

- **tomcat是用来干嘛的，你所知道的和tomcat类似的软件有哪些？**
- tomcat是一种web容器，一个免费开源的Servlet/JSP容器。它是Apache基金会的Jakarta项目中的一个核心项目，由Apache，Sun和其它一些公司及个人共同开发而成。作用有管理serverlet应用的生命周期，把客户端请求的url映射到对应的serverlet，与Servlet程序合作处理HTTP请求等等。它能解析jsp和Java。类似的软件还有jetty、jboss、weblogic、resin等等
- **tomcat的日志文件在哪？tomcat和jdk的关系是什么，tomcat要怎么配置jdk的环境？**
- tomcat的日志文件在tomcat下的logs目录夹下，一种是记录tomcat运行情况的，一种是记录用户访问情况的。tomcat是由Java编写的，运行tomcat就必须要有jre，由于还有编译java源程序的需求，所以需要通过安装jdk来得到jre还有编译Java程序的功能。tomcat可以在bin目录下的catalina.sh和setclasspath.sh文件里指定jdk的环境。
- **你当前架构的并发能达到多少？你tomcat的并发是多少，怎么得出这个结论的，tomcat如何优化？**
- 整个架构并没有测试过，但是单个tomcat服务器使用jmeter测试过，当时jvm配置是5G内存，测试页面是110k大小，以平均响应时间500ms以下为标准，当时测到了850左右的并发。
- 安全优化
- 1首先禁用欢迎界面和管理界面
 1. 删除tomcat的webapps目录下所有文件和目录
 2. 修改默认根目录路径
 3. 删除conf/tomcat-user.xml文件
- 2端口保护
- tomcat的shutdown端口，我们可以改成8000-9000之间不常用的端口号，然后更改shutdown的执行命令，如下操作

```
<Server port="8008" shutdown="HA">
```

关闭tomcat方式

```
[root@localhost conf]# telnet 127.0.0.1 8008
```

```
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
```

```
Escape character is '^['.
```

```
HA
```

```
Connection closed by foreign host.
```

-
- 3降权启动，使用非root用户启动

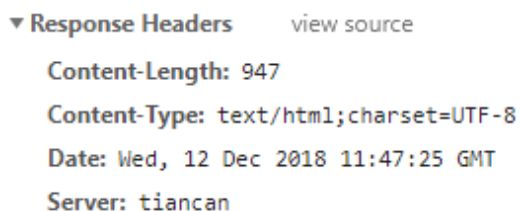
```
groupadd xfq
useradd -g xfq xfq
chown -R xfq:xfq /usr/local/tomcat
```

- 4自定义错误页面

```
<error-page>
  <error-code>404</error-code>
  <location>/404.html</location>
</error-page>
```

- 5header头信息重写(伪装)

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  server="tiancan" />
```



A screenshot of a web browser's developer tools showing the 'Response Headers' section. The headers are: Content-Length: 947, Content-Type: text/html; charset=UTF-8, Date: Wed, 12 Dec 2018 11:47:25 GMT, and Server: tiancan. There is a 'view source' link next to the 'Response Headers' title.

- 6访问限制

server.xml中的<Host>和</Host>间添加如下代码:

限制主机ip

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
  allow="172.24.204.167" deny=""/>
```

限制主机域名:

```
<Valve className="org.apache.catalina.valves.RemoteHostValve"
  allow="www.abc.com" deny=""/>
```

- 7执行文件权限

```
chmod 777 tomcat/bin/*
```

- 8关闭tomcat自动部署

tomcat默认自动开启了对war包的热部署的, 为了防止被植入木马等恶意程序

```
unpackWARs="false" autoDeploy="false">
```

- 9不使用tomcat的虚拟机, 可以使用多实例

因为tomcat是单进程多线程的, 共享内存, 其中只要有一个虚拟主机崩溃, 其它就会有影响

回答方案:欢迎界面, 端口保护, 非root启动(使root跑nginx服务, 一旦网站出现了漏洞, 用户就可以轻松的获得服务器的root权限。), 自定义404页面, 关闭热部署, 多实例。

性能优化

1操作系统调优

1. 文件句柄数

```
ulimit -n 65535 vim /etc/security/limits.conf
```

1. 端口范围

```
echo 1024 65535 > /proc/sys/net/ipv4/ip_local_port_range
```

1. tcp优化

```
echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle(临时生效)
```

```
vim /etc/sysctl.conf
```

```
net.ipv4.tcp_tw_recycle = 1
```

```
sysctl -p
```

