ApusicAS与Tomcat对比分析

金蝶中间件有限公司

2014年4月1日

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **更新人** | **版本** | **备注** |
| 2014-4-1 |  | 1.0 |  |
| 2015-3-10 |  | 1.1 |  |
|  |  |  |  |

目录

[版本历史 2](#_Toc385843670)

[目录 3](#_Toc385843671)

[第1章 概述 4](#_Toc385843672)

[1.1 厂商信息 4](#_Toc385843673)

[第2章 各方面对比 5](#_Toc385843674)

[2.1 功能 5](#_Toc385843675)

[2.2 性能 6](#_Toc385843676)

[2.3 安全性 7](#_Toc385843677)

[2.4 扩展性 7](#_Toc385843678)

[2.5 易用性 8](#_Toc385843679)

[2.6 支持服务 8](#_Toc385843680)

[2.7 典型用户 9](#_Toc385843681)

# 概述

Kingdee Apusic Application Server（下称：AAS）和Tomcat并不能直接比较，因为Tomcat只是一个Web容器，它只包含Servlets和JSP容器，而AAS还包含EJB, JMS, JNDI, JTA/JTS等J2EE中的所有规范。Tomcat是Apache软件组织的一个子项目，Apusic是金蝶中间件有限公司的商业产品。

本文试图从功能、性能、安全性、扩展性、易用性、服务支持和客户等七大方面对AAS和Tomcat进行初步比较，以便读者比较快速地了解掌握相关情况。本文所采用的数据，均来自公开场合的技术文档、资料或报导，如需进一步了解详细情况，请联系厂商进行咨询。

## 厂商信息

表一 金蝶中间件有限公司与Apache 软件组织的基本情况比较：

|  |  |  |
| --- | --- | --- |
|  | ***金蝶中间件*** | ***Apache 软件组织*** |
| ***成立时间*** | 2000年 | 1995年 |
| ***人员规模*** | 150+ | 150+ |
| ***机构设置*** | 中国境内设置北方区、华东区、西部区、南方区、北京子公司五大营销服务机构，总部位于广东深圳。 | 全球的自由软件组织，成员通过邀请制加入。 |
| ***是否JCP成员*** | 是 | 是 |
| ***产品初次发布时间*** | 2000年 | 1995年 |
| ***产品最新版本号*** | 9.0 | 7.x |
| ***产品覆盖区域*** | 中国大陆、港澳、台湾地区 | 全球各地 |
| ***产品覆盖行业*** | 电子政务、电子商务等 | 个人网站，小型商业系统 |

从表一可知，Apache 软件组织是个全球性的开源软件组织，旗下有30多个子项目。Apache 软件组织的Tomcat主要还是覆盖个人网站和小型商业系统，由于她是全球范围内免费的，不提供服务和支持，所以应用都会受限。在中国市场，而金蝶中间件作为本土的软件提供商，并凭借其母公司金蝶集团在国内软件业的影响力，能更好的为客户提供贴心的服务和支持。

