



APUSIC  
固若长城  
睿比世界

# 性能调优

金蝶Apusic应用服务器V10

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

# 目录

- 1 前言
  - 1.1 术语解释
  - 1.2 面向对象
- 2 版本变更说明
- 3 优化概述
- 4 常用配置调优
  - 4.1 堆内存配置
  - 4.2 HTTP线程池配置
  - 4.3 请求数配置
  - 4.4 监控配置
  - 4.5 日志配置
  - 4.6 IO策略
- 5 常用问题定位工具
  - 5.1 应用服务器
  - 5.2 客户端
  - 5.3 top/free/iostat
  - 5.4 jstack
  - 5.5 netstat
  - 5.6 ss
  - 5.7 tcpdump
  - 5.8 JFR
  - 5.9 全链路监控工具
- 6 实际操作示例
  - 6.1 静态页面场景
    - 6.1.1 调优配置
      - 6.1.1.1 堆内存配置
      - 6.1.1.2 HTTP 线程池配置
      - 6.1.1.3 请求数配置
      - 6.1.1.4 监控配置
      - 6.1.1.5 日志配置
      - 6.1.1.6 静态页面压缩
    - 6.1.2 应用部署

- 6.1.3 测试脚本设置
- 6.2 JSP EL表达式场景
  - 6.2.1 调优配置
    - 6.2.1.1 堆内存配置
    - 6.2.1.2 请求数配置
    - 6.2.1.3 监控配置
    - 6.2.1.4 日志配置
    - 6.2.1.5 配置取消JSP动态更新
    - 6.2.1.6 设置IO策略
    - 6.2.1.7 HTTP 线程池配置
    - 6.2.1.8 取消session创建
  - 6.2.2 应用部署
  - 6.2.3 测试脚本设置

# 1 前言

数字化时代，系统性能直接关联用户体验与企业竞争力。业务规模扩张、用户激增下，响应延迟、吞吐量不足等瓶颈频发，不仅降低用户满意度，还可能引发业务中断与经济损失，科学的性能调优成为系统稳定运行的关键。

性能调优需覆盖网络、服务器、应用、数据库等多层面，依赖监控分析、瓶颈定位等专业操作。本文档聚焦性能指标、定位工具、各层级调优策略及案例，为开发、运维、架构师提供实操指引，助力快速解决性能问题。

需注意，性能调优是持续迭代的过程，建议结合自身系统场景灵活运用方法，探索适配业务的优化路径，为业务发展筑牢技术支撑。

## 1.1 术语解释

- **并发用户数**：是指在同一时间窗口内，向系统发起有效请求并处于活跃交互状态的用户数量，是衡量系统并发处理能力的核心指标，并非物理用户总数。
- **点击量**：一次性能测试过程中各个虚拟客户端向后台发起的总的请求数。
- **每秒点击率**：每秒钟用户向web服务器提交的HTTP请求数。点击率越大，对服务器的压力也越大。
- **吞吐量**：指的是在一次性能测试过程中网络上传输的数据量的总和。
- **响应时间**：指的是客户端发出请求到得到响应的整个过程的时间。
- **每秒事务数**：Transactions per Second（每秒通过事务数/TPS）“每秒通过事务数/TPS”显示在场景运行的每一秒钟，每个事务通过、失败以及停止的数量，是考查系统性能的一个重要参数。通过它可以确定系统在任何给定时刻的时间事务负载。分析TPS主要是看曲线的性能走向。将它与平均事务响应时间进行对比，可以分析事务数目对执行时间的影响。

## 1.2 面向对象

本手册主要面向对象为使用金蝶Apsic应用服务器进行应用开发的开发人员，生成环境的系统管理员，应用发布人员，技术运维人员等。具备以下技能可能会更好理解和使用金蝶Apsic应用服务器性能调优说明内容：

- 熟悉Linux常用命令
- 基本的系统管理任务
- 安装和管理软件

## 2 版本变更说明

本手册根据产品实际更新情况同步更新，最新版本将会包括历史版本内容或作出对应的修改说明。

日期	手册版本	适用产品	更新说明
2025年10月	V10P10E05F01	AAS-V10	新建

### 3 优化概述

用户响应时间是指应用系统为用户返回请求结果所消耗的时间。一个典型的用户请求的处理时序图如下，包含了用户和应用系统的网络延迟、应用的处理时间、应用和数据库的交互时的网络延迟和数据库的服务时间等。用户响应时间受到请求链路上各个子系统的影响，比如网络延迟和带宽、系统并发用户数和请求类型、服务器 CPU 和 IO 资源使用率等。要对整个系统进行有效的优化，你需要先定位用户响应时间的瓶颈。

图1: 应用架构图



## 4 常用配置调优

性能瓶颈多集中在 JVM 资源分配、代码效率、线程调度、内存管理四大维度，调优需结合工具定位问题，通过配置优化与代码重构实现性能提升，同时适配操作系统调优后的资源环境。

### 4.1 堆内存配置

堆内存配置是指对计算机系统或应用程序中堆内存的大小、结构等参数进行设置，以优化系统性能和资源利用。

核心参数：

- `-Xms`：用于设置堆的初始大小，即 JVM 启动时分配的堆内存空间大小。
- `-Xmx`：用于设置堆的最大大小，即堆可以扩展的最大内存空间。为避免堆动态扩容导致的性能抖动，通常建议将 `-Xms` 和 `-Xmx` 设置为相同的值。

