



APUSIC
固若长城
睿比世界

产品运维手册

金蝶Apusic应用服务器V9

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。

本档如有更新，不另行通知。对本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 概述
 - .1.2 适用范围
 - .1.3 参考文档
 - .1.4 术语说明
- .2 安装部署
 - .2.1 运行环境要求
 - .2.1.1 操作系统支持
 - .2.1.2 JDK要求
 - .2.1.3 数据库兼容性
 - .2.1.4 硬件推荐
 - .2.2 安装与卸载
 - .2.2.1 安装前准备
 - .2.2.2 安装
 - .2.2.3 获取许可
 - .2.2.4 启动与停止
 - .2.2.4.1 启动应用服务器
 - .2.2.4.2 停止应用服务器
 - .2.2.5 访问管理控制台
 - .2.2.6 卸载
- .3 基础运维
 - .3.1 应用运维管理
 - .3.1.1 应用部署
 - .3.1.2 应用启停与卸载
 - .3.1.3 应用访问
 - .3.2 配置备份与恢复
 - .3.2.1 定期备份策略
 - .3.2.2 配置恢复操作
 - .3.3 授权许可管理
- .4 配置优化
 - .4.1 线程池配置
 - .4.1.1 通用线程池基础配置
 - .4.1.2 Web Server线程池专项配置
 - .4.1.3 自动调优线程池配置
 - .4.2 Session配置优化
 - .4.2.1 超时时间配置
 - .4.2.2 session储存配置
 - .4.2.3 序列化优化

- 4.3 日志配置优化
 - 4.3.1 日志级别与输出配置
 - 4.3.2 记录客户端IP
- 5 安全加固
 - 5.1 身份认证加固
 - 5.1.1 密码策略配置
 - 5.1.2 禁用匿名访问与权限控制
 - 5.2 传输安全配置
 - 5.2.1 启用SSL/TLS加密
 - 5.2.2 双向认证配置
 - 5.2.3 开启国密配置
 - 5.3 数据安全域防护
 - 5.3.1 敏感数据加密储存
 - 5.3.2 防攻击配置
- 6 性能调优
 - 6.1 JVM参数调优
 - 6.2 应用服务器性能调优
 - 6.2.1 连接优化
 - 6.2.2 缓存优化
 - 6.2.3 数据库访问优化
 - 6.3 集群性能调优
 - 6.3.1 负载均衡策略配置
 - 6.3.2 Session复制优化
- 7 系统监控
 - 7.1 监控概览
 - 7.2 监控配置
 - 7.3 监控回放
 - 7.4 JavaEE组件监控
 - 7.5 数据源监控
 - 7.6 SQL语句监控
 - 7.7 应用会话监控
 - 7.8 HTTP连接监控
 - 7.9 JTA事务监控
 - 7.10 JVM内存监控
 - 7.11 线程监控
 - 7.12 类加载监控
 - 7.13 CPU监控
 - 7.14 系统集成
- 8 使用及运维常见问题

1 前言

1.1 概述

金蝶Apusic应用服务器V9（简称AAS）是一款标准、安全、高效、集成并且具有丰富功能的企业级应用服务器，全面支持JakartaEE8技术规范，提供满足该规范的Web容器、EJB容器以及Web Service容器等。AAS为复杂应用提供了一个简便、快速的开发和运行平台，对于分布式的企业级应用，提供了易扩展、可伸缩和高安全性等特性。

本运维手册旨在为中间件运维人员提供AAS V9的标准化运维指南，涵盖安装部署、配置优化、日常运维、故障排查、安全加固等核心场景，确保 AAS产品服务稳定、高效、安全运行。

1.2 适用范围

本产品运维手册覆盖AAS V9产品的安装、配置、运维、安全、性能、监控和故障排查。适用于AAS V9产品的中间件开发及运维技术人员，同时可作为新人培训及日常运维规范参考资料。

1.3 参考文档

- 《金蝶Apusic应用服务器V9 产品白皮书》
- 《金蝶Apusic应用服务器V9 用户手册》
- 《金蝶Apusic应用服务器V9 常见问题 》

1.4 术语说明

- AAS：Apusic Application Server产品英文简称，中文名称为金蝶Apusic应用服务器。
- AMDC：Apusic In-Memeory Data Cache 产品英文简称，中文名称为金蝶Apusic分布式缓存。
- APUSIC_HOME：Apusic应用服务器的安装目录。
- DOMAIN_HOME：域主目录，`APUSIC_HOME/domains/[domainname]`。
- JAVA_HOME：Java环境变量。

2 安装部署

2.1 运行环境要求

2.1.1 操作系统支持

1. Linux

- 国产操作系统：银河麒麟、统信UOS、华为欧拉、中科方德等。
- Ubuntu Server、RedHat Enterprise Linux (RHEL)、Suse Linux Enterprise Server 等LTS版本。

2. Unix

- HP Unix系列、IBM AIX系列、Oracle Solaris系列。

3. Windows

- Windows 10、Windows 11系列。
- Windows 2008 Server、Windows 2012 Server、Windows Server 2022。

2.1.2 JDK要求

- Open JDK 8+
- Oracle JDK 8+
- IBM JDK 8+

AASv9兼容各主流JDK，常用版本建议在JDK6-11之间，推荐使用JDK8。

2.1.3 数据库兼容性

AAS V9兼容各主流数据库，如MySQL、Oracle、SQLServer、DB2，以及国产数据库如达梦、人大金仓、神舟、高斯等等。

通常情况下，数据库具备远程访问服务，且准备有对应的驱动文件，即可在AAS的【数据源】模块中配置数据源信息。

2.1.4 硬件推荐

推荐配置：CPU 4核以上，内存8GB以上，磁盘50GB以上。以上为运行最低配置，实际请根据业务应用负载调整资源配置情况。

2.2 安装与卸载

2.2.1 安装前准备

需要在服务器上提前安装并设置好Java运行环境。

查看Java版本：

```
[root@AAS-168 bin]# java -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (Apusic JDK V8.0) (build 1.8.0_422-b13)
OpenJDK 64-Bit Server VM (Apusic JDK V8.0) (build 25.422-b13, mixed mode)
```

2.2.2 安装

- Linux下安装

若安装包为 .zip 包，放置指定目录，执行命令 `unzip AAS-V9.0.0-SP10-EE-20251105.zip`，解压安装包完成安装。

获取图形化安装包，放置指定目录，执行授权命令，`chmod 755 AAS-V9.0.bin`；再执行 `./AAS-V9.0.bin`，按照安装指引完成安装。

- Windows下安装

如若安装包为 .zip 包，放置指定目录，直接解压安装包 `AAS-V9.0.0-SP10-EE-20251105.zip` 完成安装。

如若安装包为 .exe，双击执行 `AAS-V9.0.exe`，按照安装指引安装程序。

- 三元分立版本安装

如需要安装三元分立版本，解压 `AAS-V9.0.0-SP10-EE-20251105.zip`，确保 `${APUSIC_HOME}/lib` 目录下有 `admin.war`，即可完成安装。如若原本安装包下只有 `webtool.war`，需要将 `webtool.war` 删除，拷贝 `admin.war` 放置在该目录下。

2.2.3 获取许可

AAS需要有对应的许可证才能正常使用，通常情况下，金蝶天燕会根据用户购买的产品版本配套对应的许可证，如果在使用过程中出现许可证过期或无效等问题，建议优先联系对接的天燕服务人员，重新申请对应许可证。

许可证位置为：`${APUSIC_HOME}/license.xml`。

2.2.4 启动与停止

以下主要介绍生产环境Linux服务器上的启动和停止AAS的操作，windows环境操作基本类似，可以查看用户手册获取相关操作指引。

首次启动应用服务器时需要设置管理员admin的密码；并且首次启动服务器，需要使用前台启动方式，不能使用后台启动（因为后台启动的话不能输入密码，会一直报密码为空的错误）。

2.2.4.1 启动应用服务器

- 使用startapusic脚本启动

进入 `${DOMAIN_HOME}/bin/` 目录，输入命令：

```
startapusic
```

注：终端的退出将导致apusic服务的退出。

- 后台启动

进入 `${DOMAIN_HOME}/bin/` 目录，输入命令：

```
startapusic &
```

- 启动指定域

进入 `${APUSIC_HOME}/bin/` 目录，输入命令：

```
startas mydomain
```

- 系统服务方式

取决于不同的Linux发行版本，一般可以在 `/etc/rc.local` 脚本中添加apusic后台启动命令：

```
java -Xms128m -Xmx512m -XX:MaxPermSize=128m -
Dcom.apusic.domain.home=usr/apusic-/domains/mydomain -
classpath %CP%
com.apusic.server.Main -root usr/apusic-
```

注：假定 `usr/apusic-` 为服务器的安装目录，变量 `%CP%` 的值包含前述启动服务器必需的JAR文件。

2.2.4.2 停止应用服务器

在Linux平台上停止AAS应用服务器方法如下：

- 使用stopapusic脚本

进入 `${DOMAIN_HOME}/bin/` 目录，输入命令：

```
stopapusic
```

Linux终端方式启动下，按Ctrl + C 即可停止控制台前台启动运行的应用服务器。

- 杀进程

通过以下命令查找运行中的进程号：

```
ps -uax|grep java
```

然后用以下命令停止运行中的Apusic应用服务器中间件。

```
kill -9 查到的进程号
```

2.2.5 访问管理控制台

启动应用服务器后，客户端访问<http://localhost:6888/admin> 即可。

管理控制台默认认为localhost可以访问，不能进行远程访问。如果需要远程其他地址访问，在 `${DOMAIN_HOME}/config/vm.options` 文件中写：`com.apusic.webtool.allowHosts=172.2.1.1`，多个地址用英文逗号分隔；支持星号通配符，如 `172.*`

