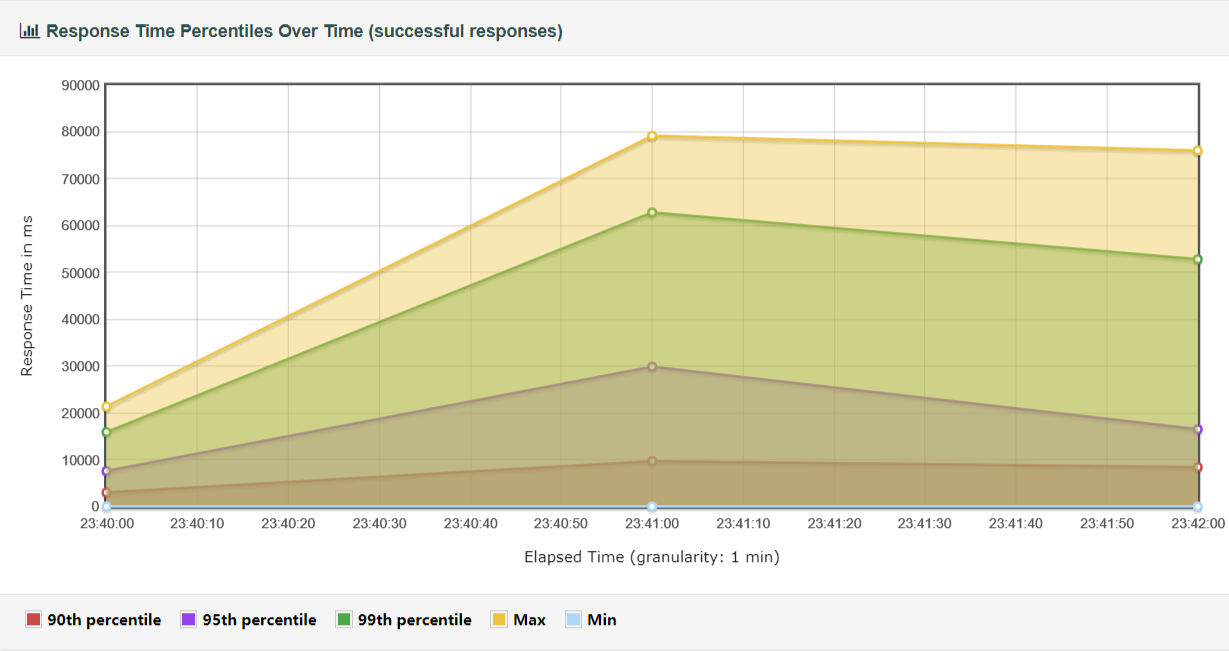
测试时间如下表

**表2. 性能测试时间**

|  |  |  |
| --- | --- | --- |
| **并发用户数量** | **测试开始时间** | **测试结束时间** |
| **50** | **18/9/6 11:35 PM** | **18/9/6 11:35 PM** |
| **100** | **18/9/6 11:40 PM** | **18/9/6 11:42 PM** |



