**SPECpower2008 安装和使用**

[ **SPECpower2008介绍** 2](#_Toc293730575)

[ **SPECpower2008安装指南** 2](#_Toc293730576)

[ **JVM安装指南** 2](#_Toc293730577)

[ **测试环境准备** 2](#_Toc293730578)

[ **SPECpower2008运行** 3](#_Toc293730579)

[runpower.bat（红字需要修改） 4](#_Toc293730580)

[runtemp.bat 4](#_Toc293730581)

[runCCS.bat 5](#_Toc293730582)

[Runssj.bat 5](#_Toc293730583)

[rundirector.bat（基本不用修改） 7](#_Toc293730584)

[SPECpower\_ssj.props（可以从SPECpower\_ssj\_EXPERT.props改名后再改值，一般不修改，一次测试需要75分钟左右） 9](#_Toc293730585)

[ccs.props(一般不需要修改) 17](#_Toc293730586)

[SPECpower\_ssj\_config.props（可以不修改） 27](#_Toc293730587)

[SPECpower\_ssj\_config\_sut.props（可以不修改） 30](#_Toc293730588)

* **SPECpower2008介绍**

SPEC是the ***S***tandard ***P***erformance ***E***valuation ***C***orporation的缩写. SPEC是一个包含计算机硬件厂商,软件公司,大学,研究机构,系统集成商, 出版发行及咨询机构的非赢利性组织. SPEC的目标是创建,维护并引导一个计算机系统及业界公认的系统指标.

SPECpower2008 是由 SPEC® 开发的一种性能/功耗比基准测试，用于评估运行基于 Java 的应用程序的服务器的功耗情况。SPECpower\_ssj2008利用标准Java的JDK计算整体服务器性能，并根据其11个不同工作负载区域段的功耗得出服务器的工作负载/能耗比的测试方式，它应用specjbb作为工作负载，先实时满负荷的运行3次，求得平均值得到系统的最高性能值，然后系统以此为参照，按100%, 90%, 80%....10%, 0%(idle)运行工作负载，其系统的利用率也依次下降, 性能运行结果会以ssj\_ops方式记录。同时连接系统电源的功率仪(Yokogawa WT210)会实时记录系统的功率状况(Average Active Power (W))，最后系统会把性能和功率做一个累加并相除得到性能功耗比(∑ssj\_ops / ∑power =Performance to Power Ratio) 。

因此SPECpower\_ssj2008成为了一个较为客观的服务器的能耗标准。

* **SPECpower2008安装指南**

如是安装盘，可以直接运行java setup, 按照安装程序的流程往下走。如是非安装盘，可直接解压文件到指定目录, 在x86平台上比较推荐运行在Windows2008-R2-x86-64bit平台上，当然linux平台上也可以的，是java程序。

* **JVM安装指南**

SPECjbb 2005需要real-time JVM的支持， 因此需要安装JVM. 现有通用的JVM有Sun JVM, Oracle/BEA JRocket和IBM 的Websphere JVM. 根据以往的测试经验，JVM的性能对specjbb2005的结果有很大的影响，因此推荐IBM JVM.

 两个名词：

* SUT: 系统测试单元就是下面图上的“被侧系统”。
* CCS: 就是下面图上的“数据收集和控制系统”。

**Windows安装（SUT和CCS安装过程及内容完全一样）:**

* 从IBM网站上下载IBM JVM J9
* <http://www-01.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=D400&q1=IBM+WebSphere+Installation+Factory+V7.0.0.5+for+Windows&uid=swg24020539&loc=en_US&cs=utf-8&lang=en>
* 在Windows平台上直接运行安装程序
* 安装完后，在DOS command下试运行”java –version”, 查看JVM的运行版本情况, 如无此命令，请在Start->Control Panel->System and security->System->Advanced system setup

(开始->控制面板->系统和安全->系统->高级系统设置) 在”Environment Variables”(”环境变量”)中的path加入JVM的安装路径.

* **测试环境准备**

根据测试情况，系统和BIOS的设置对性能结果的影响也非常大，因此为取得最好性能，需要对系统和BIOS做以下设定。

系统BIOS设定

|  |  |
| --- | --- |
|  | Intel® Xeon® 5600/5500 series |
| EIST | Enabled(Default) |
| Intel Turbo Boost | Enabled (Default) |
| SMT | Enabled |
| Hardware Prefetcher | Disabled (default is enabled) |
| Advanced Cache Line Prefetcher | Disabled (default is enabled) |
| NUMA | Enabled |
| System Acoustics and Performance Configuration | CLTT |
| Node interleave | Disable |
| Power management(取决于每家OEM firmware) | DELL: system activeHP: balance |

* **SPECpower2008运行**
1. PC通过串口线(COM)和功耗仪通讯得到功率数据，PC通过网络和测试服务器相连来控制整个测试流程. PC通过USB口连接温度传感器得到环境温度（没有的话，直接启动温度进程，会模拟固定在21度）。



1. 调试功耗仪器保证能正确显示服务器电压，电流和功率运行状况
2. 在PC控制端依次运行以下批处理文件
* Runpower.bat (\specpower2008\ptd目录下，使用前需要配置它本身)
* Runtemp.bat (\specpower2008\ptd目录下)
* rundirector.bat(\specpower2008\ssj目录下，使用前需要配置SPECpower\_ssj.props)
1. 在测试服务器端运行以下批处理文件
* runssj.bat(\specpower2008\ssj目录下，使用前需要配置它本身)

运行完后过几秒钟可以看到起了几个java的命令行窗口并且和PC端的rundirector.bat有协议握手，表示和服务器连上了。

1. 在PC控制端运行以下批处理文件完成整个测试
* runCCS.bat(\specpower2008\ccs目录下，使用前需要配置ccs.props)

运行大约75分钟，结束后可以在result目录下发现ssj.000x-main.html 文件显示结果，如果需要结果显示的地方符合实际，需要修改一些SPECpower\_ssj\_config.props和SPECpower\_ssj\_config\_sut.props的值（不修改对结果没影响）

## runpower.bat（红字需要修改）

:: Windows batch file to run ptd in power mode

:: See the Hardware Setup Guide for advanced configurations including GPIB usage

@echo off

echo.

