Upgrading Platform Analytics

Platform Analytics
Version 8.0.2
November 2011

Platform
Computing

Copyright

We'd like to hear from you

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to doc@platform.com.

Your comments should pertain only to Platform documentation. For product support, contact support@platform.com.

Document redistribution and translation

Internal redistribution

Trademarks

Third-party license agreements

http://www.platform.com/Company/third.part.license.htm

Contents

CHAPTER

1

Platform Analytics upgrade packages

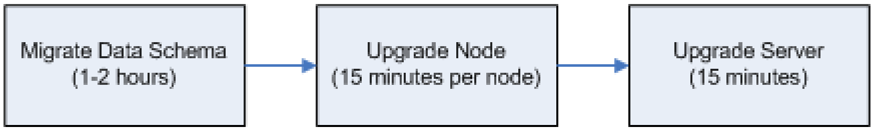
About the Platform Analytics upgrade packages

The Platform Analytics 8.0.2 upgrade packages allow you to upgrade your Platform Analytics system from version 8.0 to version 8.0.2. The upgrade packages include updates to the Vertica database schema, the Platform Analytics node, and the Platform Analytics server. You can upgrade Platform Analytics 8.0 with any patches or solution packages installed.

You do not need a separate license for Platform Analytics 8.0.2 if you already have a license for Platform Analytics 8.0.

The Platform Analytics upgrade packages will help improve the upgrade process:

- The time to migrate the Vertica database schema and Platform Analytics data depends on the performance of your database host and the amount of Platform Analytics-related data in the database. For example, it will take approximately one to two hours to migrate the database schema and data for a cluster with 1.5 million jobs a day, 2000 records per sampling interval, and a data retention period of two weeks for raw data, 26 weeks for hourly data, and 52 weeks for daily data.
- Upgrading the Platform Analytics node will typically take 15 minutes. The data loaders will be down for up to five minutes during this time while being upgraded.
- Upgrading the Platform Analytics server will typically take 15 minutes.



Customization issues

The Platform Analytics 8.0.2 upgrade packages can handle certain customizations that you applied to certain built-in files.

Database schema

The following customizations to the database schema will remain in the upgraded database schema:

- New tables added to the Platform Analytics database
- New columns added to built-in tables in the Platform Analytics database

The Platform Analytics database schema upgrade package cannot handle certain customizations to the database schema. After the upgrade, your Platform Analytics node and server hosts may encounter problems, which are sent as events. The following customizations may result in problems after the upgrade:

- Dropped columns in built-in tables
- Modified built-in table names
- Modified column names or types in built-in tables

Platform Analytics node

Customizations to the following configuration files (all in the `conf` directory) will remain in the upgraded Platform Analytics node:

- `datasource.xml`
- `log4j.properties`
- `plc.xml`
- `perf.conf`
- All `*.properties` files in the `dataloader` subdirectory.
- All `*.xml` files in the `plc` subdirectory.
- `wsm_plc.conf` in the `wsm` subdirectory.

Platform Analytics server

Customizations to the following configuration files (all in the `conf` directory) will remain in the upgraded Platform Analytics server:

- `datasource.xml`
- `log4j.properties`
- `Config.xml`
- `ItemLists.xml`
- `pi.conf`
- All `*.xml` files in the `purger` subdirectory.
- `Package.xml` in the `packages/workload` subdirectory.

Platform Analytics upgrade packages

CHAPTER

2

Prepare for the Platform Analytics upgrade

Obtain the upgrade package files and upgrade the Vertica database to prepare for the Platform Analytics upgrade.

What you need to do

Check the following:

- The Vertica database host is properly configured and running.
- You have a user name, password, and URL to access the database.

  The Platform Analytics database user must have the following:

  - Roles: Connect and resource.
  - System privileges: Create view.
- The Vertica database and the database host meet the Platform Analytics system requirements.

  Refer to the *Release Notes for Platform Platform Analytics* for the latest system requirements for the database host.