可以使用 `top` 命令查看内存够用，一般可以设置不超过物理机的一半，建议为物理内存的 `1/4~1/2`。

设置方法：

以金蝶Apusic应用服务器企业版为例，添加JVM选项：`-Xms` 和 `-Xmx`。启动应用服务器。

```
<jvm-options>-Xms5120m</jvm-options>
<jvm-options>-Xmx5120m</jvm-options>
```

管控平台设置：

The screenshot shows the Apusic application server configuration interface. The top navigation bar includes the Apusic logo and the text '金蝶Apusic应用服务器'. On the right side of the navigation bar, there is a user profile 'admin', a power icon, and links for '注销' (Logout) and '帮助中心' (Help Center).

The left sidebar contains a tree view of configuration categories: 集群管理, 会话缓存, 应用管理, 监视管理, 资源管理, 配置管理, default-config, server-config, 管理服务, **JVM 配置** (highlighted), 线程池, HTTP 服务, 网络配置, 日志配置, 监视配置, 虚拟主机, Web 容器, EJB 服务, JMS 服务, ORB 配置, 系统属性, 安全服务, 事务处理服务, 连接器服务.

The main content area is titled 'JVM 选项' (JVM Options) and includes tabs for '一般信息', '路径设置', 'JVM 选项', and '分析器'. Below the tabs, there is a '保存' (Save) button. The text indicates that JVM options are used to manage the server and must be enclosed in double quotes. The configuration name is 'server-config'.

Under '选项 (100)', there is a table with columns for '选择' (Select), '值' (Value), and '说明' (Description). The table contains several rows of JVM options, such as:

选择	值	说明
<input type="checkbox"/>	-Xms5120m	
<input type="checkbox"/>	-server	
<input type="checkbox"/>	[9]-Xbootclasspath/a:\${com.apusic.aas.installRoot}/lib/jaxb-impl-2.3.6.jar	
<input type="checkbox"/>	[9]-Xbootclasspath/a:\${com.apusic.aas.installRoot}/lib/activation-1.1.1.jar	
<input type="checkbox"/>	[9]-Xbootclasspath/a:\${com.apusic.aas.installRoot}/modules/endorsed/jakarta.xml.bind-api.jar	
<input type="checkbox"/>	[9]-add-opens=java.base/sun.net.www=ALL-UNNAMED	
<input type="checkbox"/>	[9]-add-opens=java.base/sun.security.util=ALL-UNNAMED	
<input type="checkbox"/>	[9]-add-opens=java.base/sun.security.provider=ALL-UNNAMED	
<input type="checkbox"/>	[9]-add-exports=java.base/jdk.internal.misc=ALL-UNNAMED	
<input type="checkbox"/>	[9]-add-opens=java.management/javax.management.openmbean=ALL-UNNAMED	
<input type="checkbox"/>	-Dorg.glassfish.gmbal.noMultipleUpperBoundsException=true	
<input type="checkbox"/>	[9]-add-opens=java.management/javax.management.openmbean=ALL-UNNAMED	

## 4.2 HTTP线程池配置

在性能调优中，应用服务器线程池的设置需要结合应用场景、硬件资源（CPU 核心数、内存）及业务特性（并发量、响应时间要求）综合考量。核心目标是减少线程上下文切换、避免资源浪费，同时最大化吞吐量和响应速度。