:: Use a full path name for the ptd executable if it is not in the current directory

set PTD=ptd-windows-x86.exe

:: Set NETWORK\_PORT if needed. 8888 is the default used by CCS for the power device

set NETWORK\_PORT=8888

:: Set DEVICE to the power analyzer device you will use (0=dummy device)

:: use the numeric value found in the help output of the ptd executable

set DEVICE=8 一定要是8哦，因为实际不知道这个设备是几，8的话会自己找到。

:: Set DEVICE\_PORT to the serial port you will connect your power analyzer to

set DEVICE\_PORT=COM1

%PTD% -p %NETWORK\_PORT% %DEVICE% %DEVICE\_PORT%

## runtemp.bat

:: Windows batch file to run ptd in temperature mode

:: See the Hardware Setup Guide for advanced configurations including GPIB usage

@echo off

echo.

::

:: NOTE: make sure your sensor is located per Run and Reporting Rules 2.13.3

:: "temperature must be measured no more than 50mm in front of (upwind of)

:: the main airflow inlet of the SUT"

::

echo NOTE: make sure your sensor is located per Run and Reporting Rules 2.13.3

echo "temperature must be measured no more than 50mm in front of (upwind of)

echo the main airflow inlet of the SUT"

echo.

:: Use a full path name for the ptd executable if it is not in the current directory

set PTD=ptd-windows-x86.exe

:: Set NETWORK\_PORT if needed. 8889 is the default used by CCS for the temperature sensor

set NETWORK\_PORT=8889

:: Set DEVICE to the sensor device you will use (1000=dummy temp sensor)

:: use the numeric value found in the help output of the ptd executable

set DEVICE=1000

:: Set DEVICE\_PORT to the serial port you will connect your sensor to

set DEVICE\_PORT=COM1

%PTD% -t -p %NETWORK\_PORT% %DEVICE% %DEVICE\_PORT%

## runCCS.bat

:: Use a full path name for JAVA if the default JRE is not appropriate

set JAVA=java

:: Set JAVAOPTIONS to override any JVM settings when running CCS

set JAVAOPTIONS=

:: Set SSJHOME to the directory containing ssj.jar

set SSJHOME=..\ssj

set CP=.\ccs.jar;.\check.jar;%SSJHOME%\ssj.jar;%SSJHOME%\lib\jfreechart-1.0.10.jar;%SSJHOME%\lib\jcommon-1.0.13.jar

%JAVA% -classpath %CP% %JAVAOPTIONS% org.spec.power.ccs.SpecPowerCCS ccs.props

## Runssj.bat

:: SPECpower\_ssj runssj.bat

::

:: This is an example of what a run script might look like

::

@echo off

:: Set the number of JVMs to run

set JVMS=1

:: Set to TRUE if jvm Director is on this host

set LOCAL\_DIRECTOR=FALSE

set DIRECTOR\_HOST=\_hostname\_or\_IP\_of\_Director\_system\_

:: The SETID is used to identify the descriptive configuration properties

:: that will be used for the system under test. For example, with a SETID

:: of "sut", the descriptive configuration properties will be read from the

:: file SPECpower\_ssj\_config\_sut.props from the Director system.

set SETID=sut

:: Properties file to be passed to Director

set DIRECTOR\_PROPFILE=SPECpower\_ssj.props

:: Benchmark run rules require a list of active OS services be retained for publishable runs.

:: For Windows, this can be accomplished by uncommenting the following line.

:: net start > services.txt

:: Set java options for ssj and director

set JAVAOPTIONS\_SSJ=-Xms1024m -Xmx1024m

set JAVAOPTIONS\_DIRECTOR=-Xms64m -Xmx256m

:: Set JAVA to Java.exe path.

set JAVA=java

:: if JAVA not set, let's find it.

if $%JAVA%$ == $$ goto findjava

goto foundjava

:findjava

:: Note, this algorithm finds the last occurance of java.exe in path.

echo Attempting to find java...

for %%p in ( %PATH% ) do if exist %%p\java.exe set JAVA=%%p\java

if $%JAVA%$ == $$ goto nojava

echo Found java: %JAVA%

:foundjava

@echo on

%JAVA% -version

@echo off

goto stage1

:nojava

echo No java? Please make sure that the path to java is set in your environment!

echo Current PATH: %PATH%

goto egress

:stage1

set SSJJARS=.\ssj.jar;.\check.jar;.\lib\jcommon-1.0.13.jar;.\lib\jfreechart-1.0.10.jar

if "%CLASSPATHPREV%" == $$ set CLASSPATHPREV=%CLASSPATH%

set CLASSPATH=

goto stage2

:stage2

set CLASSPATH=%SSJJARS%;%CLASSPATHPREV%

echo Using CLASSPATH entries:

for %%c in ( %CLASSPATH% ) do echo %%c

:stage3

set I=0

IF %JVMS% == 0 GOTO END

:LOOP

set /a I=%I + 1

@echo on

@echo.

@echo Starting instance %I%

start "SPECpower\_ssj, jvm %I% of %JVMS%" %JAVA% %JAVAOPTIONS\_SSJ% org.spec.power.ssj.SpecPowerSsj -jvmId %I% -numJvms %JVMS% -director %DIRECTOR\_HOST% -setid %SETID%

@echo off

IF %I% == %JVMS% GOTO END

GOTO LOOP

:END

IF NOT %LOCAL\_DIRECTOR% == TRUE GOTO egress

@echo on

@echo.

@echo Starting Director

%JAVA% %JAVAOPTIONS\_DIRECTOR% org.spec.power.ssj.Director -propfile %DIRECTOR\_PROPFILE%

@echo off

goto egress

:egress

## rundirector.bat（基本不用修改）

:: SPECpower\_ssj rundirector.bat

::

:: This is an example of what a run script might look like

::

@echo off

:: Number of hosts

set NUM\_HOSTS=1

:: Set java options for director

set JAVAOPTIONS\_DIRECTOR=-Xms64m -Xmx256m

:: Properties file to be passed to Director

set PROPFILE=SPECpower\_ssj.props

:: Set JAVA to Java.exe path.

set JAVA=java

:: if JAVA not set, let's find it.

if $%JAVA%$ == $$ goto findjava

goto foundjava

:findjava

:: Note, this algorithm finds the last occurance of java.exe in path.

echo Attempting to find java...

for %%p in ( %PATH% ) do if exist %%p\java.exe set JAVA=%%p\java

if $%JAVA%$ == $$ goto nojava

echo Found java: %JAVA%

:foundjava

@echo on

%JAVA% -version

@echo off

goto stage1

:nojava

echo No java? Please make sure that the path to java is set in your environment!

echo Current PATH: %PATH%

goto egress

:stage1

set SSJJARS=.\ssj.jar;.\check.jar;.\lib\jcommon-1.0.13.jar;.\lib\jfreechart-1.0.10.jar

if "%CLASSPATHPREV%" == $$ set CLASSPATHPREV=%CLASSPATH%

set CLASSPATH=

goto stage2

:stage2

set CLASSPATH=%SSJJARS%;%CLASSPATHPREV%

echo Using CLASSPATH entries:

for %%c in ( %CLASSPATH% ) do echo %%c

:stage3

@echo on

@echo.