如果是三员分立版，则在 `${DOMAIN_HOME}/config/vm.options` 文件中写：`com.apusic.admin.allowHosts=172.2.1.1`。

添加后启动AAS。

2.2.6 卸载

- Linux下卸载

若安装方式为直接解压的zip包，可进入安装目录，直接删除安装目录即可卸载。若安装方式为图形化安装，可点击双击“卸载AAS”。

注：卸载前需停止应用服务器。

- Windows下卸载

若安装方式为直接解压的zip包，可进入安装目录，直接删除安装目录即可卸载。

若安装方式为图形化安装，可点击开始 - > 程序 - > 金蝶Apusic应用服务器-> 卸载Apusic应用服务器。

注：卸载前需停止应用服务器。

3 基础运维

3.1 应用运维管理

3.1.1 应用部署

- 自动部署

1. 将 WAR/EAR 包拷贝到 `DOMAIN_HOME/applications` 目录。
2. 服务器自动检测文件变化, 5 秒内完成部署, 无需额外操作。

- appctl工具部署

1. 切换到工具目录: `cd APUSIC_HOME/bin`
2. 部署命令: `./appctl install 应用名 应用包路径 -vhost 虚拟主机名 -context 上下文路径`

示例: `./appctl install myapp /opt/apps/myapp.war -vhost localhost -context /myapp`

3. 远程部署: `./appctl -s 192.168.1.200:6888 install myapp /opt/apps/myapp.war -remote`

- 管理控制台部署

1. 访问控制台: `http://服务器IP:6888/admin`
2. 登录后→应用管理→部署应用→选择应用包→设置上下文路径→点击部署

注: 默认情况下管控台只能本地访问, 如果要远程访问需要在 `${DOMAIN_HOME}/config/vm.options` 文件中写:

`com.apusic.webtool.allowHosts=172.2.1.1`, 多个地址用英文逗号分隔。支持星号 通配符, 如 `172.*`; 配置完成启动应用服务器。

3.1.2 应用启停与卸载

1. 启动应用: `appctl start 应用名` 或控制台→应用管理→选择应用→启动
2. 停止应用: `appctl stop 应用名` 或控制台→应用管理→选择应用→停止
3. 重启应用: `appctl restart 应用名`
4. 卸载应用: `appctl uninstall 应用名` (卸载前需先停止应用)
5. 查看应用列表: `appctl list` (远程查看: `appctl -s 服务器IP:6888 list`)

3.1.3 应用访问

1. 直接在客户端输入访问地址, `http://[IP]:[port]/[context]`, 如在浏览器输入 `http://127.0.0.1:6888/myapp`。
2. 访问管控平台, 登录后→应用管理→查看应用模块, 点击“访问应用”。
3. 如果启用SSL, 则在客户端使用HTTPS访问, `https://[IP]:[port]/[context]`, 如在浏览器输入 `https://127.0.0.1:6889/myapp`

3.2 配置备份与恢复

3.2.1 定期备份策略

1. 备份内容: `${DOMAIN_HOME}/config` 目录下所有配置文件 (`apusic.conf`、`datasources.xml`、`logging.xml` 等)
2. 备份频率: 每日自动备份, 重大配置变更后手动备份
3. 备份方式 (bash脚本) :

```
# Linux脚本备份 (可加入定时任务)
BACKUP_DIR=/opt/apusic/backup/$(date +%Y%m%d)
mkdir -p $BACKUP_DIR
cp -r DOMAIN_HOME/config $BACKUP_DIR
```

3.2.2 配置恢复操作

1. 停止服务器: `${DOMAIN_HOME}/bin/stopapusic`
2. 覆盖配置文件: 将备份的 config 目录下文件复制到 `${DOMAIN_HOME}/config`
3. 启动服务器: `${DOMAIN_HOME}/bin/startapusic`
4. 验证: 访问控制台确认配置生效 (如数据源、线程池参数)

3.3 授权许可管理

License许可证位置: `${APUSIC_HOME}/license.xml`。若临时许可到期或临时许可更换为正式许可, 参考如下更新步骤:

1. 联系金蝶天燕服务人员获取新许可证文件。文件格式为 `.xml`, 需要重命名为 `license.xml`
2. 停止服务器: `stopapusic`
3. 替换许可: 将新 `license.xml` 覆盖原文件
4. 启动服务器: `startapusic`
5. 验证: 查看启动日志 (`${DOMAIN_HOME}/logs/apusic.log`), 确认无许可证报错

4 配置优化

4.1 线程池配置

线程池是金蝶Apusic应用服务器并发处理能力的核心，需根据 CPU 核心数、内存大小和业务并发量精准配置，避免线程过多导致上下文切换开销增大，或线程不足导致请求排队拥堵。

4.1.1 通用线程池基础配置

编辑 `${DOMAIN_HOME}/config/apusic.conf` 文件，调整默认公共线程池参数，适配通用场景的并发需求。

```
<SERVICE CLASS="com.apusic.util.ThreadPoolService"
NAME="apusic:service=ThreadPool,name=default">
  <ATTRIBUTE NAME="MinSpareThreads" VALUE="10" /> <!-- 最小备用线程数，确保基础并发响应 -->
  <ATTRIBUTE NAME="MaxSpareThreads" VALUE="50" /> <!-- 最大备用线程数，避免空闲线程占用资源 -->
  <ATTRIBUTE NAME="MaxThreads" VALUE="200" /> <!-- 最大线程数，根据CPU核心数调整（建议CPU核心数的2-4倍） -->
  <ATTRIBUTE NAME="MaxQueueSize" VALUE="1000" /> <!-- 等待队列最大长度，超出则拒绝请求 -->
  <ATTRIBUTE NAME="IdleTimeout" VALUE="60" /> <!-- 空闲线程超时时间（秒），超时自动关闭 -->
</SERVICE>
```

4.1.2 Web Server线程池专项配置

针对 Web 请求的突发性和高并发特性，单独配置 WebServer 线程池，优化 HTTP 请求处理效率。

```
<SERVICE CLASS="com.apusic.util.ThreadPoolService"
NAME="apusic:service=ThreadPool,name=HTTPHandler">
  <ATTRIBUTE NAME="MinSpareThreads" VALUE="15" /> <!-- Web请求最小备用线程，高于通用线程池 -->
  <ATTRIBUTE NAME="MaxSpareThreads" VALUE="80" /> <!-- Web请求最大备用线程 -->
  <ATTRIBUTE NAME="MaxThreads" VALUE="300" /> <!-- Web请求最大线程数，根据日常峰值并发量+20%冗余配置 -->
  <ATTRIBUTE NAME="MaxQueueSize" VALUE="2000" /> <!-- Web请求等待队列，容忍短期流量峰值 -->
  <ATTRIBUTE NAME="IdleTimeout" VALUE="30" /> <!-- Web线程空闲超时，快速释放临时资源 -->
</SERVICE>
```

4.1.3 自动调优线程池配置

启用 `SelfTuneThreadPoolService` 实现动态调优，根据实时并发量自动调整线程数，适配流量波动场景。

```
<SERVICE CLASS="com.apusic.util.SelfTuneThreadPoolService"
NAME="apusic:service=ThreadPool,name=HTTPHandler">
  <ATTRIBUTE NAME="MinSpareThreads" VALUE="5" /> <!-- 调优区间最小值 -->
  <ATTRIBUTE NAME="MaxSpareThreads" VALUE="30" /> <!-- 调优区间中间值 -->
  <ATTRIBUTE NAME="MaxThreads" VALUE="400" /> <!-- 调优区间最大值 -->
  <ATTRIBUTE NAME="MaxQueueSize" VALUE="5000" /> <!-- 队列容量适配高并发 -->
  <ATTRIBUTE NAME="IdleTimeout" VALUE="300" /> <!-- 空闲超时保持稳定 -->
</SERVICE>
```

配置指引：自动调优线程池适合流量波动大的场景（如工作日业务期峰值访问、节假日大促），MinSpareThreads 和 MaxThreads 需覆盖日常并发区间，避免频繁触发极限调优。

4.2 Session配置优化

Session 管理直接影响用户体验和服务器资源占用，需结合存储方式、超时时间和序列化效率综合优化。

4.2.1 超时时间配置

通过 `apusic.conf` 设置全局 Session 超时时间，同时支持应用单独配置覆盖。

```
<SERVICE CLASS="com.apusic.web.session.SessionService">
  <ATTRIBUTE NAME="DefaultSessionTimeout" VALUE="1800" /> <!-- 全局默认超时30分钟，替代默认1小时 -->
  <ATTRIBUTE NAME="SessionInvalidateCheckInterval" VALUE="60" /> <!-- 每60秒检查失效Session -->
</SERVICE>
```

应用单独配置：在应用的 `WEB-INF/web.xml` 中配置专属超时时间。

```
<session-config>
  <session-timeout>20</session-timeout> <!-- 该应用Session超时20分钟，优先级高于全局配置 -->
</session-config>
```

4.2.2 session储存配置

高并发或集群场景下，推荐使用Redis或AMDC存储 Session，避免内存溢出和 Session 丢失。以下为使用Redis为例，编辑 `apusic.conf` 配置。

```
<SERVICE CLASS="com.apusic.web.session.RedisSessionStoreService">
  <ATTRIBUTE NAME="ConfigPath" VALUE="config/redis_session.properties" /> <!-- Redis配置文件路径 -->
  <ATTRIBUTE NAME="ConnectionMode" VALUE="single" /> <!-- 部署模式: single (单机)、shard (分片)、sentinel (哨兵) -->
</SERVICE>
```