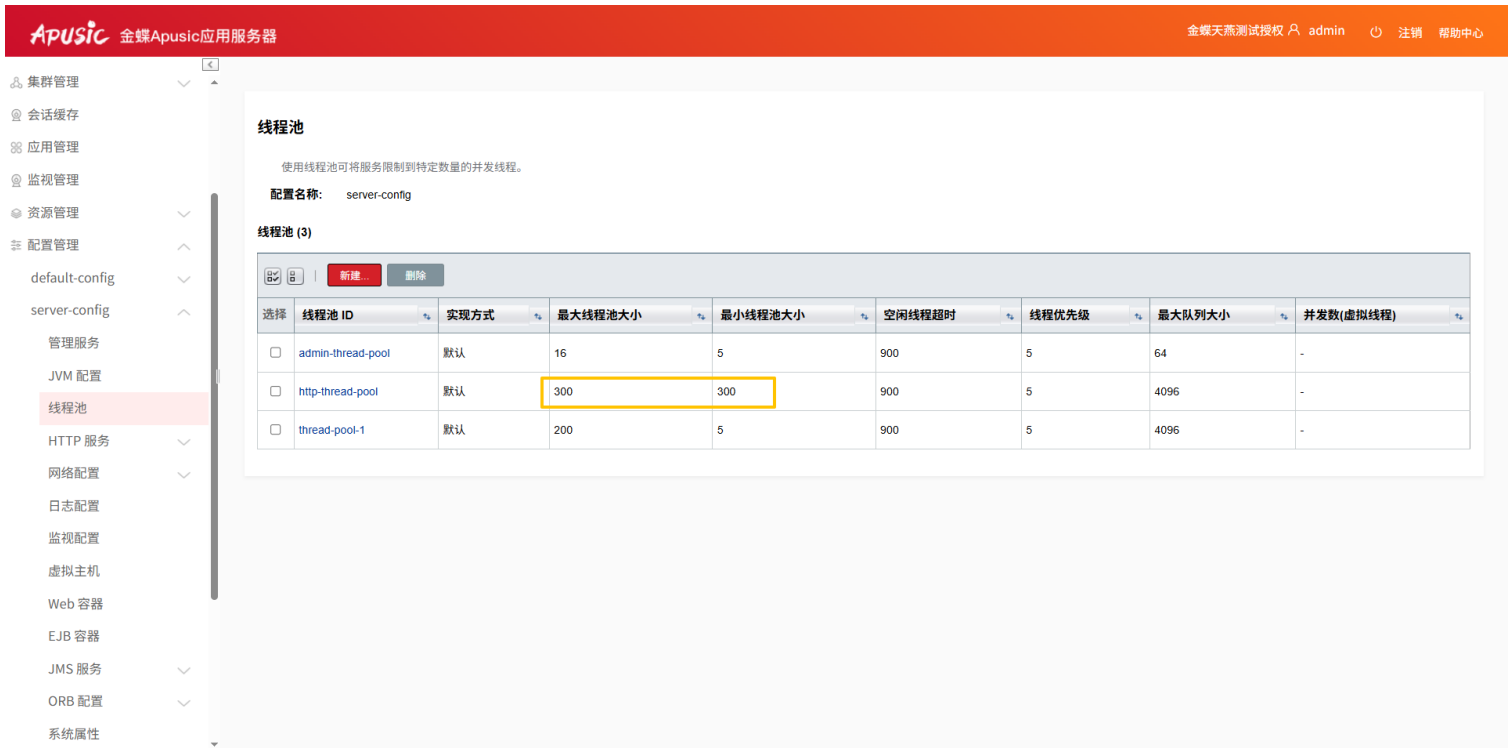
### 1、通用配置

一般情况下，核心线程池大小或最小线程池大小建议设置为 `CPU核数+1`；最大线程池大小建议与核心线程数接近。

### 2、连接外部资源

如果有连接外部资源，例如数据库，设置最大线程池大小和最小线程池大小，一般可以设置跟 `并发数` 一样，后续结合堆栈信息调整到最优大小值。

管控平台设置线程池信息位置：



## 4.3 请求数配置

在 keepalive 模式下，有每连接可以处理多少请求的设置，达到数量后会关闭重新创建连接，AAS会主动断开连接，把该参数设置为比较大，表示不限制，避免重复创建连接，可以提升性能。

设置方法：

以金蝶Apusic应用服务器企业版为例，设置监听协议中的“最大请求数”，`max-connections`。

```
<network-config>
  <protocols>
    <protocol name="http-listener-2" security-enabled="true">
      <http http2-enabled="false" max-connections="1000000">
        <file-cache></file-cache>
      </http>
      <ssl
classname="com.sun.enterprise.security.ssl.ApusicSSLImpl" client-
auth="want" cert-nickname="server"></ssl>
    </protocol>
  </protocols>
```

&lt;/network-config&gt;

管控平台设置位置：



## 4.4 监控配置

在性能测试中，禁用应用服务器的监视服务及相关监控组件可以有效提高性能。

设置方法：

以金蝶Apusic应用服务器企业版为例，系统管理员登录管控平台，进入【配置管理】 - 【server-config】（根据实际选择配置文件） - 【监视配置】，取消勾选“监视服务”、“监视MBean”。

管控平台设置位置：

Apusic 金蝶Apusic应用服务器 金蝶天燕测试授权 admin 注销 帮助中心

配置管理

- default-config
- server-config
  - 管理服务
  - JVM 配置
  - 线程池
  - HTTP 服务
  - 网络配置
  - 日志配置
  - 监视配置**
  - 虚拟主机
  - Web 容器
  - EJB 容器
  - JMS 服务
  - ORB 配置
  - 系统属性
  - 安全服务
  - 事务处理服务
  - 连接器服务
  - 配置中心
- 事务管理

