```
</i><code>-1]</code>,
 * leaving elements <code>b[</code><i>k</i><code>]</code> through
 * <code>b[b.length-1]</code> unaffected.
 *
 * <p> The <code>read(b)</code> method for class
<code>InputStream</code>
 * has the same effect as: <pre><code> read(b, 0, b.length)
</code></pre>
 *
 * @param      b   the buffer into which the data is read.
 * @return     the total number of bytes read into the buffer, or
 *             <code>-1</code> if there is no more data because the end
of
 *             the stream has been reached.
 * @exception  IOException  If the first byte cannot be read for any
reason
 * other than the end of the file, if the input stream has been closed, or
 * if some other I/O error occurs.
 * @exception  NullPointerException  if <code>b</code> is
<code>null</code>.
 * @see        java.io.InputStream#read(byte[], int, int)
 */
public int read(byte b[]) throws IOException {
    return read(b, 0, b.length);
}

```

> 如果JDK>=9,可以使用`readAllBytes`方法，更为便捷。内部实现其实也是按照8K进行读取的。

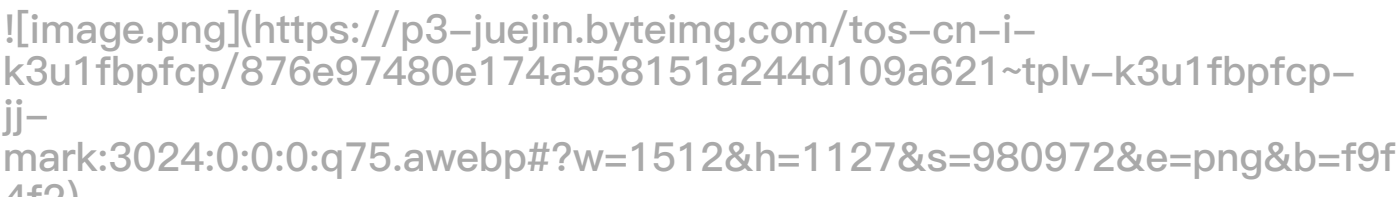
> 文件上传接口通常仅对业务逻辑做处理，文件存储往往会调用专门的存储服务。有2种处理思路：1、接收到完整文件数据，存储至内存中，然后调用存储接口；2、用流的方式，一边read ServletRequest#InputStream，一边write到存储服务的Stream中。个人认为方式2更合理，节约内存。

问题二、tomcat暂存性能瓶颈

接口采用`multipart/form-data`方式上传文件，tomcat接收到请求后会将请求

内容暂存至本地磁盘，目录通常位于tomcat basedir目录下，比如我本地路径为`{basedir}\work\Tomcat\localhost\ROOT`。受限于磁盘写入速率瓶颈，限制了接口性能上限。

> 机械硬盘写入速率预估100MB/s，则在千兆组网场景不存在性能瓶颈，如果是固态硬盘，则写入速率更高。所以此项配置在2G以上组网才需考虑配置。



修改方法为修改sizeThreshold,`默认值为0`。如下所示修改为`1MB`,即内容大于1MB才存入磁盘，小于直接存入内存。

> 关于sizeThreshold，catalina包中处理逻辑为：如果对servlet做了配置，会使用配置的值。如果未配置，默认值为0。util包中DiskFileItemFactory默认值为10k。

```
servlet:
  multipart:
    file-size-threshold: 1MB
```

Tomcat中的相关处理逻辑,`parseRequest`方法按照`RFC 1867`规范对request进行处理。

```
// org.apache.tomcat.util.http.fileupload.disk.DiskFileItemFactory.java

/**
 * <p>The default {@link org.apache.tomcat.util.http.fileupload.FileItemFactory}
 * implementation. This implementation creates
 * {@link org.apache.tomcat.util.http.fileupload.FileItem} instances which keep
 * their
```

