Sawmill Best Practice Guide

Essential reading for anyone installing Sawmill

SAWMILL ANALYTICS Mulberry House Swindon UK SN4 9LZ +44 (0)845 250 4470 sales@sawmill.co.uk www.sawmill.co.uk

Log File Management

Log files are your company's asset and irreplaceable. We strongly recommend that you retain your historical log files for as long as possible. If space is a concern consider compressing your log files. Log files typically compress down very nicely to anywhere between 90-98% of their original size. Sawmill is also capable of natively reading compressed files so there's no need to spend time decompressing files to process them with Sawmill. Compressed log files also reduce bandwidth if you are reading log files remotely, or moving files between remote servers. The most important reason to keep log files is to be able to rebuild your Sawmill database in the event of a catastrophic failure. If you retain your historical log files recovering a database is always possible, and it's a simple operation to rebuild your database. If you choose to follow only one of the best practices and guidelines, make the choice to retain your historical log data. Historical logs are cheap insurance (and typically use very little disk space when compressed) and having them will save you from data loss if you ever need to recover a database from the ground up.

Backup Your Database

We strongly recommend that if you build larger datasets that you create a backup plan for your Sawmill databases. Having a backup of your database will save you many hours (or even days) of reprocessing logs and rebuilding your database in the event of a database corruption or catastrophic failure. Though database corruptions or failures are infrequent, they can happen, and if these types of situations do occur you'll be glad you had a backup of your Sawmill databases so that you can be back up and running quickly.

The Sawmill internal database is a flat file database and can be backed up by most any commonly available backup software. It's important that you schedule backups when the Sawmill database is not actively updating. If you use an external database like MySQL, Oracle or MS SQL check with your Database Administrator about what options exist for backing up or restoring your Sawmill database.

Both the Sawmill internal database, and external databases created by Sawmill can also be exported to plain text files manually, and then compressed, and later decompressed and reloaded manually if no better backup facility is available.

Sawmill Scheduled Tasks

Sawmill's built-in scheduler is very powerful and very flexible and allows you to setup and run many types of operations in an unattended manner. With this power it's possible to quickly run into trouble and overwhelm your server if you set up too many concurrent tasks. We recommend that you do not schedule multiple database builds or multiple database updates concurrently. Sawmill tasks use a large amount of resources (memory, processor, disk space and disk access) on your server when you are processing and building large datasets. To avoid issues it's best to schedule large operations consecutively, even on systems with multiple processors.

Disable Antivirus Active Scanning of the Sawmill Installation Directory

Active-scanning antivirus software (and automatic disk defragmentation software, or indexing software), which runs continually and monitors all files on the disk, can interfere with Sawmill's operation. This software may also incorrectly conclude that Sawmill's database files contain viruses (they don't), causing it to modify or delete the files, resulting in database corruption. This can cause log processing, database building and report generations to fail. This type of software also considerably slows down Sawmill processing and reporting. In very extreme cases making it as much as 20 times slower.

So, you must disable scanning of the directory where Sawmill is installed, or where Sawmill databases are located in order for Sawmill to operate properly. It is not necessary to disable the active scanning software entirely; just configure active scanning so it excludes the Sawmill installation directory and subdirectories and files. Most antivirus products can be configured to exclude scanning of certain directories.

Best Practices for setting-up your Server

Operating System and CPU Platform System requirements

Sawmill will run on any platform, but we always recommend x64 (64-bit) Linux. You can use any distribution, but Red Hat Enterprise Linux is a good choice for maximum speed, compatibility and stability. On other distributions it may be necessary to build Sawmill from source code. Other x64 operating systems are also reasonable choices, including x64 Windows, x64 Mac OS, x64 FreeBSD, and x64 Solaris. CPU architectures other than x64 will work, but overall we see better performance with x64 than with SPARC and other RISC architectures. 32-bit architectures are not recommended for any large dataset (>10GB). The address space limitations of 32-bit operating systems can cause errors in Sawmill when processing large datasets.

Apart from the smallest projects Sawmill should always be hosted on a dedicated system, a 'Sawmill server'. Sawmill can run on a virtual system, but performance can be considerably slower than running Sawmill on physical server. We occasionally see errors with Sawmill running in a virtual environment, so physical hardware is always recommended, especially for larger datasets.

