IBM SPSS Analytic Server
Version 3.1.1.1

Administrator's Guide

Note
Before using this information and the product it supports, read the information in "Notices" on page 23.

Product Information

This edition applies to version 3, release 1, modification 1 of IBM SPSS Analytic Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Tenant management

Tenants provide a high-level division of users, projects, and data sources so that objects cannot be shared between tenants. Each user accesses the system in the context of a tenant to which they are assigned.

You manage tenants, and assign users to tenants, in the Analytic Server console. The view of the Tenants page depends upon the role of the user that is logged in to the console:

- The "super user" administrator that is set up during installation is the tenant manager. Only this user can create new tenants and edit the properties of any tenant.
- Users with the Administrator role can edit the properties of the tenant they are logged in to.
- Users with the User role cannot edit tenant properties. The Tenants page is hidden from them.
- Users with the Reader role cannot edit data sources, or even log in to the Analytic Server console.

Administrators can access the Projects and Data sources pages and manage any project or data source for cleanup and administration. See the *IBM® SPSS® Analytic Server User's Guide* for more information.

Tenant listing

The main Tenants page displays the existing tenants in a table. Only the "super user" administrator can make edits on this page.

- Click a tenant's name to display its details and edit its properties.
- Click a tenant's URL to open the console in the context of that tenant.

  Note: You will be logged out of the console and will need to log in with valid credentials for the tenant.
- Type in the search area to filter the listing to display only tenants with the search string in their name.
- Click **New** to create a new tenant with the name you specify in the **Add new tenant** dialog. See "Naming rules" on page 2 for restrictions on the names you can give to tenants.
- Click **Delete** to remove the selected tenant(s).
- Click **Refresh** to update the listing.

Individual tenant details

The content area is divided into several collapsible sections.

Details

Name An editable text field that displays the name of the tenant.

Description
An editable text field that allows you to provide explanatory text about the tenant.

URL This is the URL to give to users to log in to the tenant through the Analytic Server console, and to use to configure SPSS Modeler server. See *IBM SPSS Analytic Server Installation and Configuration Guide* for details on configuring SPSS Modeler.

Status **Active** tenants are currently in use. Making a tenant **Inactive** prevents users from logging in to that tenant, but does not delete any of the underlying information.

Principals

Principals are users and groups that are drawn from the security provider that is set up during installation. You can add principals to a tenant as Administrators, Users, or Readers.

- Typing in the text box filters on users and groups with the search string in their name. Select **Administrator**, **User**, or **Reader** from the drop-down list to assign their role within the tenant. Click **Add participant** to add them to the list of authors.
- To remove a participant, select a user or group in the member list and click **Remove participant**.

Metrics

Allows you to configure resource limits for a tenant. Reports the disk space currently used by the tenant.

- You can set a maximum disk space quota for the tenant; when this limit it reached, no more data can be written to disk on this tenant until enough disk space is cleared to bring the tenant disk space usage below the quota.
- You can set a disk space warning level for the tenant; when the quota is exceeded, no analytic jobs can be submitted by principals on this tenant until enough disk space is cleared to bring the tenant disk space usage below the quota.
- You can set a maximum number of parallel jobs that can be run at a single time on this tenant; when the quota is exceeded, no analytic jobs can be submitted by principals on this tenant until a currently running job completes.
- You can set the maximum number of fields a data source can have. The limit is checked whenever a data source is created or updated.
- You can set the maximum file size in megabytes. The limit is checked when a file is uploaded.

Security provider configuration

Allows you to specify the user authentication provider. **Default** uses the default tenant's provider, which was set up during installation and configuration. **LDAP** allows you to authenticate users with an external LDAP server such as Active Directory or OpenLDAP. Specify the settings for the provider and optionally specify filter settings to control the users and groups available in the Principals section.

Naming rules

For anything that can be given a unique name in Analytic Server, such as data sources and projects, the following rules are applied to those names.

- Within a single tenant, names must be unique within objects of the same type. For example, two data sources cannot both be named insuranceClaims, but a data source and a project could each be named insuranceClaims.
- Names are case-sensitive. For example, insuranceClaims and InsuranceClaims are considered unique names.
- Names ignore leading and trailing white space.
- The following characters are invalid in names.

  `~, #, %, &, *, {, }, \\, :, <, >, ?, /, |, ", \t, \r, \n`

Chapter 2. Getting users started

Tell users to navigate to `http://<host>:<port>/<context-root>/admin/<tenant>` and enter their username and password to log on to the Analytic Server console.

<host>
The address of the Analytic Server host.

<port>
The port that Analytic Server is listening on. By default this is 9080.

<context-root>
The context root of the Analytic Server. By default this is analyticserver.

<tenant>
In a multi-tenant environment, the tenant you belong to. In a single-tenant environment, the default tenant is **ibm**.