1. *Obtain the necessary upgrade package files* on page 10
2. *Update the database schema* on page 10

Obtain the necessary upgrade package files

1. Obtain the necessary files for upgrading Platform Analytics.

   - Platform Analytics server upgrade package

     `platform_analytics8.0.2_server_upgrade_${OS}.zip`
   - Platform Analytics node upgrade package

     `platform_analytics8.0.2_node_upgrade_${OS}.zip`
   - Platform Analytics database schema upgrade package

     `platform_analytics8.0.2_upgrade_dbschema.zip`

Update the database schema

Update the Vertica database schema and the data to prepare the database host for the Platform Analytics upgrade.

Note:

The database schema upgrade package does not support upgrading non-partitioned schema to a partitioned schema, and vice versa.

Therefore, if you are using a non-partitioned schema, you cannot upgrade to a partitioned schema using the database schema upgrade package.

Update the database schema in the Vertica database

1. Log into the Vertica database host.
2. (Optional) Set `VERTICA_ROOT` environment variable to the Vertica installation directory. If you have not installed Vertica on the default location, set it to `/opt/vertica`.
3. Extract the database schema upgrade package.

   ```
   unzip platform_analytics8.0.2_server_upgrade_${OS}.zip
   ```

4. Navigate to the directory containing your database schema scripts.

   `migration/schema/Vertica`

5. Run the script `upgrade_schema.sh` to upgrade the database.

   sh upgrade_schema.sh <*user_name*> <*password*> <*database_name*> <*port*>

   where

   - *user_name* is the name of the Platform Analytics database user.
   - *password* is the password for the database user.
   - *database_name* is the name of the database.
   - *port* is the TCP port number or the local socket file extension in which the server is listening for connections. The default port number is 5433

CHAPTER

3

Upgrade Platform Analytics

Upgrade Platform Analytics from version 8.0 to version 8.0.2.

What you need to do

Check the following:

- The Platform Analytics server and node hosts are running Platform Analytics 8.0.

  The Platform Analytics upgrade packages will work with any patches or solution packages that you installed, but will only work on version 8.0.
- The Platform Analytics server and node hosts meet the detailed system and software requirements for Platform Analytics 8.0.2.

  Refer to the *Release Notes for Platform Platform Analytics* for the latest requirements.

1. *Upgrade the Platform Analytics server* on page 14
2. *Verify the Platform Analytics server upgrade* on page 15
3. *Upgrade the Platform Analytics node* on page 16
4. *Verify the Platform Analytics node upgrade* on page 17
5. *Upgrade the Platform Analytics report package* on page 18
6. *Verify the reports package upgrade* on page 18
7. *Data migration - for those who have upgraded LSF 7.x to LSF 8* on page 18

After completing the Platform Analytics upgrade, you can optionally optimize the Platform Analytics node and server hosts.

Upgrade the Platform Analytics server

Upgrade the Platform Analytics server on a UNIX or Windows host.

Check the following:

- You migrated the Vertica database schema to Platform Analytics 8.0.2.

Important:

The Platform Analytics upgrade package cannot detect whether you updated the database schema. If you proceed with the upgrade without migrating the database schema, some Platform Analytics server tasks will encounter problems and send events.

Upgrade the Platform Analytics server using the Platform Analytics server upgrade package. The upgrade package will make backups of the previous version of Platform Analytics before upgrading your server to the latest version.

1. Log into the Platform Analytics server host with the same account that you used to install the previous version of Platform Analytics.
2. Extract the server patch.

   unzip <*server_upgrade_package*>
3. If you connected to the UNIX host, grant the execution permission to the `analytics8.0.2_upgrade_server_${OS}/patch_tools` directory..
4. Run the Platform Analytics server upgrade package.
   - UNIX: **analytics8.0.2_upgrade_server_${OS}/patch_tools/pinstall.sh**
   - Windows: **analytics8.0.2_upgrade_server_${OS}\patch_tools\pinstall.bat**

Remember:

`pinstall` will automatically back up the original files from `ANALYTICS_TOP`, copying the files from the patch directory to this directory.

For the Analytics server, `pinstall` will automatically stop the Platform Task Scheduler, which may cause running Analytics tasks to report a failed status. However, this does not indicate any data loss.