创建 `config/redis_session.properties` 配置 Redis 连接。

```
# 单机模式配置
host=192.168.1.100
port=6379
password=redis123
timeout=2000
# 连接池配置
maxTotal=50
maxIdle=10
minIdle=5
testWhileIdle=true
timeBetweenEvictionRunsMillis=30000
```

4.2.3 序列化优化

选择高效的 Session 序列化方式，提升存储和传输效率，编辑 `/${DOMAIN_HOME}/config/vm.options` 添加参数。

```
apusic.http.session.serializer=kryo <!-- 可选: jdk (默认)、fst、kryo, 推荐kryo (速度最
快、字节最小) -->
```

序列化方式对比：

1. kryo: 速度最快，生成字节小，需序列化类有无参构造函数
2. fst: 速度较快，兼容 jdk 序列化，需实现 Serializable 接口
3. jdk: 兼容性最好，速度最慢，生成字节大

4.3 日志配置优化

合理的日志配置既能满足问题排查需求，又能避免日志冗余占用磁盘空间和 IO 资源。

4.3.1 日志级别与输出配置

编辑 `DOMAIN_HOME/config/logging.xml`，调整日志级别和输出规则。

```
<logging>
  <!-- 控制台日志: 仅输出WARNING及以上级别, 避免开发环境冗余 -->
  <handler class="com.apusic.logging.ConsoleHandler">
    <property name="level" value="WARNING" />
    <formatter class="com.apusic.logging.SimpleFormatter" />
  </handler>
  <!-- 文件日志: 输出INFO及以上级别, 按大小轮转 -->
  <handler class="java.util.logging.FileHandler">
    <property name="level" value="INFO" />
    <property name="limit" value="10485760" /> <!-- 单个日志文件最大10MB -->
```

```
<property name="count" value="10" /> <!-- 最多保留10个轮转文件 -->
<property name="pattern" value="%h/logs/apusic.log" /> <!-- 日志存储路径 -->
<property name="append" value="true" /> <!-- 追加写入, 不覆盖历史日志 -->
<formatter class="com.apusic.logging.SimpleFormatter" />
</handler>
<!-- 特定模块日志: 按需调整级别, 如数据源模块DEBUG级便于排查连接问题 -->
<logger name="apusic.jdbc" level="DEBUG" />
<logger name="" level="WARNING" /> <!-- 全局默认日志级别 -->
</logging>
```

4.3.2 记录客户端IP

启动时添加 JVM 参数, 在日志中记录客户端 IP, 便于定位异常请求来源: 编辑 `${DOMAIN_HOME}/config/vm.options`, 添加:

```
-Dapusic.log.clientIP=true
```

5 安全加固

5.1 身份认证加固

5.1.1 密码策略配置

首次启动服务器时，强制设置管理员密码，需符合以下复杂度要求：

1. 长度不小于 8 位
2. 包含大小写字母、数字和特殊字符（如！@#%\$）
3. 配置密码定期更换：
4. 登录管理控制台→系统配置→密码策略
5. 启用“密码有效期”→设置有效期（如 90 天）
6. 启用“密码历史限制”→禁止重复使用前 5 次密码

5.1.2 禁用匿名访问与权限控制

禁用匿名访问敏感资源：编辑 `apusic.conf`，在连接工厂、消息目的地等配置中添加。

```
<ATTRIBUTE NAME="anonymous" VALUE="false" /> <!-- 禁止匿名用户访问 -->
```

多因素认证集成：结合 JAAS 框架集成 LDAP 认证，编辑 `security.xml` 配置。

```
<realm>
  <realm-name>LDAPRealm</realm-name>
  <provider-type>ldap</provider-type>
  <attribute name="host" value="192.168.1.150" /> <!-- LDAP服务器IP -->
  <attribute name="port" value="389" /> <!-- LDAP端口 -->
  <attribute name="rootDN" value="cn=admin,dc=apusic,dc=com" /> <!-- LDAP管理员DN -->
  <attribute name="password" value="ldap123" /> <!-- LDAP管理员密码 -->
</realm>
```

5.2 传输安全配置

5.2.1 启用SSL/TLS加密

编辑 `apusic.conf` 中的 Muxer 服务配置，启用 HTTPS 加密传输。

```
<SERVICE CLASS="com.apusic.net.Muxer">
  <ATTRIBUTE NAME="Port" VALUE="6888" /> <!-- HTTP端口 -->
  <ATTRIBUTE NAME="SSLEnabled" VALUE="True" /> <!-- 启用SSL -->
  <ATTRIBUTE NAME="SecurePort" VALUE="6889" /> <!-- HTTPS端口 -->
```

```
<ATTRIBUTE NAME="KeyStore" VALUE="config/sslserver.jks" /> <!-- 密钥库路径 -->
<ATTRIBUTE NAME="KeyPassword" VALUE="changeit" /> <!-- 密钥库密码 -->
<ATTRIBUTE NAME="SSLProtocol" VALUE="TLSv1.2" /> <!-- 禁用弱协议, 仅启用TLS1.2 -->
</SERVICE>
```

密钥库生成: 使用 JDK 的 keytool 工具生成自签名证书, 如下。

```
keytool -genkey -alias apusic -keyalg RSA -keysize 2048 -keystore
config/sslserver.jks -validity 3650
```

5.2.2 双向认证配置

1. 生成客户端证书并导入服务器信任库

```
keytool -import -alias clientca -file client.cer -keystore config/truststore.jks -
storepass changeit
```

2. 配置服务器要求客户端认证

```
<SERVICE CLASS="com.apusic.net.Muxer">
  <ATTRIBUTE NAME="NeedClientAuth" VALUE="True" /> <!-- 强制客户端认证 -->
  <ATTRIBUTE NAME="TrustStore" VALUE="config/truststore.jks" /> <!-- 信任库路径 -->
  <ATTRIBUTE NAME="TrustStorePassword" VALUE="changeit" /> <!-- 信任库密码 -->
</SERVICE>
```

5.2.3 开启国密配置

AAS支持配置国密证书, 开启国密需在 `apusic.conf` 配置文件中的Muxer服务中添加SSLProtocol属性, 设置值为: "SMv1.1" (旧版本为SM1.1), 并分别配置密钥库和信任库。

1. 配置证书信息

```
<SERVICE CLASS="com.apusic.net.Muxer">
  <ATTRIBUTE NAME="Port" VALUE="6888"/>
  <ATTRIBUTE NAME="Backlog" VALUE="1024"/>
  <ATTRIBUTE NAME="Timeout" VALUE="30"/>
  <ATTRIBUTE NAME="LookAheadTimeout" VALUE="30"/>
  <ATTRIBUTE NAME="MaxWaitingClients" VALUE="200"/>
  <ATTRIBUTE NAME="WaitingClientTimeout" VALUE="5"/>
  <ATTRIBUTE NAME="SecurePort" VALUE="6889"/>
  <ATTRIBUTE NAME="MutualAuthPort" VALUE="6887"/>
  <ATTRIBUTE NAME="NeedClientAuth" VALUE="false"/>
  <ATTRIBUTE NAME="SSLEnabled" VALUE="true"/>
  <ATTRIBUTE NAME="SSLProtocol" VALUE="SMv1.1"/>
```

```
<ATTRIBUTE NAME="KeyStore" VALUE="config/keystore.p12"/>
<ATTRIBUTE NAME="KeyStoreType" VALUE="PKCS12"/>
<ATTRIBUTE NAME="KeyPassword" VALUE="123456"/>
    <ATTRIBUTE NAME="TrustStore" VALUE="config/trust.p12"/>
<ATTRIBUTE NAME="TrustStoreType" VALUE="PKCS12"/>
<ATTRIBUTE NAME="TrustStorePassword" VALUE="123456"/>
</SERVICE>
```

2. 使用 `./startapusic.sh gm` 以国密模式启动AAS应用服务器

5.3 数据安全域防护

5.3.1 敏感数据加密储存

数据库密码加密：使用服务器内置加密工具加密数据库密码，编辑 `datasources.xml`。

```
<datasource name="mysqlDB" jndi-name="jdbc/mysql" driver-
class="com.mysql.jdbc.Driver" url="jdbc:mysql://192.168.1.10:3306/mydb">
    <property name="user" value="root" />
    <property name="password" value="${ENCRYPT:encrypted_password}" /> <!-- 加密后的
密码 -->
</datasource>
```

加密工具使用：`${APUSIC_HOME}/bin/encrypt.sh` 明文密码，生成加密后的字符串。

5.3.2 防攻击配置

1. 关闭不必要的服务和端口：禁用未使用的协议（如 RMI、IIOP），仅保留 HTTP/HTTPS 端口
2. 过滤弱安全套件：编辑 `vm.options`，添加

```
-
Dcom.apusic.ciphersuite.exclude=SSL_RSA_WITH_DES_CBC_SHA,SSL_DHE_RSA_WITH_DES_CBC_SHA
# 过滤弱加密套件
```

3. 隐藏服务器版本信息：编辑 `vm.options`，添加

```
-Dcom.apusic.errorPage.hideServerInfo=true # 错误页面不显示服务器版本
```