For example, if the host machine has IP address 9.86.44.232, you have created a "mycompany" tenant and added users to it, and the other settings have been left to their defaults, then users should navigate to `http://9.86.44.232:9080/analyticserver/admin/mycompany` to access the Analytic Server console.

Chapter 3. Analytic Server job names

Analytic Server produces map-reduce and Spark jobs, which can be monitored through your Hadoop cluster's Resource Manager user interface.

The map-reduce job name has the following structure.

```
AS/{tenant name}/{user name}/{algorithm name}
```

{tenant name}
: This is the name of the tenant under which the job is run.

{user name}
: This is the user who requested the job.

{algorithm name}
: This is the primary algorithm in the job. Note that a single stream may generate multiple map-reduce jobs; likewise, several operations within a stream can be contained within a single map-reduce job.

All map-reduce jobs display in the Resource Manager user interface. A single Spark application is started for each Analytic Server. Open the Spark application's user interface to monitor the Spark jobs (the Job names display in the **Description** column).

Chapter 4. IBM SPSS Analytic Server best practices and recommendations

The following sections provide Analytic Server best practices and recommendations regarding data sources, cluster configuration, and IBM SPSS Modeler streams.

Data sources

Analytic Server supports the following data source types:

- File based data sources, such as delimited, fixed text and Microsoft Excel files.
- Relational databases, such as Db2, Oracle, Microsoft SQL Server, Teradata, Postgres, Netezza, MySQL, and Amazon Redshift.
- Hive/HCatalog data sources that include all built-in data types (for example, ORC and Parquet), as well any custom data type for which appropriate Hive Serializer-Deserializer implementation is available. Additionally, Analytic Server can be configured to access NoSQL databases such as HBase, MongoDB, Accumulo, Cassandra, Oracle NoSQL, and other databases for which appropriate Hive Storage Handler implementation is available.
- Geospatial type data sources (shape file based and map services based).

Analytic Server limitations on Hive/HCatalog data sources

- If Hive pushback is required for the SPSS Modeler Select Node, the filtering expression can reference only partitioned columns of type STRING. Starting with Analytic Server 3.0, data type support was added for the following partitioned columns: TINYINT, SMALLINT, INT, BIGINT. The static filtering expression that is specified for the Hive data source can have filtering expressions for partitioned columns of any data type.
- Analytic Server does not support data sources that are based on Hive views.

Cluster configuration - security

Kerberos impersonation

Prior to version 3.0.1, Analytic Server instances utilized a User Principal Name in the Analytic Server keytab for authenticating HDFS operations when Kerberos security was enabled. Starting with version 3.0.1, Analytic Server utilizes a Service Principal Name in the Analytic Server keytab along with the requesting user name (of the user who makes the rest request) to authenticate HDFS operations that utilize Kerberos impersonation. Analytic Server 3.0.1, or greater, is required to add impersonation configuration attributes to HDFS (or the Hive service configurations) when running in a Kerberos enabled cluster. In the case of HDFS, the following properties must be added to the HDFS `core-site.xml` file:

```
hadoop.proxyuser.<analytic_server_service_principal_name> .hosts = *
hadoop.proxyuser.<analytic_server_service_principal_name> .groups = *
```

where `<analytic_server_service_principal_name>` is the default `as_user` value that is specified in the Analytic Server configuration's `Analytic_Server_User` field.

The following properties must also be added to the HDFS `core-site.xml` file in cases where data is accessed from HDFS via Hive/HCatalog:

```
hadoop.proxyuser.hive.hosts = *
hadoop.proxyuser.hive.groups = *
```

Kerberos cross-realm authentication

Analytic Server supports Kerberos cross-realm authentication. To enable this feature, you must first ensure that KDC cross-realm authentication is enabled and then add the following setting to the Analytic Server Ambari configuration's **Custom analytics.cfg** section:

```
kerberos.user.realm.trim = true
```

Cluster configuration - performance tuning settings and results

Spark configuration

Analytic Server uses `yarn-client` mode to interact with YARN and run Spark jobs on the Hadoop cluster.

Analytic Server custom configuration:

- Ambari settings are defined in the Analytic Server Ambari configuration's **Custom analytics.cfg** section.
- Cloudera settings are located in the Cloudera Manager's **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** section.

1. Consider increasing the value of the **spark.driver.memory** configuration setting by adding a configuration item in the Analytic Server custom configuration (when not set explicitly, the default value is `1g`). For example:

   ```
   spark.driver.memory=2g
   ```

2. Select from one of the following Analytic Server with Spark resource usage options.
   - **Option A: Static resource allocation configuration**

     There are 3 parameters that must be configured in the Analytic Server custom configuration:

     ```
     spark.executor.instances
     spark.executor.cores
     spark.executor.memory
     ```

     The following steps describe how to determine the parameter values.

     a. Establish the percentage, in terms of CPU and memory, that Analytic Server can permanently allocate for Spark. This results in a specific number of cores (C) and a fixed amount of memory that can be used in each machine (M).

     b. Establish the number of executors (E) that each machine can run. These executors run as separate Hadoop containers (processes) on each cluster node. Usually a value greater than 2 is appropriate, but the value must be lower than the total number of cores. The memory that is allocated for Spark is divided between these executors, so selecting a high value for this parameter will decrease the amount of memory that is allocated for each container.

     c. Establish the number of cores that are used per each executor (CE). Usually this value is C/E (the number of cores from each machine that are allocated for Spark application, divided by the total number of executors).

     d. Establish the amount of memory that is used for each executor (ME). This is usually M/E.