5. Copy Platform Analytics 8.0.2 license to `ANALYTICS_TOP/conf`.
6. Modify the task scheduler of the Platform Analytics Server.
   a) Open the Platform Analytics Console.
   b) Navigate to Scheduled Tasks.
   c) Add the following Scheduled Tasks

   - DailyReportETL

   To add a new task, Select Action -> Add Scheduled Task from the task bar.

   - Scheduled Task—task name **DailyReportETL**
   - Script File—`bin/dataaggdaily.js`
   - Script Function— **doit**
   - Select Enable Scheduling
   - Set the Run Interval—1 day
7. (LSF 8 users only) Set `LSF_VERSION` in the `pi.conf` file.
   - Edit `ANALYTICS_TOP/conf/pi.conf`
   - Append the following line at the end of the file:

     LSF_VERSION=8
8. Start Platform Analytics server.

The changed files from your previous version of Platform Analytics are backed up to the following location:

- UNIX: *`ANALYTICS_TOP`*`/upgrade_backup/`*`version_number`*
- Windows: *`ANALYTICS_TOP`*`\upgrade_backup\`*`version_number`*

Verify the Platform Analytics server upgrade

Verify that you have successfully upgraded the Platform Analytics server on a UNIX or Windows host.

You upgraded the Platform Analytics server and are still logged into the host.

1. Verify that there are no errors generated in the event log file after the upgrade.

   The event log file is named `event.log.`*`host_name`* and is located in the PERF log directory:

   - UNIX: *`ANALYTICS_TOP`*`/log`
   - Windows: *`ANALYTICS_TOP`*`\log`
2. Start the Platform Analytics Console.
   - UNIX: ***ANALYTICS_TOP*/bin/runconsole.sh**
   - Windows: Start > Programs > Platform Platform Analytics Server > Platform Platform Analytics Console
3. Click Scheduled Tasks in the navigation tree and verify that there are no failed tasks.

4. After at least an hour, verify that the data is collected and you can view reports.
5. If your Platform Analytics installation has customizations, check that the customizations are still present.

Upgrade the Platform Analytics node

Upgrade the Platform Analytics node on a UNIX or Windows host.

Check the following:

- You migrated the Vertica database schema to Platform Analytics 8.0.2.

Important:

The Platform Analytics upgrade package cannot detect whether you updated the database schema. If you proceed with the upgrade without migrating the database schema, the Platform Analytics data loaders will encounter problems and send events.

Upgrade the Platform Analytics node using the Platform Analytics node upgrade package. The upgrade package will make backups of the previous version of Platform Analytics before upgrading your node to the latest version.

The upgrade package stops the Platform Analytics data loaders to upgrade these components. These data loaders will not be stopped for longer than five minutes under normal circumstances.

Note:

You cannot upgrade Platform Analytics node on the Windows host. Choose a Unix host to upgrade.

1. Log into the Platform Analytics node host with the same account that you used to install the previous version of Platform Analytics.
2. Extract the node patch.

   unzip *<node_upgrade_package>*
3. If you connected to the UNIX host, grant the execution permission to the `upgrade_node_${OS}/patch_tools` directory..
4. Run the Platform Analytics node upgrade package.
   - UNIX: **analytics_upgrade8.0.2_node_${OS}/patch_tools/pinstall.sh**

   Remember:

   `pinstall` will automatically back up the original files from `ANALYTICS_TOP,` copying the files from the patch directory to this directory.

   For the Analytics node, `pinstall` will automatically restart the data loader (plc), which may cause data loss in some data loaders that are currently used.
5. Copy Platform Analytics 8.0.2 license to `ANALYTICS_TOP/conf`.
6. Enable `lsbpendingreasonloader/lsfjobstatusloader` data loader.

