



Performance Tuning ScrumWorks® Pro Server

Summary

Audience: ScrumWorks Pro Server Systems Administrators

This document provides a summary of performance tuning measures for ScrumWorks Pro Servers with large data sets and/or high volume use.

In general, performance is affected by two main considerations:

1. The amount of data in the system (the number of Products and the number of PBIs and Tasks per Product); and
2. The number of users accessing the system simultaneously.

Since customer server hardware and software configurations vary greatly and performance tuning the ScrumWorks Pro Server results in increased consumption of available server resources, ScrumWorks Pro Server is shipped by default with moderate performance capabilities to satisfy average usage scenarios with modest hardware specifications. ScrumWorks Pro Server must therefore undergo configuration modifications for heavy use and large data sets.

Benchmark: Using the guide below in conjunction with recommended hardware configurations, the ScrumWorks Pro Team has successfully load tested the current stable ScrumWorks Pro release with extremely large data sets (> 10,000 PBIs and Tasks) and 300 simultaneous users. However, your results may vary since simulated load tests do not accurately reflect real users.

This guide is divided into four sections, each of which addresses different ways to increase performance:

- **RAM Allocation Configuration**
- **Database Connection Pool Configuration**
- **Multi-Core/CPU Processor Configuration**
- **Linux-Specific Performance Enhancements**

Please refer to the Recommended Hardware Guide for specific recommendations on hardware to house ScrumWorks Pro Servers:

http://danube.com/docs/scrumworks/pro/latest/ScrumWorks_Server_Hardware_Recommendations.pdf



RAM Allocation Configuration

By default the ScrumWorks Pro Server configuration limits RAM use to 768 MB. This value can be adjusted to make use of additional RAM. This particular configuration directive is important for those customers using the default Hypersonic database. Because Hypersonic is an “in-memory” database, large data sets may cause high CPU loads with allocated RAM resources are too low. Further, we recommend using 64-bit server platforms and the 64-bit version of the JVM to fully take advantage of the configuration option since the 32-bit JVM cannot be allocated more than 1,536 MB of RAM.

Warning: the server will fail to start if memory is over-allocated (depending upon hardware and other applications in use on the machine). It is generally recommended that no more than 70% of free physical RAM be allocated to the ScrumWorks Pro server to avoid over-allocation.

The method described in this document require ScrumWorks Pro release 2.1 and above.

Linux / Mac OS X server platform:

This limit can be increased through the run.conf file found in the 'bin' directory. This option is only available under a Unix platform (Linux and Mac OS X)

Note: If you have a 32-bit JVM installed the upper limit is 1536m, 64-bit may be as high as 4096m.

1. Navigate to INSTALLDIR/bin
2. Edit the run.conf file
3. Locate the line: `JAVA_OPTS = "-server -Xms256m -Xmx768m ..."`
4. Change the "768m" and "256m" portions after -Xms and -Xmx to reflect how much RAM you wish to allocate to ScrumWorks Pro Server. Note, your server may have a different value by default like "512m".

Example: If you wish to allocate 1.5 GB of RAM to the ScrumWorks Pro Server you would change the line to read:

```
JAVA_OPTS = "-server -Xms1536m -Xmx1536m"...
```

5. Restart the ScrumWorks Pro Server for the configuration changes to take effect.

**Windows server platform:**

For ScrumWorks 3.0+:

The RAM can be increased by using the memory.exe application located in the INSTALLDIR\utils\ directory.

Select the memory you want to allocate to ScrumWorks Pro Server and choose the Update button, then select "Close". Please note that on 32-bit platforms (or 32-bit JVMs) the upper limit is "1536m". The settings will take effect after the server has been restarted.

RAM Allocation Limitations on Windows

Quoted from Oracle's Java FAQ:

http://www.oracle.com/technetwork/java/hotspotfaq-138619.html#gc_heap_32bit

"Why can't I get a larger heap with the 32-bit JVM?"

The maximum theoretical heap limit for the 32-bit JVM is 4G. Due to various additional constraints such as available swap, kernel address space usage, memory fragmentation, and VM overhead, in practice the limit can be much lower. On most modern 32-bit Windows systems the maximum heap size will range from 1.4G to 1.6G. On 32-bit Solaris kernels the address space is limited to 2G. On 64-bit operating systems running the 32-bit VM, the max heap size can be higher, approaching 4G on many Solaris systems."



Database Connection Pool Sizes

ScrumWorks Pro Server can be configured to utilize either the Hypersonic or MySQL database systems. Under both circumstances, the default database connection pool size may be insufficient for heavy simultaneous use. If this particular issue is encountered, the server.log will contain the following error: No ManagedConnections available

Since available database resources vary case-by-case, please consult your database administrator in determining a target database pool size before changing your configuration.

MySQL Database Pool Size Configuration

1. Shut down your ScrumWorks Pro server
2. Edit the file
`INSTALL_DIR/server/scrumworks/deploy/scrumworks-mysql-ds.xml`
3. Find the option `<max-pool-size>50</max-pool-size>`
4. Change "50" to the value supplied by your MySQL database administrator
5. Save your changes
6. Restart your ScrumWorks Pro server

Hypersonic Database Pool Size Configuration

1. Shut down your ScrumWorks Pro server
2. Edit the file
`INSTALL_DIR/server/scrumworks/deploy/scrumworks-hsqldb-ds.xml`
3. Find the option `<max-pool-size>100</max-pool-size>`
4. Change "100" to a new value based on simultaneous use projections and server resources
5. Save your changes
6. Restart your ScrumWorks Pro server

Multi-Core/CPU Processor Configuration

On server hardware with **four cores (total across all CPUs)** or more, a configuration change can be made to take advantage of parallel processing. While this configuration can be made to dual core servers, the empirical evidence is that the benefits are outweighed by the overhead of parallelization.

Linux servers

1. Edit the file:
`INSTALLDIR/bin/run.conf`
2. Locate the section:
`# Linux servers are sometimes headless`
`JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true"`
3. Modify it to include `-server -XX:+UseParallelGC`. For example:
`JAVA_OPTS="$JAVA_OPTS -server -XX:+UseParallelGC -Djava.awt.headless=true"`

Windows Servers

1. Start > Run >
2. At the prompt type:
`regedt32`
3. Select OK
4. Edit the registry key:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ScrumWorks Pro Server [Server Version]\Parameters\JVM Option Count`

Increase the value by 1
5. Add a new String Value:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ScrumWorks Pro Server [Server Version]\Parameters\JVM Option Number [The value of JVM Option Count minus 1]`

Set the value to `"-XX:+UseParallelGC"`
6. After changing this, you will have to restart the ScrumWorks service.
Control Panel > Administrative Tools > Services > select the ScrumWorks service and choose Restart.



Linux-Specific Performance Enhancements

Headless Servers

On Linux machines performance can be enhanced by modifying the following line in your INSTALLDIR/bin/run.conf file:

```
# Linux servers are sometimes headless
```

```
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true"
```

Change this line to include -server:

```
JAVA_OPTS="$JAVA_OPTS -server -Djava.awt.headless=true"
```