**监视服务**    监控告警    SNMP监听

**监视服务** 保存

选择“低”或“高”启用组件或服务的监视功能。必须同时启用监视服务和监视 MBean 才能使用管理控制台监视功能。

配置名称: server-config

---

**监视服务:**  为 Apusic Server 启用监视

**监视 MBean:**  部署监视所需的所有 MBean

**监控采集周期(秒):**

**监控回放:**

**启用采集器:**

**监控数据总量:**

**监控回放抽样时间(秒):**

**组件级别设置 (16)**

级别:  更改级别

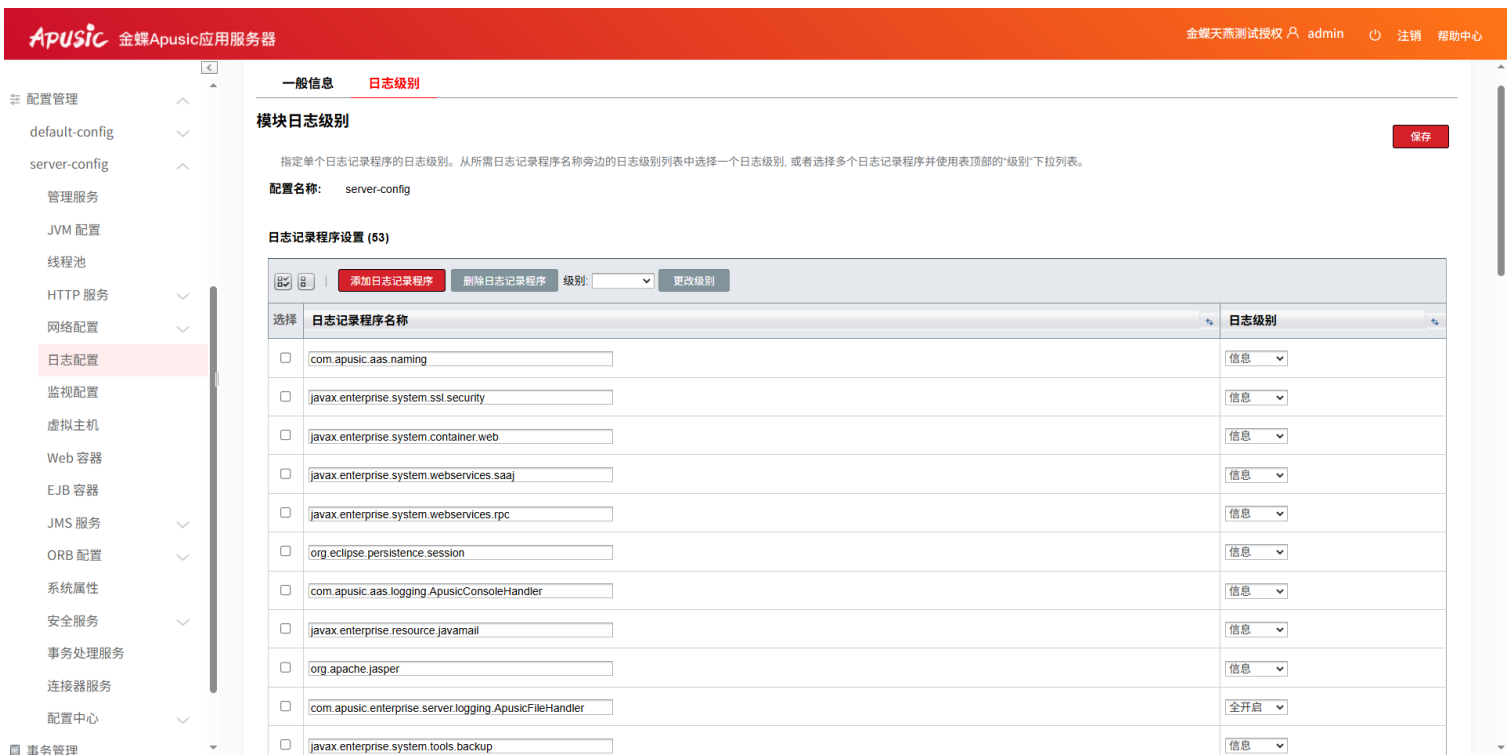
选择	组件	监视级别
<input type="checkbox"/>	Jvm	关闭

## 4.5 日志配置

在性能测试中，可设置日志排除多余的字段输出、刷新频率、文件大小。

如果是没有产生日志的应用场景，日志配置可以保持默认配置或全选设置为错误级别。

管控平台设置位置：



## 4.6 IO策略

通常情况下, 使用默认的 `WorkerThreadIOStrategy`。 `SameThreadIOStrategy` 这个策略只可用于测试目的, 不可用于生产环境, 也不可用于管控。

设置方法:

在Apusic应用服务器企业版中，管控平台和http-listener使用同一个传输，为了不影响管控的使用，需要新建一个传输配置。

- 1、系统管理员登录管控平台，进入【配置管理】 - 【server-config】（根据实际选择配置文件） - 【网络管理】 - 【传输配置】，点击“新建”。属性“IO策略”选择 `org.glassfish.grizzly.strategies.SameThreadIOStrategy`；“接受方程式”设置为 `CPU核数+1`。
- 2、然后新建【监听程序】，设置“传输”为新建的传输配置；或通过修改配置文件 `domain.xml` 将 `http-listener-1` 的“transport”修改为新建的传输配置。
- 3、部署的应用需要使用对应传输配置的监听程序。

管控平台新建传输配置位置：

Apusic 金蝶Apusic应用服务器

金蝶天燕测试授权 admin 注销 帮助中心

配置管理

default-config

server-config

管理服务

JVM 配置

线程池

HTTP 服务

网络配置

监听程序

协议配置

传输配置

日志配置

监视配置

虚拟主机

Web 容器

EJB 容器

JMS 服务

ORB 配置

系统属性

安全服务

事务处理服务

### 传输

单击“新建”以定义新的传输。单击现有传输的名称以修改其设置。

配置名称: server-config

传输 (1)

选择	传输名称	类名称	字节缓冲区类型
<input type="checkbox"/>	tcp	org.glassfish.grizzly.nio.transport.TCPNIOTransport	heap

