# Springboot项目相关参数优化

## 关于JVM参数

启动springboot项目，在java –jar 后面带上以下相关参数.

1. -server (启用jdk 的 server 版)
2. -XX:MetaspaceSize=128m （元空间默认大小）
3. -XX:MaxMetaspaceSize=128m （元空间最大大小）
4. **-Xms1024m （堆最大大小）**
5. **-Xmx1024m （堆默认大小）**
6. -Xmn256m （新生代大小）
7. -Xss256k （棧最大深度大小）
8. -XX:SurvivorRatio=8 （新生代分区比例 8:2）
9. -XX:+UseConcMarkSweepGC （指定使用的垃圾收集器，这里使用CMS收集器）

**关于Jdk8版本改动**:

JDK8之后把-XX:PermSize 和 -XX:MaxPermGen移除了，取而代之的是

-XX:MetaspaceSize=128m （元空间默认大小）

-XX:MaxMetaspaceSize=128m （元空间最大大小）

JDK 8开始把类的元数据放到本地化的堆内存(native heap)中，这一块区域就叫Metaspace，中文名叫元空间。

使用本地化的内存有什么好处呢？最直接的表现就是java.lang.OutOfMemoryError: PermGen 空间问题将不复存在，因为默认的类的元数据分配只受本地内存大小的限制，也就是说本地内存剩余多少，理论上Metaspace就可以有多大（貌似容量还与操作系统的虚拟内存有关？这里不太清楚），这解决了空间不足的问题。不过，让Metaspace变得无限大显然是不现实的，因此我们也要限制Metaspace的大小：使用-XX:MaxMetaspaceSize参数来指定Metaspace区域的大小。JVM默认在运行时根据需要动态地设置MaxMetaspaceSize的大小。

<https://zhuanlan.zhihu.com/p/31803182>

## Springboot-Tomcat参数优化

Springboot使用application配置设定Tomcat参数主要有以下几点：

1. 最小线程数(初始化线程数)

server.tomcat.min-spare-threads =100

1. 最大线程数

server.tomcat.max-threads = 300

1. 最大连接数

server.tomcat.max-connections = 200

1. 指定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。

server.tomcat.accept-count =50

1. 连接超时时间

server.connection-timeout = 3000

1. 编码方式

server.tomcat.uri-encoding = UTF-8

1. 最大post提交数据大小，设置为0不限制

server.tomcat.max-http-post-size= 0

1. 设定Tomcat日志相关

server.tomcat.accesslog.buffered=true

server.tomcat.accesslog.directory=logs

server.tomcat.accesslog.enabled=false

server.tomcat.accesslog.file-date-format=.yyyy-MM-dd

server.tomcat.accesslog.pattern=common

server.tomcat.accesslog.prefix=access\_log

server.tomcat.accesslog.rename-on-rotate=false

server.tomcat.accesslog.request-attributes-enabled=false

server.tomcat.accesslog.rotate=true

server.tomcat.accesslog.suffix=.log

## 三．MySQL JDBC 连接参数.

(一).其连接属性主要分为以下N个大类：

1. **Connection（**连接**）**
2. **Session.（会话）**
3. **Networking. （网络）**
4. **Security. （安全）**
5. **Statements. （声明）**
6. **Prepared Statements. （预先声明）**
7. **Result Sets. （结果集）**
8. **Metadata. （元数据）**
9. **BLOB/CLOB processing. （BLOB / CLOB处理）**
10. **Datetime types processing. （日期时间类型处理）**
11. **High Availability and Clustering. （高可用性和群集）**
12. **Performance Extensions. （性能扩展）**
13. **Debugging/Profiling. （调试/剖析）**
14. **Exceptions/Warnings. （异常/警告）**
15. **Tunes for integration with other products. （调整与其他产品的集成）**
16. **JDBC compliance. （JDBC合规性）**
17. **X Protocol and X DevAPI. （X协议和X DevAPI）**

具体类别参数设定参考：

<https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-configuration-properties.html>