@echo Starting Director

%JAVA% %JAVAOPTIONS\_DIRECTOR% org.spec.power.ssj.Director -numHosts %NUM\_HOSTS% -propfile %PROPFILE%

@echo off

goto egress

:egress

## SPECpower\_ssj.props（可以从SPECpower\_ssj\_EXPERT.props改名后再改值，一般不修改，一次测试需要75分钟左右）

#########################################################################

# #

# Control parameters for SPECpower\_ssj2008 benchmark #

# #

#########################################################################

# Each parameter is preceded by an expanatory comments section

# containing the following information:

# - a terse description of the parameter

# - the default value (used if the parameter is commented out)

# - optional additional information on meaning or usage

# - optional examples of parameter usage.

#

# This file has 2 sections: changeable parameters and fixed parameters.

# The fixed parameters exist so that you may run tests any way you want,

# however in order to have a valid, reportable run of SPECpower\_ssj2008,

# you must reset them to their original values or comment them out.

#

# Modification of the values of parameters in the fixed parameters

# section or modification of the values in the changeable parameters

# section to values outside the compliant ranges, will result in

# benchmark runs and results which are not compliant with the benchmark

# run rules. Information, data or conclusions produced from runs which

# are not in compliance with the run rules should not be represented as

# SPECpower\_ssj2008 benchmark results either privately or publicly. It

# is a violation of your license agreement to do so. Please consult the

# run rules documentation for additional requirements for compliant

# benchmark runs and publication.

#########################################################################

# #

# Changeable input parameters #

# #

#########################################################################

# The total number of emulated warehouses.

# Default: Number of logical processors on the SUT (obtained by calling

# Runtime.getRuntime.availableProcessors()).

# Compliant: Any change from the default value automatically obtained

# (by uncommenting this parameter) requires justification, refer to the

# additional information below and the run rules documentation.

# Additional: If this property is explicitly set to another value, an

# explanation must be included in the Notes section. Refer to the

# benchmark run rules for details. In multi-JVM runs, the value of

# this parameter is divided by the number of JVM instances to determine

# the number of warehouses used by each individual JVM.

#input.load\_level.number\_warehouses=8

# The number of calibration intervals to perform.

# Default: 3

# Compliant: any integer >= 3 or <= 10

# Additional: The final calibration value is calculated as the average

# of the last 2 of these calibration intervals. A value of at least 3

# is recommended because the first interval usually has a lower result

# than the following intervals.

#input.calibration.interval\_count=3

# The host on which the benchmark director will run.

# Default: localhost

# Compliant: any

# Additional: Only used when the benchmark director is enabled

# (i.e. input.director.enabled=true). Normally the director

# hostname is passed on the command-line (with the -d or -director

# parameter) rather than using this property. Set

# config.director.location in SPECpower\_ssj\_config.props to indicate

# whether the Director ran on the SUT or the Controller system.

#input.director.hostname=localhost

# The port used for synchronization and data communications from all SSJ

# instances and the benchmark director.

# Default: 1500

# Compliant: any valid and available port value

# Additional: Not used when the benchmark director and CCS are disabled

# (i.e. input.director.enabled=false and input.ccs.enabled=false).

# If the director is disabled and ccs is enabled, the port on which the

# workload listens for CCS to connect.

# (i.e. input.director.enabled=false and input.ccs.enabled=true).

#input.status.port=1500

# Connection timeout in seconds for all SSJ connections.

# Default: 300

# Compliant: any non-negative integer

# Additional: A value of 0 indicates no timeout, i.e. wait indefinitely.

# Only used when the benchmark director is enabled (i.e.

# input.director.enabled=true).

#input.director.connect\_timeout=300

# The name of the descriptive properties file.

# Default: null (note this parameter is uncommented by default)

# Compliant: any valid filename

# Additional: Descriptive properties define aspects of the system

# configuration and other conditions that remain constant during the

# benchmark run. The common file name is "SPECpower\_ssj\_config.props.

# On systems where the file separator is "\", use "\\" as the file

# separator here (refer to the third example).

# Examples:

# input.include\_file=SPECpower\_ssj\_config.props

# input.include\_file=/path/to/SPECpower\_ssj\_config.props

# input.include\_file=c:\\path\\to\\SPECpower\_ssj\_config.props

#input.include\_file=SPECpower\_ssj\_config.props

# The directory in which output files are stored.

# Default: results

# Compliant: any valid directory name

# Additional: On systems where the file separator is "\", use "\\" as

# the file separator here (refer to the third example).

# Examples:

# input.output\_directory=results

# input.output\_directory=/path/to/results

# input.output\_directory=c:\\path\\to\\results

#input.output\_directory=results

#########################################################################

# #

# Fixed input parameters #

# #

# Changes to these values will result in a non-compliant run. #

# #

#########################################################################

# The workload name.

# Default: null (note this parameter is uncommented by default)

# Compliant: SPECpower\_ssj

# Additional: Reserved for future work. Changing the parameter value or

# commenting it out will result in benchmark failure to execute.

input.suite=SPECpower\_ssj

# Number of load levels.

# Default: 10

# Compliant: 10

# Additional: Defines the number of load levels to run, including peak

# (100%) but excluding active idle (0%). The default value results

# in a run consisting of 10 load levels (100%, 90%, 80%, ..., 10%) and

# active idle, a total of 11 measurement intervals.

#input.load\_level.count=10