## 5 常用问题定位工具

系统性能问题定位需覆盖网络、服务器、应用、数据库等全链路，不同层级对应不同工具，以下介绍常用问题定位工具及适用场景。

### 5.1 应用服务器

通常情况下，遇到问题可首选通过应用服务器本身定位问题。例如查看日志文件，查看报错信息，定位问题。

金蝶Apusic应用服务器企业版日志文件默认存储在 `${DOMAIN_HOME}/mydomain/logs` 目录下，默认命名为 `server.log`。

巧用管控平台。系统管理员登录管控平台，通过可视化界面查看日志信息、监控信息等，可生成快照文件、火焰图，分析性能，排查问题。

The screenshot displays the Apusic Application Server Management Console. The top navigation bar includes the Apusic logo and the text '金蝶Apusic应用服务器'. On the right side of the header, there is a search icon, the user 'admin', and links for '注销' (Logout) and '帮助中心' (Help Center).

The left sidebar contains a navigation menu with the following items: '金蝶Apusic应用服务器', '域配置', '服务器 (管控服务器)', '节点管理', '独立实例', '集群管理', '会话缓存', '应用管理', '监视管理', '资源管理', '配置管理', '事务管理', '生命周期', and '升级管理'.

The main content area is titled '一般信息' (General Information) and includes several tabs: '一般信息', '资源', '属性', '监视管理', '批处理', and '证书管理'. Under the '一般信息' tab, there are buttons for '停止', '重新启动', '查看日志文件', '查看原始日志', '滚动日志', '恢复事务处理...', '安全管理...', and '查看访问日志'.

The '一般信息' section displays the following details:

- 名称: server
- 状态: 正在运行 (Running)
- JVM: [JVM 报告](#)
- 配置: [server-config](#)
- 安装目录: /home/apusic/testz/v10/ApusicAS/iaas
- 已安装的版本: Apusic Application Server 10.0.8 EE.SP10.ST. (build 202508211458)
- 安全管理: 已启用
- 调试: 不启用
- 运行时间: 5 天
- License 类型: 临时
- License 授权开始时间: 2024-12-24 00:00
- License 授权过期时间: 2025-12-31 00:00
- License 版本标识: 46
- License 文件: 

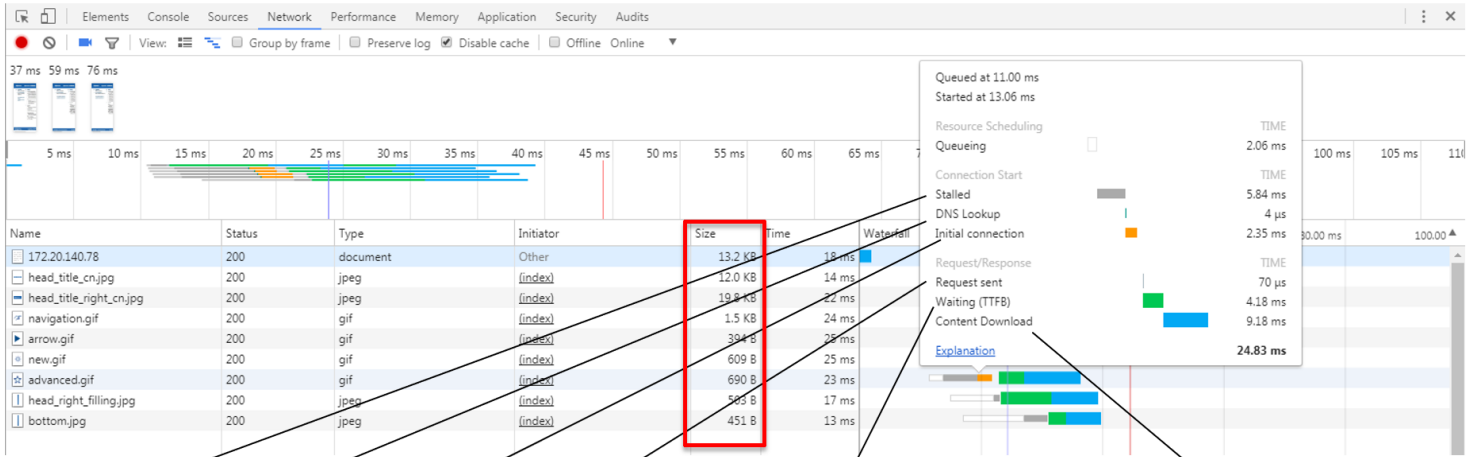
```
<?xml version="1.0" encoding="UTF-8"?>
<license
product="Apusic Application Server"
edition="Enterprise Edition"
version="10.0"
licensee="金蝶天燕测试授权"
connections="10"
datefrom="2024-12-24"

```

At the bottom right of the license file section, there is a button labeled '更新 License 文件' (Update License File).