     Note: The number of executors and cores that are used must be balanced in such a way that the amount of each executor memory should be greater than 3G * CE. Each core from each executor must be allocated at least 3G of memory that will be used as storage or computation memory.

     ```
     spark.executor.instances = <E>*N /<E> // value established in step b where N is the number of compute nodes
     spark.executor.cores = <CE> // value established in step c
     spark.executor.memory = <ME> // value established in step d
     ```

Figure 1. Custom analytics.cfg Spark settings

- **Option B: Dynamic resource allocation configuration**

  When using this option, all of the executors that are allocated by YARN are increased/decreased dynamically according to the entire cluster's actual available resources.

  The minimum configuration is:

  ```
  spark.dynamicAllocation.enabled = true
  spark.shuffle.service.enabled = true
  ```

  A typical configuration is:

  ```
  spark.default.emitter.class = com.spss.ae.spark.ListEmitter
  spark.default.emitter.compressed = false
  spark.dynamicAllocation.enabled = true
  spark.executor.cores = 4
  spark.executor.memory = 16g
  spark.io.compression.codec = snappy
  spark.rdd.compress = true
  spark.shuffle.service.enabled = true
  ```

  Notes:

  – `spark.executor.instances = <E>` should not be used, otherwise Static resource allocation is employed.

  – The considerations regarding the executor cores and memory values are the same as Option A.

3. You can disable Spark cache in the Analytic Server custom configuration using the following settings:

   ```
   spark.cache=false
   spark.storage.memoryFraction = 0.3
   ```

Figure 2. Custom analytics.cfg Spark cache settings

Spark cache should not be disabled when large IBM SPSS Modeler streams are used. Disabling Spark cache in this instance results in slower running streams, but avoids out-of-memory conditions that can occur when the specified amount of memory per executor is small.

JVM configuration

Ambari settings:

1. In the Analytic Server Ambari configuration set the amount of memory that the server can use for local processing. The default value (2 GB) can be safely used for small to medium streams but a higher value heap size (for example, 10 GB) should be used for larger streams.

   Analytic Server > **Configuration** > **Advanced analytic-jvm-options**

2. Replace `-Xmx2048M` with `-Xmx10G`, save the configuration, and restart Analytic Server.

Figure 3. Advanced analytic-jvm-options settings

Cloudera settings:

1. In Cloudera Manager, navigate to the Analytic Server service's **Configuration** tab and update the `jvm-options` control to set the amount of memory that the server can use for local processing. The default value (2 GB) can be safely used for small to medium streams but a higher value heap size (for example, 10 GB) should be used for larger streams.

   Analytic Server service > **Configuration** > **jvm-options**
2. Replace `-Xmx2048M` with `-Xmx10G`, save the configuration, and restart Analytic Server.

Yarn MapReduce2 configuration:

- If you must run MapReduce jobs in parallel with Spark jobs for Analytic Server execution, the Yarn cluster must be configured to have at least 4 GB memory per each Yarn container.

Zookeeper configuration:

- Cloudera requires you to manually update the Zookeeper configuration. For more information, see https://www.cloudera.com/documentation/enterprise/5-4-x/topics/cdh_ig_zookeeper_server_maintain.html.
- If you use complex SPSS Modeler streams or wide data (a large number of fields), you may run into problems with failing jobs due to a broken Analytic Server–Zookeeper connection. The problem is the result of the large program size that SPSS Modeler Server sends to Analytic Server. The problem is less likely to occur in Analytic Server 3.0 (or greater). Use the following steps to resolve the issue:
  1. In the Ambari console, navigate to the Zookeeper service **Configs** tab, add the following line to the `zookeeper-env` template under **Advanced zookeeper-env**, and then restart the Zookeeper service.

     ```
     export JVMFLAGS="-Xmx2048m -Djute.maxbuffer=2097152"
     ```

Figure 4. zookeeper-env template settings

  2. In the Ambari console, navigate to the Analytic Server service's **Configs** tab, add the following to **Advanced analytics-jvm-options**, and then restart the Analytic Server service.

     ```
     -Djute.maxbuffer=2097152
     ```

Figure 5. Advanced analytics-jvm-options settings

Note: If the problem persists, increase the `-Djute.maxbuffer` value from `2097152` to `4194304` in both places.