# Benchmark logging level.

# Default: INFO

# Compliant: INFO

# Additional: Used for debugging, possible values in order from lowest

# to highest verbosity are: INFO, SEVERE, WARNING, CONFIG, FINE, FINER,

# FINEST.

#input.log\_level=INFO

# Should a deterministic random seed be used?

# Default: false

# Compliant: false

# Additional: The random seed is used to compute the arrival rate of

# benchmark transactions.

#input.deterministic\_random\_seed=false

# Pre-measurement interval.

# Default: 30

# Compliant: 30

# Additional: Time in seconds running each load point before performance

# and power measurement begins.

#input.load\_level.pre\_measurement\_seconds=30

# Post-measurement interval.

# Default: 30

# Compliant: 30

# Additional: Time in seconds running each load point after performance

# and power measurement has completed.

#input.load\_level.post\_measurement\_seconds=30

# Calibration measurement interval.

# Default: 240

# Compliant: 240

# Additional: Time in seconds that each calibration interval is run.

#input.calibration.length\_seconds=240

# Load level measurement interval.

# Default: 240

# Compliant: 240

# Additional: Time in seconds that each load level interval is run.

#input.load\_level.length\_seconds=240

# Load level delay.

# Default: 10

# Compliant: 10

# Additional: Time in seconds between each load level.

#input.load\_level.delay\_between=10

# Target throughput.

# Default: self-calibrating

# Compliant: self-calibrating

# Additional: Specifying an absolute throughput value will override the

# calibration value. Target throughput for each JVM at each measurement

# point will be based on the specified value. Calibration intervals will

# still be run. Set input.calibration.interval\_count=0 if no calibration

# intervals are desired when specifying this parameter.

#input.load\_level.target\_max\_throughput=100000

# Absolute throughput sequence (input.load\_level.throughput\_sequence).

# Percentage throughput sequence (input.load\_level.percentage\_sequence).

# Default: 10 levels plus active idle from highest to lowest with self-

# calibrating maximum.

# Compliant: the default

# Additional: By default, the benchmark will perform a calibration and

# then run measurements at several fractions of the calibrated max

# throughput. This behavior can be overridden by listing a specific

# sequence of throughputs (input.load\_level.throughput\_sequence) or

# sequence of percentages (input.load\_level.percentage\_sequence) of

# the calibrated maximum throughput (or of the target throughput if

# input.load\_level.target\_max\_throughput is defined, see above). If

# input.load\_level.count is not specified then the number of load

# points defined here will be used; if it is specified then it controls

# the number of load points and this list will be reused until that

# number of points has been measured. The value "0" represents an

# active idle interval. The sequence of throughputs will be executed

# by each JVM; in multi-JVM runs, the overall throughput will be the

# sum of these per-JVM throughputs. Only one of these two parameters may

# be specified.

#input.load\_level.throughput\_sequence=20000 80000 40000 100000 60000

#input.load\_level.percentage\_sequence=20 40 60 80 100 90 70 50 30 10

# Use a single scheduler queue for all warehouses?

# Default: false

# Compliant: false

# Additional: By default a dedicated scheduler is used for each ware-

# house/thread. Setting this parameter to "true" causes a single

# scheduler to be used by all warehouses in a JVM instance, with a

# thread pool size set by input.scheduler.number\_threads.

#input.scheduler.single\_queue=false

# Size of the thread pool for single-scheduler mode.

# Default: Runtime.getRuntime().availableProcessors()

# Compliant: not used in compliant runs

# Additional: Used in single-scheduler mode only.

#input.scheduler.number\_threads=2

# Fine grained transactions contained in each batch of high-level

# transactions.

# Default: 1000

# Compliant: 1000

#input.scheduler.batch\_size=1000

# Maximum scheduled delay in seconds between the starts of consecutive

# transaction batches.

# Default: 10

# Compliant: 10

# Additional: Transaction batch arrival times follow a negative

# exponential distribution. This parameter sets the maximum arrival

# time so that an abnormally long delay will not significantly impact

# benchmark results.

#input.scheduler.max\_arrival\_delay=10

# Create a log file with actual transaction batch arrival rates.

# Default: false

# Compliant: false

# Additional: For debugging purposes.

#input.scheduler.log\_arrival\_rates=false

# Include an active idle measurement interval prior to calibration?

# Default: false

# Compliant: false

# Additional: Include an active idle measurement after the benchmark has

# been initialized but before any calibration or load level intervals

# have taken place.

#input.idle.pre\_calibration=false

# Include an active idle measurement after calibration, before the first

# load level interval?

# Default: false

# Compliant: false

# Additional: Include an active idle measurement after the final

# calibration interval but before any load level intervals have begun.

#input.idle.post\_calibration=false

# Include an active idle measurement after the final load level interval?

# Default: true

# Compliant: true

# Additional: Include an active idle measurement after the final load

# level interval has completed. This is the default and run rules

# compliant placement of the active idle measurement interval.

#input.idle.post\_run=true

# Active idle settle period in seconds.

# Default: 0

# Compliant: 0

# Additional: The settle period can be used to allow the system to

# adjust to an idle state. No measurements are taken during the settle

# period.

#input.idle.settle\_seconds=0

# Active idle measurement interval in seconds.

# Default: 240

# Compliant: 240

# Additional: Time in seconds that the active idle interval is run.

#input.idle.length\_seconds=240

# Use CCS to monitor and coordinate benchmark runs?

# Default: true

# Compliant: true

# Additional: If false, the workload will run in stand-alone mode. If

# true, the workload (or the workload director if enabled) will listen

# on the CCS port (defined below) and will not begin until CCS connects

# to the port and starts the run.

#input.ccs.enabled=true