**图1. 50并发用户时，所有测试响应时间总和随测试进行的变化**

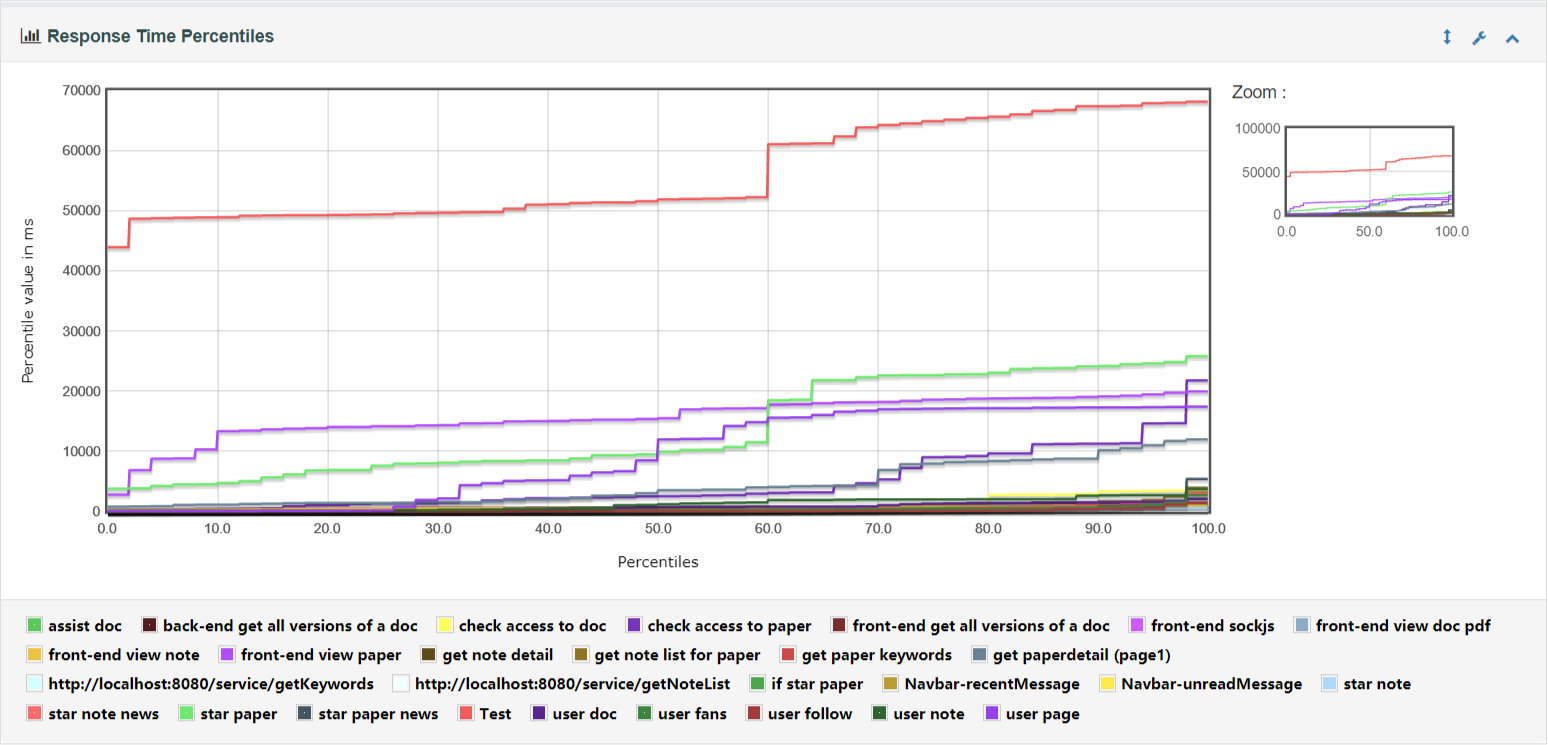


**图2. 100并发用户时，所有测试响应时间总和随测试进行的变化**

注：

