

Advanced Configuration and Performance Tuning

Introduction

There are a number of parameters within GreenLight's configuration files that can be tuned to improve performance or simply adapt the default configuration to increasing load.

Here is a quick overview of the related configuration files:

- **ramkin-config.properties:** advanced application level settings
- **quartz.properties:** scheduler (job handler) configuration
- **tomcat.properties:** settings and configuration parameters for the web server
- **postgresql.conf:** configuration of the PostgreSQL database server

Editing Files

One way to edit these files is connecting to the system via [Putty](#) and using a text editor like "vim" or "nano" (for people who don't have experience with vi). An alternative is opening the virtual machine console and logging in with the "admin" user. Within the graphical interface, open a console / terminal window and enter "sudo thunar". You can now navigate in an explorer like interface to the file locations and edit them with double click using a graphical text editor.

ramkin-config.properties

Location

/opt/panagenda/ramkin-config.properties

Notable Parameters

Property	Description
greenlight.notes.id.filename	Lotus Notes ID file (filename only)
greenlight.notes.id.password	Lotus Notes ID password (unencoded)
greenlight.heartbeat.filename	Filename of the GreenLight heartbeat file
greenlight.scheduler.config.file	Filename of the scheduler (quartz) config file. Must start with 'file:/' eg. 'file:/D:\panagenda\GreenLight\quartz.properties' or 'file:/opt/panagenda/quartz.properties'.

quartz.properties

Location

/opt/panagenda/quartz.properties

Notable Parameters

Parameter	Description
org.quartz.threadPool.threadCount	Number of worker threads for sensor measurements. Default value is 10, increase only in small steps.
org.quartz.jobStore.misfireThreshold	Time in milliseconds to wait for a free worker thread if none is available at schedule measurement time. Default value is 60000.

Additional Information

For details on quartz properties see <http://www.quartz-scheduler.org/docs/configuration/>

64bit Appliance - postgresql.conf

Location

/opt/postgres/conf/postgresql.conf

recommended settings

4 GB Memory

Parameter	Value
shared_buffers	1GB
work_mem	24MB
effective_cache_size	2GB
maintenance_work_mem	128MB
wal_buffers	-1
checkpoint_segments	32
checkpoint_completion_target	0.8

8 GB Memory

Parameter	Value
shared_buffers	2GB
work_mem	48MB
effective_cache_size	4GB
maintenance_work_mem	256MB
wal_buffers	32
checkpoint_segments	64
checkpoint_completion_target	0.8

64bit Appliance - tomcat7

Location
/etc/default/tomcat7

recommended settings

4 GB Memory

JAVA_OPTS="-server -Djava.awt.headless=true -Xmx1024m -XX:MaxPermSize=128m -XX:+UseConcMarkSweepGC"

8 GB Memory

JAVA_OPTS="-server -Djava.awt.headless=true -Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:InitialCodeCacheSize=64m -XX:ReservedCodeCacheSize=64m -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled"

***** old 32 Bit Appliance *****

32bit Appliance - tomcat.properties

Location

/opt/panagenda/tomcat.properties

Notable Parameters

Parameter	Description
STATIC_XMS	Minimum memory heap size for Tomcat's jvm. The default value is calculated depending on the machine's memory. This value overrides that detection. Possible values are (e.g.): 256m, 512m, 1g
STATIC_XMX	Maximum memory heap size for Tomcat's jvm. The default value is calculated depending on the machine's memory. This value overrides that detection. Possible values are (e.g.): 256m, 512m, 1g
ADD_JAVA_OPTS	Additional java options for Tomcat's jvm. Example value (e.g.): -XX:+UseParallelGC

Additional Information

If this file should not exist copy/paste the existing file "tomcat.properties.default" to create it. Note: these values should only be increased, if a corresponding amount of physical memory is available. At least 512MB to 768MB have to be kept free for other processes like the database server or the operating system itself.

32bit Appliance - postgresql.conf

Location

/opt/postgres/conf/postgresql/8.3/main/postgresql.conf

Notable Parameters

Parameter	Description
shared_buffers	<p>Sets the amount of memory the database server uses for shared memory buffers. The default value is 96MB, but on systems with more than 1GB of memory it should be raised to 10-15% of your total RAM. Larger settings for shared_buffers usually require a corresponding increase in checkpoint_segments, in order to spread out the process of writing large quantities of new or changed data over a longer period of time.</p> <p>If this value is increased above 100MB the corresponding value in the Linux OS has to be adapted as well. The file <code>/etc/sysctl.conf</code> holds system wide configuration parameters for shared memory minimums and maximums. Edit it and adapt <code>kernel.shmmax</code> and <code>kernel.shmall</code> (both specified in bytes) to accommodate the new db settings. Example: if shared_buffers is raised to 128MB a value of 201326592 (196MB) is recommended in <code>kernel.shmmax</code>.</p>
checkpoint_segments	Maximum number of log file segments between automatic WAL checkpoints (each segment is normally 16 megabytes). The default value is 8 and should only be increased slowly.
effective_cache_size	Should be set to an estimate of how much memory is available for disk caching by the operating system and within the database itself, after taking into account what's used by the OS itself and other applications. This is a guideline for how much memory you expect to be available in the OS and PostgreSQL buffer caches, not an allocation! This value is used only by the PostgreSQL query planner to figure out whether plans it's considering would be expected to fit in RAM or not. If it's set too low, indexes may not be used for executing queries the way you'd expect.

There are a lot more configuration parameters for the database system, and we have only listed a fraction of those here. For more details see the annotations in the config file itself and for detailed explanations in the PostgreSQL [online documentation](#) and the [tuning section](#) of the PostgreSQL Wiki.