# Use the workload director?

# Default: true

# Compliant: true

# Additional: Although single-JVM workloads can be run without the

# workload director, compliant runs require use of the director.

#input.director.enabled=true

# Print per-warehouse details in the raw results file?

# Default: false

# Compliant: false

# Additional: Per-warehouse statistics are not used in the final report

# but may be useful when analyzing the behavior of JVMs. When running

# a large number of warehouses, changing this parameter to "true" will

# result in very large raw files. For submission to SPEC, this

# parameter should be set to "false" (the default).

#input.show\_warehouse\_detail=false

# The number of orderlines included in each order.

# Default: 10

# Compliant: 10

#input.orderlines\_per\_order=10

# The number of distinct items in the in-memory database.

# Default: 20000

# Compliant: 20000

#input.override\_itemtable\_size=20000

# Should transaction output be written to the screen?

# Default: false

# Compliant: false

# Additional: Setting this value to true will result in a huge amount of data

# written to the screen, with a large impact on performance. Generally only

# useful for debugging purposes.

#input.screen\_write=false

# The number of customers per district.

# Default: 60

# Compliant: 60

# Additional: Changing this parameter impacts the size of the in-memory

# database.

#input.warehouse\_population=60

# Allow the workload to measure power?

# Default: false

# Compliant: false

# Additional: A compliant run requires power be measured by CCS.

#input.power\_meter.enabled=false

# Hostname of system running the power meter daemon.

# Default: localhost

# Compliant: not used in a compliant run

# Additional: The hostname or IP address of the system running the power

# daemon. Only used if the workload is measuring power (i.e.

# input.power\_meter.enabled=true).

#input.power\_meter.hostname=127.0.0.1

# Port the power meter daemon is listening on.

# Default: 8888

# Compliant: not used in a compliant run

# Additional: Only used if the workload is measuring power (i.e.

# input.power\_meter.enabled=true).

#input.power\_meter.port=8888

# Transaction mix

# Default: new\_order=10, payment=10, order\_status=1, delivery=1

# stock\_level=1, cust\_report=10

# Compliant: default values

#input.transaction\_mix.new\_order=10

#input.transaction\_mix.payment=10

#input.transaction\_mix.order\_status=1

#input.transaction\_mix.delivery=1

#input.transaction\_mix.stock\_level=1

#input.transaction\_mix.cust\_report=10

# Method of measuring individual transaction response time.

# Default: none

# Compliant: none

# Additional: Valid values are "none", "nanos", or "millis". If "none" is

# used, the response time of individual transactions will not be measured.

# With "nanos", System.nanoTime() will be used to measure transaction

# response times. With "millis", System.currentTimeMillis() will be used.

# "nanos" will give the more precise values, but may have significantly more

# overhead than "millis" or "none".

#input.transaction.response\_time=none

## ccs.props(一般不需要修改)

#####################################################

##

## CCS properties file - sample

##

#####################################################

#

# "#" character in column 1 designates a comment line.

#

#####################################################

##

## Global Properties

##

#####################################################

#

# runId is an arbitrary string to describe your test.

# timeout value is in seconds, default is 300.

#

ccs.runId = 0000-0000-0000

ccs.timeout = 300

#####################################################

##

## Data Source Properties

##

#####################################################

#

# There are two categories of data sources.

# 1. Workload

# 2. Power and/or Temperature

#

# 1. Workload - also defines the system upon which the

# Director (in ssj.jar) runs.

# 1.1 Example data source: ccs.wkld = ssj\_dir

# 1.2 "ssj\_dir" is the user-supplied 'tag' that ties to

# detail configuration entries.

# 1.3 Only one "ccs.wkld =" entry is expected

#

# 2. Power and/or Temperature - The identification of a

# particular power and/or temperature measurement device

# 2.1 Examples of define power analyzer

# 2.1.1 Example of single power data source: ccs.ptd = pwr1

# 2.1.2 Example of two power data sources: ccs.ptd = pwr1, pwr2

# 2.1.3 Multiple power analyzers may be specified, each

# will require a data source entry (as above) and a

# matching set of detail configuration entries.

# 2.1.4 In the examples above; "pwr1" and "pwr2" are user selected.

# 2.1.5 Other power analyzer results are logged in

# ssj.nnnn.ccs-log.csv

#

# 2.2 Examples of temperature sensor definition

# 2.2.1 single temperature data source: ccs.ptd = tmp1

# 2.2.2 two temperature sources: ccs.ptd = tmp1, tmp2

# 2.2.3 Multiple temperature sensors may be specified, each

# will require a data source entry and a

# matching set of detail configuration entries.

# 2.2.4 In the examples above; "tmp1" and "tmp2" are user selected.

# 2.2.5 Other temperature sensor readings are logged in

# ssj.nnnn.ccs-log.csv

#

# 2.3 Examples of power and temperature sensor definition

# 2.3.1 Example of one power analyzer and one temperature sensor

# data source: ccs.ptd = pwr1, tmp1

# 2.3.2 Multiple power/temperature devices may be specified, each

# will require a data source entry and a

# matching set of detail configuration entries.

# 2.3.3 In the examples above; "pwr1" and "tmp1" are user selected.

#

# For each data source entry, a detail section-set

# is required to provide:

# Device Type (reserved words, see below)

# Tag (alphanumeric string of users choice, no spaces),

# IP address (name or IP)

# TCP/IP port # (must match port # of other program)

#

# Example detail section-set:

# ## configure connection to Power analyzer 1

# ccs.ptd.pwr1.type = PowerAnalyzer

# ccs.ptd.pwr1.IP = localhost

# ccs.ptd.pwr1.Port = 8888

#

# Notes on examples above:

# "PowerAnalyzer" is the device type,

# --> "type" is always a reserved word.

# "pwr1" is the tag in the example above, user selected.

#

# NOTE:

# For a given data source type, all entries must be on

# one line, for example, two power analyzers, two temp meters:

# ccs.ptd = pwr1, pwr2, pwr3, temp1, temp2

#

# Reserved Words:

# ccs, SSJ, PowerAalyzer, TempSensor,

# wkld, ptd, type, IP, Port, RunID, timeout, VAM.

# Reserved words are not case sensitive.

#

#####################################################