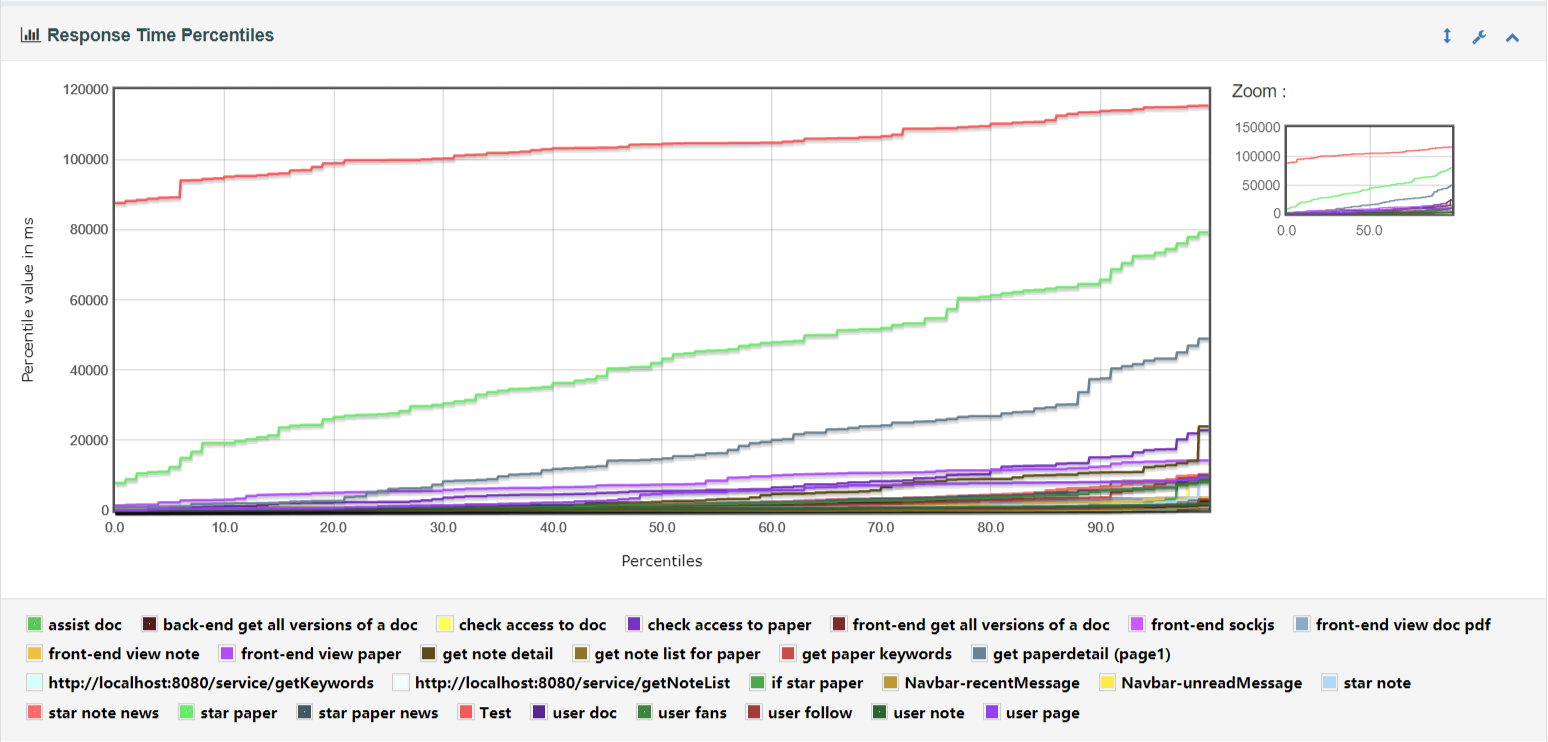
90th percentile表示90%的用户响应时间最大值，95th percentile和99th percentile类似

