



IBM China Development Lab

# WebSphere Process Server V6 性能调优之一

## 调优参数介绍

高明  
WPS 二线技术支持

# 目录

- 系统调优方法
- 追踪和日志相关的参数
- JVM相关的参数
- MDB 激活规范相关的参数
- MQ Listener Port 相关的参数
- 线程池大小相关的参数
- JMS 连接池相关的参数
- 数据源连接池大小相关的参数
- 数据源预处理语句（prepared statement）缓冲大小相关的参数
- 消息引擎相关的参数
- BPC相关的参数
- 开发模式
- 总结
- Q&A

# WPS系统调优方法

- 系统调优是一个反复循环的过程
  - 1. 设置一些合理的初始参数，并运行系统
  - 2. 监控系统，获得性能指标，判断是否需要性能调优
  - 3. 根据监控获得的性能指标进行进一步系统调优
  - 4. 重复进行上述过程直到性能满足要求为止

# WPS追踪和日志调优相关参数

## ■ 关闭追踪日志

- 减少I/O带来的性能下降
- 管理控制台：  
故障诊断 > 记录和跟踪 > server 名称 > 更改日志详细信息级别
- 建议在配置和运行时页面同时设置\*=all=disabled

## ■ 开闭PMI日志

- **PMI**：提供解释运行时和应用程序资源行为的一组全面的数据
- 建议在不需要监控的时候禁用所有统计信息，以提高性能
- 管理控制台：监视和调整 > 性能监视基础结构 (PMI) > server名称

The image contains two screenshots from the IBM management console. The left screenshot shows the 'Records and Tracking > server1 > Change Log Detail Level' page. It has a 'Configuration' tab selected and a text input field containing '\*=info: SCA.\*=all'. The right screenshot shows the 'Performance Monitoring Infrastructure (PMI) > server1' page. It has a 'Configuration' tab selected and the 'Current Statistics Information' section is set to 'None' (无), which is highlighted with a red box. Below it, the 'Basic' option is also visible.

## WPS追踪和日志调优相关参数（续）

- 关闭跨组件追踪（Cross-Component Tracing）
  - 跨组件追踪是WPS特有的一个日志追踪功能，您可以使用此功能从systemout.log 或 trace.log日志追踪中识别与WPS或WESB的模块（modules）和组件（components）相关的数据
  - 启动跨组件追踪将消耗I/O资源，尤其是启动数据快照模式将会有更大的I/O操作将快照输出到文件
  - 建议在不需要调试时关闭跨组件追踪功能
  - 管理控制台：故障诊断 > 跨组件跟踪
- 关闭IBM服务日志（service logs）
  - IBM 服务日志包含写入 System.out 流的 WebSphere Application Server 消息和一些包含扩展服务信息的特殊消息
  - 多并发情况下启动IBM 服务日志会严重影响性能
  - 管理控制台：故障诊断 > 日志和跟踪 > server name > IBM 服务日志

### 跨组件跟踪

在此页面上，可选择要从收集跟踪数据的服务器。在“配置”列中，对服务器指定服务器启动或重新启动时将使用的跟踪设置。在“运行时”列中，对正在运行的服务器指定跟踪设置。

服务器	状态	节点	配置	运行时
server1	+	widNode	disable	disable

总数 1

应用 确定 复位 取消



# JVM相关参数调优

## ■ 调整 Java 对象垃圾回收（GC）规则

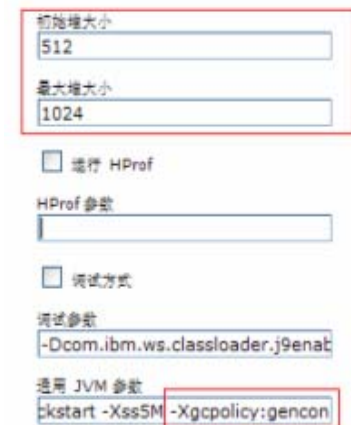
- IBM JVM默认的Java GC规则是标记-清除-整理（mark-sweep-compact）
- generational concurrent（gencon）规则在许多情况下通过调整短生命周期对象和长生命周期对象所占用Java堆的空间大小可以提高性能
- 您可以通过在通用 JVM 参数（Generic JVM arguments）中输入-Xgcpolicy:gencon来启动generational concurrent（gencon）规则