#

# Data Source Entries;

# Three are defined by default.

# 1. the workload "ssj\_dir"

# 2. ptd sources (power and temperature)

# 3. VAM source (Visual Activity Monitor);

# VAM is commented out by default.

#

ccs.wkld = ssj\_dir

ccs.ptd = pwr1, temp1

#ccs.vam = vam1

#####################################################

##

## Detail Section-topology of the systems and software

##

#####################################################

#

# Workload Data Source #################################

#

# Configure the connection to SSJ Director.

# If the director is run on a system other than the CCS

# system, replace "localhost" with the IP address of the

# "director" system. If the SSJ director is started with

# a non-standard port, change the port number here to match.

#

ccs.wkld.ssj\_dir.type = SSJ

ccs.wkld.ssj\_dir.IP = localhost

ccs.wkld.ssj\_dir.Port = 8886

#

# Power Analyzer 1 Data Source ########################

#

# If the PTDaemon is run on a system other than the CCS

# system, replace "localhost" with the name or IP address

# of that system.

# If PTDaemon is started with a TCP/IP port other than the

# default, change the port number below to match the

# port number in the PTDaemon script file.

#

ccs.ptd.pwr1.type = PowerAnalyzer

ccs.ptd.pwr1.IP = localhost

ccs.ptd.pwr1.Port = 8888

#

# Power Analyzer Range Settings ##########################

#

# Select Power analyzers have multiple ranges that can be

# set by the front panel or programmatically.

# Ranges can be set for current (amps) and voltage,

# and can be set or changed for each benchmark load level.

#

# Range settings are sent to the power analyzer,

# only if the device supports programmatic setting.

#

# A range may be specified for each load level.

# The range value should be the maximum of

# the power analyzer's range setting.

# If a range setting is left blank,

# or no setting property is

# present for that meter (or commented out),

# then existing meter settings will be used.

# Some analyzers must be set up on the front panel.

# If range settings are entered at the analyzer,

# those range(s) must be recorded in the

# configuration section below.

# For compliant benchmark runs, and by default,

# the SPECpower\_ssj2008 benchmark generates

# 14 load levels;

# 3 calibration, and 11 graduated levels.

# Ranges for the levels are set by a comma separated

# list that may contain one or more range values.

# The position of a range in the string is a 1:1

# match to the load level.

# Spaces are ignored.

# Comma separators are required except after the

# last value.

# Example 1: Auto for all levels

# ptd.pwr1.current\_range\_settings = auto

# Example 2: 4 amps for calibration and 100%,

# then 2 for all other load levels

# ptd.pwr1.current\_range\_settings = 4, 4, 4, 4, 2

# Example 3; 5 amps for levels down to 80%,

# then 2 amps for levels 70% and below:

# ptd.pwr1.current\_range\_settings= 5, 5, 5, 5, 5, 5, 2

# The load levels are in this order:

# cal1, cal2 cal3, 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, idle

ccs.ptd.pwr1.current\_range\_settings=auto

# Given voltage is expected to be constant for a given

# run of the benchmark, voltage\_range need only be set

# for the first level; for example:

# ccs.ptd.pwr1.voltage\_range\_settings = 120

ccs.ptd.pwr1.voltage\_range\_settings=auto

#

# Temperature Sensor 1 connection ########################

#

# If the PTDaemon is run on a system other than the CCS

# system, replace "localhost" with the name or IP address

# of the system running the PTDaemon.

# If PTDaemon is started with a TCP/IP port other than the

# default, change the port number below to match the

# port number in the PTDaemon script file.

ccs.ptd.temp1.type = TempSensor

ccs.ptd.temp1.IP = localhost

ccs.ptd.temp1.Port = 8889

#

# Power Analyzer 2 Data Source ########################

#

# See the comments for power analyzer 1 above. Note that

# for these entries to be used, "pwr2" must be added to

# the ccs.ptd data sources above.

#

#ccs.ptd.pwr2.type = PowerAnalyzer

#ccs.ptd.pwr2.IP = localhost

#ccs.ptd.pwr2.Port = 8890

#

# VAM Connection Configuration #######################

#

# If VAM is run on a system other than the CCS

# system, replace "localhost" with the host name or

# or IP address of that system.

# The port number below is the default for one of three

# potential instances of CCS from which VAM can display data.

#

# If this instance of CCS is the second or third for VAM,

# the port number below must match that of the port number

# in the vam.props file.

#

ccs.vam.vam1.type = VAM

ccs.vam.vam1.IP = localhost

ccs.vam.vam1.Port = 8905

#

# To enable synchronization of multiple data sources read by

# VAM, the "syncStart" parameter must be set to True

# in the ccs.props file for each instance of CCS,

# and in the vam.props file.

# SyncStart causes CCS and therefore all Directors to

# wait until all are ready.

#

ccs.vam.syncStart = false

#####################################################

##

## CCS Configuration Info Sections

##

#####################################################

#

# The following properties will be used by the 'reporter'.

#

# Change as needed to reflect the details of the

# equipment in your setup.

#

#####################################################

##

## CCS system (platform) hardware and software info

##

#####################################################

#

# Company which sells/manufactures the hardware.

#

ccs.config.hw.vendor=IQ2 Corporation

#

# Model name of CCS platform / system

#

ccs.config.hw.model=Meridian 38

#

# Make and model of the processor(s) in the CCS system.

#

ccs.config.hw.cpu=Saturn ULP

#

# Basic info on the processor, e.g. single/dual core, clock speed,

# cache sizes.

#

ccs.config.hw.cpu.characteristics=3.2 GHz, dual core, 2M L2 cache, 4M L3 cache

#

# The amount of main memory in the CCS system in GigaBytes.

#

ccs.config.hw.memory.gb=1.5

#

# CCS host Operating System

#

ccs.config.sw.os=SPEC Open Doors 2006 F500

#

# JVM vendor name and version used on CCS system.

#

ccs.config.sw.jvm.vendor=IQ2 Corporation

ccs.config.sw.jvm.version=IQ2 Java VM v1.0

#####################################################

##

## Basic info on Power Analyzer

##

#####################################################

#

# Power analyzer manufacturer.

#

ptd.pwr1.config.analyzer.vendor=Energy Minder, Inc.