结合上表可知，在各线程不断发送请求时，前端、后端压力越来越大，导致最大响应时间越来越长，值得庆幸的是大多数用户的响应时间并没有上升太多。测试进行到后半部分时，由于已经完成任务的线程开始沉寂，前后端压力减小，响应时间也在相应降低。



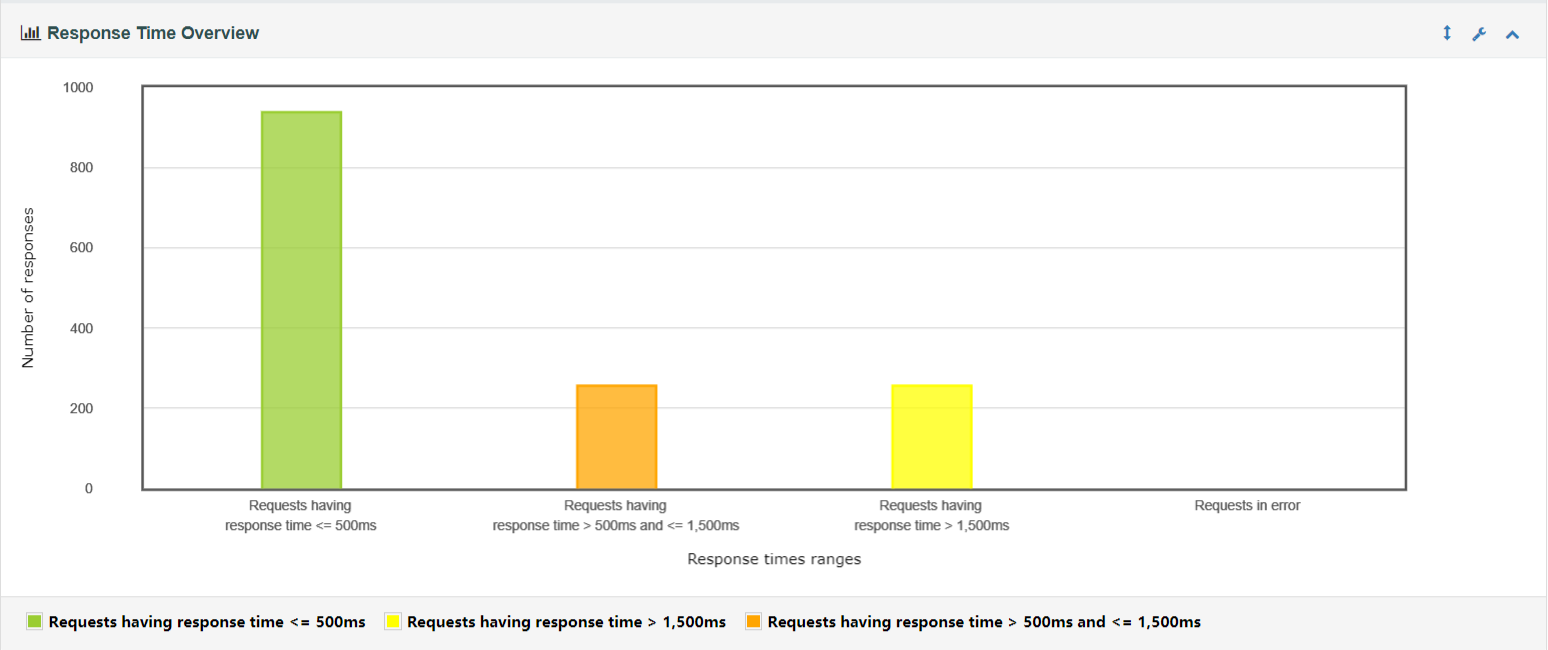
**图3. 50并发用户时，各功能响应时间随测试进行的变化**

注：最上面一条为时间总和

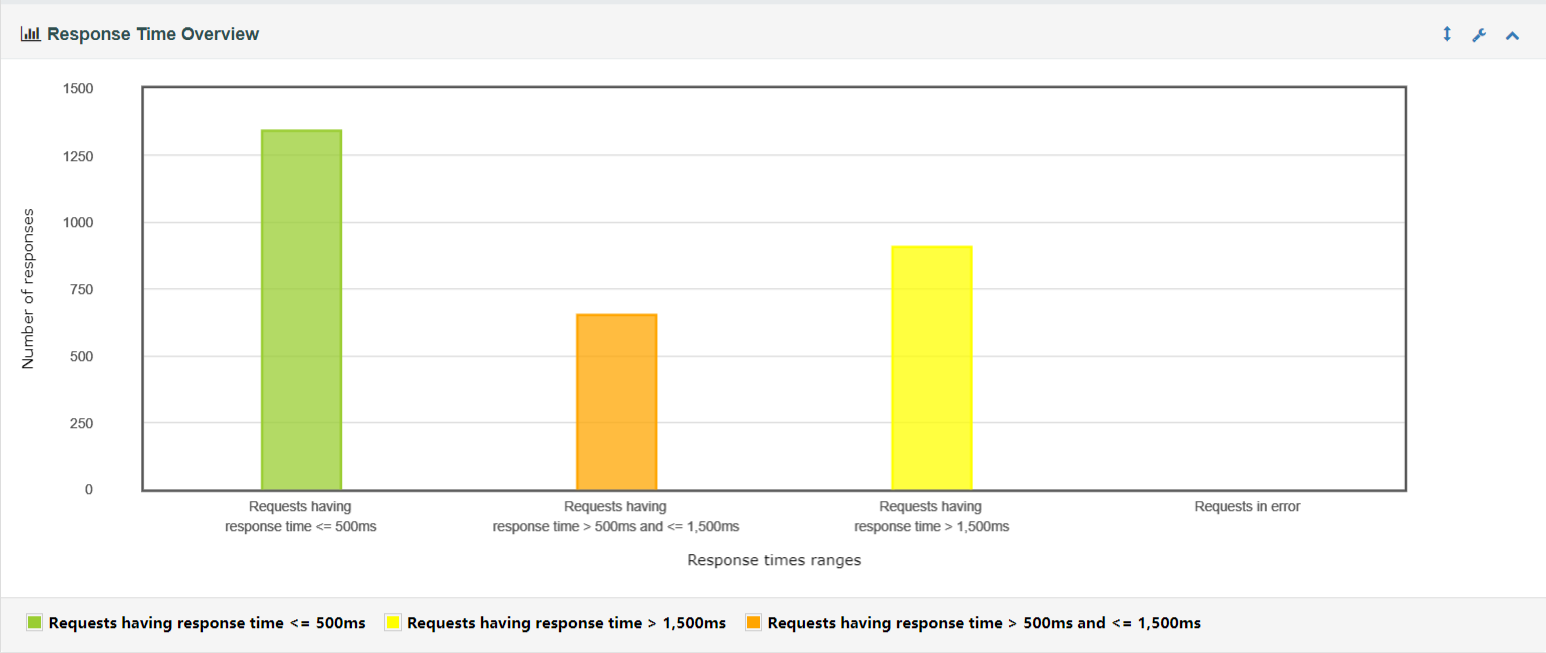


**图4. 100并发用户时，各功能响应时间随测试进行的变化**

由以上两张图可以发现，在并发量上升后，查看用户收藏论文列表服务的响应时间和查看论文PDF服务响应时间显著增加，将在第六点详细讨论。



**图5. 50并发用户时，所有请求响应时间归类**



**图6. 100并发用户时，所有请求响应时间归类**

上面两张图将请求响应时间按照小于500ms、500-1500ms、大于1500ms和错误进行分类。（通常认为500ms为Toleration threshold，1500ms为Frustration threshold）。可以看出，并发用户较少时，响应时间小于500ms的请求占了大多数，用户体验尚可，当并发数上升后，响应时间大于500ms的请求数量大大增多，降低了用户体验，这时尚未出现错误。解决这种情况的建议在6.中会详细说明。