## ■ 调整Java 堆栈大小

- 主要考虑三个方面：
  - 初始Java 堆栈大小
  - 最大Java 堆栈大小
  - 短生命周期对象所占用的Java 堆栈大小
    - 你可以在Generic JVM arguments中输入三个参数：
      - -Xmns<size>: 初始化的短生命周期对象所占用的Java 堆栈大小
      - -Xmx<size>: 最大的短生命周期对象所占用的Java 堆栈大小
      - -Xmn<size>: 固定短生命周期对象所占用的Java 堆栈大小

## ■ 调整途径

- 管理控制台：服务器 > 应用程序服务器 > server名称 > 服务器基础结构 > Java 和进程管理 > 进程定义 > Java 虚拟机



# MDB 激活规范相关参数调优

## ■ 有两种MDB激活规范

### — J2C 激活规范

- 资源 > 资源适配器 > J2C 激活规范 > 激活规范名称 > J2C 激活规范定制属性
- 资源 > 资源适配器 > 资源适配器名称 > 其他属性 > J2C 激活规范 > J2C 激活规范名称 > J2C 激活规范定制属性

### — JMS激活规范

- 资源 > JMS > 激活规范 > 激活规范名称

## ■ 两个关键参数

- maxConrurrency: 用于调整消息驱动的 bean 的最大实例数
- maxBatchSize: 用于调整消息驱动的 bean 的最大批处理大小

### J2C 激活规范 > failedevent\_AS > 定制属性

使用此页面来指定定制属性。对于您配置的资源提供程序和资源工厂，企业信息系统（EIS）需要这些属性。例如，大多数数据库供应商需要访问该数据库的数据源的其他定制属性。

☐ 首选项

名称	值	描述	必需
destinationType	Queue		true
subscriptionName			false
messageSelector	(SI_CorrelationID IS NULL) AND (user_scaStoredMsgId IS NULL)		false
useServerSubject	true		false
targetType			false
maxConcurrency	10		false
userName			false
discriminator			false
target			false
durableSubscriptionHome			false
messageDeletionMode	Single		false
busName	SCA.SYSTEM.widCell.Bus		false
maxBatchSize	1		false

最大批次大小

1

最大并发端点数

10

# MQ侦听器端口（Listener Port）相关参数调优

- WPS使用MQ 侦听器端口为MQ或MQ JMS绑定（binding）读入消息
- 关键参数
  - 最大会话（maximum sessions）：控制读入消息的并发性能，每个会话相应于一个单独的侦听器线程，并因此控制并发处理的消息数
    - 如果您希望按严格的消息顺序来处理消息，则将值设置为 1，这样在任何时候都只有 1 个线程在处理消息
    - 如果您希望同步处理多个消息（称为“消息并发性”），则将此属性设置为大于 1 的值
  - 管理控制台：服务器 > 应用程序服务器 > 通信 > 消息传递 > 消息侦听器服务 > 其他属性 > 侦听器端口 > 侦听器端口名称





## 线程池相关参数调优

- 对于每个server，有三个线程池的参数需要调整：
  - Default: Default 线程池中的线程用于分配给MDB的实例使用
  - ORB.thread.pool: ORB（对象请求代理，Object Request Broker）线程池用于运行ORB请求
  - WebContainer: Web Container 线程池中的线程用于处理HTTP和Web服务的请求
- 两个Work Manager需要调整线程池参数
  - BPENavigationWorkManager: 用BPC流程扭转的异步Bean的线程池
  - DefaultWorkManager: 默认异步Bean的线程池
- 需要根据需要调整相关线程池的大小
  - 服务器 > 应用程序服务器 > 其他属性 > 线程池
  - 资源 > Asynchronous beans > 工作管理器 > BPENavigationWorkManager
  - 资源 > Asynchronous beans > 工作管理器 > DefaultWorkManager

选择	名称	描述	最小大小	最大大小
<input type="checkbox"/>	Default		5	20
<input type="checkbox"/>	ORB.thread.pool		10	50
<input type="checkbox"/>	SIBFAPInboundThreadPool	Service integration bus FAP inbound channel thread pool	4	50
<input type="checkbox"/>	SIBFAPThreadPool	Service integration bus FAP outbound channel thread pool	4	50
<input type="checkbox"/>	WebContainer		10	50