6 性能调优

6.1 JVM参数调优

JVM 参数直接影响服务器内存使用和垃圾回收效率，需根据物理内存大小合理配置，以下为通用优化配置：

编辑 `${DOMAIN_HOME}/bin/startapusic` (Linux) 或 `startapusic.cmd` (Windows) ，修改 JVM 参数。

```
# Linux示例（物理内存16GB服务器）
JAVA_OPTS="-Xms8192m -Xmx8192m -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=512m -
XX:+UseG1GC -XX:MaxGCPauseMillis=200 -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=../logs/heapdump.hprof"
```

- 堆内存配置

-Xms（初始堆内存）和 -Xmx（最大堆内存）设置为物理内存的 1/2~2/3，避免频繁扩容缩容。

示例：16GB 内存服务器设置为 8GB，32GB 内存设置为 16GB

- 元空间配置

JDK1.8 及以上使用 -XX:MetaspaceSize 和 -XX:MaxMetaspaceSize，避免类加载过多导致内存溢出。

推荐配置：256m~512m，根据应用依赖 jar 包数量调整。

- 垃圾回收优化

使用 G1 垃圾回收器（ -XX:+UseG1GC ），适合大堆内存场景。

设置最大 GC 停顿时间（ -XX:MaxGCPauseMillis=200 ），确保业务响应稳定。

启用 OOM 时堆转储（ -XX:+HeapDumpOnOutOfMemoryError ），便于排查内存泄漏。

6.2 应用服务器性能调优

6.2.1 连接优化

- HTTP连接复用

编辑 `apusic.conf` 中的 `WebService` 配置，优化长连接。

```
<SERVICE CLASS="com.apusic.servlet.http.WebService">
  <ATTRIBUTE NAME="KeepAlive" VALUE="true" /> <!-- 启用长连接 -->
  <ATTRIBUTE NAME="KeepAliveTimeout" VALUE="30" /> <!-- 长连接超时时间30秒 -->
  <ATTRIBUTE NAME="MaxKeepAliveConnections" VALUE="500" /> <!-- 最大长连接数 -->
</SERVICE>
```

- GZip 压缩启用

压缩文本类响应内容，减少网络传输量。

```
<SERVICE CLASS="com.apusic.servlet.http.WebService">
  <ATTRIBUTE NAME="EnableGZip" VALUE="true" /> <!-- 启用GZip压缩 -->
  <ATTRIBUTE NAME="CompressableMimeTypes"
VALUE="text/html,text/xml,text/css,application/json" /> <!-- 压缩类型 -->
<ATTRIBUTE NAME="GZipMinSize" VALUE="2048" /> <!-- 内容大于2KB才压缩 -->
</SERVICE>
```

6.2.2 缓存优化

- 静态资源缓存

编辑 `apusic.conf`，调整静态文件缓存大小。

```
<SERVICE CLASS="com.apusic.servlet.http.WebService">
<ATTRIBUTE NAME="FileCacheSize" VALUE="4096" /> <!-- 静态资源缓存4GB, 默认1GB -->
... (省略其他配置)
</SERVICE>
```

- JDBC结果集缓存

编辑 `datasources.xml`，启用结果集缓存，减少数据库查询次数。

```
<datasource name="mysqlDB" ...>
  <property name="pre-fetch-size" value="200" /> <!-- 结果集缓存大小 -->
  <property name="stmt-cache-size" value="100" /> <!-- 预编译SQL缓存数量 -->
</datasource>
```

6.2.3 数据库访问优化

- 连接池参数优化

编辑 `datasources.xml`，适配数据库并发连接能力。

```
<datasource name="mysqlDB" jndi-name="jdbc/mysql" driver-
class="com.mysql.jdbc.Driver" url="jdbc:mysql://192.168.1.10:3306/mydb"
  min-spare-connections="10" <!-- 最小空闲连接数 -->
  max-spare-connections="50" <!-- 最大空闲连接数 -->
  max-connections="200" <!-- 最大连接数, 不超过数据库最大连接数(默认151) -->
  idle-timeout="600" <!-- 空闲连接超时10分钟, 自动关闭 -->
  test-before-reused="true" <!-- 借用连接前检测可用性 -->
  test-command="select 1" <!-- 检测SQL语句 -->
>
  <property name="user" value="root" />
  <property name="password" value="mysql123" />
</datasource>
```

6.3 集群性能调优

6.3.1 负载均衡策略配置

编辑 `loadbalancer.conf`，选择合适的负载均衡算法。

```
<SERVICE CLASS="com.apusic.web.loadbalancer.LoadBalancer">
  <ATTRIBUTE NAME="BackendServers" VALUE="192.168.1.201:6888,192.168.1.202:6888"
/> <!-- 集群节点 -->
  <ATTRIBUTE NAME="BalancePolicy" VALUE="Weight" /> <!-- 负载均衡算法: Random (随
机)、Round-Robin (轮询)、Weight (权重) -->
  <ATTRIBUTE NAME="LoadWeight" VALUE="3,2" /> <!-- 节点权重, 与BackendServers顺序对应
-->
</SERVICE>
```

6.3.2 Session复制优化

集群场景使用配对复制策略，减少网络风暴：编辑 `apusic.conf`，配置 `SessionService`。

```
<SERVICE CLASS="com.apusic.web.session.SessionService">
  <ATTRIBUTE NAME="Replicable" VALUE="True" /> <!-- 启用Session复制 -->
  <ATTRIBUTE NAME="Anycasting" VALUE="False" /> <!-- 禁用广播复制, 使用配对复制 -->
</SERVICE>
```

