VM options

REMOTE_DEBUG="-agentlib:jdwp=transport=dt_socket,address=14025,server=y,suspend=n"

MEMORY="-Xms24g -Xmx24g -XX:MaxMetaspaceSize=256m"

GC="-XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70 -XX:+ScavengeBeforeFullGC -XX:+CMSScavengeBeforeRemark -XX:+UseParNewGC"

GC_LOG=" -verbose:gc -Xloggc:./logs/gc_pid_%p.log -XX:+PrintGCDateStamps -XX:+PrintGCDetails -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M"

DUMP=”-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=./dumps/”

java $REMOTE_DEBUG $MEMORY $GC $GC_LOG $JMX $DUMP -cp libs com.my.Main

REMOTE_DEBUG

-Xdebug makes VM about 5% slower in debug mode because it can't do some kinds of optimizations.

Debugger actually starts when someone connects to debugging port.

-agentlib:jdwp=transport=dt_socket,address=14025,server=y,suspend=n

Java Debug Wire Protocol (JDWP) is the protocol used for communication between a debugger and the VM

suspend=y the Java-process will wait until the debugger connects,

suspend=n you will be also able to debug the application servers startup process.

server=y opens a socket and listens for incoming debugger requests.

server=n the debugged application will try to connect actively to a debugger and run therefore as a client.

MEMORY

-Xms24g -Xmx24g -XX:MaxMetaspaceSize=256m

Xmx – max memory

Xms – min memory

-XX:MaxMetaspaceSize allows to limit the amount of native memory used for class metadata.

By default class metadata allocation is limited by the amount of available native memory

GC

The Concurrent Mark Sweep (CMS) collector is designed for applications that prefer shorter garbage collection pauses and that can afford to share processor resources with the garbage collector while the application is running.

-XX:+UseConcMarkSweepGC	//GC for old generation
-XX:+CMSParallelRemarkEnabled	//reduce remark pauses
-XX:CMSInitiatingOccupancyFraction=70	//GC trigger
-XX:+UseCMSInitiatingOccupancyOnly	//use OccupancyFraction for all collections
-XX:+UseParNewGC	//GC for young generation
-XX:+CMSScavengeBeforeRemark	//force young collection before remark phase
-XX:+ScavengeBeforeFullGC	//do young generation GC prior to a full GC

GC_LOG

-verbose:gc

-Xloggc:./logs/gc_pid_%p.log

-XX:+PrintGCDateStamps

-XX:+PrintGCDetails

-XX:+UseGCLogFileRotation

-XX:NumberOfGCLogFiles=10

-XX:GCLogFileSize=100M

DUMP

-XX:+HeapDumpOnOutOfMemoryError

-XX:HeapDumpPath=./dumps/