# JMS连接池相关参数调优

- 有四种JMS连接池
  - J2C 连接工厂：资源适配器通过J2C 连接工厂将应用程序连接到企业信息系统（EIS）
  - JMS连接工厂：JMS 连接工厂用于创建与 JMS 队列和主题的连接
  - JMS队列连接工厂：JMS 连接工厂用于创建与 JMS 目标的连接
  - JMS主题连接工厂：JMS 连接工厂用于创建与 JMS 主题的连接
- 需要调整的参数
  - 最大连接数



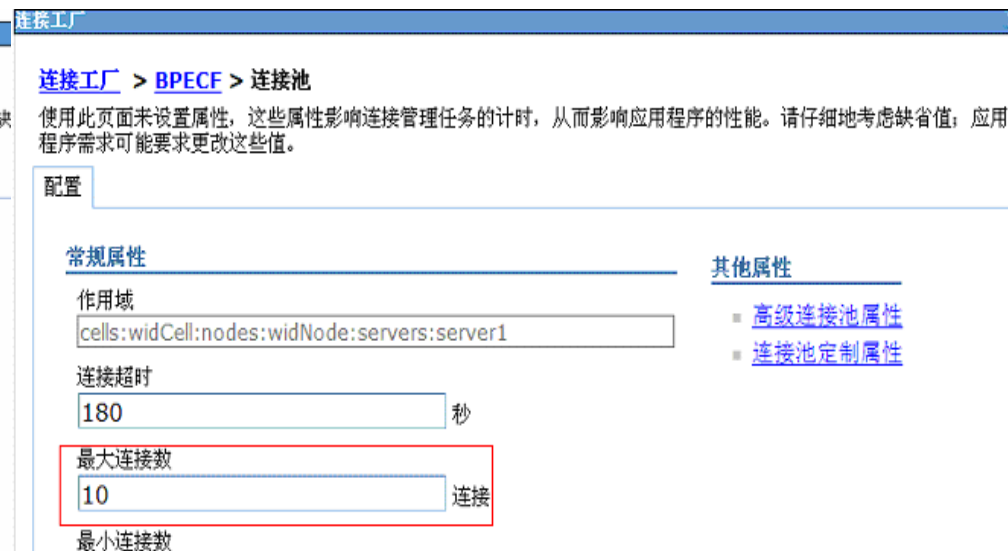
**J2C 连接工厂**

[J2C 连接工厂](#) > [test](#) > 连接池

使用此页面来设置属性，这些属性影响连接管理任务的计时，从而影响应用程序的性能。请仔细地考虑缺省值；应用程序需求可能要求更改这些值。

配置

常规属性	其他属性
作用域 <input type="text" value="cells:widCell:nodes:widNode"/>	<ul style="list-style-type: none"><li>高级连接池属性</li><li>连接池定制属性</li></ul>
连接超时 <input type="text" value="180"/> 秒	
最大连接数 <input type="text" value="10"/> 连接	
最小连接数 <input type="text" value="1"/> 连接	



**连接工厂**

[连接工厂](#) > [BPECF](#) > 连接池

使用此页面来设置属性，这些属性影响连接管理任务的计时，从而影响应用程序的性能。请仔细地考虑缺省值；应用程序需求可能要求更改这些值。

配置

常规属性	其他属性
作用域 <input type="text" value="cells:widCell:nodes:widNode:servers:server1"/>	<ul style="list-style-type: none"><li>高级连接池属性</li><li>连接池定制属性</li></ul>
连接超时 <input type="text" value="180"/> 秒	
最大连接数 <input type="text" value="10"/> 连接	
最小连接数 <input type="text" value="1"/> 连接	

# 数据源连接池相关参数调优

## ■ 主要参数

### — 最大连接数:

- 指定您可以在数据源连接池中创建的最大物理连接数
- 最大连接数会影响数据库操作的并发能力

### — 管理控制台: 资源 > JDBC > 数据源 > 数据源名称 > 其他属性 > 连接池属性

## ■ 四个WPS相关的数据源连接池需要调节

- BPEDataSource BPE 数据源
- SCA Application Bus ME 数据源
- SCA System Bus ME 数据源
- CEI Bus ME 数据源