### 5.2 客户端

应用系统响应慢时，可通过客户端，如浏览器本身进行性能分析；确定有问题的URL，请求数据量大小，以及确定问题所属方。



等待时间    DNS查找时间    TCP握手时间    请求发送时间    请求发送完成到接收到第一个响应字节的时间    接收到第一个响应字节到接收响应完成的时间

### 5.3 top/free/iostat

Linux系统自带工具。

**top** : 实时监控 CPU、内存、进程占用与系统负载。

**free** : 查看内存使用 (含缓存 / 交换区)。

**iostat** : 监控磁盘 IO 读写速率、等待时间 (% util)。

适用场景:

快速定位 CPU 瓶颈、内存泄漏、磁盘 IO 饱和问题。

### 5.4 jstack

Java 线程分析工具, JDK 自带。实时打印 Java 进程的线程栈信息, 包含线程状态 (RUNNABLE/BLOCKED/WAITING)、调用链路、锁持有情况。

常用实操命令:

场景需求	命令示例	关键分析点
查看当前线程栈	jstack 1234 (1234为Java进程ID)	搜索 "BLOCKED" 定位阻塞线程, 查看 "waiting to lock <0xxxxx>" 确认锁竞争对象
检测死锁	jstack -l 1234 > thread_stack.log (-l 显示锁详情)	日志中搜索 "Found one Java-level deadlock", 查看死锁线程的锁依赖链

持续监控线程状态	<pre>while true; do jstack 1234 gre p "RUNNABLE"; sleep 5; done</pre>	<p>该命令通过 while 循环不断执行 jstack 命令，并筛选出状态为 RUNNABLE 的线程，每 5 秒执行一次。通过持续监控，可以观察到线程在运行过程中的动态变化，例如是否存在线程长时间处于 RUNNABLE 状态却未有效执行任务的情况，帮助发现潜在的性能瓶颈或线程调度问题。</p>
----------	---	---

### 适用场景：

- Java 应用响应缓慢（如接口超时），怀疑线程阻塞或死锁；
- 应用 CPU 使用率过高，定位高占用线程的具体方法（结合 top 命令：top -Hp 1234 找到高 CPU 线程 ID，再用 jstack 分析）；
- 生产环境线程异常（如线程数量突增、线程池任务堆积）。

## 5.5 netstat

网络连接状态查看工具，系统自带。查看系统当前的网络连接状态，包含：

- 连接类型（TCP/UDP）、源 / 目的 IP: 端口、连接状态（ESTABLISHED/TIME\_WAIT/CLOSE\_WAIT）；
- 网络接口的流量统计（如接收 / 发送数据包数、错误数）；
- 监听端口（LISTEN 状态）及对应的进程。

### 常用命令：

#### 1. 查看所有活动连接

```
netstat -a # 显示所有TCP/UDP连接（包括监听和非监听状态）
netstat -at # 只显示TCP连接（t: TCP）
netstat -au # 只显示UDP连接（u: UDP）
```

#### 2. 查看监听端口

```
netstat -l # 显示所有监听状态的端口（TCP的LISTEN、UDP的未连接状态）
netstat -lt # 只显示TCP监听端口（如服务端程序监听的端口）
netstat -lu # 只显示UDP监听端口
netstat -ltn # 不解析域名和服务名（n: numeric, 加速输出, 显示IP:端口数字）
```

#### 3. 显示进程与连接的关联

```
netstat -p # 显示每个连接对应的进程ID (PID) 和进程名 (需root权限)
netstat -tlnp # 结合监听端口, 查看哪个进程在监听指定端口 (如排查端口占用)
```

适用场景:

- 应用无法连接外部服务, 排查端口是否可达、连接是否被拒绝;
- 服务器端口耗尽 (如报 "Address already in use"), 定位 TIME\_WAIT/CLOSE\_WAIT 连接堆积原因;
- 怀疑存在异常连接 (如外部 IP 频繁连接敏感端口), 排查连接来源与状态。

## 5.6 ss

ss (Socket Statistics)是 Linux 系统中用于替代 netstat 的高性能网络统计工具, 直接读取内核套接字信息, 在高并发场景下效率远高于 netstat。显示活动的 TCP/UDP 连接、监听端口、套接字统计、网络状态等。

常用命令:

1. 查看所有活动连接

```
ss -a # 显示所有TCP/UDP连接 (包括监听和非监听状态)
ss -t # 只显示TCP连接
ss -u # 只显示UDP连接
ss -tn # 不解析域名和服务名 (n: numeric, 加速输出, 显示IP:端口数字)
```

2. 查看监听端口

```
ss -l # 显示所有监听状态的端口 (TCP的LISTEN、UDP的未连接状态)
ss -lt # 只显示TCP监听端口
ss -lu # 只显示UDP监听端口
ss -ltn # 不解析域名, 快速查看监听的TCP端口
```

3. 显示进程与连接的关联

```
ss -p # 显示每个连接对应的进程ID (PID) 和进程名 (需root权限)
ss -tlnp # 结合监听端口, 查看哪个进程在监听指定端口 (排查端口占用)
```

适用场景:

- 统计不同状态的 TCP 连接数;
- 排查端口耗尽风险 (TIME-WAIT 连接过多);
- 端口占用与服务排查;

- 网络性能与故障排查

## 5.7 tcpdump

网络数据包捕获工具，Linux 系统自带。捕获指定网卡、IP、端口的 TCP/UDP 数据包，支持按协议（如 TCP）、方向（如入站 / 出站）过滤；解析数据包内容，包含源 / 目的 IP、端口、TCP 状态（SYN/ACK/FIN）、数据 payload（可选明文展示）。输出格式支持文本（实时打印）或 pcap 文件（后续用 Wireshark 可视化分析）。