# 各方面对比

## 功能

JavaEE应用服务器必须要遵循Java EE规范，因此衡量JavaEE应用服务器功能是否完备，首先要看其对标准的支持程度。

表2 AAS和Tomcat对Java EE规范的支持情况

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** | ***Tomcat*** |
| ***Java EE 6 (JSR 316)*** | √ |  |
| ***Web Services技术*** |  |  |
| ***Java API for RESTful Web Services (JAX-RS) 1.1 (JSR 311)*** | √ |  |
| ***Implementing Enterprise Web Services 1.3 (JSR 109)*** | √ |  |
| ***Java API for XML-Based Web Services (JAX-WS) 2.2 (JSR 224)*** | √ |  |
| ***Java Architecture for XML Binding (JAXB) 2.2 (JSR 222)*** | √ |  |
| ***Web Services Metadata for the Java Platform (JSR 181)*** | √ |  |
| ***Java APIs for XML Messaging 1.3(JSR 67)*** | √ |  |
| ***Java API for XML Registries (JAXR) 1.0(JSR 93)*** | √ |  |
| ***Web应用技术*** |  |  |
| ***Java Servlet 3.0(JSR 315)*** | √ | √ |
| ***JavaServer Faces 2.0(JSR 314)*** | √ | √ |
| ***JavaServer Pages 2.2/Expression Language 2.2(JSR 245)*** | √ | √ |
| ***Standard Tag Library for JavaServer Pages (JSTL) 1.2(JSR 52)*** | √ | √ |
| ***Enterprise JavaBeans (EJB) 3.1 Lite*** | √ |  |
| ***企业应用技术*** |  |  |
| ***Contexts and Dependency Injection for Java (Web Beans 1.0) (JSR 299)*** | √ |  |
| ***Dependency Injection for Java 1.0(JSR 330)*** | √ |  |
| ***Bean Validation 1.0(JSR 303)*** | √ |  |
| ***Enterprise JavaBeans 3.1 (includes Interceptors 1.1) (JSR 318)*** | √ |  |
| ***Java EE Connector Architecture 1.6(JSR 322)*** | √ |  |
| ***Java Persistence 2.0(JSR 317)***  | √ |  |
| ***Common Annotations for the Java Platform 1.1(JSR 250)*** | √ |  |
| ***Java Message Service API 1.1(JSR 914)*** | √ |  |
| ***Java Transaction API (JTA) 1.1(JSR 907)*** | √ |  |
| ***JavaMail 1.4(JSR 919)*** | √ |  |
| ***管理与安全技术*** |  |  |
| ***Java Authentication Service Provider Interface for Containers(JSR 196)*** | √ |  |
| ***Java Authorization Contract for Containers 1.3(JSR 115)*** | √ |  |
| ***Java EE Application Deployment 1.2(JSR 88)*** | √ | √ |
| ***J2EE Management 1.1(JSR 77)*** | √ |  |
| ***其他相关规范*** |  |  |
| ***Java API for XML Processing (JAXP) 1.3(JSR 206)*** | √ |  |
| ***Java Database Connectivity 4.0(JSR 221)*** | √ | √ |
| ***Java Management Extensions (JMX) 2.0(JSR 255)*** | √ | √ |
| ***Streaming API for XML (StAX) 1.0(JSR 173)*** | √ |  |
| ***Java API for WebSocket (JSR 356)*** | √ |  |

除标准功能（对规范的支持）外，一般情况下Java应用服务器还将提供覆盖应用软件生命周期的扩展功能（工具），以便客户更好地使用。Apusic和Tomcat扩展功能的情况请参考表3所示。

表3 Apusic和Tomcat扩展功能的比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** | ***Tomcat*** |
| ***开发工具*** | √ |  |
| ***远程管理工具*** | √ | √ |
| ***命令行管理工具*** | √ | √ |
| ***Eclipse开发插件*** | √ | √ |
| ***性能监控工具*** | √ |  |
| ***集中管控工具*** | √ |  |

从标准功能和扩展功能的比较中，可以看出，AAS与Tomcat相差比较大，因为Tomcat只是一个Web容器，很多Java EE的功能都不具备。Tomcat的管理工具也是比较简单，大多数都是通过手工的配置文件来解决，无疑增加了维护成本。

## 性能

决定Java EE应用系统性能表现的因素涉及网络、硬件、数据访问层、中间件、开发框架及应用设计实现等多个方面，其中Java应用服务器本身的性能表现是相当重要的一方面，Java应用服务器应尽可能提供各种性能优化技术。而Tomcat只是Web容器，所以只有比较其中的几项了，AAS和Tomcat性能优化技术比较请参考表4所示。

表4 Apusic和Tomcat性能优化技术比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** |  ***Tomcat*** |
| ***JSP预编译技术*** | √ | √ |
| ***数据库连接池*** | √ | √ |
| ***对象实例池*** | √ |  |
| ***集群配对复制*** | √ |  |
| ***支持Client Side Session Cache技术（for cluster）*** | √ | √ |
|  |  |  |

在性能优化方面，AAS提供更多有效的技术手段。经过测试，在相同的硬件配置基础下，Tomcat最多支持1000人同时在线，而AAS支持>5000人同时在线。据统计，很多网站开始时使用Tomcat，但是随着访问量的上升，Tomcat的表现都不如意。