# 数据源预处理语句相关的参数调优

## ■ 主要参数

— 缓冲大小：指定每个连接可以高速缓存的语句数

- 在用户关闭语句后，WebSphere Application Server 将对该语句进行高速缓存
- WAS 数据源通过对活动连接中未使用的预编译语句和可调用语句进行高速缓存来优化对那些语句进行的处理
- 预编译语句是存储在 PreparedStatement 对象中的预编译 SQL 语
- Application Server 使用此对象来根据应用程序运行时的需要将 SQL 语句运行多次，并且值由运行时确定
- 如果语句高速缓存不够大，就会废弃有用的条目以便为新条目腾出空间
- 将高速缓存大小设置为该值意味着您不会再有高速缓存废弃的情况发生
- 通常，您的应用程序具有的语句越多，高速缓存应该越大。

■ BPC DB大量用到数据库预处理，至少要设为128

数据源 > BPEDataSourceDerby > WebSphere Application Server 数据源属性

使用此页面来设置 WebSphere(R) Application Server 连接特定于管理的属性，它们会影响连接池。

配置

常规属性

语句高速缓存大小

128

语句

# 消息引擎相关参数调优

- 消息引擎的两个参数会影响消息引擎的性能
  - DiscardableDataBufferSize: 在内存中尽最大可能保存非持久化(最佳非持久)消息的缓冲区大小, 如果超过这个大小就会将消息丢弃 (bytes)
    - 默认值是320000
  - CachedDataBufferSize: 在内存中保存JMS消息(除了最佳非持久消息)的缓冲区大小 (bytes)
    - 默认值是320000
- 设置
  - 管理控制台: 服务集成 > 总线 > 总线名称 > 消息传递引擎 > 消息传递引擎名称 > 定制属性





# BPC相关参数调优

## 新的导航模式，WorkManager-based 导航

- 一个WorkManager是一个J2EE的线程池。通过使用WorkManager，BPC使用WAS提供的异步Bean实现流程的扭转，而不是用JMS提供的消息方式。从而提高BPC流程的运行速度。

## 启用高级性能优化

- 当清除了此选项时，将使用 JMS 导航
- 当选择了此选项时，将使用异步 Bean。
  - 消息池大小：当生成的消息比使用的消息更多时，消息池将用作缓冲区。消息是由工作管理器中的导航线程以及进入系统的API 和 SCA 请求所生成的。而工作管理器中的导航线程将使用消息。如果消息池变满，那么会将后续产生的消息直接发送至JMS 并使用 JMS 导航来处理。
  - 事务间高速缓存大小：此字段指定在各个事务中要保留在高速缓存中的数据库对象（例如 BPEL 变量和流程上下文）的数目。应该将此参数设置为两倍于并发运行的流程实例数。默认值为BPENavigationWorkManager的线程池大小。

### 业务流程导航性能

启用高级性能优化

#### 基于工作管理器的导航

##### 消息池大小

使用工作管理器线程池大小 \* 10

消息池大小值

延期消息的最大生存期（以秒计）

延期消息的恢复时间间隔（简单日历）

线程上的最大处理时间（以秒计）

#### 事务间高速缓存

使用工作管理器线程池大小

事务间高速缓存大小

# 开发模式

- WPS server 可以运行于开发模式
  - 开发模式通过禁止字节码验证减少JIT编译时间并减少JVM启动时间
  - 在开发模式下还可以不用停止流程模板就可以卸载应用程序，从而减少了开发调试时间
- 尽管开发模式有助于提高性能，在生产系统中不要选择此选项

The screenshot displays the 'Integrated Solutions Console' interface. On the left is a navigation tree with categories like 'Servers', 'Applications', and 'Resources'. The main content area is titled 'Application servers' and shows the configuration for a specific server named 'server1'. The 'Configuration' tab is active, and the 'Run in development mode' checkbox is checked and highlighted with a red box. Other visible settings include 'Node name' (widNode), 'Parallel start' (checked), and 'Access to internal server classes' (set to 'Allow').

# 总结

- 系统调优方法
- 主要参数
  - 追踪和日志相关的参数
  - JVM相关的参数
  - MDB 激活规范相关的参数
  - MQ Listener Port 相关的参数
  - 线程池大小相关的参数
  - JMS 连接池相关的参数
  - 数据源连接池大小相关的参数
  - 数据源预处理语句 (prepared statement) 缓冲大小相关的参数
  - 消息引擎相关的参数
  - BPC相关的参数
  - 开发模式
- 调优原则：根据需求，反复调整和试验

# Q&A

谢谢！！