GWME-7.1.0-2 - Archive Patch

Background

This patch applies to GroundWork Monitor 7.1.0.

Problem

The archive database in GroundWork Monitor 7.1.0 requires schema changes for regular archival to complete successfully. Consequently, archiving is broken in the 7.1.0 release. This affects both fresh installs of 7.1.0 and for upgrades to 7.1.0.

A secondary issue, is that archiving was overly aggressive removing records required for the Availability Graphs in Status Viewer to display properly after older data had been aged off.

Solution

To address the above issues we are providing a patch. Bundled into this patch are upgraded archiving scripts that perform more selective delete operations in the runtime database, preserving the important data points even though they would have otherwise aged out.

This fix means the archiving software will no longer remove useful data, but by itself it does not replace the important data points which have previously been deleted from the runtime database.

The `TB7.1.0-2.archive-fixes.tar.gz` tarball attached to this article provides replacement files for the 7.1.0 release to get daily archiving back in order.

Name	Size	Creator	Creation Date	Comment
TB7.1.0-2.archive-fixes.tar.gz	121 kB	NotSupportContact-Mark Carey	Oct 06, 2016 16:58	MD5: 0ef1e0805eaaeca33fb3ca50011ec994

Run these commands as the nagios user.
Whether for a freshly-installed 7.1.0 or for a 7.1.0 which was upgraded from some previous release, run all the following steps while logged in as the `nagios` user.

Steps for all types of 7.1.0 installs

Before you install the replacement files, back up the existing files:

```
cd /usr/local/groundwork/config
cp -p log-archive-receive.conf log-archive-receive.conf.orig
cp -p log-archive-send.conf    log-archive-send.conf.orig
mkdir -p /usr/local/groundwork/gw-backup
tar cfz  /usr/local/groundwork/gw-backup/archiving-backup-files.tar.gz \
    -C /usr/local/groundwork \
    config/log-archive-receive.conf \
    config/log-archive-send.conf \
    core/archive/bin/log-archive-receive.pl \
    core/archive/bin/log-archive-send.pl \
    core/databases/postgresql/Archive_GWCollageDB_extensions.sql \
    core/migration/postgresql/pg_migrate_archive_gwcollagedb.pl
```

Next, install the new files, here assuming that the tarball has been placed into the `/tmp` directory:

```
tar xvf /tmp/TB7.1.0-2.archive-fixes.tar.gz -C /usr/local/groundwork
```

Then compare the old and new config files. Resolve differences between any previously localized settings and those that came in with this patch. The patch includes support for the `auditlog`, `hostname`, `hostidentityid`, and `hostblacklistid` tables to be archived, so you should keep that setup as-is in the new config files.

```
cd /usr/local/groundwork/config
diff log-archive-receive.conf.orig log-archive-receive.conf
diff log-archive-send.conf.orig    log-archive-send.conf
```

Edit as needed:

```
vim log-archive-receive.conf
vim log-archive-send.conf
```

Additional steps for a freshly-installed 7.1.0 release

If your system was installed with 7.1.0 without upgrading that system from a previous release, follow these steps. This will re-create the archive database from scratch.

If you upgraded to 7.1.0 or migrated a `gwcollagedb` and/or `archive_gwcollagedb` database from an earlier version, you must instead follow the procedure in the Additional steps for systems upgraded to 7.1.0 from a previous release section, below.

Run the following commands, one at a time, in a `bash` shell. Each run of `$psql` will ask for a password. Respond with the administrative password of the PostgreSQL-database `postgres` user.

```
psql=/usr/local/groundwork/postgresql/bin/psql
scriptdir=/usr/local/groundwork/core/databases/postgresql
$psql -W -U postgres -d postgres            -f $scriptdir/create-fresh-archive-databases.sql
$psql -W -U postgres -d archive_gwcollagedb -f $scriptdir/GWCollageDB.sql
$psql -W -U postgres -d archive_gwcollagedb -f $scriptdir/Archive_GWCollageDB_extensions.sql
$psql -W -U postgres -d archive_gwcollagedb -f $scriptdir/GWCollage-Version.sql
```

Do not worry about the following message that appears when you run the `Archive_GWCollageDB_extensions.sql` script:

> *NOTICE: constraint "host_hostname_key" of relation "host" does not exist, skipping*

It is only a NOTICE, not a WARNING or ERROR, and it is normal and expected.

Additional steps for systems upgraded to 7.1.0 from a previous release

If you either upgraded your present 7.1.0 server from a previous release of GroundWork Monitor, or you installed a fresh 7.1.0 release and then imported databases from a previous release of GroundWork monitor, this section is for you.

Run the scripting needed to repair the structure and content of the archive database, in this order:

```
cd /usr/local/groundwork/core/migration/postgresql
./pg_migrate_archive_gwcollagedb.pl
./conflicting_archive_service_rows.pl -m show
./conflicting_archive_service_rows.pl -m remove
./conflicting_archive_service_rows.pl -m show
```

The `conflicting_archive_service_rows.pl` script, when run with the `-m show` options, will show you if any archive-database rows need to be deleted to synchronize with the runtime database. You will be asked to type in the PostgreSQL administrative password to perform these steps. If you see this message in the results:

```
There are no colliding rows to worry about.
```

there will be no reason to run the script again with the `-m remove` options. However, if you see rows something like this:

```
+-----------------+--------------------------+------------------+------------------+
| servicestatusid |    servicedescription    |   old_hostname   |   new_hostname   |
+-----------------+--------------------------+------------------+------------------+
|           15137 | Arbitrary.HealthCheck    | IJONES-ATO       | ijones-ato       |
|           13957 | External.Response.Period | BOOKS.domain.com | books.domain.com |
|           14453 | Arbitrary.HealthCheck    | TSMITH-ATO       | tsmith-ato       |
+-----------------+--------------------------+------------------+------------------+
```

then you will need to run the `-m remove` form of the command to clean up these rows. You should see:

```
3 rows were deleted.
```

Afterward, run the `-m show` form again to demonstrate that all of the required cleanup has been done.