 * content either in memory, for smaller items, or in a temporary file on disk,
 * for larger items. The size threshold, above whic ，计算controller总耗时。如果确定是业务逻辑耗时长，再层层递进排查缩小范围，找到罪魁祸首。

测试性能汇总
======

测试环境
----

* 服务器主机、客户机

测试环境所限，服务器主机、客户机使用同一台开发主机。操作系统：windows10,CPU：Intel(R) Xeon(R) Gold 6242R CPU @ 3.10GHz，内存16G
* 磁盘

RND512KQ1T1 Read1219.86Mb/s Write44.88Mb/s
* Jmeter

400线程，60s拉起全部线程
* tomcat

tomcat9，做了如下配置

```
tomcat:
    threads:
        max: 400
    max-connections: 10000
    accept-count: 1000
```
* jar启动参数

配置了初始堆内存
`java -Dfile.encoding=UTF-8 -jar .\xxx.jar -server -Xms4096m -Xmx9000m`

测试结果

类型	平均响应时间 ms	吞吐量/s
原始状态	22081	0.18
优化Byte[]	3966	89
优化file-size-threshold	1203	265
基准-（form-data）	1279	279
基准-（优化file-size-threshold）	109	2930
基准-空接口	28	12401

> **原始状态**：现场报性能问题时的版本，性能太过炸裂，Jmeter线程数调整为4，测试上传文件5KB
>
> **优化Byte[]**：优化了从stream读取存入优化Byte[]方法，测试上传文件5KB。此时网络吞吐量45MB/s，生产环境服务器配置性能至少比当前测试机器高2倍，接口性能至少提高1倍，对于千兆组网场景无须进一步优化，并发瓶颈是网络带宽
>
> **优化file-size-threshold**：优化为>1MB文件才存入磁盘，测试场景文件全部读入内存，测试上传文件5KB。此时网络吞吐量已大于100MB/s
>
> **基准-（form-data）**：form-data配置简单key参数，不上传文件，服务端接口直接返回简单字符串。相当于默认情况下form-data参数类型接口的性能基准，性能瓶颈是磁盘写入速率
>
> **基准-（优化file-size-threshold）**：form-data配置简单key参数，不上传文件，服务端接口直接返回简单字符串，优化为>1MB文件才存入磁盘。可以对比看出磁盘与内存的速率差异
>
> **基准-空接口**：普通的get无参接口，直接返回“hello”，作为当前配置环境下，tomcat接口性能极限

现场问题处理方案
========

经过定位现场性能瓶颈是`网络`。现场采用分布式架构，客户端、服务端部署多个节点，客户端通过本地回环地址调用服务端，降低网络压力。

原架构

```
classDiagram
存储服务 <|-- Server0
Server0 <|-- Server1
Server0 <|-- Server2
Server0 <|-- Server3
存储服务:+获取上传地址()
存储服务:+上传文件()
class Server0{
+服务端
+获取上传地址()
+上传文件()
}
class Server1{
-客户端
}
class Server2{
-客户端
}
class Server3{
-客户端
}
```

新架构

```
classDiagram
存储服务 <|-- Server0
存储服务 <|-- Server1
存储服务 <|-- Server2
存储服务 <|-- Server3
存储服务:+获取上传地址()
存储服务:+上传文件()
class Server0{
+服务端
-客户端
+获取上传地址()
+上传文件()
}
class Server1{
```

+服务端
-客户端
+获取上传地址()
+上传文件()
}
class Server2{
+服务端
-客户端
+获取上传地址()
+上传文件()
}
class Server3{
+服务端
-客户端
+获取上传地址()
+上传文件()
}

```

参考资料
====


tomcat 中是怎么处理文件上传的?

[Apache Tomcat 9 Configuration Reference](http://cxyroad.com/”https://tomcat.apache.org/tomcat-9.0-doc/config/http.html”)

[4-高并发运行环境优化-Tomcat](http://cxyroad.com/”https://zhuanlan.zhihu.com/p/672751242”)
```