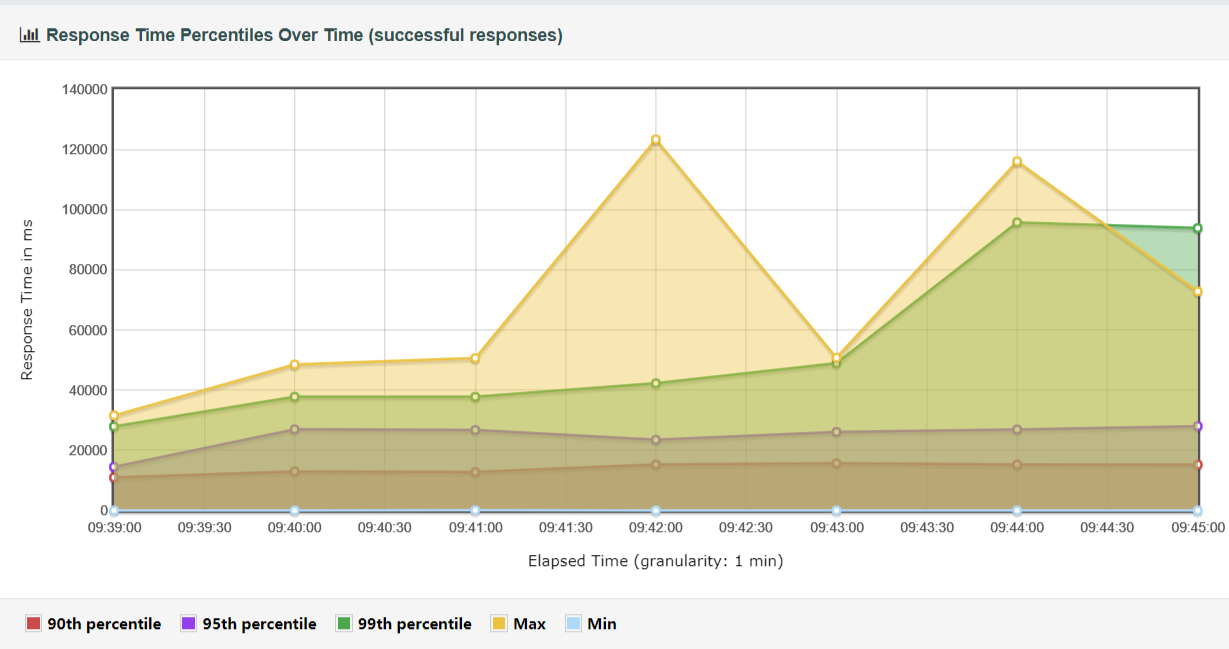
**压力测试**

压力测试的规模为4000在线用户，200并发用户不断请求系统能够提供的各种操作

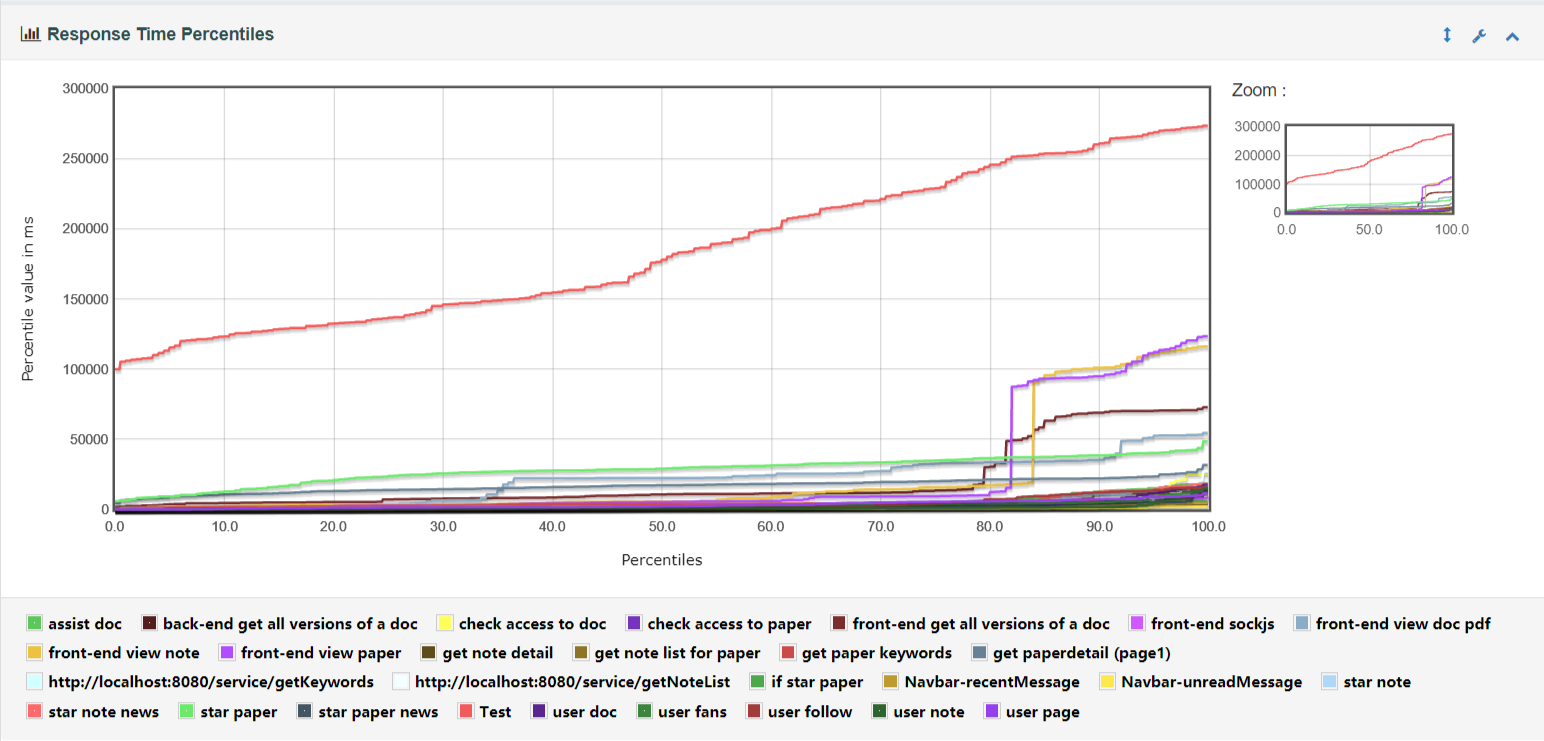
测试时间如下表

**表3. 压力测试时间**

|  |  |  |
| --- | --- | --- |
| **并发用户数量** | **测试开始时间** | **测试结束时间** |
| **200** | **18/9/7 9:39 AM** | **18/9/7 9:45 AM** |