高并发场景推荐使用 Redis 分布式 Session，替代复制策略，避免复制开销。

7 系统监控

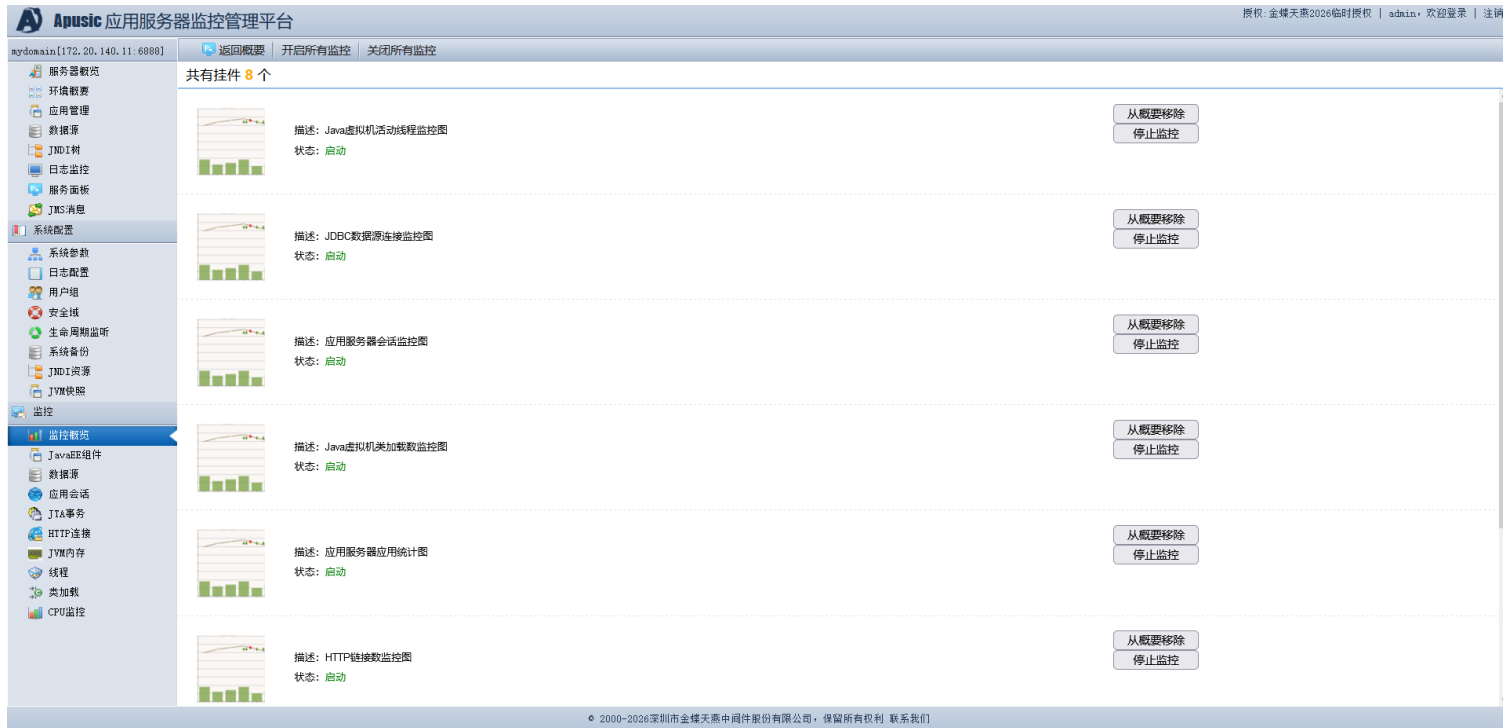
7.1 监控概览

打开Apusic应用服务器监控管理平台，进入“监控”\“监控概览”操作界面下，可以查看活动线程数、应用会话统计等信息。



7.2 监控配置

打开Apusic应用服务器监控管理平台，进入“监控”\“监控概览”操作界面，点击“监控配置”，可设置关停监控组件。



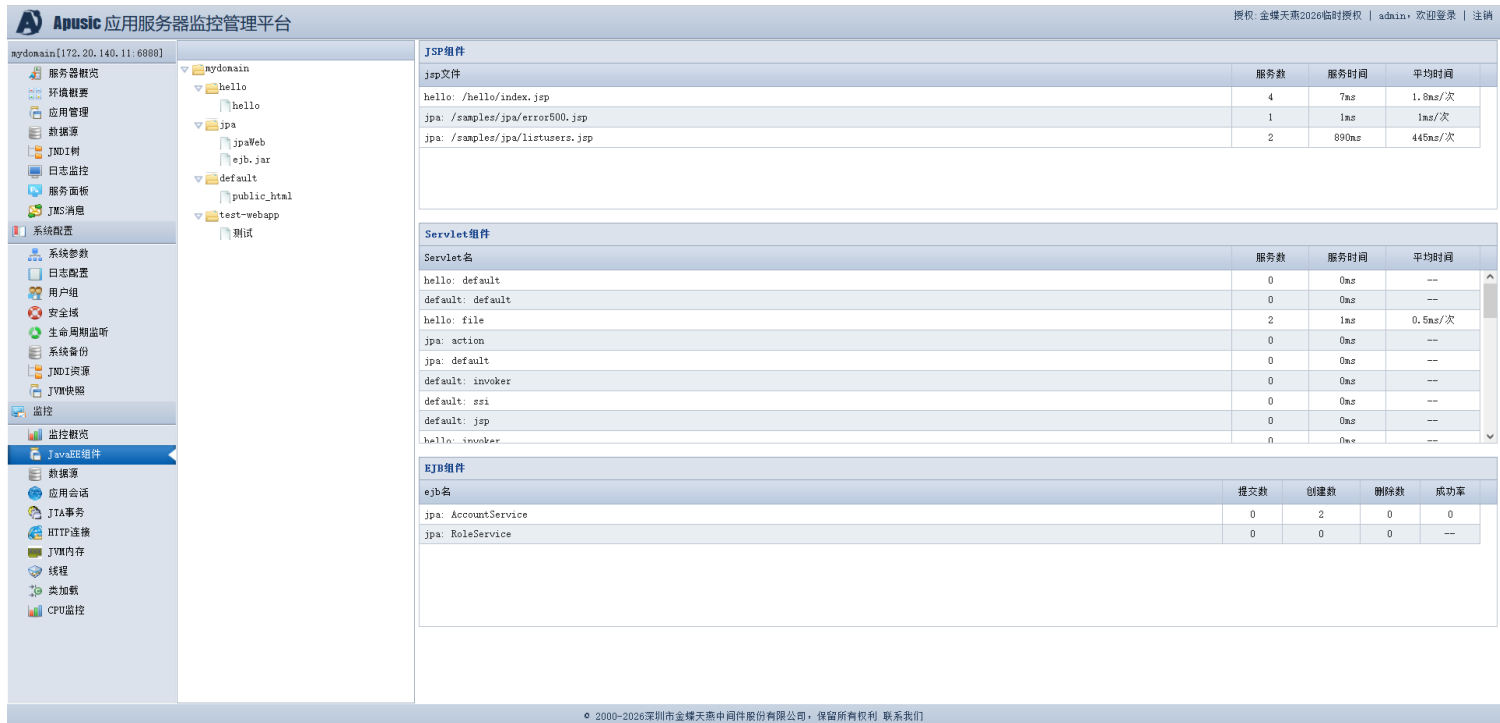
7.3 监控回放

打开Apusic应用服务器监控管理平台，进入“监控”\“监控概览”操作界面，点击“监控回放”，可查看监控项回放信息。



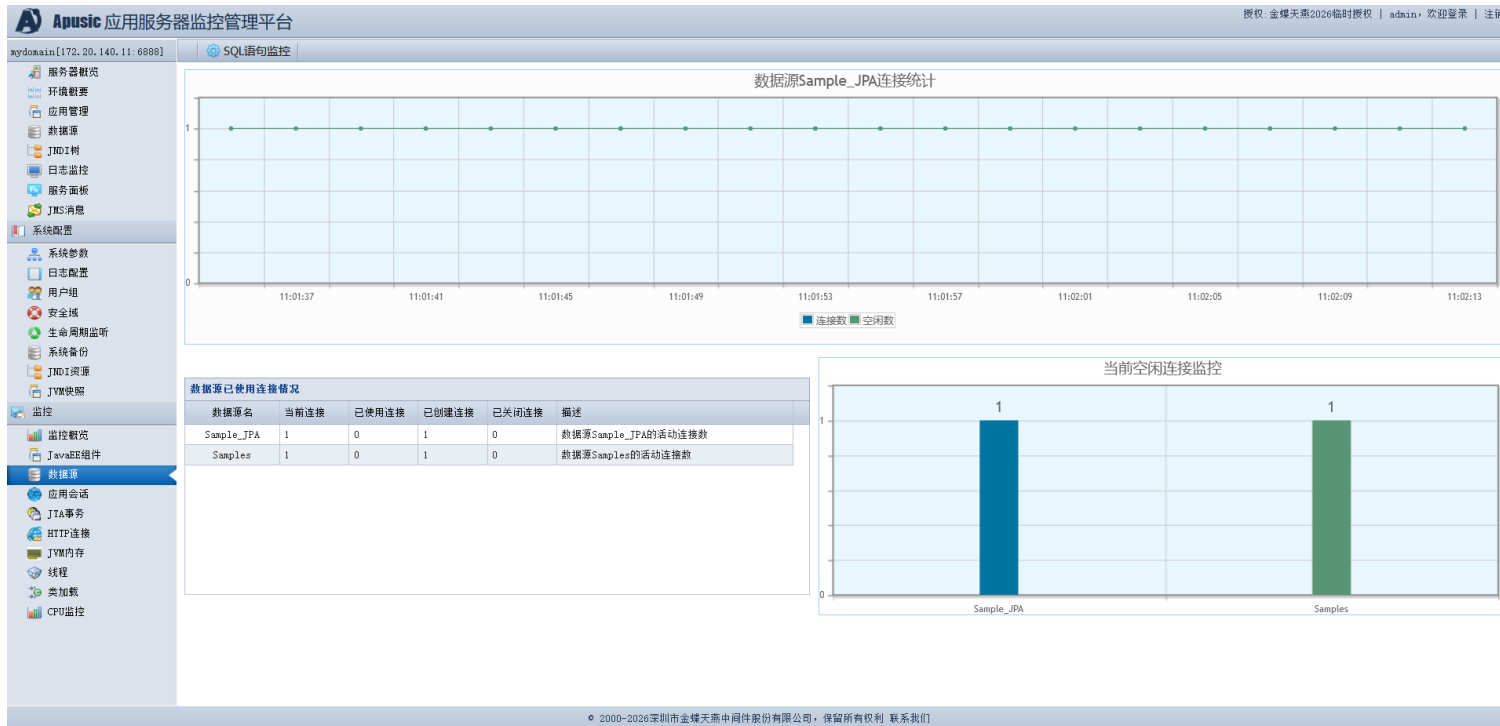
7.4 JavaEE组件监控

打开Apusic应用服务器监控管理平台，进入“监控”\“javaEE组件”操作界面下。在此可以查看每一个应用下javaEE组件的服务数，服务时间，平均时间等。



7.5 数据源监控

打开Apusic应用服务器监控管理平台，进入“监控”>“数据源”操作界面下。在此可以查看数据源总的连接数和空闲数以及每个数据源的连接数和空闲数和活动连接数。



7.6 SQL语句监控

打开Apusic应用服务器监控管理平台，进入“监控”>“数据源”操作界面，点击“SQL语句监控”。点击“启用SQL监控”按钮，开启SQL语句监控；点击“启用SQL堆栈监控”，开启SQL堆栈监控，输出堆栈信息。

Apusic 应用服务器监控管理平台 授权: 金蝶天燕2026临时授权 | admin, 欢迎登录 | 注销

mydomain[172.20.140.11:6888]

返回数据源监控 | 刷新 | 禁用SQL监控 | 启用SQL堆栈监控

数据源名称: 事件类型: ALL

事件类型	数据源名称	SQL语句	执行次数	执行时长(毫秒)	执行时间	执行堆栈
1	Connection.close	Sample_JPA	1	0	2026-02-26 11:03:37	
2	Statement.Execut	Sample_JPA	2	2	2026-02-26 11:03:37	
3	Connection.close	Sample_JPA	1	0	2026-02-26 11:03:37	
4	Statement.Execut	Sample_JPA	3	1	2026-02-26 11:03:37	
5	Statement.Execut	Sample_JPA	3	1	2026-02-26 11:03:37	
6	Connection.close	Sample_JPA	1	0	2026-02-26 11:03:24	
7	Connection.close	Sample_JPA	1	0	2026-02-26 11:03:02	com.apusic.jdbc.trace.JDBCTracer.trace(JDBCTracer.java:101) com.apusic.jdbc.trace.JDBCTracer.trace(JDBCTracer.java:98) com.apusic.jdbc.adapter.ConnectionHandle.close(ConnectionHandle.java:183) org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.closeDataSourceCon org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.closeConnection(O org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.closeConnection(O org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.decrementCallCou org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.closeStatement(Dat org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.releaseStatement(D org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.basicExecuteCall(D org.eclipse.persistence.internal.databaseaccess.DatabaseAccessor.executeCall(Databa org.eclipse.persistence.internal.sessions.AbstractSession.basicExecuteCall(Abstract org.eclipse.persistence.sessions.server.ServerSession.executeCall(ServerSession.jav org.eclipse.persistence.internal.queries.DataSourceCallQueryMechanism.executeCall(D org.eclipse.persistence.internal.queries.DataSourceCallQueryMechanism.executeCall(D org.eclipse.persistence.internal.queries.DataSourceCallQueryMechanism.executeSelect org.eclipse.persistence.internal.queries.DataSourceCallQueryMechanism.selectAllRows org.eclipse.persistence.internal.queries.ExpressionQueryMechanism.selectAllRows(Exp org.eclipse.persistence.queries.ReadAllQuery.executeObjectLevelReadQuery(ReadAllQue org.eclipse.persistence.queries.ObjectLevelReadQuery.executeDatabaseQuery(ObjectLev org.eclipse.persistence.queries.DatabaseQuery.execute(DatabaseQuery.java:852) org.eclipse.persistence.queries.ObjectLevelReadQuery.execute(ObjectLevelReadQuery.j org.eclipse.persistence.queries.ReadAllQuery.execute(ReadAllQuery.java:393) org.eclipse.persistence.queries.ObjectLevelReadQuery.executeUnitOfWorkWork(ObjectLev org.eclipse.persistence.internal.sessions.UnitOfWorkImpl.internalExecuteQuery(UnitO org.eclipse.persistence.internal.sessions.AbstractSession.executeQuery(AbstractSess org.eclipse.persistence.internal.sessions.AbstractSession.executeQuery(AbstractSess