常用实操命令：

场景需求	命令示例	关键分析点
捕获指定端口数据包	<code>tcpdump -i eth0 port 6888 -w packet.pcap</code> (-i 指定网卡, -w 保存为 pcap 文件)	用 Wireshark 打开 pcap 文件, 查看响应耗时
分析 TCP 三次握手 / 四次挥手	<code>tcpdump -i eth0 host 192.168.1.100 and port 6888 -v</code> (-v 显示详细信息)	观察 TCP 包的标志位, 如Flags [S] (SYN, 用于发起连接)、Flags [A] (ACK, 用于确认收到数据), 判断是否存在握手超时 (如发送 SYN 后无 ACK 响应) 或挥手异常, 从而定位连接层的问题
排除网络丢包	<code>tcpdump -i eth0 tcp port 6888 -n -vv -w 6888_traffic.pcap</code>	捕捉 6888 端口的 TCP 重传、窗口异常等特征包, 结合 Wireshark 分析序列号和重传趋势, 可定位丢包是源于网络链路、接收端处理能力不足还是连接异常。关键是对比发送端和接收端的抓包结果, 缩小问题范围

适用场景：

- 应用与数据库 / 第三方服务通信异常 (如连接超时、数据传输中断)；
- 网络延迟高 (如跨机房调用耗时超预期), 定位延迟发生在传输层 (如 TCP 重传频繁)；
- 怀疑网络数据篡改 (如明文传输的密码被拦截), 验证数据包内容完整性。

## 5.8 JFR

Java Flight Recorder, Java 全链路性能记录, JDK 自带。生成 jfr 格式日志, 可通过 JDK 自带的 JMC (Java Mission Control) 可视化分析。

常用实操命令：

场景需求	命令示例	关键分析点
------	------	-------

启动实时记录 (10 分钟)	<pre>jcmd 1234 JFR.start duration=10m filename=app_perf.jfr (jcmd 为 JDK 工具)</pre>	用 JMC 打开 app_perf.jfr, 查看 “Method Profiling” 定位 top 慢方法
配置事件过滤 (仅记录 IO)	<pre>jcmd 1234 JFR.start settings=io.jfc filename=io_perf.jfr (io.jfc 为自定义配置文件)</pre>	JMC 中 “IO” 面板查看文件 / 网络 IO 耗时排行, 定位 IO 瓶颈 (如频繁小文件读写)
故障后分析 (预先开启)	<pre>jcmd 1234 JFR.start delay=1h duration=2h filename=post_crash.jfr (延迟 1 小时启动, 记录 2 小时)</pre>	应用崩溃后, 通过日志回溯崩溃前的 JVM 状态 (如堆内存溢出前的分配趋势)

#### 适用场景:

- 复杂 Java 应用 (如微服务、中间件) 的性能瓶颈定位, 需关联 JVM 与应用层数据;
- 生产环境偶发性性能问题, 需长期低开销记录;
- 应用崩溃 / 内存溢出后的根因分析, 需完整的性能上下文日志。

## 5.9 全链路监控工具

在进行性能问题定位时, 可借助第三方全链路监控工具进行定位。例如金蝶Apusic监控平台软件、Prometheus、Grafana、Skywalking等。

## 6 实际操作示例

### 6.1 静态页面场景

静态页面场景聚焦于服务器处理 HTML、CSS、JS、图片等静态资源的能力，核心测试指标包括并发请求数、每秒处理请求数（TPS）、响应时间（RT）及资源利用率（CPU、内存、网络 IO）。

业务价值在于验证服务器在高并发静态资源访问下的稳定性，优化缓存策略（如 CDN、浏览器缓存）和服务器配置（如连接数、线程池），确保用户快速加载页面，提升访问体验，避免因静态资源阻塞导致的业务流失。

#### 6.1.1 调优配置

##### 6.1.1.1 堆内存配置

根据物理机实际情况调整堆内存，一般可以设置不超过物理机的一半，设置最大启动内存（-Xmx）和最小启动内存（-Xms）一样。例如物理机16G内存，可设置为5120m。

```
<jvm-options>-Xms5120m</jvm-options>
<jvm-options>-Xmx5120m</jvm-options>
```

##### 6.1.1.2 HTTP 线程池配置

调整线程池大小，可以设置为 CPU核数+1，例如8核的物理机设置为9。核心/最小线程池大小和最大线程池大小设置为一致。

##### 6.1.1.3 请求数配置

在 keepalive 模式下，有每连接可以处理多少请求的设置，达到数量后会关闭重新创建连接，AAS会主动断开连接，把该参数设置为比较大，表示不限制，避免重复创建连接，可以提升性能。