   Edit the `ANALYTICS_TOP/conf/plc/plc_lsf_advanced.xml` file and add the line below into node <`Dataloaders`>.

```
<DataLoader Name="lsbpendingreasonloader" Interval="600" Enable="true"
LoadXML="dataloader/lsbpendingreasons.xml" />
```

```
<DataLoader Name="lsfjobstatusloader" Interval="600" Enable="true"
LoadXML="dataloader/lsbstatus.xml" />
```

7. Edit the `ANALYTICS_TOP/conf/dataloader/lsbevents.properties` file.

   Add event type `JOB_FINISH2` and `JOB_STARTLIMIT` to `ALLOW_EVENT_TYPE` field.

   This is for LSF 8.0.1 QPK users only. Add event type `JOB_STATUS2` and enable this in the LSF configuration file `lsb.params`.

   For example, `ALLOW_EVENT_TYPE=JOB_NEW JOB_FINISH JOB_FINISH2 JOB_STARTLIMIT JOB_STATUS2`
8. (Optional) Edit the `ANALYTICS_TOP/conf/dataloader/lsbpendingreasons.properties [lsbstatus.properties]` file.
   - If you do not want the `lsbpendingreasonloader/lsfjobstatusloader` data loader to delete historical `lsb.pendingreaons/lsb.status` files after loading the file, add the following line:

     AUTO_DELETE_STREAM_FILE=N
   - Set `REMAINED_FILE_NUM` to define the number of historical `lsb.pendingreaons/lsb.status` file. The minimum value is 1. The maximum value should be less than the value of `MAX_EVENT_STREAM_SIZE`, which is defined in the `lsb.params` file.
   - Set `ARCHIVE_PATH` to define the place to backup historical `lsb.pendingreasons/lsb.status` file.

     For example, `ARCHIVE_PATH=/tmp`
9. Verify that the `plc` service has started. If not, restart the loader controller.

   perfadmin stop plc

   perfadmin start plc

The changed files from your previous version of Platform Analytics are backed up to the following location:

- UNIX: *`ANALYTICS_TOP/`*`.patch_backup`

Verify the Platform Analytics node upgrade

Verify that you have successfully upgraded the Platform Analytics node on a UNIX or Windows host.

You upgraded the Platform Analytics node and are still logged into the host.

1. Verify that the data loaders are running.
   - UNIX: **plcclient.sh -s**
2. After at least one hour, verify that there are no errors in any of the data loader log files.

   You need to wait at least one hour because this is the longest data sampling interval of the data loaders.

   The data loader log files are located in the `dataloader` subdirectory of the PERF log directory:
   - UNIX: *`ANALYTICS_TOP/`*`log/dataloader`
3. Verify that there are no errors in the loader controller log file.

   The loader controller log file is named `plc.log.`*`host_name`* and is located in the PERF log directory:
   - UNIX: *`ANALYTICS_TOP/`*`log`

Upgrade the Platform Analytics report package

Upgrade the Platform Analytics report package.

Uninstall previous reports and the `parb` service

1. Run `rptinstall.bat` to publish the reports and install the `parb` service.
2. Reinstall PAC 8.0.2 as Platform Analytics8.0.2 supports only the 8.0.2 version of Platform Applicaqtion Center.
3. Log into the Platform Analytics reporting server with the same account that you used to install the previous version of Platform Analytics.
4. Extract the report patch.

   unzip *<report_upgrade_package>*
5. Follow the dialog prompts to complete the installation. Specify the following to complete the installation.
   - Database type
   - Database server name
   - Database name
   - Database port number
   - Database user name and password
   - Tableau server directory
   - Whether or not to use SSL
   - Tableau server administrator name and password
   - Tableau server port number

   Note:

   To publish customized workbooks, you have to install and use the Platform Analytics Designer or run the the `tabcmd` tool provided by the Tableau server.

Verify the reports package upgrade

Once data is collected in the database, you can view reports using the Analytics reporting server. Optionally, you can even view reports using Platform Analytics Designer or Platform Application Center.

1. Log in to the Platform Analytics reporting server.

   ```
   http://<host_name>:<port>
   ```

   where <*host_name*> is the name of the system where Tableau server is installed and <*port*> is the number which you entered during the Tableau server installation.
2. See if you can view workbooks, worksheets, and dashboards. Try to change measures and dimensions and see if you can view new reports.