(二).列举几个重要的参数，如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 缺省值 | 最低版本要求 |
| user | 数据库用户名（用于连接数据库） |  | 所以版本 |
| password | 用户密码（用于连接数据库） |  | 所以版本 |
| useUnicode | 是否使用Unicode字符集，如果参数characterEncoding设置为gb2312或gbk，本参数值必须设置为true | false | 1.1g |
| characterEncoding | 当useUnicode设置为true时，指定字符编码。比如可设置为gb2312或gbk | false | 1.1g |
| autoReconnect | 当数据库连接异常中断时，是否自动重新连接？ | false | 1.1 |
| autoReconnectForPools | 是否使用针对数据库连接池的重连策略 | false | 3.1.3 |
| failOverReadOnly | 自动重连成功后，连接是否设置为只读？ | true | 3.0.12 |
| maxReconnects | autoReconnect设置为true时，重试连接的次数 | 3 | 1.1 |
| initialTimeout | autoReconnect设置为true时，两次重连之间的时间间隔，单位：秒 | 2 | 1.1 |
| connectTimeout | 和数据库服务器建立socket连接时的超时，单位：毫秒。 0表示永不超时，适用于JDK 1.4及更高版本 | 0 | 3.0.1 |
| socketTimeout | socket操作（读写）超时，单位：毫秒。 0表示永不超时 | 0 | 3.0.1 |

## 四．关于springboot DataSource连接池

自动配置生产数据库连接DataSource。以下是选择特定实现的算法：

我们更喜欢Tomcat池DataSource的性能和并发性，所以如果可用，我们总是选择它。

否则，如果HikariCP可用，我们将使用它。

如果Tomcat池数据源和HikariCP都不可用，并且Commons DBCP可用，我们将使用它，但我们不建议在生产中使用它，并且不支持它的支持。

最后，如果Commons DBCP2可用，我们将使用它。

如果你使用spring-boot-starter-jdbc或spring-boot-starter-data-jpa ''starters'，你将自动获得依赖tomcat-jdbc。

（摘自springboot1.5.14官方文档）

配置Tomcat特定的连接池设置，Spring Boot提供了spring.datasource.tomcat.\* 在application.properties文件中使用的前缀:

**1. initialSize**  
initialSize 是池启动时创建的初始连接数。它配置如下。

spring.datasource.tomcat.initial-size=15

**2. maxWait**  
maxWait 是池在抛出异常之前等待返回连接的最大毫秒数。它配置如下。

spring.datasource.tomcat.max-wait=20000

**3. maxActive**  
maxActive 是池可以同时分配的最大活动连接数。它配置如下。

spring.datasource.tomcat.max-active=50

**4. maxIdle**  
maxIdle 是应始终保留在池中的最大连接数。它配置如下。

spring.datasource.tomcat.max-idle=15

**5. minIdle**  
minIdle 是应始终保留在池中的最小连接数。它配置如下。

spring.datasource.tomcat.min-idle=8

**6. defaultAutoCommit**  
defaultAutoCommit 属性配置此池创建的连接的默认自动提交状态。它是布尔值，默认值为**true**。

spring.datasource.tomcat.default-auto-commit=true

**7. testOnBorrow**  
testOnBorrow 是一个布尔值，指示在从池中借用对象之前是否验证对象。默认值为**false**。

spring.datasource.tomcat.test-on-borrow=false

## 参考链接：

1. <https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html#common-application-properties> （springboot所有的application配置）
2. <https://docs.oracle.com/middleware/11119/wls/PERFM/jvm_tuning.htm#i1146060> （jvm调优官方参考）
3. <https://www.concretepage.com/spring-boot/spring-boot-tomcat-connection-pool>

<https://docs.spring.io/spring-boot/docs/1.5.14.RELEASE/reference/html/boot-features-sql.html>

（springboot – DataSource相关参考文档）

1. <https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-configuration-properties.html> （MySQL jdbc 连接属性官方文档）
2. <https://www.oschina.net/question/16_2400> (jdbc-url重要参数)