IBM SPSS Modeler stream recommendations

Note: Most of the following recommendations also apply to small data.

Prototype on small data

When you're experimenting with a stream, you often add a few nodes, test the stream to that point, maybe add a node to check out some tabular or graphical output, and then continue building the stream. You typically cannot afford to do a data pass of your big data every time you test your stream.

Creating a suitable data sample of your big data allows you to test the stream against actual data without incurring the time penalty that is required when you perform a complete data pass. The data sample must contain enough data to the successful run your stream. For example, if you plan to analyze transactions at stores that are located in Minnesota, your data sample must contain transactions from stores in Minnesota.

After sampling, you can:

- Create a cache of the data sample on the cluster where the big data resides, or

  Pros -Simple and does not require switching source nodes

  Cons - The cache disappears when the session is over
- Create a new Analytic Server data source that contains the data sample, or

  Pros - Permanent data source

  Cons -Requires editing/switching source nodes
- Download the data sample to your local system and create a local data source

  Pros -Does not consume cluster resources when prototyping; the SPSS Modeler client is more efficient than Analytic Server when you work with small data.

  Cons - Requires switching source nodes

Create separate Type and Filter nodes from the Source nodes

Every SPSS Modeler source node also has the combined functionality of the Filter and Type nodes. This is useful for keeping the canvas streamlined, but makes it difficult when you switch to different Source node types. Additionally, it obscures the fact that Type and Filter operations are occurring.

Place Filter and Select nodes as close to the Source node as possible

This reduces the number of records in downstream operations.

Avoid the Sort node whenever possible

Analytic Server does not support the optimizations in nodes that depend upon the data being sorted (such as the Merge node). As such, a mid-stream Sort node is rarely doing anything useful. The Sort node does have value when followed immediately by a Sample node in order to get the Top N (or Bottom N) records.

Compute only the fields that will be used

Do not compute a field and then immediately filter it.

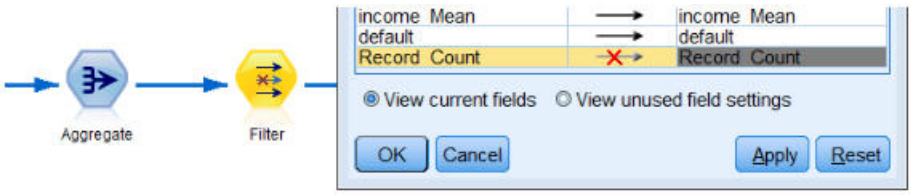

Figure 6. Modeler field options

Whenever possible, without making the expressions hard to understand, avoid creating numerous, temporary fields. For example, instead of defining the following example:

```
now = datetime_now()
birthdate = datetime_date(bYear, bMonth, bDay)
age = date_years_difference(birthdate, now)
```

define the following example instead:

```
age = date_years_difference(datetime_date(bYear, bMonth, bDay), datetime_now())
```

Folding temporaries into inline expressions in this manner can make increase performance when a large numbers of fields are transformed.

Set storage in the data source

Operations that change a field's storage type (for example, string to integer) mid-stream can be detrimental to overall performance. You can set the storage for fields, when defining data sources in the Analytic Server Console, to avoid repeating these conversions.

Use SPSS Modeler when you work with small data

Manipulate big data with Analytic Server and then use SPSS Modeler to finish computations on small data.

Select the appropriate Analytic Server-related stream properties

Configure relevant stream properties (**Tools** > **Options** > **Stream Properties** > **Analytic Server**) and decide whether to allow data processing to drop out of Analytic Server and continue in SPSS Modeler (when a node cannot be run in Analytic Server).

By default, SPSS Modeler is configured to report an error and stop running in this situation. You can bypass the error by changing the setting from `Error` to `Warn` and by adjusting the limit of how much data can be processed in SPSS Modeler. For example, you can update the data transfer rate from the default 10000 record value (if necessary). Note this limit also applies when viewing results that use the SPSS Modeler table node. If the limit is exceeded SPSS Modeler reports that Data fetch exceeded the limit set in the stream properties.

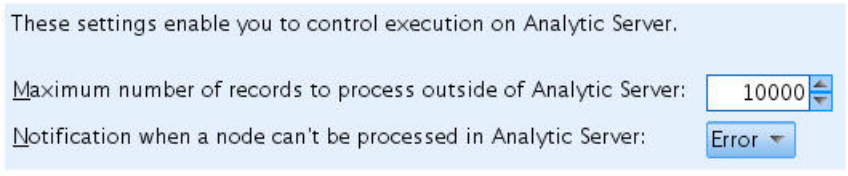

Figure 7. Analytic Server settings

Use Analytic Server Source nodes

Analytic Server can connect to different database data sources but SPSS Modeler requires that all Source nodes be Analytic Server Source nodes (in order for the entire stream to run as an Analytic Server job). For the entire stream to run in Analytic Server, the database source node must be changed to an Analytic Server Source node, and an Analytic Server database data source must be created in the Analytic Server Console.