Data migration - for those who have upgraded LSF 7.x to LSF 8

This is the mandatory step for LSF 8 users who have applied the LSF 8 add-on data solution and have enabled new data set: JOB_FINISH2, JOB_STATUS2, and JOBS_PENDINGREASONS events.

1. Log on to Platform Analytics server host.

2. Make sure that the value of `LSF_VERSION` which in `pi.conf` file is 7.
3. Start the Platform Analytics server.
4. Monitor the log files that are in the `ANALYTICS_TOP/log/datatransformer` directory and wait untill you see "`Records written: 0`" in all these log files.
    - ClusterCapacity.log.*HOSTNAME*
    - WorkloadStatistics.log.*HOSTNAME*
    - Hardware.log.*HOSTNAME*
    - WorkloadAccounting.log.*HOSTNAME*
    - FlexlmLicUsage.log.*HOSTNAME*
5. Change the value of `LSF_VERSION` which in `pi.conf` file from 7 to 8.
6. Restart the Platform Analytics server.

CHAPTER

4

Optimize the Platform Analytics node

Optional. Modify the Platform Analytics node to enhance performance.

What you need to do

1. *Increase JVM memory* on page 22
2. *Disperse the Platform Analytics node workload* on page 22
3. *Optimize specific data loaders* on page 25

Increase JVM memory

Increase the Java Virtual Machine (JVM) memory to at least 2 GB for the Platform Analytics node instead of using the default value to prevent the JVM from running out of memory.

Note:

Java cannot use more than 1638 MB of memory on 32-bit platforms. If you are using 32-bit Windows, you can only increase the JVM memory to 1638 MB.

1. Log into the Platform Analytics ndoe host.
2. Edit the `ANALYTICS_TOP/conf/wsm/wsm_plc.conf` file.
3. Change the Java starting options to increase the JVM memory.

   Navigate to JAVA_OPTS and increase the JVM memory to at least 2 GB.

   For example,

   JAVA_OPTS=-Xms64m -Xmx2048m

Disperse the Platform Analytics node workload

You can reduce performance issues by dispersing the Platform Analytics node workload among multiple hosts or multiple loader controllers within a host so that each host or loader controller is responsible for a specific type (or types) of data loading.

Based on the type of data, you can categorize all the data loaders into different types, and enable each node or loader controller to be responsible for only one type of data loader. The following table describes the different data loader categories and their corresponding loader controller configuration files:

Data loader category	Loader controller configuration file
Host-related	• `plc_ego.xml` • `plc_coreutil.xml`
Job-related	• `plc_lsf.xml` • `plc_bjobs-sp012.xml`
Advanced job-related	• `plc_lsf_advanced.xml`
License-related	• `plc_license.xml`

- *Disperse the Platform Analytics node workload to multiple hosts* on page 23

Disperse the workload to multiple hosts if your Platform Analytics node is experiencing performance issues and cannot handle all the workload by itself.

- *Disperse the Platform Analytics node workload to multiple loader controllers in one node* on page 23

  Disperse the workload to multiple loader controllers within a host to reduce performance bottlenecks if your Platform Analytics node host is powerful enough to handle the workload (for example, the host has at least four cores and at least 6 GB of memory).

Disperse the Platform Analytics node workload to multiple hosts

A single node might experience performance issues when handling the high workload. To resolve this issue, you can disperse the node workload to multiple hosts so that each host is responsible for a specific type of data loading.

1. Install and configure the Platform Analytics node to other hosts in the same LSF cluster.

   For example, if you plan to have one node for each data loader category, you should install and configure three additional Platform Analytics node hosts. Therefore, your original node would handle host-related data loaders, while each of the other three nodes would handle each of the other three categories.

   Alternatively, you can have fewer nodes and still disperse workload to some degree. For example, you could have two nodes: your original node could handle host-related and job-related data loaders while another node could handle advanced job-related and license-related data loaders.
2. For each Platform Analytics node, disable the data loading categories that the node will not be handling.
   a) Navigate to the `ANALYTICS_TOP/conf/plc` directory.
   b) Move or rename the configuration files for the data loader categories that the Platform Analytics node will no longer handle.