#

# Model name of the power analyzer - usually on front panel.

#

ptd.pwr1.config.analyzer.model=EM1000+ USB

#

# Serial number of the power analyzer.

#

ptd.pwr1.config.analyzer.serial=ser001122

#

# Data connection from power analzer to the

# host system, e.g. USB, RS-232, GPIB, LAN.

# If a USB to serial converter is used, the brand

# and model should be entered in the Notes.

#

ptd.pwr1.config.analyzer.connectivity=USB2

#

# Power Analyzer Calibration and Certification

#

# Name of the national metrology institute or organization

# which specifies calibration specs and standards

#

ptd.pwr1.config.calibration.institute=NIST

#

# Name of the organization that performed the power analyzer calibration.

# Could be the analyzer manufacturer, a third party company, or an

# organization within your own company.

#

ptd.pwr1.config.calibration.accredited\_by=IQ2 Calibration Laboratory

#

# The number which uniquely identifies this device calibration event.

# May appear on the certification certificate or on a sticker applied

# to the power analyzer. The format of this number is specified by

# the calibration institute.

#

ptd.pwr1.config.calibration.label=N-32768

#

# Date of calibration, from the calibration paperwork or sticker.

# Day-Month-Year

#

ptd.pwr1.config.calibration.date=1-Jan-2007

#

# Manufacturer and model number of the computer system to which the power

# analyzer data cable is connected, and the operating system of that

# computer. This is the system specified by ccs.ptd.pwr1.IP above. It

# may be the same as the CCS system.

#

ptd.pwr1.config.ptd.system=same as CCS

ptd.pwr1.config.ptd.os=same as CCS

#

# input\_connection: input used to connect the load, if several options are

# available, or "Default" if not.

# current\_range: value of power analyzer current range configuration if

# PTD cannot automatically determine the range. Or "Auto" if none

# voltage\_range: value of power analyzer voltage range configuration if

# PTD cannot automatically determin ther range. Or "Auto" if none

#

ptd.pwr1.config.analyzer.input\_connection = Default

ptd.pwr1.config.analyzer.current\_range = Auto

ptd.pwr1.config.analyzer.voltage\_range = Auto

#

# The 'analyzer.setup\_description' property describes

# which device or devices are measured for this instance

# of a power analyzer and the PTDaemon.

#

# This should be a textual description

# of the device or devices measured.

# Example:

# analyzer.setup\_description= SUT Power Supplies 1 and 2

#

# The default is a null string (blank)

#

ptd.pwr1.config.analyzer.setup\_description=

#####################################################

##

## Basic info on Temperature Sensor

##

#####################################################

#

# Temperature sensor manufacturer.

#

ptd.temp1.config.sensor.vendor=EnviroBits, Inc.

#

# Temperature sensor model name.

#

ptd.temp1.config.sensor.model=Tsense USB-2

#

# Name of and version number of temperature sensor driver.

#

ptd.temp1.config.sensor.driver=1.2.3.4

#

# Data connection from temperature sensor to the

# host system, e.g. USB, RS-232, GPIB, LAN.

# If a USB to serial converter is used, the brand

# and model should be entered in the Notes.

#

ptd.temp1.config.sensor.connectivity=USB

#

# Manufacturer and model number of the computer system to which the

# temperature sensor data cable is connected, and the operating system

# of that computer. This is the system specified by ccs.ptd.temp1.IP

# above. It may be the same as the CCS system.

#

ptd.temp1.config.ptd.system=same as CCS

ptd.temp1.config.ptd.os=same as CCS

#

#

# The 'sensor.setup\_description' property describes

# which device or devices are measured for this instance

# of a temperature sensor and the PTDaemon.

#

# This should be a textual description

# of the device or devices measured and the

# approximate location of the temperature sensor.

# Example:

# sensor.setup\_description= 5 mm in front of SUT main intake

#

# The default is a null string (blank)

#

ptd.temp1.config.sensor.setup\_description=

#

#####################################################

##

## End of CCS properties

##

#####################################################

## SPECpower\_ssj\_config.props（可以不修改）

#

# SPECpower\_ssj2008 properties file

#

# This is a SAMPLE file which you can use to specify characteristics of

# a test run. Details of the system or systems under test are defined

# separately in SPECpower\_ssj\_config\_<set\_id>.props files.

# Director Location: SUT or Controller or Other (describe in config.notes)

config.director.location=Controller

# Any additional testbed configuration notes. (Compliant XHTML fragments allowed.)

config.notes=None

#########################################################################

#

# Tester information

#

#########################################################################

# The next four properties specify the Line Voltage Standard

# The voltage (integer: 100, 110, 120, 208, 220, 230 -- if other, describe

# in config.notes)

config.line.standard.voltage=120

# Frequency of the voltage in Hz (50, 60 -- must be an integer)

config.line.standard.frequency=60

# Phases used to provide the voltage (must be an integer)

config.line.standard.phase=1

# Wires used to provide the voltage (must be an integer)

config.line.standard.wires=2

# Elevation (meters)

config.test.elevation=132

# The company sponsoring the test

config.test.sponsor=SPEC Company

# The testing company's SPEC license number

config.test.spec\_license=0

# The company that ran the benchmark

config.test.tested\_by=SPEC Company

# The person who ran and submitted this result (optional -- name does not go on

# public pages)

config.testx.tested\_by.name=J.D.C. Powers Sr.

# The physical location of the SUT (city, state/province, country)

config.test.location=Warrenton, VA, USA

# A web page where people within the aforementioned company might get

# more information about this result.

config.testx.internal\_reference=Run 1234

config.testx.internal\_reference.url=http://internal.spec.org/specpubs/jan2008/run1234

#########################################################################

#

# Shared Hardware (this section is ignored if there is only a single node)

#

#########################################################################

# Cabinet/Housing/Enclosure

config.shared.enclosure=A really big cardboard box

# Full SUT form factor (including all nodes and any shared hardware). For

# rack-mounted systems, specify the number of rack units. For other types

# of enclosures, specify "Tower" or "Other".

# For example:

# config.shared.form\_factor=4U

# config.shared.form\_factor=Tower

config.shared.form\_factor=0U

# Power Supply Details (N/A if there are no shared power supplies)

config.shared.psu.description=N/A

# Power Supplies Installed (0 if there are no shared power supplies)

config.shared.psu.installed=0

# Power Supply Rating (0 if there are no shared power supplies)

config.shared.psu.rating=0

# Network Switch

config.shared.network.switch=None

# Network Switch Description

config.shared.network.switch.description=None

# KVM Switch / Management Blade

config.shared.kvm=None

# KVM Switch details

config.shared.kvm.description=N/A

# Other Common Hardware

config.shared.other=None

# Common Hardware Comments

config.shared.comment=None

## SPECpower\_ssj\_config\_sut.props（可以不修改）

#

# SPECpower\_ssj2008\_config\_sut properties file

#

# This is a SAMPLE file which you can use to specify characteristics of

# a particular system under test (or set of identical systems). You can reuse

# this file repeatedly, and have a version for each system setup you use. You

# should edit the reporting fields appropriately. All of this can still be

# edited in the output properties file after you run the test, but putting the

# values in here can save you some typing for attributes which do not change

# from test to test.

# A description of the set.

config.set.description=System Under Test

config.set.comment=Here is a comment for this set.