Consider how unsupported nodes are used

Analytic Server does not support all nodes (the Transpose node is a good example). In order to merge the results of a transpose operation with the rest of the stream, and have it run in Analytic Server, a substream that includes a Transpose node should be written out to an Analytic Server data source that uses an Analytic Server Export node. You can then attach an Analytic Server Source node where the stream was broken to write to Analytic Server.

Note: The transpose operation is suitable for one-time or rarely run operations, but should not be used for routine stream operations.

Determine if a stream will work in Analytic Server before it is run

After you prepare a stream for running in Analytic Server, select a terminal node and use the SPSS Modeler preview feature (the **Preview Run** control on the toolbar) to verify that any nodes that are involved in running the terminal node will work in Analytic Server (without running the stream). Issues are reported in the messages window.

Combine back-to-back Merge operations

Combine a series of Merge nodes into a single node when they have the same keys and join type.

Combine identical substreams

Try to combine identical substreams whenever possible, especially if they contain expensive operations (for example, merge and sort). SPSS Modeler performs these operations once and uses cache to improve performance. In the following example, the streams are identical up to the **newField** node.

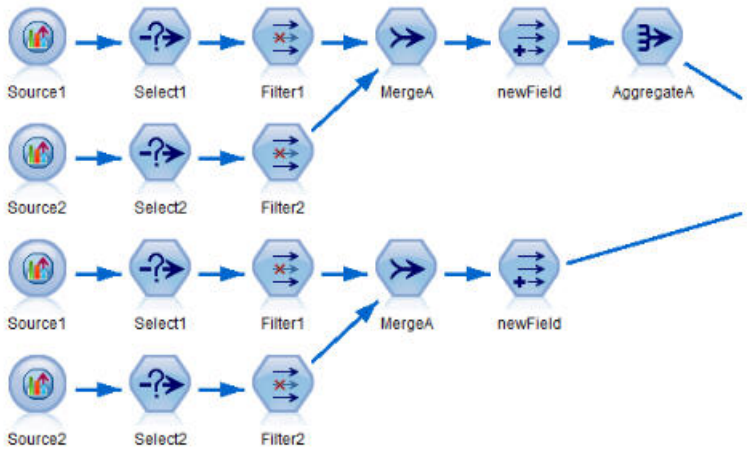

Figure 8. Example stream

It is more efficient (and easier to maintain) if the substream is instead structured as follows:

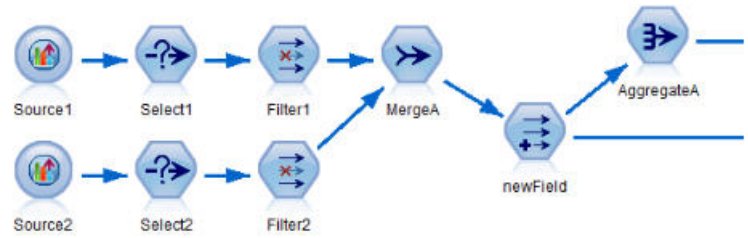

Figure 9. Example stream

Remove extra Type nodes

Figure 10. Example stream

Avoid unnecessary Type nodes when running against Analytic Server. The Type node's `Read Values` operation starts a MapReduce job. This is typically a one-time savings, unless you clear the Type node values.

Fully document each stream

The following example shows a complex stream that contains a number of substreams.

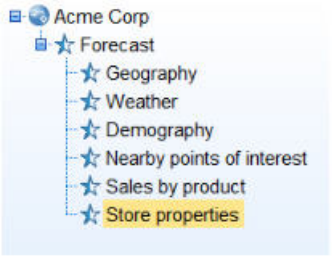

Figure 11. Substream example

In such cases, it is important to properly name the supernodes and document the stream (just as you would document code). A clear comment can provide invaluable information to other analysts who read or maintain the stream. For example:

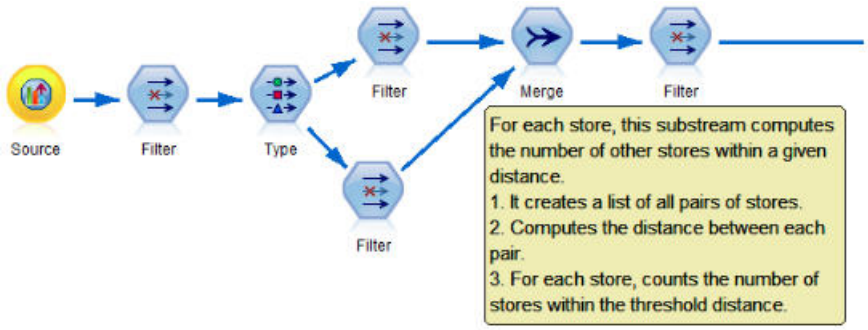

Figure 12. Stream example with comments

When developing streams, use SPSS Modeler caches to quickly store intermediate results

In streams that run against Analytic Server, node caching works by storing the data in a specific part of the stream to temporary files on HDFS (as opposed to storing on the SPSS Modeler server). Caches work well with big data and are safe to use in streams that run on Analytic Server.