共12条数据, 显示1-10条

© 2000-2026深圳市金蝶天燕中间件股份有限公司, 保留所有权利 联系我们

7.7 应用会话监控

打开Apusic应用服务器监控管理平台, 进入“监控”)“应用会话”操作界面下。在此可以从图表和比例图以及文字描述中监控到每个应用的话连接数的变化。

Apusic 应用服务器监控管理平台 授权: 金蝶天燕2026临时授权 | admin, 欢迎登录 | 注销

mydomain[172.20.140.11:6888]

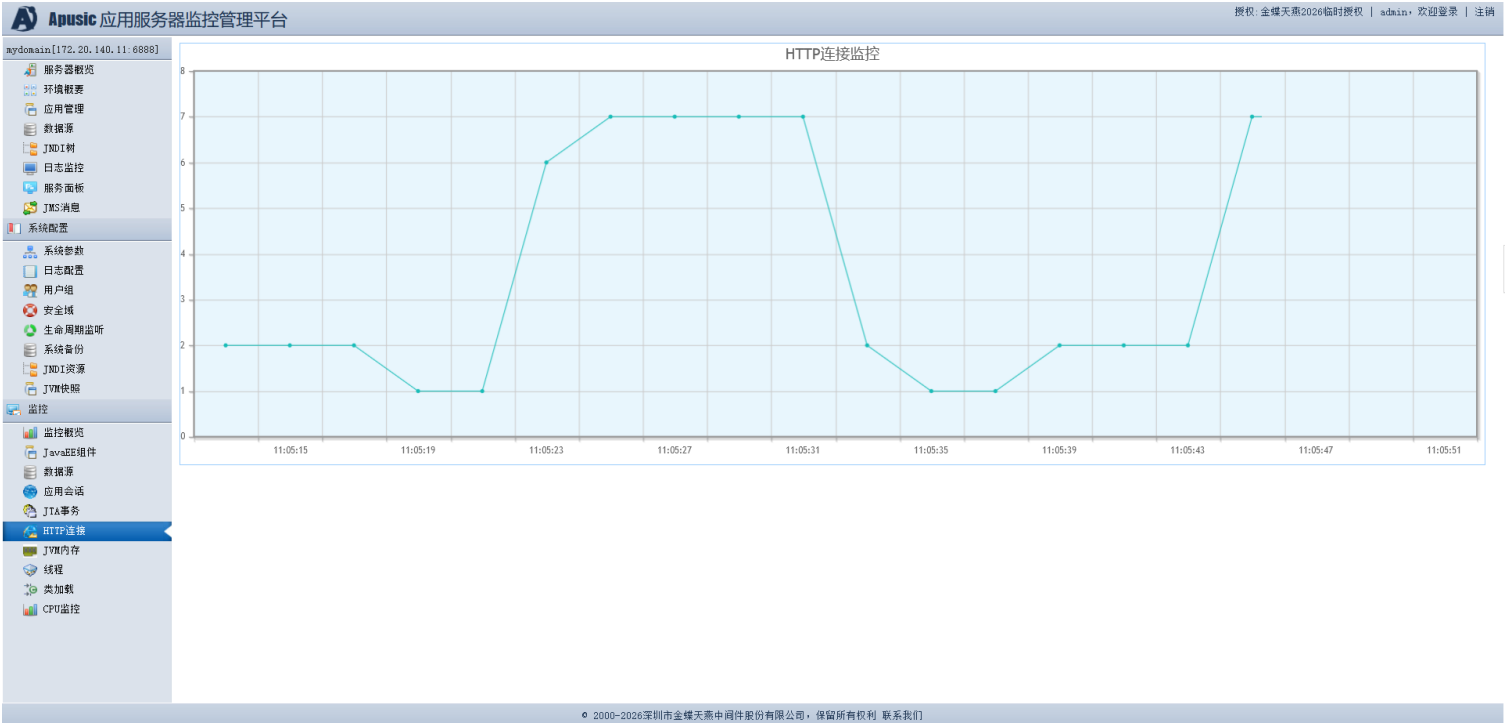
应用会话统计

Legend: default, jpa, hello, test-webapp

© 2000-2026深圳市金蝶天燕中间件股份有限公司, 保留所有权利 联系我们

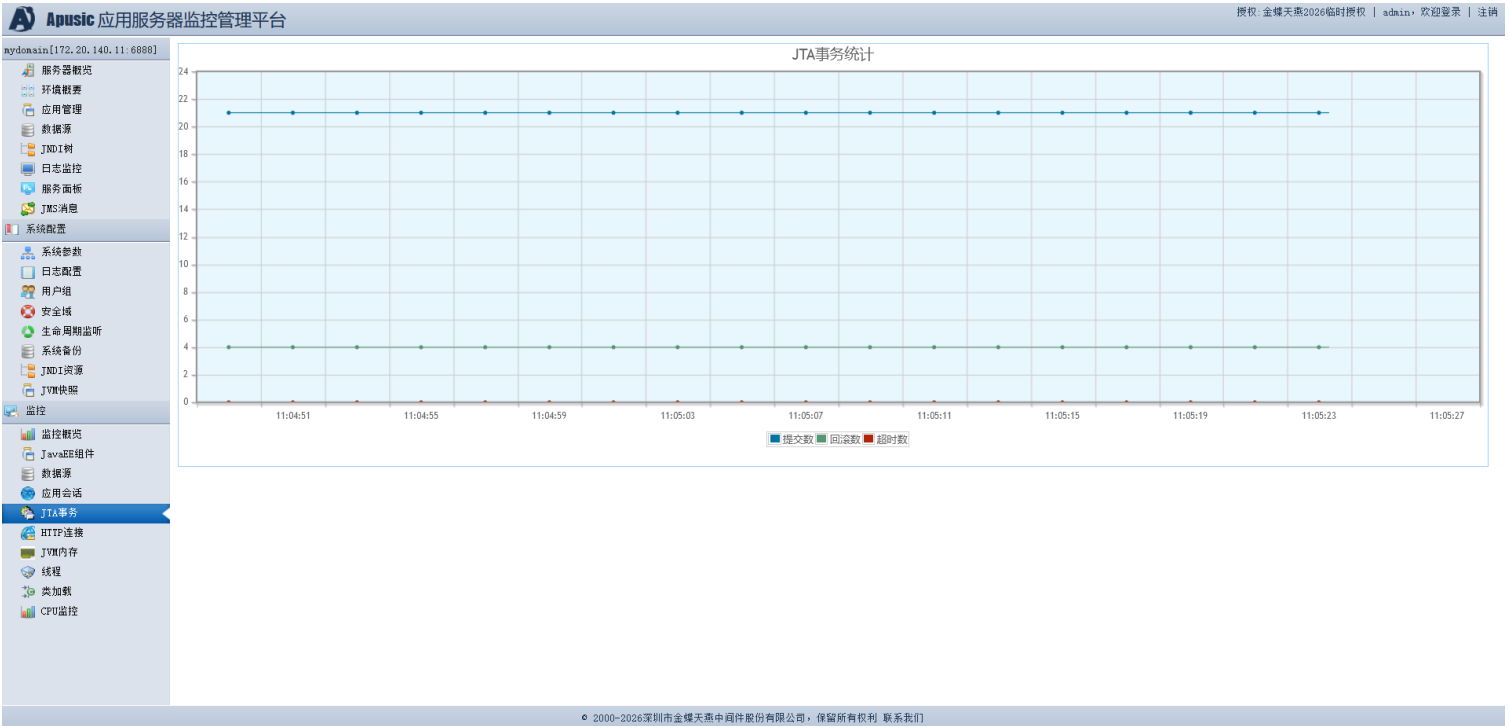
7.8 HTTP连接监控

打开Apusic应用服务器监控管理平台, 进入“监控”)“HTTP连接监控”操作界面下。在此可以监控HTTP连接数以及随时间变化的曲线图。