2tomcat配置调优

1. 关闭DNS查询

```
enableLookup="false"
```

1. 打开tomcat压缩

```
useSendfile="false"
```

```
compression="on"
```

```
compressionMinSize="2048"
```

```
compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain"  哪些资源类型需要压缩
```

1. tomcat并发优化

maxThreads: tomcat可用于请求处理的最大线程数, 默认是200

minSpareThreads: tomcat初始线程数, 即最小空闲线程数

maxSpareThreads: tomcat最大空闲线程数, 超过的会被关闭

acceptCount: 当所有可以使用的处理请求的线程数都被使用时, 可以放到处理队列中的请求数, 超过这个数的请求将不予处理.默认100

参考配置

```
<Connector port="8080" protocol="HTTP/1.1"
```

```
    connectionTimeout="20000"
```

```
    redirectPort="8443"
```

```
    server="tiancan"
```

```
    maxThreads="800"
```

```
minSpareThreads="100"
maxSpareThreads="200"
acceptCount="1000"
maxConnections="800"
enableLookup="false"
compression="on"
compressionMinSize="2048"
```

```
compressableMimeType="text/html,text/xml,application/javascript,text/css,text/plain"
```

```
/>
```

3jvm调优

```
JAVA_OPTS="-server -Xms5120m -Xmx5120m"
```

tomcat部署web项目的三种方式

- 静态部署，部署解压包到webapps目录
 - 热部署，动态部署，打包的war文件，放到webapps目录下，自动解压
 - manager web应用程序，通过web来管理自己的web项目
-

tomcat工作模式和运行模式

运行模式

1. bio 即阻塞式I/O操作，表示Tomcat使用的是传统的Java I/O操作(即java.io包及其子包)。一个线程处理一个请求，缺点：并发量高时，线程数较多，浪费资源。
1. nio Java nio是一个基于缓冲区、并能提供非阻塞I/O操作的Java API，因此nio也被看成是非阻塞I/O的缩写。它拥有比传统I/O操作(bio)更好的并发运行性能。利用Java的异步请求IO处理，可以通过少量的线程处理大量的请求。
1. apr Tomcat将以JNI的形式调用Apache HTTP服务器的核心动态链接库来处理文件读取或网络传输操作，从而大大地提高Tomcat对静态文件的处理性能。Tomcat apr也是在Tomcat上运行高并发应用的首选模式。

连接器

http连接器 AJP连接器，主要用来和apache配合使用

- 压力测试

首页300毫秒，其它页面500毫秒

ab命令

```
yum install httpd-tools
```

命令	介绍
-n	表示要发送多少次测试请求
-t	表示测试持续多长时间;
-c	(concurrency) 表示用多少并发来进行测试;
-p	发送post请求的时候所请求的参数的文件

- **如果tomcat报错outofmemory, 怎么解决? 如果修改了内存大小还没有解决, 怎么办?**

- 1.OutOfMemoryError: [Java](#) heap space
- 是堆溢出。

在JVM中如果98%的时间是用于GC且可用的 Heap size 不足2%的时候将抛出此异常信息。没有内存泄露的情况下, 调整-Xms -Xmx参数可以解决。

-Xms: 初始堆大小

-Xmx: 最大堆大小

但堆的大小受下面三方面影响:

- 1.相关操作系统的数据模型 (32-bit还是64-bit) 限制; (32位系统下, 一般限制在1.5G~2G; 我在2003 server 系统下 (物理内存: 4G和6G, jdk: 1.6) [测试](#) 1612M, 64为操作系统对内存无限制。)
- 2.系统的可用虚拟内存限制;
- 3.系统的可用物理内存限制。

堆的大小可以使用 java -Xmx***M version 命令来[测试](#)。支持的话会出现jdk的 版本号, 不支持会报错。

-Xms -Xmx一般配置成一样比较好比如set JAVA_OPTS= -Xms1024m -Xmx1024m

- OutOfMemoryError: PermGen space
- 永久保存区域溢出

PermGen space的全称是Permanent Generation space, 是指内存的永久保存区域。这一部分用于存放Class和Meta的信息, Class在被 Load的时候被放入PermGen space区域, 它和存放Instance的Heap区域不同, GC (Garbage Collection) 不会在主程序运行期对PermGen space进行清理, 所以如果你的APP会LOAD很多CLASS的话, 就很可能出现PermGen space错误。

这一个一般是加大-XX: PermSize -XX: MaxPermSize 来解决问题。

-XX: PermSize 永久保存区域初始大小

-XX: PermSize 永久保存区域初始最大值

这一般结合第一条使用, 比如 set JAVA_OPTS= -Xms1024m -Xmx1024m -XX:

PermSize=128M -XX: PermSize=256M

- OutOfMemoryError: unable to create new native thread.