      For example, if you want the Platform Analytics node to only handle host-related data loaders, move or rename all configuration files except `plc_ego.xml` and `plc_coreutil.xml`.

Disperse the Platform Analytics node workload to multiple loader controllers in one node

If your node host is powerful enough to handle the workload (for example, the host has at least four cores and at least 6 GB of memory), you can reduce the performance bottlenecks by creating multiple loader controllers and having each loader controller be responsible for a specific type of data loading. This enables each of the multiple cores in your host to control a single loader controller and be more efficient in sharing the workload among multiple cores.

1. For each data loader category beyond the first, create a new loader controller on the same host.

   Therefore, since there are four data loader categories, create three additional loader controllers.

   In this section, *PLC_NAME* represents the name of the new loader controller that you will create.

   For example, you can create `plc_job` to handle job-related data loaders, `plc_advanced_job` to handle advanced job-related data loaders, and `plc_license` to handle license-related data loaders, while the original loader controller (`plc`) handles host-related data loaders.
   a) From the `ANALYTICS_TOP/conf/wsm` directory, copy the `wsm_plc.conf` file to a new file, one for each new loader controller.

      For each new loader controller, name the new file `wsm_PLC_NAME.conf`

For example, copy `wsm_plc.conf` to `wsm_plc_job.conf`, `wsm_plc_advanced_job.conf`, and `wsm_plc_license.conf`.

b) Edit each new `wsm_`*`PLC_NAME`*`.conf` file and specify the new loader controller name.

   Navigate to the `SERVICE_NAME`, `SERVICE_COMMAND`, and `LOG_PREFIX` parameters to use the new loader controller name as follows:

```
SERVICE_NAME=PLC_NAME
SERVICE_COMMAND=com.platform.perf.dataloader.Main –f PLC_NAME.xml
LOG_PREFIX=PLC_NAME
```

   For example, for the `wsm_plc_job.conf` file:

```
SERVICE_NAME=plc_job
SERVICE_COMMAND=com.platform.perf.dataloader.Main –f plc_job.xml
LOG_PREFIX=plc_job
```

   Make similar edits to the `wsm_plc_advanced_job.conf` and `wsm_plc_license.conf` files.

c) From the *`ANALYTICS_TOP`*`/conf` directory, create a new subdirectory for each new loader controller.

   Name each directory *`PLC_NAME`*.

   For example, create three new directories named `plc_job`, `plc_advanced_job`, and `plc_license`.

d) Move the loader controller configuration files, corresponding to the data loader categories, from *`ANALYTICS_TOP`*`/conf/plc` to the subdirectory corresponding to the new loader controller that will handle the category data loaders.

   For example,

   1. Move `plc_lsf.xml` and `plc_bjobs-sp012.xml` from *`ANALYTICS_TOP`*`/conf/plc` to *`ANALYTICS_TOP`*`/conf/plc_job`.
   2. Move `plc_lsf_advanced.xml` from *`ANALYTICS_TOP`*`/conf/plc` to *`ANALYTICS_TOP`*`/conf/plc_advanced_job`.
   3. Move `plc_license.xml` from *`ANALYTICS_TOP`*`/conf/plc` to *`ANALYTICS_TOP`*`/conf/plc_license`.

e) From the *`ANALYTICS_TOP`*`/conf` directory, copy the the loader controller file (`plc.xml`) to a new file, one for each new loader controller.

   For each new loader controller, name the new file *`PLC_NAME`*`.xml`.

   For example, copy `plc.xml` to `plc_job.xml`, `plc_advanced_job.xml`, and `plc_license.xml`.

f) Edit each new loader controller file and change the `Port` parameter to a new port and change the `PLCDir` parameter to the new loader controller directory.

   For example,

   1. In `plc_job.xml`, change `Port` to from **4046** to **4047** and change `PLCDir` from **plc** to **plc_job**.
   2. In `plc_advanced_job.xml`, change `Port` to from **4046** to **4048** and change `PLCDir` from **plc** to **plc_advanced_job**.
   3. In `plc_license.xml`, change `Port` to from **4046** to **4049** and change `PLCDir` from **plc** to **plc_license**.