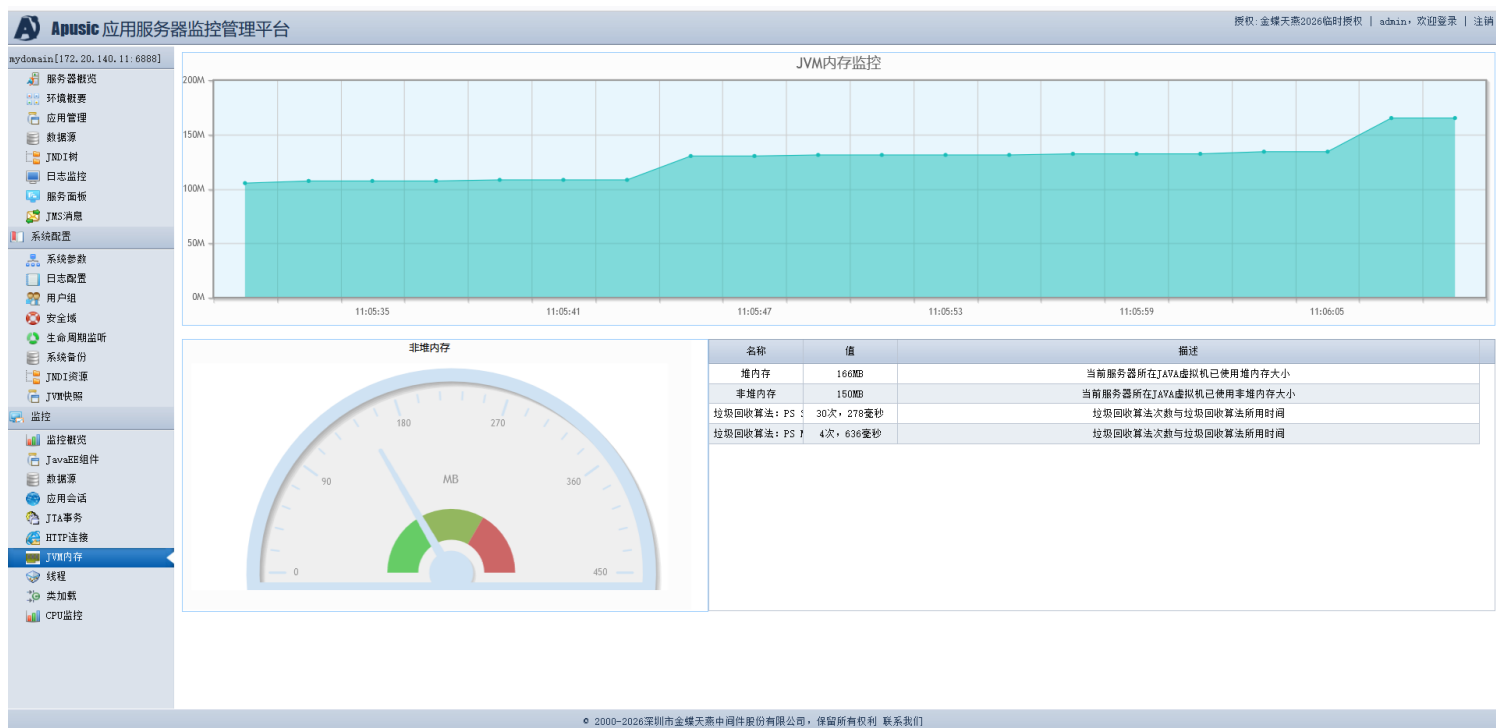
7.9 JTA事务监控

打开Apusic应用服务器监控管理平台, 进入“监控”)“JTA事务”操作界面下。在此可以监控到JTA事务的提交数, 回滚数以及超时数随时间的变化等。



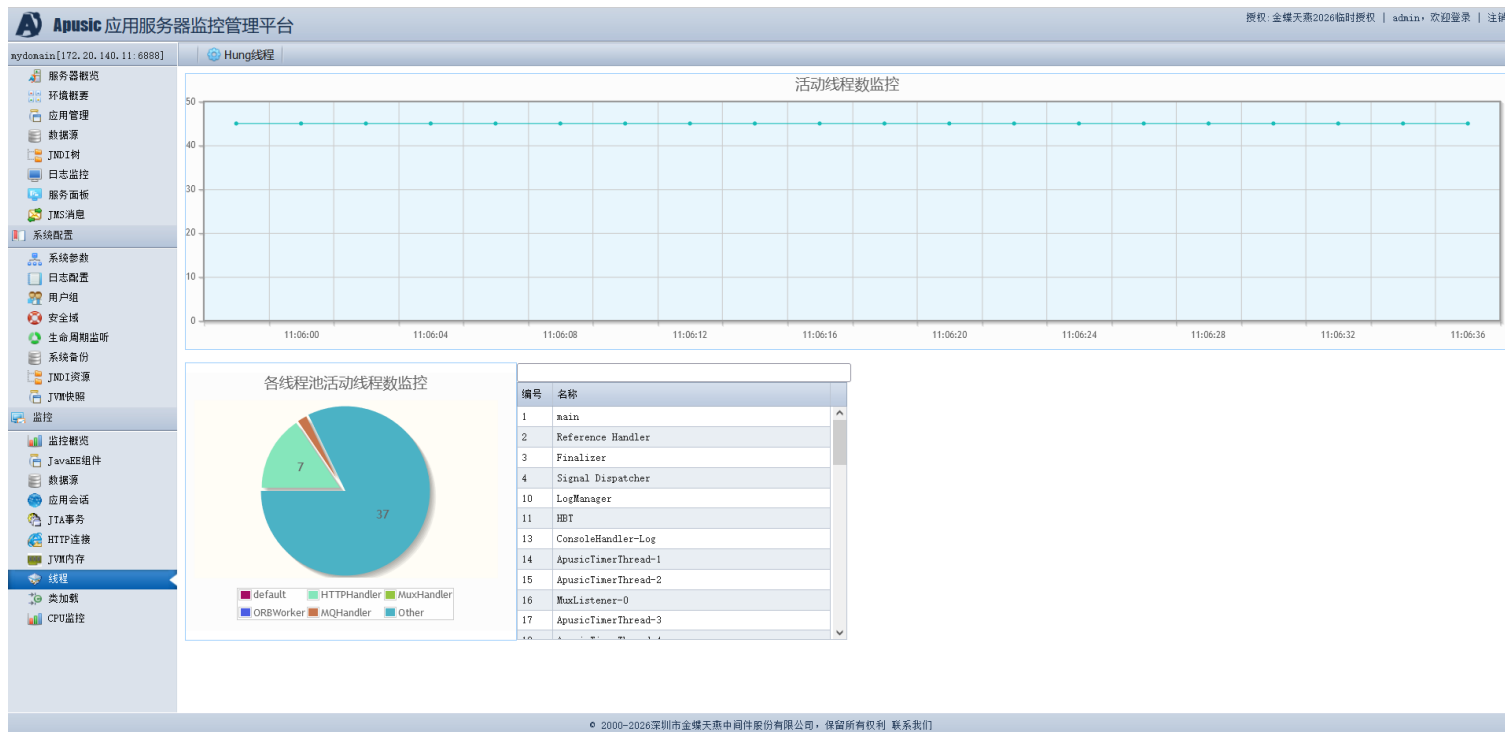
7.10 JVM内存监控

打开Apusic应用服务器监控管理平台，进入“监控”\“JVM内存”操作界面下。在此可以查看当前服务器所在java虚拟机已使用的堆内存的使用量及随时间变化的曲线图，还可以查看到java虚拟机内存使用的情况，包含堆和非堆。



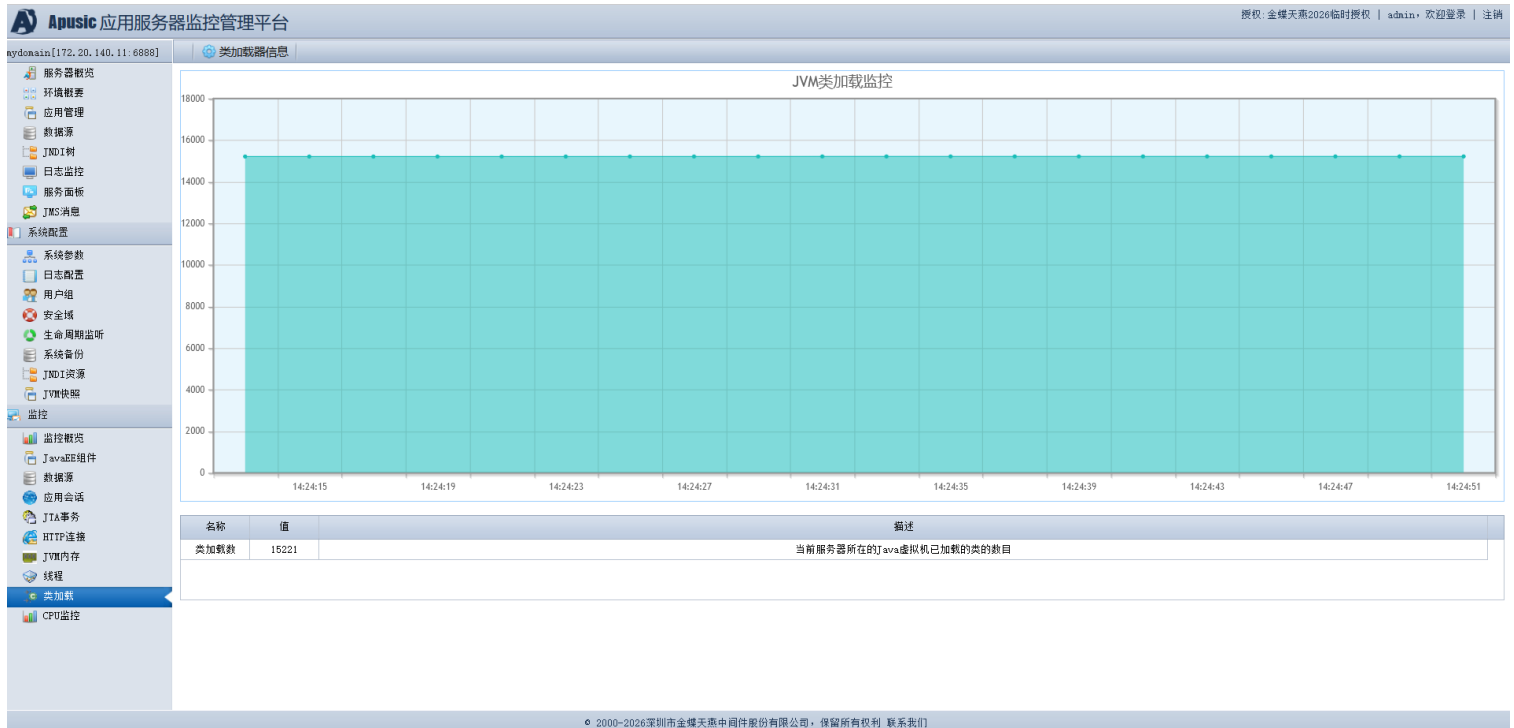
7.11 线程监控

打开Apusic应用服务器监控管理平台，进入“监控”\“线程”操作界面下。在此可以监控当前服务器所在java虚拟机的活动线程数及其随时间的变化活动曲线。另外以饼状图的形式显示了线程池中各线程的比例。