Chapter 5. Troubleshooting

Analytic Server provides several helpful tools for problem determination.

Logging

Analytic Server creates customer log files and trace files that are helpful for diagnosing problems. With the default Liberty installation, you can find the log files in the {AS_ROOT}/ae_wlpserver/usr/servers/aeserver/logs directory.

The default logging configuration produces two log files that roll over on a daily basis.

as.log This file contains the high-level summary of informational warning and error messages. Check this file first when server errors occur that cannot be resolved by using the error message that is displayed in the User interface.

as_trace.log
This file contains all the entries from ae.log, but adds more information that is primarily targeted to IBM support and development for debugging purposes.

Analytic Server uses Apache LOG4J as its underlying logging facility. Using LOG4J, the logging can be dynamically adjusted by editing the {AS_SERVER_ROOT}/configuration/log4j.xml configuration file. You may be asked to do this by Support to help diagnose problems, or you may want to modify this to limit the number of log files kept around. Changes to the file are detected automatically within a few seconds so the Analytic Server does not need to be restarted.

For more information about log4j and the configuration file, see documentation at the official Apache website at http://logging.apache.org/log4j/.

Version information

You can determine what version of Analytic Server is installed by checking the {AS_ROOT}/properties/version folder. The following files contain version information.

IBM_SPSS_Analytic_Server-*.swtag
Contains detailed product information.

version.txt
Version and build number for the installed product.

Log collector

When problems cannot be resolved by directly reviewing the log files, you can bundle all the logs and send them to IBM support. There is a utility that is provided to make collecting all the necessary data simpler.

Using a command shell, run the following commands:

```
cd  {AS_ROOT}/bin
run  >sh ./logcollector.sh
```

These commands create a compressed file under {AS_ROOT}/bin. The compressed file contains all the log files and product version information.

Common issues

This section describes some common administration issues and how you can fix them.

Security

Kerberos authentication fails when trying to access HCatalog data source

If you see errors in the log like the following:

```
cause:javax.security.sasl.SaslException: Failure to initialize security context

com.spss.analyticframework.api.exceptions.ComponentException: Cannot access HCatalog
```

You must ensure that the HDFS user's Kerberos TGT is cached and available on the Analytic Server server host. To do this:

1. Stop the Analytic Server process.
2. Run `kinit -f $hdfs.user` from the Analytic Server host, where **`$hdfs.user`** is as defined in the `config.properties` file and has write permission to analytic root
3. Start Analytic Server.

Analytic Server Console

Accessing the Analytic Server Console from Safari on iOS

The tenant status dropdown list does not work, and you cannot update the data model for File-based data sources. Use another browser when performing these actions.

Running streams

R jobs translate non-English words to Unicode

On Cloudera clusters, if the system encoding of Hadoop servers is not UTF-8, R translates the non-English words into Unicode.

1. Navigate to the YARN configuration tab in the Cloudera Manager console.
2. Add the following settings in the "NodeManager Environment Advanced Configuration Snippet (Safety Valve)" field.

```
LC_ALL=""
LANG=en_US.utf8
```

PySpark jobs fail to run

Ensure that the Spark service is deployed in all Analytic Server nodes and in all node managers.

PySpark jobs fail to run on Kerberos enabled environments

You must run the `kinit` command and then restart Analytic Server before PySpark tests will run successfully. For example:

HDP Kerberos

```
cd /etc/security/keytabs/
sudo -u as_user kinit -k -t as_user.headless.keytab as_user/lyrh1.fyre.ibm.com@IBM.COM
```

CDH Kerberos

```
cd /run/cloudera-scm-agent/process/387-analyticserver-ANALYTIC_SERVER
sudo -u as_user kinit -k -t analyticserver.keytab as_user/cdh12-1.fyre.ibm.com@IBM.COM
```

Memory errors

Configuring YARN after executor memory errors

The following error can occur when the required executor memory is above the maximum threshold:

```
Caused by: com.spss.mapreduce.exceptions.JobException:
    java.lang.IllegalArgumentException: Required executor memory (1024+384  MB) is above the max
    threshold (1024 MB) of this cluster! Please  increase the value of
    'yarn.scheduler.maximum-allocation-mb'.
```

The following steps provide the YARN configuration settings that are required to resolve the issue.