**图7. 并发量为200时，响应时间随测试进行的变化**

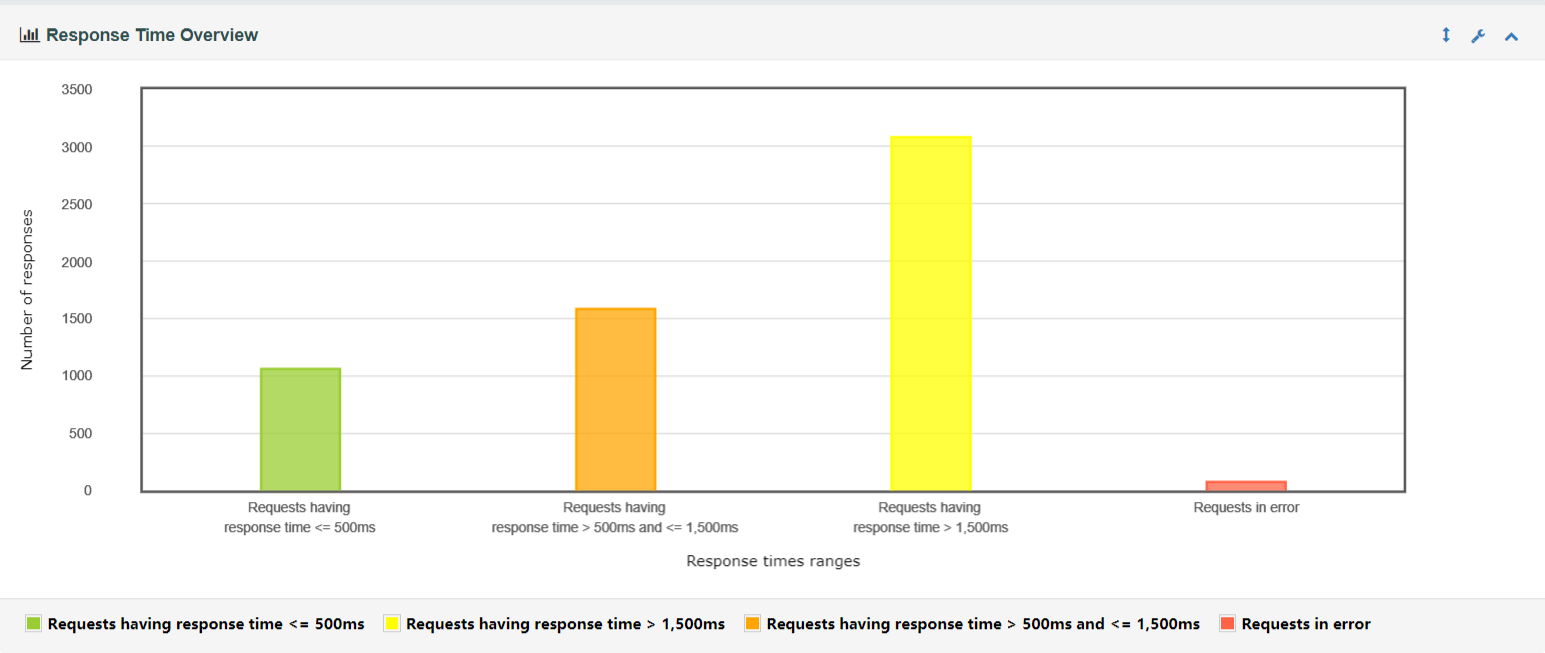


**图8. 并发量为200时，各服务响应时间随测试进行的变化**

注：图中最上面的一条红线代表的是测试总计。

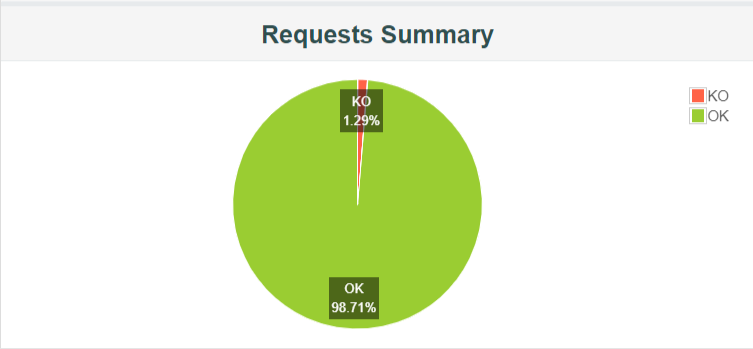
图中，除去在并发量为100时出现的响应时间大幅增加的两个服务（查看）外，这里还出现了其余高峰，这些高峰有前端查看论文的PDF、前端查看笔记、前端查看文档的PDF。由于传输PDF、加载附加在PDF的block批注及评论信息（即与文章中的某段话耦合的批注即评论）数据量非常大，因此并发量非常大时，会十分耗费时间（服务器端数据读取、传输速度受限）。在实际使用中由于我们实现的缓存机制（在浏览本页时加载下一页PDF和批注、笔记等信息），使用体验不会如测试一样糟糕。

响应时间在测试进行到80%时这几个服务开始出现陡峰的原因，在查看测试结果树之后发现，是JMeter线程按照脚本顺序执行，因此在进行到这个阶段时才开始大幅请求这些服务，从而造成了先平后陡升的图像。



**图9. 并发量为200时，所有请求响应时间归类**

可以发现，在并发量达到200时，大多数服务都不能使用户满意，有少量请求出现错误。在查看测试过程检查错误的请求后发现，错误全部为拒绝连接，没有使服务器或者客户端出现重大错误，也没有改变数据库。我们认为目前的性能测试已经可以满足SRS中2000在线用户的需求，如果要满足200个并发用户，需要升级硬件。但总体来说，高压测试的结果是可以接受的。我们认为系统可以在短时间内服务较多用户，尽管会有一些用户的连接被拒绝（见图10.），但是不会对系统造成不可逆转的损失。



**图10. 并发量为200时，请求错误率**。

## 需求覆盖率及缺陷分布

**表4. 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **注册** | 4 | 100% | 0 | 0.00% |  |
| **登录** | 2 | 100% | 0 | 0.00% |  |
| **注销** | 1 | 100% | 0 | 0.00% |  |
| **访问控制** | 1 | 100% | 0 | 0.00% |  |
| **输入验证** | 4 | 100% | 0 | 0.00% |  |
| **操作验证** | 6 | 100% | 0 | 0.00% |  |
| **增加操作验证** | 5 | 100% | 0 | 0.00% |  |
| **删除操作验证** | 3 | 100% | 0 | 0.00% | 批注和对批注的评论不支持删除 |
| **修改操作验证** | 2 | 100% | 0 | 0.00% | 批注和对批注的评论不支持修改 |
| **导出PDF验证** | 2 | 100% | 0 | 0.00% |  |
| **DAL类验证（单元测试）** | 15 | 100% | 0 | 0.00% |  |
| **功能项小计** | *42* | *100%* | *0* | *0.00%* |  |
| **非功**  **能项** | **性能** | 32 | 100% | 6 | 18.75% |  |
| **易用性** | 5 | 100% | 0 | 0.00% |  |
| **兼容性** | 1 | 100% | 0 | 0.00% |  |
| **界面** | 4 | 100% | 0 | 0.00% |  |
| **压力** | 15 | 100% | 4 | 26.67% |  |
| **安全性** | 2 | 100% | 0 | 0.00% |  |
|  | **非功能项小计** | *59* | *100%* | *10* | *16.95%* |  |
|  | **总计** | *105* | *100%* | *10* | *9.52%* |  |