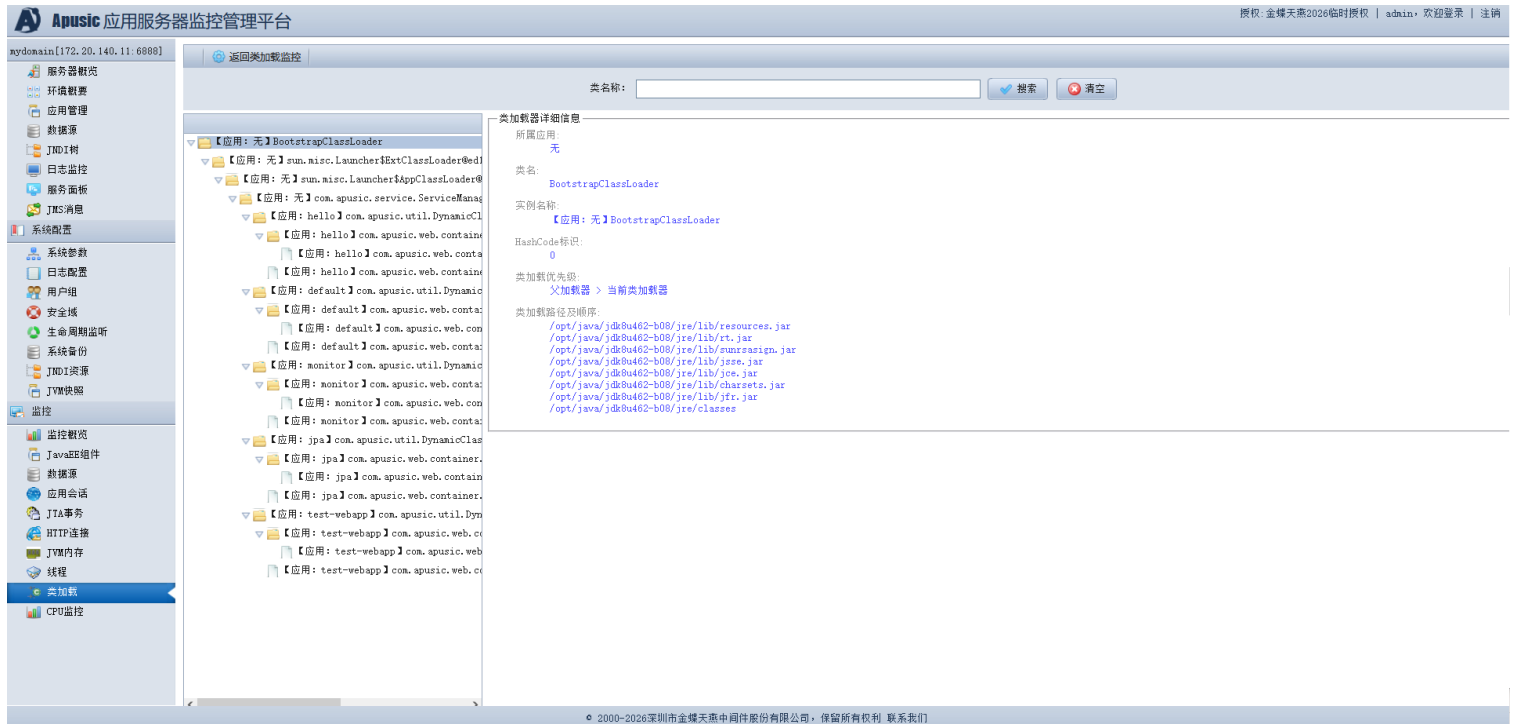


7.12 类加载监控

打开Apusic应用服务器监控管理平台，进入“监控”>“类加载”操作界面下。在此界面下可以查看当前服务器所在java虚拟机已加载的类的数目以及其随时间的变化图。

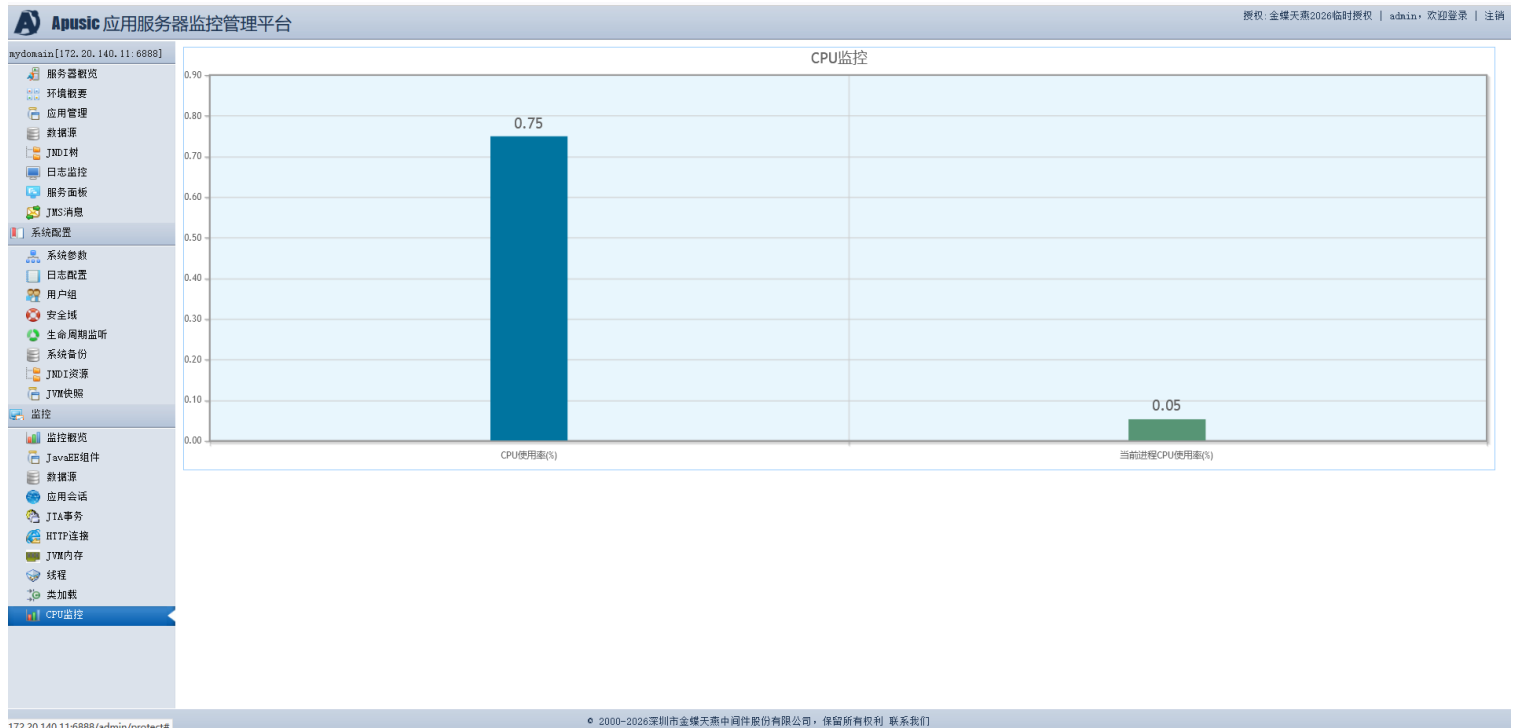


点击“类加载信息”，可查看类加载情况。



7.13 CPU监控

打开Apusic应用服务器监控管理平台，进入“监控”\“CPU监控”操作界面，查看CPU使用情况。



7.14 系统集成

AAS支持与Zabbix、Prometheus等流行的可观测系统实现集成，能无缝融入已有的IT监控运维体系，实现对服务器主机、网络、数据库、中间件及应用一体化融合监控。

具体集成操作请参考用户手册相关章节内容，本手册文档不再赘述。

8 使用及运维常见问题

产品使用及运维常见问题，详情请参考《金蝶Apsic应用服务器V9常见问题》手册文档。

注：更多AAS V9产品使用及运维常见问题，请访问金蝶云社区获取更多解决方案。 访问地址：

<https://vip.kingdee.com/knowledge/specialDetail/581874822318363904?category=581880019329094912&productLineId=14&lang=zh-CN>

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