#########################################################################

#

# System Under Test hardware

#

#########################################################################

# Hardware availability date

config.hw.available=Jan-2008

# Single Supplier or Parts Built

config.hw.system\_source=Single Supplier

# System Designation (Server or Personal System)

config.hw.system\_designation=Server

# Power Provisioning (Line-powered or Battery-powered)

config.hw.power\_provisioning=Line-powered

# Company which sells the hardware

config.hw.vendor=SPEC

# Home page for company which sells the hardware

config.hw.vendor.url=http://www.spec.org

# What type of system was used

config.hw.model=System EM65

# Form factor for this system. In multi-node configurations, this is the

# form factor for a single node. For rack-mounted systems, specify the

# number of rack units. For blades, specify "Blade". For other types of

# systems, specify "Tower" or "Other".

# For example:

# config.hw.form\_factor=4U

# config.hw.form\_factor=Tower

config.hw.form\_factor=0U

# What type of processor(s) the system had

config.hw.cpu=Tacoma Spi

# Technical characteristics to help identify the processor

config.hw.cpu.characteristics=Tri-Core, 1.3GHz, 37MB L3 Cache

# MegaHertz rating of the chip (integer).

config.hw.cpu.mhz=1300

# Number of chips in the system as configured

config.hw.cpu.chips=2

# Number of cores in the system as configured

config.hw.cpu.cores=6

# Number of cores on each chip

config.hw.cpu.cores\_per\_chip=3

# Hardware threads per core

config.hw.cpu.threads\_per\_core=2

# The number of CPU chips that can be ordered for this system model

config.hw.cpu.orderable=1,2 chips

# Amount of physical memory in the system, in Gigabytes. (possibly fractional)

config.hw.memory.gb=2.5

# Number and size of memory modules used

config.hw.memory.dimms=5 x 512 MB

# Detailed description of the system main memory technology, sufficient for

# identifying the memory used in this test. The preferred format is described at

# http://www.spec.org/power\_ssj2008/docs/SPECpower\_ssj2008-Result\_File\_Fields.html#MemDetails

config.hw.memory.description=2GB 2Rx4 PC2-5300F ECC CL5; slots 1, 2, 3, 7, and 8 populated

# Description (size and organization) of the CPU's primary cache. This cache

# is also referred to as "L1 cache".

config.hw.cache.primary=64 KB I + 64 KB D on chip per core

# Description (size and organization) of the CPU's secondary cache. This cache

# is also referred to as "L2 cache".

config.hw.cache.secondary=1 MB I+D on chip per chip

# Description (size and organization) of the CPU's tertiary, or "L3" cache.

config.hw.cache.tertiary=37 MB I+D off chip per chip

# Description (size and organization) of any other levels of cache memory.

config.hw.cache.other=None

# A description of the disk drive(s) (size, type, and RAID level if any)

# included in the system.

config.hw.disk=4 x 73GB 15K RPM SAS

# The manufacturer and model number of the controller used to drive the

# disk(s).

config.hw.disk.controller=Integrated SAS controller

# Number and type of Network Interface Cards (NICs) installed

config.hw.network.controller=4 x SuperFast NIC

# NICs enabled in Firmware

config.hw.network.controller.enabled.firmware=2

# NICs enabled in the OS

config.hw.network.controller.enabled.os=2

# NICs which are connected

config.hw.network.controller.connected=1

# Actual network speed (Mbit)

config.hw.network.speed=100

# Number of power supplies installed in the SUT (0 if there is no private

# power supply (i.e. there is a shared power supply)

config.hw.psu.installed=1

# Power rating for each installed power supply (integer, in watts) (0 if

# there is no private power supply)

config.hw.psu.rating=750

# PSU Brand and Model (N/A if there is no private power supply)

config.hw.psu.description=PS-0815

# Keyboard: USB / PS2 / KVM / None

config.hw.keyboard=PS2

# Mouse: USB / PS2 / KVM / None

config.hw.mouse=USB

# Monitor: Yes / KVM / None

config.hw.monitor=Yes

# Optical Drives (CD, DVD): Yes / None

config.hw.optical=Yes

# Any other equipment added to achieve the reported power and performance

# scores.

config.hw.other=ReallyFast Java Accelerator Card

#########################################################################

#

# System Under Test software

#

#########################################################################

# The date (month and year) by which all system software will be shipping and

# generally available to the public

config.sw.available=Jan-2008

# The company that makes the JVM software

config.sw.jvm.vendor=SPEC

# Home page for the company that makes the JVM software

config.sw.jvm.vendor.url=http://www.spec.org

# Name and version of the JVM software product

config.sw.jvm.version=SPEC Java VM 5.0 (build 1.2.3.4-tricore 20071111)

# JVM command-line options used when invoking the benchmark.

config.sw.jvm.options=-Xms3500m -Xmx3500m -XrunFast -XconsumeLessPower -Xmn3100m

# Commands used to configure affinity for each JVM (possibly "None")

# For example, with 4 JVMs on a 16-core Windows system, the following command might be used:

config.sw.jvm.affinity=start /affinity [0F,F0,F00,F000]

# Memory initially allocated for the JVM heap (MB). Values such as

# "Unlimited" or "Dynamic" are allowed for JVMs that adjust automatically.

config.sw.jvm.heap.initial=3500

# Maximum memory allowed for the JVM heap (MB). Values such as "Unlimited"

# or "Dynamic" are allowed for JVMs that adjust automatically.

config.sw.jvm.heap.max=3500

#Is the JVM 32- or 64-bit? (or some other number?)

config.sw.jvm.bitness=64

# Operating system name

config.sw.os=SPEC Open Doors 2006 F500

# Operating system version

config.sw.os.version=1.2.3.4 (64-bit)

# The type of the filesystem used to contain the run directories

config.sw.filesystem=SPECFS

# Shows whether power management features of the SUT were enabled or disabled.

config.sw.power\_management=Enabled (see SUT Notes)

# System bootstrap software or firmware, often referred to as BIOS (for systems

# where this software is or may be installed at the end-user site). May be "none".

config.sw.boot\_firmware.version=1.2.3.4

# Bootstrap software or firmware settings. May be "none". If there are many

# settings or long descriptions, use "See SUT Notes" and include the details

# in config.sut.notes.

config.sw.boot\_firmware.settings=none

# Baseboard management controller software or firmware, often referred to as BMC

# (for systems where this software is or may be installed at the end-user site).

# May be "none".

config.sw.mgmt\_firmware.version=1.2.3.4

# Baseboard management controller software or firmware settings. May be

# "none". If there are many settings or long descriptions, use "See SUT

# Notes" and include the details in config.sut.notes.

config.sw.mgmt\_firmware.settings=none

# Any additional software that is needed to reproduce these power and

# performance results.

config.sw.other=None

# Additional tuning of the SUT hardware or software needed to achieve the

# result. (Compliant XHTML fragments allowed.)

config.sut.notes=<ul><li>Each JVM instance was affinitized to a socket.</li><li>Using the local security settings console, &quot;lock pages in memory&quot; was enabled for the user running the benchmark.</li><li>HT Technology disabled in BIOS.</li></ul>