## 安全性

有关安全性的比较见表5所示。

表5 AAS和Tomcat安全性比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** |  ***Tomcat*** |
| ***支持Java Authentication and Authorization Service (JAAS)*** | √ | √ |
| ***支持Java Secure Sockets Extensions (JSSE)*** | √ | √ |
| ***支持Java Cryptography Extensions (JCE)*** | √ | √ |
| ***支持Java Authorization Contract for Containers (JACC)*** | √ | √ |
| ***提供安全实现源码备查*** | √ | √ |
| ***支持HTTPS*** | √ | √ |
| ***敏感配置项加密*** | √ |  |

在对Java标准安全技术的支持方面，AAS比Tomcat提供良好支持。Tomcat作为Web容器，在安全性方面只是提供基本的功能，没有像AAS那样要求严格。AAS还提供自助研发的安全产品，提供更安全的解决方案。

## 扩展性

扩展性比较参考表6。

表6 Apusic和Tomcat扩展性比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** |  ***Tomcat*** |
| ***支持异构系统集群*** | √ | √ |
| ***支持Web负载均衡*** | √ | √ |
| ***支持EJB负载均衡*** | √ |  |
| ***支持JNDI集群*** | √ |  |
| ***支持Session失效恢复*** | √ | √ |
| ***支持Apache服务器*** | √ | √ |
| ***参加集群的服务器数量不受限制*** | √ | √ |
| ***支持分布式缓存*** | √ | √ |
| ***提供纯Java的负载均衡器*** | √ |  |
| ***支持BIO与NIO接入方式*** | √ | √ |
| ***提供节点管理器*** | √ |  |
| ***提供远程管理脚本工具*** | √ |  |

AAS与Tomcat均提供了对系统扩展的良好支持，提供了全面的WEB集群和负载均衡、失效恢复技术。但Tomcat的集群只提供WEB集群，而AAS支持EJB、JNDI的集群。

## 易用性

易用性比较参考表7。

表7 Apusic和Tomcat易用性比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS***  |  ***Tomcat*** |
| ***提供开发工具*** | √ |  |
| ***提供远程管理工具*** | √ | √ |
| ***提供运行期监控工具*** | √ | √ |
| ***支持热部署*** | √ | √ |
| ***支持目录级部署*** | √ | √ |
| ***提供完善的中文手册*** | √ |  |
| ***提供集中管控工具*** | √ |  |
| ***封装的管理监控API*** | √ |  |
| ***一个安装支持多个域管理*** | √ |  |

Tomcat作为开源项目，在易用性方面，都还停留在供开发人员研究的层面，而且文档基本都是英文的，中文翻译也不及时准确。AAS更多结合了中国的实际情况和使用习惯，因此更易为国内用户所接受。

## 支持服务

易用性比较参考表8。

表8 Apusic和Tomcat服务支持比较

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** |  ***Tomcat*** |
| ***提供7x24小时客户支持*** | √ |  |
| ***提供专业培训服务*** | √ |  |
| ***提供专业咨询服务*** | √ |  |
| ***提供开发级服务*** | √ |  |
| ***提供源码级服务*** | √ |  |

AAS作为商业性产品，提供了相当完善的服务支持体系，而且金蝶中间件薀含的本地化服务能力具有更大的吸引力。Tomcat作为国际开源组织，不对中国地区提供服务，遇到问题只能自己解决，这样就会存在比较大的风险，增加更多开发维护成本。

## 典型用户

比较参考表9。

表9 Apusic和Tomcat分行业典型用户

|  |  |  |
| --- | --- | --- |
|  | ***AAS*** |  ***Tomcat*** |
| ***金融*** | 人民银行招商证券申银万国证券香港亨达集团 |  |
| ***电信*** | 中国移动广东电信辽宁移动通信 |  |
| ***政府*** | 国家质检总局国家民政部国家监察部国家信访局海南省电子政务湖北省电子政务广东工商南宁市政府 |  |
| ***工商企业*** | 深圳万科香港中旅香港华润宁煤集团海尔集团广汽集团 |  |
| ***其他*** | TOM户外传媒华为联想金蝶集团九牧王集团安踏集团 |  |

AAS在电子政务、工商企业树立了比较多典型用户，精彩用户案例接踵而来。Tomcat在商业应用上都还只是充当小型Web容器的角色，个人网站或者要求不严格的小系统才采用。