For Ambari

1. In the Ambari user interface, go to **YARN** > **Configs** > **Settings**.
2. Increase the memory **node (the memory that is allocated for all YARN containers)** to `8192MB`.
3. Increase the container values:
   - **Minimum Container Size (Memory)** to `682MB`
   - **Maximum Container Size (Memory)** to `8192MB`
4. Increase the **Maximum Container Size (VCores)** to `3`.
5. Restart YARN, Spark, and the Analytic Server service.

For Cloudera

1. Increase the `yarn.nodemanager.resource.memory-mb` to 8GB
   - In the Cloudera Manager user interface, go to **Yarn service** > **Configurations** > **Search Container Memory** and increase the value to `8GB`.
2. In the Cloudera Manager user interface, go to **YARN service** > **Quick Links** and select **Dynamic Resource Pools**.
3. Under **Configuration**, click **edit** for each of the available pools, and under **YARN** set the **Max Running Apps** value to `4`.
4. Restart YARN, Spark, and the Analytic Server service.

HCatalog errors

HCatalog cases can fail with the error: "org.apache.adopt.hive.ql.oi.Acidulous.isTablePropertyTransactional(Ljava/util/Map;)"

The error is the result of a *.jar file conflict issue. Because Analytic Server cannot modify code to control which control classes are loaded, the solution is to manually remove the related class information from the `spark-assembly-*.jar` file.

1. Retrieve the `spark-assembly-*.jar` file from the following directory: `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/modules/spark`.

   Note: For Hortonworks Data Platform 2.6, the `spark-assembly-*.jar` file is located in the following directory: `/usr/hdp/2.6.1.0-129/spark/lib`.
2. Use the `jar` command to decompress the package. For example:
   ```
   jar xvf spark-assembly-1.6.3.2.6.1.0-129-hadoop2.7.3.2.6.1.0-129.jar
   ```
3. Remove `org\apache\hadoop\hive\ql\io\AcidUtils.class` and all `AcidUtils$*.class` references.
4. Use the `jar` command to reassemble the package. For example:
   ```
   jar cfv spark-assembly-1.6.3.2.6.1.0-129-hadoop2.7.3.2.6.1.0-129.jar *
   ```
5. Copy the updated `*.jar` file to the Analytic Server server, replacing the existing file.
6. Restart Analytic Server.

Hadoop with Apache Spark 2.x

- Most `forcespark` and `forcehadoop` jobs fail when Hadoop and Apache Spark 2.x exist in the same environment. The issue can be resolved by manually editing the `/etc/spark2/conf/spark-defaults.conf` file as follows:
  ```
  #spark.hadoop.mapreduce.application.classpath=
  #spark.hadoop.yarn.application.classpath=
  ```
- When two JDK versions are installed on the same system, Cloudera uses JDK 1.7 while Spark 2.x uses JDK 1.8. Running `forcespark` or `forcehadoop` jobs with Apache Spark 2.x may result in all jobs failing with the following error message:

```
Execution failed. Reason: org/apache/spark/api/java/function/PairFunction : Unsupported major.minor version 52.0
```

For Cloudera, add the following line in Cloudera Manager's **Analytic Server Advanced Configuration Snippet (Safety Valve) for server.env** section:

```
JAVA_HOME=/usr/java/jdk1.8.0_152
```

Performance tuning

This section describes ways to optimize the performance of your system.

Analytic Server is a component in the Ambari framework that utilizes other components such as HDFS, Yarn and Spark. Common performance tuning techniques for Hadoop, HDFS and Spark apply to Analytic Server workloads. Every Analytic Server workload is different therefore tuning experimentation is required based on your specific deployments workload. The following properties and tuning tips are key changes that have impacted the results of the Analytic Server benchmarking and scaling tests.

When the first job runs on Analytic Server, the server will start a persistent Spark application that will be active until the Analytic Server is shut down. The persistent Spark application will allocate and hold onto all the cluster resources allocated to it for the duration of the Analytic Server running, even if an Analytic Server job is not actively running. Careful thought should be given to the amount of resources allocated to the Analytic Server Spark application. If all cluster resources are allocated to the Analytic Server Spark application, then other jobs could be delayed or not run. These jobs could be queued waiting for sufficient free resources and those resources will be consumed by Analytic Server Spark application.

If multiple Analytic Server services are configured and deployed, each service instance could potentially allocate its own persistent Spark application. For example, if two Analytic Server services are deployed to support high availability failover, then you could see two persistent Spark applications active, each allocating cluster resources.

An additional complexity is that in certain situations, Analytic Server may start a map reduce job that will require cluster resources. These map reduce jobs will require resources that are not allocated to the Spark application. The specific components that require map reduce jobs are PSM model builds.

The following properties can be configured to allocate resources to Spark application. If they are set in the spark-defaults.conf of the Spark installation, then they are allocated for all Spark jobs run in the environment. If they are set in the Analytic Server configuration as custom poperties under the “Custom analytic.cfg” section, then they are allocated for the Analytic Server Spark application only.

spark.executor.memory

Amount of memory to use per executor process.

spark.executor.instances

The number of executor processes to start.

spark.executor.cores

The number of executor worker threads per executor process. This value should be between 1 and 5.