Disk and Space

As a guide you will need between 200% and 400% of the size of your uncompressed log data to store the Sawmill database. Databases tend towards the high side (400%) on 64-bit systems, especially when tracking a very large number of numerical fields (more than ten or so). For instance, if you want to report on 1 terabyte (TB) of log data in a single profile you would need up to 4 TB of disk space for the database. This is the total data in the database, not the daily data added. if you have 1 TB of log data per day, and want to track 30 days, then that is a 30TB dataset, and requires between 60TB and 120TB of disk space.

IMPORTANT:: If your profile uses a database filter which sorts the main table, e.g., a "sessions" analysis (most web server logs do this by default), there is a stage of database build where the entire main table of the database is copied and sorted. During this period, the database will use

up to twice the disk space, and after the build the disk space will drop back down to the final size. So for the example above, if there is a "sessions" analysis or other sorting database filter (another example is a "concurrent streams" analysis), it will temporarily use between 4TB and 8TB of space per day, before returning to 2TB-4TB at the end of the build.

If you are using a separate SQL database server, you will need space to accommodate the server; the databases use this disk space, and the remainder of Sawmill will fit in a smaller space, so 1GB should be sufficient.

Sawmill uses the disk intensively during database building and report generation, for best performance use a fast disk. Ideally use a RAID 10 array of fast disks. RAID 5 or RAID 6 will hurt performance significantly (about 2x slower than RAID 10 for database builds) and is not recommended. Write buffering on the RAID controller should be turned on if possible as it provides an additional 2x performance for database builds.

Network mounts will usually work for storage of the Sawmill database but are not recommended for performance reasons. We sometimes see errors apparently due to locking and synchronization issues with network mounts.

Memory

On the Sawmill server we recommend a minimum of 2GB of RAM per core for large datasets, although more is always helpful

Processor(s)

To estimate the amount of processing power you need, start with the assumption that Sawmill Enterprise processes 2000 log lines per second, per processor core for Intel or AMD processors; or 1000 lines per second for SPARC or other processors. Note: This is a conservative assumption; Sawmill can be much faster than this on some datasets reaching speeds of 10,000-20,000 lines per second per core in some cases. However for sizing your processor it is best to use a conservative estimate to ensure that the specified system is sufficient.

Compute the number of lines in your daily dataset, 200 bytes per line is a good estimate. This will tell you how many seconds Sawmill will require to build the database. Convert that to hours, if it is more than 6 hours you will need more than one processor. You should have enough processors that when you divide the number of hours by the number of processors, it is less than 6.

For example:
50 Gigabytes (GB) of uncompressed log data per day
divide by 200 -> ~268 million lines of log data
divide by 2000 -> ~134 million seconds
divide by 3600 -> ~37 hours
divide by 6 -> 6 processors

The use of six hours is based upon the assumption that you don't want to spend more than six hours per night updating your database to add the latest data. A six hour nightly build time is a good starting point. It provides some flexibility to modify or tune the database and filters that can slow down processing and keep within the processing time available each day. The dataset above could be processed in 9 hours using four processors; if a 9 hour nightly build time is acceptable.

Profile Tuning For High Volume

A default profile created in Sawmill is optimized for maximum report generation speed. These settings can be very problematic for very large datasets (more than 200 million lines of log data), requiring very large amount of time and memory and disk space to build, and should be modified before any large import begins. It is best to start by turning off most of the database indices (all fields are indexed by default) in Config->Database Fields, and most of the cross-references (in Config->Cross-reference Groups). Turning off cross-reference groups will make the database build faster, but will make the corresponding report much slower (minutes instead of seconds). Turning off indices will make the database build faster and the database smaller, but will make filtering on the corresponding field slower. The date/time cross-reference group should generally be left enabled, as it is used in nearly every report, and is not very expensive. Field complexity must also

be managed for large datasets. Any normalized (non-numerical) field with more than about 10 million unique values can become a performance liability, and should be simplified or removed if possible. Fields can be simplified using a Log Filter, for instance setting the field value to a single constant value if it is not needed. Database fields can be removed completely using the command line “remove_database_field” action. During a large database build, it is good to monitor the web interface progress display for the build (click Show Reports on the profile), to see if there are any performance warnings appearing. These warnings also appear in the TaskLog. Single monolithic databases can also be a performance issue. Whenever possible very large datasets should be segmented into separate profiles, each with its own database, for instance one profile per server, or one profile per month, or one profile per region.