2. Restart the new loader controllers.

   perfadmin stop all

   perfadmin start all

Note:

To stop or start an individual data loader, use **perfadmin stop** *PLC_NAME* and **perfadmin start** *PLC_NAME*

For example, to stop the loader controller that handles license-related data loaders, run **perfadmin stop plc_license**.

Optimize specific data loaders

If you are encountering specific problems or are working under specific environments, you can optimize certain data loaders to enhance the working performance of these individual data loaders.

1. Optimize the FLEXnet usage data loader (`flexlicusageloader`) to improve data loading time if it cannot finish loading the data within one sampling interval.

   If the FLEXnet usage data loader cannot finish the data loading of one sampling interval within the time of one sampling interval (typically five minutes), the data loader will be behind. You need to enable the multi-threads setting to catch up with the multi-servers and daemons workload scale.

   a) Log into the Platform Analytics node host that is running the FLEXnet usage data loader.
   b) Edit the `ANALYTICS_TOP/conf/dataloader/flexlicusage.properties` file.
   c) Edit the `ThreadNumber` parameter to increase the number of threads for collecting data concurrently.

      For example,

      ThreadNumber=10

2. Optimize the FLEXnet Manager data loader (`fnmloader`) to achieve a larger capacity if you need to handle more than 1 million events per day.

   Using the default configuration means that you do not have to modify the data schema of the FLEXnet Manager reporting database, but the capacity of this data loader is limited to 1 million events per day.

   If you need to achieve a larger capacity, you need to modify the data schema of the FLEXnet Manager reporting database to increase performance. For more details, refer to the `ANALYTICS_TOP/conf/dataloader/FNMLOADER.readme` file.

3. Optimize the LSF events data loader (`lsfeventsloader`) to prevent data loss if you are using LSF versions 7.0.3 or earlier.

   The LSF events data loader reads data from the `lsb.stream` file. If you are using LSF versions 7.0.3 or earlier, you should increase the size of the `lsb.stream` file to 2 GB or more to prevent data loss.

   a) Log into the Platform Analytics node host that is running the LSF events data loader.
   b) Edit the `LSF_ENVDIR/lsbatch/cluster_name/configdir/lsb.params` file.
   c) In the `Parameters` section, define a new parameter named `MAX_EVENT_STREAM_SIZE`.

      For example,

      ```
      Begin Parameters
      ...
      MAX_EVENT_STREAM_SIZE = 2048
      ...
      End Parameters
      ```

   d) Reconfigure the LSF cluster for your changes to take effect.

      lsadmin reconfig

CHAPTER

5

Optimize the Platform Analytics server

Modify the Platform Analytics server to enhance performance.

What you need to do

The following is optional.

1. *Change the data retention period* on page 28
2. *Split the data transformer tasks to disperse workload* on page 29

Change the data retention period

A long data retention period can have a significant impact on performance and the data volume. You can tailor the data retention period according to your business requirements to maximize the performance of your Platform Analytics server.

The data purger consists of multiple scheduled tasks (`PartitionMaintenanceGroup*`), which are enabled by default.

1. Launch the `vsql` command line.
   a) Navigate to the `bin` subdirectory of the Vertica installation directory.

      By default, this is `/opt/vertica/bin`.
   b) Run `vsql` to connect to the database.

      ./vsql -d *database_name* **-p** *port* **-U** *username* **-w** *password*

      where

      - *database_name* is the name of the database
      - *port* is the TCP port number or the local socket file extension in which the server is listening for connections. The default is port number 5433.
      - *username* is the name of the user with which to connect to the database, instead of the default user (the database administrator).
      - *password* is the password for the database user.

      Alternately, you can run `vsql` with no options to accept the defaults and specify the administrator password at the prompt.
2. Examine the current data retention periods of the database tables for Platform Analytics.