Notes for all types of 7.1.0 installs

Below is background data to help understand what to expect from the updated archiving process.

Archiving is normally scheduled to run at 00:30 each morning, via a `nagios` user `cron` job. You can check to see whether it succeeded or failed by looking at the archiving log files:

```
/usr/local/groundwork/foundation/container/logs/log-archive-receive.log
/usr/local/groundwork/foundation/container/logs/log-archive-send.log
```

A quick check of the status of the last run can be made this way, without examining the whole files:

```
tail /usr/local/groundwork/foundation/container/logs/log* | egrep -i 'SUCCEEDED|FAILED'
```

Archiving cycles cannot be run immediately back-to-back. The `minimum_additional_hours_to_archive` parameter in the `config/log-archive-send.conf` enforces a minimum delay period between cycles. This setting should not be altered as it will not achieve any speedup in the archival process.

After the steps listed above, daily archiving should run without error. If there number of records ready to be archived is of sufficient size, it should be expected that the first few runs of archiving may take a fair amount of time to run.

For safety reasons, records are not deleted from the runtime database until that data has lain in the archive database for a few days; this is controlled by the `post_archiving_retention_days_for_messages` and `post_archiving_retention_days_for_performance_data` configuration parameters in the `config/log-archive-send.conf` file. Normal operation will not require these settings to be changed.

First archiving run after fixes are installed

The first run of log archiving after a hiatus is special.
The initial run of archiving after you install the fixes described above might be best run by hand (see the instructions below).

The initial run of archiving on a production system or after a hiatus is special, because it needs to sweep up all the accumulated legacy data at one time. See the Initial application of archiving section below for the considerations to address before the first run. Subsequent runs will only archive data which has not yet been archived (subject to configured redundancy). If a run is skipped, there will be no loss of continuity; generally speaking, all the data which would have been archived in the skipped run will simply be picked up in the following run. Missing a run could happen due to both major events (say, you have a power outage one night), or minor events (say, you have a temporary failure in hostname resolution, so the scripting cannot connect to the archive database at the moment it first needs to do so).

Initial application of archiving

The initial run of the scripting on a production system, or a run after a long hiatus, should have `dump_days_maximum` set to `10000`, which is the default value for this parameter in the shipped send-side config file (`config/log-archive-send.conf`). This setting will archive all data from the Pleistocene era until midnight last night, in the local timezone. For a large site which has been running a long time with fine-grained performance data, this could require a lot of space in the filesystem, and a lot of CPU and disk activity. You should plan for that, perhaps by running the sending script manually the first time (as the `nagios` user) during a weekend maintenance window. To do that, you may need to temporarily disable the software by either commenting out the `nagios`-user cron job that runs the nightly archiving:

```
30 0 * * * /usr/local/groundwork/core/archive/bin/log-archive-send.pl -a
```

or via the `enable_processing` directive in the (`config/log-archive-send.conf`) config file.

Have enough disk space available for staging data transfers
When archiving is first put into play, or after a long hiatus, there will be a very large backlog of message and performance data to be transferred to the archive database. This data is staged in files, and the same data will appear in multiple sets of those files (for safety purposes). Make sure you have enough space in the filesystem on the source and target machines where these files will be stored (listed as the values of the `log_archive_source_data_directory` and `log_archive_target_data_directory` in the `config/log-archive-send.conf` and `config/log-archive-receive.conf` configuration files).

For purposes of estimating the amount of space needed, the sizes of the `logmessage` and `logperformancedata` tables in the `gwcollagedb` database are most critical:

```
select count(*) from logmessage;
select count(*) from logperformancedata;
```

While the details will vary from site to site, a reasonable first estimate is to allow 200 bytes per `logmessage` row, and 61 bytes per `logperformancedata` row, for each copy of these tables that will be retained in the filesystem. Multiply the space so determined for one copy by the `source_dumpfile_retention_days` parameter in the `log-archive-send.conf` config file on the source machine, for the space needed on the source machine. If your target machine is different from your source machine (this is not a common scenario), multiply the space so determined for one copy by the `target_dumpfile_retention_days` parameter in the `log-archive-receive.conf` config file on the target machine, for the space needed on the target machine.

Example: We have 768539 `logmessage` rows, and 118465 `logperformancedata` rows in our runtime database. Dumping out this data into files will take around:

```
768539 message_rows * 200 bytes/message_row
    + 118465 perf_rows * 61 bytes/perf_row = 160934165 bytes
```

With `source_dumpfile_retention_days` set to 11.5 (the default value, as shipped), this yields:

```
160934165 bytes/dataset * 12 datasets = 1931209980 bytes
```

Which is to say, you would need to ensure that about 2 GB of space is available on the source machine, in the `log_archive_source_data_directory` filesystem (normally configured as `/usr/local/groundwork/core/archive/log-archive`), for the initial period of archiving this example database.

To run archiving manually, you must do so as the `nagios` user:

```
su - nagios
/usr/local/groundwork/core/archive/bin/log-archive-send.pl -a
# Check to see whether the archiving worked as expected.
tail /usr/local/groundwork/foundation/container/logs/log* | egrep -i 'SUCCEEDED|FAILED'
exit
```