## 缺陷严重程度

**表5. 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 10 | 0 | 10 |
| **占缺陷百分比** | 0 | 0 | 100% | 0 | 100% |

# 缺陷清单

## 功能性缺陷

**表6. 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
|  |  |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

**表7. 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序  号 | 缺陷  编号 | 严重程度 | 测试类型 | 缺陷标题 | 缺陷描述 | 测试用例编号 |
| 1 | 1 | 一般 | 用户主页 | 用户主页低压负载响应时间缺陷 | 在并发用户数较大时，用户主页响应时间较长。主要原因是用户主页需要加载的模块较多，包括用户头像、用户名及描述、用户收藏的论文笔记用户等最近动态 | paperclip-testcase-054 |
| 2 | 2 | 一般 | 用户主页 | 用户主页增量负载响应时间缺陷 | 在并发用户数较大时，用户主页响应时间较长。主要原因是用户主页需要加载的模块较多，包括用户头像、用户名及描述、用户收藏的论文笔记用户等最近动态 | paperclip-testcase-070 |
| 3 | 3 | 一般 | 用户收藏论文 | 用户收藏论文页增量负载响应时间缺陷 | 在并发用户数较大时，用户收藏论文页响应时间过长 | paperclip-testcase-071 |
| 4 | 4 | 一般 | 查看笔记详情 | 查看笔记详情页增量负载响应时间缺陷 | 在并发用户数较大时，由于笔记详情页要加载的数据较多导致响应时间较长 | paperclip-testcase-078 |
| 5 | 5 | 一般 | 查看文档版本列表 | 查看文档版本列表增量负载响应时间缺陷 | 在并发用户数较大时，用户文档的版本列表页响应时间过长 | paperclip-testcase-079 |
| 6 | 6 | 一般 | 查看文档PDF | 查看文档PDF增量负载响应时间缺陷 | 在并发用书数量过大时，由于读取、传输数据速度的限制，导致查看文档PDF页面响应时间过长（本页面需要数据量较大） | paperclip-testcase-080 |
| 7 | 7 | 一般 | 用户收藏论文页 | 用户收藏论文页高压负载响应时间缺陷 | 在并发用户数较大时，用户收藏论文页响应时间过长 | paperclip-testcase-087 |
| 8 | 8 | 一般 | 查看笔记详情 | 查看笔记详情高压负载响应时间缺陷 | 在并发用户数较大时，由于笔记详情页要加载的数据较多导致响应时间较长 | paperclip-testcase-094 |
| 9 | 9 | 一般 | 查看文档版本列表 | 查看文档版本列表高压负载响应时间缺陷 | 在并发用户数较大时，用户文档的版本列表页响应时间过长 | paperclip-testcase-095 |
| 10 | 10 | 一般 | 查看文档PDF | 查看文档PDF高压负载响应时间缺陷 | 在并发用书数量过大时，由于读取、传输数据速度的限制，导致查看文档PDF页面响应时间过长（本页面需要数据量较大） | paperclip-testcase-096 |

# 测试结论与建议

测试开始前一段时间，我们排查了一遍bug，并修复了已发现的bug，因此在测试过程中没有测出功能性缺陷。测试测出了较多的性能缺陷，并且性能缺陷全部集中在响应时间缺陷上。与其他小组测试人员交流沟通后，我认为造成这个现象主要有两个原因。

**原因：**

* 1. 我的测试计划（这里的测试计划指的是JMeter中，一个线程所进行的所有操作）囊括了绝大部分操作，而不是一个测试计划只测一个操作。这使得并发测试时，服务器承担了比并发数（50,100,200）更多的并发操作，这也就导致响应时间被大大拖慢；
  2. 有许多页面需要加载大量的数据，例如浏览论文时，需要发送多次请求，需要获得的数据有论文标题，论文关键字，论文相关的笔记，论文本身的PDF，论文PDF中所有block对应的批注；大量数据查询、读取、传输的要求超过了服务器的承受能力，因此造成了响应时间的缺陷。

**建议：**

1. 囊括大多数操作的测试计划更贴近与真实使用场景，是无错的，但应调整各操作的比例，不应该均分，而是应该更贴近真实使用场景。
2. 对于PDF数据及用户主页头像传输的问题，有三个方面的解决途径
   * 1. 压缩图片和PDF（已应用）
     2. 缓存（在用户浏览第n页PDF时，提前缓存第n+1页PDF）（已应用）
     3. 换用性能更好的服务器

大多数缺陷都能用上述分析解释，但在所有的缺陷记录种有很独特的一条——查看用户收藏的论文列表页也出现了响应时间过长的现象，然而这个页面并不需要传输大量数据。为了查看原因，我又进行了一个针对此页面的单独测试。

下面是对此页面单独测试的结果

**表8. 对用户收藏论文页面单独测试的结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试名称 | 平均响应时间/ms | 最小响应时间/ms | 最大响应时间/ms | 并发数 |
| 用户收藏论文页面的增量并发测试 | 27789.08 | 10267 | 34874 | 125 |

由上表可知，在测试中，该页面响应时间过久并不是收到其他请求的影响，而是有自身的问题。于是，我借助IDEA查看了该请求调用的函数以及log记录。

在log记录中，我发现该页面对应的请求产生了大量的查询操作，进一步分析源码后找到了原因。

**原因：**

因为数据库中存放论文信息的table中并没有储存论文的批注量，而是存在了block（PDF组成部分）的表中。因此每一次需要论文的批注数量时，都会调用下面的函数来计算结果

*/\* 计算与一篇论文有关的批注数量 \*/***private** Integer getPostilNo(Paper paper){  
 Integer num = 0;  
 List<PaperPage> l1 = **paperPageRepo**.findByPaper(paper);  
 Iterator<PaperPage> it1 = l1.iterator();  
 **while**(it1.hasNext()){  
 PaperPage pp = it1.next();  
 List<Block> l2 = **blockRepo**.findByPaperPage(pp);  
 num = num + **blockPRepo**.findDistinctPostilByBlock(l2).size();  
 }  
 **return** num;  
  
}

该函数的while循环部分要查询的数据量非常大，因此每一次调用该函数都会花费很长时间。收藏论文的界面需要展示论文的批注数量，因此访问该界面时就会为每一篇论文调用一次该函数，造成了响应速度慢。

**建议：**

优化该函数降低时间消耗，或者将论文的批注总数储存进数据库中，避免每一次需要该数字都有计算一次。