##### 6.1.1.4 监控配置

禁用监控功能。

##### 6.1.1.5 日志配置

静态页面性能测试过程中，不会产生日志，不需要关闭日志（若压测过程中，产生错误，不会打印错误的日志信息）。日志配置可以保持默认配置或全选设置为错误级别。

##### 6.1.1.6 静态页面压缩

静态文件压缩缓存可以节省处理时间和网络带宽，提高吞吐量。

应用服务器默认情况下静态文件是打开压缩的，此压缩只进行一次压缩。起始压缩阈值为2k,压缩缓存大小为20M，压缩默认 mimeType 为 " text/html,text/xml,txt/plain "。

注意：当AAS压缩时，使用jmeter进行压测时，jmeter脚本要记得在jmeter线程组中添加“http信息头管理器”，并添加名称：Accept-Encoding值: gzip,deflate。（jmeter不这样配置，依旧不会进行压缩）

## 6.1.2 应用部署

获取静态页面场景的应用程序包，例如 `test1.war`。部署至应用服务器中，确认部署无误，能正常访问操作即可。

## 6.1.3 测试脚本设置

通常情况下，建议性能工具添加两个以上的压力机，具体配置方式参考性能工具官方说明。

使用jmeter进行压测时，jmeter脚本要记得在jmeter线程组中添加“http信息头管理器”，并添加名称：Accept-Encoding值: gzip,deflate。

设置并发数，持续时间。

设置测试HTTP 请求。

根据需要添加监听器。

## 6.2 JSP EL表达式场景

JSP EL 表达式场景聚焦于动态页面解析效率，核心测试 EL 表达式解析耗时、页面渲染性能及并发下的响应稳定性。

通过模拟高并发请求含复杂 EL 表达式的 JSP 页面，监测 TPS、响应时间及服务器 CPU / 内存占用，优化表达式复杂度与 JSP 编译策略，避免动态解析成为瓶颈，保障用户访问动态页面时的流畅体验，提升应用交互效率。

### 6.2.1 调优配置

#### 6.2.1.1 堆内存配置

根据物理机实际情况调整堆内存，一般可以设置不超过物理机的一半，设置最大启动内存（-Xmx）和最小启动内存（-Xms）一样。例如物理机16G内存，可设置为5120m。

```
<jvm-options>-Xms5120m</jvm-options>
<jvm-options>-Xmx5120m</jvm-options>
```

#### 6.2.1.2 请求数配置

在 keepalive 模式下，有每连接可以处理多少请求的设置，达到数量后会关闭重新创建连接，AAS会主动断开连接，把该参数设置为比较大，表示不限制，避免重复创建连接，可以提升性能。

#### 6.2.1.3 监控配置

禁用监控功能。

#### 6.2.1.4 日志配置

JSP EL表达式场景性能测试过程中，不会产生日志，不需要关闭日志（若压测过程中，产生错误，不会打印错误的日志信息）。日志配置可以保持默认配置或全选设置为错误级别。

#### 6.2.1.5 配置取消JSP动态更新

在 `default-web.xml` 的 `JspServlet` 下面配置取消JSP动态更新。。

取消每次访问都判断jsp是否需要动态更新，适用于所有jsp页面的场景，该参数已经默认出厂已经加上。

```
<init-param>
  <param-name>reload-interval</param-name>
  <param-value>-1</param-value>
</init-param>
<init-param>
  <param-name>development</param-name>
  <param-value>>false</param-value>
</init-param>
```

#### 6.2.1.6 设置IO策略

1、系统管理员登录管控平台，进入【配置管理】 - 【server-config】（根据实际选择配置文件） - 【网络管理】 - 【传输配置】，点击“新建”。属性“IO策略”选择 `org.glassfish.grizzly.strategies.SameThreadIOStrategy`；“接受方程式”设置为 `CPU核数+1`。

2、然后新建【监听程序】，设置“传输”为新建的传输配置；或通过修改配置文件 `domain.xml` 将 `http-listener-1` 的“transport”修改为新建的传输配置。

#### 6.2.1.7 HTTP 线程池配置

调整线程池大小，可以设置为 `CPU核数+1`，例如8核的物理机设置为9。核心/最小线程池大小和最大线程池大小设置为一致。

#### 6.2.1.8 取消session创建

性能测试时，查看堆栈信息会发现会一直创建session，可以增加JVM系统参数，不创建session。

注意：该配置需要在应用部署前配置,部署后再配置无效（jsp页面已经编译）。

```
<jvm-options>-Dcom.apusic.jsp.session.default.enable=false</jvm-
options>
```

### 6.2.2 应用部署

获取静态页面场景的应用程序包，例如 `test_el.war`。

该应用需要创建JDBC，JNDI名称输入 `jdbc/derby`。

部署应用至应用服务器中，确认部署无误，访问[http://ip:6888/test\\_el/jdbctest](http://ip:6888/test_el/jdbctest) 能正常访问操作即可。

需注意，如果新建了HTTP监听程序，该应用需要使用新建的HTTP监听程序。

### 6.2.3 测试脚本设置

通常情况下，建议性能工具添加两个以上的压力机，具体配置方式参考性能工具官方说明。

设置并发数，持续时间。

设置测试HTTP 请求。

根据需要添加监听器。

全国统一服务热线  
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

**Apusic**  
金蝶天燕

云计算国家标准制定企业  
金蝶集团旗下基础软件企业  
信息技术应用创新核心企业  
官网: [www.apusic.com](http://www.apusic.com)

