Tuning Recommendations for SAS Web Application Server-SAS Midtier Tuning Series (IV)

Option	Component	Default Value	Recommended Value	Desc.
The following options can be modified in the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\conf\server.xml** file				
maxThreads Option	tomcatThreadPool Executor	300	300–1024	specifies the number of threads in the executor thread pool that is used to process incoming requests. If you have multiple instances of SAS Web Application Server, make the same changes in each of the files for all the servers that you want to tune
maxPoolSize Option	Various JDBC Resources, such as sas/jdbc/SharedServices	100	50–512	Specifies the maximum number of pooled connections to the database that is associated with a given data source reference **Additional Information**: When the pool becomes exhausted, the SAS web applications log files (for example, SASWIPServices9.4.log and SASPrincipalServices9.4.log) might contain messages like the following example: Could not get JDBC Connection; nested exception is com.atomikos.jdbc.AtomikosSQLException: Connection pool exhausted - try increasing 'maxPoolSize' and/or 'borrowConnectionTimeout' on the DataSourceBean. Also, the configuration of the database server might need to be modified, along

Option	Component	Default Value	Recommended Value	Desc.
				with this setting. For example, the default configuration of the SAS Web Infrastructure Platform Data Server allows a maximum of 256 connections. You can change the max_connections setting to be the same value as the sum of the maxPoolSize option value across all data sources on all server instances. If you change the value, a system restart is required. For more information about the max_connections setting,
Following option can be added to the ***SAS-configuration-directory*\Levn\Web\WebAppServer\SASServer*n_m*\lib\jta.properties file**:				
com.atomikos.icatch.checkpoint_interval Option	Atomikos transaction manager	500	50	Specifies the interval between transaction manager checkpoints. During checkpoints, completed transactions are removed from the recovery log. More frequent checkpoints can reduce the contention on the log file across threads in high volume usage scenarios