- 无法创建新的线程。这种现象比较少见，也比较奇怪，主要是和jvm与系统内存的比例有关，这种情况应该结合操作系统适当减少分配给jvm的最大内存。

- **你使用的是什么数据库？**

- 数据库分为关系型数据库和非关系型数据库。
- 关系型数据库使用的最多的是mysql数据库。MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。
- 非关系型数据库使用最多的是redis数据库。Redis 是一个开源（BSD许可）的，内存中的数据结构存储系统，它可以用作数据库、缓存和消息中间件。它支持多种类型的数据结构，如 字符串（strings），散列（hashes），列表（lists），集合（sets），有序集合（sorted sets）与范围查询，bitmaps，hyperloglogs 和 地理空间（geospatial）索引半径查询。Redis 内置了 复制（replication），LUA脚本（Lua scripting），LRU驱动事件（LRU eviction），事务（transactions）和不同级别的 磁盘持久化（persistence），并通过 Redis哨兵（Sentinel）和自动 分区（Cluster）提供高可用性（high availability）。

- **数据库主从复制的工作原理是什么？**

- 数据库使用InnoDB引擎，打开二进制日志。从库生成I/O和SQL两个线程，I/O线程去请求主库的bin-log，并将得到的二进制日志文件写入到relay-log（中继日志）中，主库开启一个log dump线程，来为从库的I/O线程传递binlog；从库的SQL线程则来将relaylog中的信息解析成为sql语句然后执行，从而实现主从操作一致，所以主从的数据也会一致。

- **主从架构中，昨天主从同步还好好的，今天就不行了，可能是什么原因，怎么解决？**

- 主库方面可能是因为主库负载过大，sleep过多，show variable like '%wait_timeout%',将超时时间设置更短。从库方面可能是因为出现一些缓慢查询语句，查看show variables like '%slow%'找到缓慢查询设置到1秒，这样缓慢查询日志就会有一些记录。在应用方面可能是因为超量的链接，有些链接没有及时的释放。字符识别问题，可以通过set global sql_slave_skip_counter =1跳过此条错误。最终解决方法还是stop slave ,change master ,start slave。

- **你一般使用什么工具来备份数据库，如何操作？**

- 使用mysqldump工具来进行备份。mysqldump --all-databases --lock-all-tables > backup.sql

- **mysql中有哪些存储引擎？InnoDB和MyISAM的区别有哪些？**

- InnoDB、MyISAM等等。区别：
- 1、表结构的区别（表空间+日志）（MYD+MYI+frm）
- 2、InnoDB支持事务处理，MyISAM不支持
- 3、InnoDB支持到行级锁，MyISAM支持到表级锁
- 4、大量插入更新操作时，InnoDB更优，大量查询操作时，MyISAM更优。
- 5、InnoDB支持外键，MyISAM不支持。
- **数据库有宕机了，如何在客户不知觉的情况下恢复？**
- 我们的架构是一主多从，数据库宕机可能是从库也可能是主库，从库使用了haproxy的四层代理负载均衡，因此当从库宕机的时候对客户的影响几乎是没有什么的，这个时候首先要主库配置防火墙将这个从库踢下线，以免主库每次都询问该从库白白占用资源，防火墙使用reject策略而不是drop，因为reject会反馈给主库拒绝信息而drop不会。接下来将从库数据库重新开启，stop slave然后查看master.info文件，从最新的pos位置开始同步。接下来就是change master to master_host 主机ip地址 master_username master_password连接master的用户名密码 master_log_file 同步的主库的二进制日志文件名master_log_pos查看info文件的来的位置 然后重新start slave这样子基本就能恢复了，当然还有可能从库宕机太久数据不同步的地方过多，这个时候就需要将最新的备份导入到该从库中再开始建立同步。
- 主库宕机的情况比较麻烦，因为用户肯定多少是有影响的，但是我们能让这个影响降到最低，在三分钟左右就能恢复过来。流程就是从库升级为主库。将所有的从库stop slave，然后查看它们的master.info文件选择其中pos最高的那个升级，因为pos越高同步的越完整。确定完新主库后将它的relay-log.info文件删除，修改配置文件，将有关从库的设置注释，比如说readonly等等，然后reset master。将其他从库重设master，将masterhost改为新的主库的ip，对应的logfile和logpos也改为新主库的。最后开启slave，查看slavestatus是否已经更改成功。当然，我们也可以使用mha中间件实现快速升级，这个我没有学习过，不过大致原理还是知道一些，它是将之前的那些操作系统化整理结合然后实现快速切换主从的。