An example of setting the three key Spark properties. There are 10 data nodes in a HDFS cluster and each data node has 24 logical cores and 48 GB of memory and is only running HDFS processes. Here is one way to configure the properties for this environment, assuming you are only running Analytic Server jobs on this environment and desire maximum allocation to a single Analytic Server Spark application.

- Set spark.executor.instances=20. This would attempt to run 2 Spark executor processes per data node.
- Set spark.executor.memory=22G. This would set the max heap size for each Spark executor process to 22 GB, allocating 44 GB on each data node. Other JVMs and the OS need the extra memory.

- Set spark.executor.cores=5. This will provide 5 worker threads for each Spark executor, for a total of 10 worker threads per data node.

Monitor the Spark UI for running jobs

If you see Spill to disk that could impact performance. Some possible solutions are:

- Increase memory and allocate it to Spark executors via **spark.executor.memory**.
- Reduce the number of **spark.executor.cores**. This will reduce the number of concurrent work threads allocating memory, but it will also reduce the amount of parallelism for the jobs.
- Change the Spark memory properties. **spark.shuffle.memoryFraction** and **spark.storage.memoryFraction** allocation percentage of the Spark executor heap for Spark.

Ensure the name node has enough memory

If the number of blocks in HDFS is large and growing, ensure you name node heap increases to accommodate this growth. This is a common HDFS tuning recommendation.

Alter the amount of memory used for caching

By default, **spark.storage.memoryFraction** has value 0.6. This can be increased up to 0.8 in case the HDFS block size of the data is 64MB. If the HDFS block size of the input data is greater than 64MB then this value could be increased only if the memory allocated per task is greater than 2GB.

Tuning Performance of Model Scoring

You can improve the performance of model scoring jobs on big datasets with the Apache Spark engine by using the following steps. Note that these steps should not impact the operation of non-Analytic Server services on the cluster.

1. Check if libtcmalloc_minimal.so{/version} is already installed on each node in the cluster.
   ```
   whereis libtcmalloc_minimal.so.*
   ```
2. If libtcmalloc_minimal.so is not installed, either install the operating system specific package containing the libtcmalloc_minimal library on each node in your cluster, or manually build and install libtcmalloc_minimal. For example:

   Ubuntu:

   ```
   sudo apt-get install libgoogle-perftools-dev
   ```

   Red Hat Enterprise Linux 6.x (x64):

   a. Install the EPEL repository for RedHat (if not already installed)
      ```
      wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
      sudo rpm -Uvh epel-release-6*.rpm
      ```
   b. `sudo yum install gperftools-libs.x86_64`

   Manual Build:

   a. Download the gperftools-2.4.tar.gz from the link https://github.com/gperftools/gperftools/releases
   b. `tar zxvf gperftools-2.4.tar.gz`
   c. `cd gperftools-2.4`
   d. `./configure --disable-cpu-profiler --disable-heap-profiler --disable-heap-checker --disable-debugalloc --enable-minimal`
   e. `make`
   f. `sudo make install`
3. Note one of the location(s) of the installed library file libtcmalloc_minimal.so{.version}, as returned from running the following command on one or more of the nodes.

```
whereis libtcmalloc_minimal.so.*
```

If the cluster has nodes running a mixture of operating systems, there could be multiple locations for this file.

4. In the Ambari console, go to the Analytic Server configuration and under section Custom analytics.cfg, configure the key spark.executorEnv.LD_PRELOAD using the location of the library as the value. After making this change restart the Analytic Server service. For example, if the library is installed to /usr/lib64/libtcmalloc_minimal.so.4, the configuration would be:

   ```
   spark.executorEnv.LD_PRELOAD=/usr/lib64/libtcmalloc_minimal.so.4
   ```

   If multiple locations are required, use a space to separate them, as in the following example.

   ```
   spark.executorEnv.LD_PRELOAD=/usr/lib64/libtcmalloc_minimal.so.4 /usr/lib/libtcmalloc_minimal.so
   ```

   If any nodes do not have the `libtcmalloc_minimal.so` library installed at one of the configured locations, this will not cause an error, but performance of model scoring may be slower on these nodes.

Spark map-side join

The Analytic Server Spark join implementation does not support map-side join functionality (Spark join is mainly a reduce side). The implementation does not take advantage of map-side joins to optimize joins when one input is small. Not taking advantage of map-side join results in an extremely resource intensive Spark job that eventually fails.

To optimize joins when running Analytic Server Spark map-side joins (or native Spark jobs that are based on the smallest RDD size), you can add the `spark.msj.maxBroadcast` property to the `analytics.cfg` file (`SPSS Analytic Server/Configs/Custom analytics.cfg`), or to `analytics-meta`.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Printed in USA