   - To examine the retention periods for all Platform Analytics database tables, run the following from the `vsql` command line:

     SELECT TABLE_NAME, DATA_DAYS_RANGE

     FROM SYS_TABLES_TO_PARTITION;
   - To examine the retention periods for a specific database table, run the following from the `vsql` command line:

     SELECT TABLE_NAME, DATA_DAYS_RANGE

     FROM SYS_TABLES_TO_PARTITION

     WHERE TABLE_NAME='*table_name***';**

     where *table_name* is the name of the table you want to examine.

   The output displays the name of the table and the corresponding data retention period in days.

For example, to view the data retention period for the `RESOURCE_METRICS_BUILTIN` table, run the following from the `vsql` command line:

SELECT TABLE_NAME, DATA_DAYS_RANGE

FROM SYS_TABLES_TO_PARTITION

WHERE TABLE_NAME='RESOURCE_METRICS_BUILTIN';

3. Change the data retention period for the appropriate database tables.
   a) For each database table to change, run the following from the `vsql` command line:

      UPDATE SYS_TABLES_TO_PARTITION

      SET DATA_DAYS_RANGE='*retention_period***'**

      WHERE TABLE_NAME='*table_name***';**

      where

      - *retention_period* is the new retention period, in days
      - *table_name* is the name of the table you are changing

   b) Commit the changes to the database.

      Run the following from the `vsql` command line:

      COMMIT;

For example, to change the data retention period of the `RESOURCE_METRICS_BUILTIN` table to 2192 days, run the following from the `vsql` command line:

UPDATE SYS_TABLES_TO_PARTITION

SET DATA_DAYS_RANGE='2192'

WHERE TABLE_NAME='RESOURCE_METRICS_BUILTIN';

COMMIT;

Split the data transformer tasks to disperse workload

By default, there are two default scheduled tasks that control data transformers. Four scheduled tasks might not be enough to be able to run all the data transformers within one hour, so to enhance performance, you can split these data transformers into more tasks.

The following is a recommended format for splitting your data transformers into five tasks. The examples will make use of this table (specifically, with Task 1).

Task	Data transformer name	Data flow entry
1	ClusterCapacity	main_cluster_capacity.xml
2	WorkloadStatistics	main_workload_statistics.xml
3	FlexlmLicenseUsage	main_flexlm_licusage.xml
4	Hardware	main_hardware.xml
5	Jobmart	main_jobmart.xml

1. Log into the Platform Analytics server host.

2. Create and enable a new scheduled task in the Platform Analytics Console.
   a) Launch the Platform Analytics Console.
      - UNIX: ***ANALYTICS_TOP*/bin/runconsole.sh**
      - Windows: Start > Programs > Platform Analytics Server > Platform Analytics Console

   b) Click Scheduled Tasks in the navigation tree.

   c) Right-click on the main window and select Add Scheduled Task.

   d) Complete the required fields for the new task.
      - Scheduled Task: Specify the name of this task.
      - Script File: Specify **bin/dataagghourly.js** for hourly tasks or **bin/dataaggdaily.js** for daily tasks.
      - Script Function: Specify **doit**.

      For example, if you are creating Task 1from the table with the recommended format of splitting data transformers, specify the following:

      - Scheduled Task: Specify Task1 as the name of the scheduled task.
      - Script File: Specify **bin/dataagghourly.js** as the path to the script file.
      - Script Function: Specify **doit** as the script function.

   e) Enable the new scheduled task that you created.
3. In the `tasks` subdirectory of *`ANALYTICS_TOP`*, create a new directory with the same name as the name of the new scheduled task and navigate to the new directory.

   For example, for Task 1 on a UNIX host,

   cd *ANALYTICS_TOP*/tasks

   mkdir Task1

   cd Task1
4. From the new directory, create a text file of any name with the `.tsk` extension.

   For example, create `task1.tsk`.
5. In the new `.tsk` text file, for each data transformer that you would like the scheduled task to control, add its corresponding data flow entry as a new file to the line.

   You can also add a comment with the name of the data transformer if you start the line with the **#** character.

   For example, for Task 1, the `task1.tsk` file should contain the following lines:

```
# Cluster Capacity
datatransformer/flow/clustercapacity/main_cluster_capacity.xml
```