Pivotal™

Rev: A02
Updated: November 12, 2013

Welcome to Pivotal Advanced Database Services 1.1.3

Pivotal Advanced Database Services (ADS), extends Pivotal Hadoop (HD) Enterprise, adding rich, proven parallel SQL processing facilities. These SQL processing facilities enhance productivity, rendering Hadoop queries faster than any Hadoop-based query interface on the market. Pivotal ADS enables data analysis for a variety of Hadoop-based data formats using the Pivotal Extension Framework (PXF), without duplicating or converting HBase files. For the best performance, you can use an optimized format for Pivotal ADS table storage.

About Pivotal, Inc.

Greenplum is currently transitioning to a new corporate identity (Pivotal, Inc.). We estimate that this transition will be completed in Q2 2013. During this transition, there will be some legacy instances of our former corporate identity (Greenplum) appearing in our products and documentation. If you have any questions or concerns, please do not hesitate to contact us through our web site: `http://www.greenplum.com/support-transition`.

About PADS Components

PADS comprises the following components:

- HAWQ
- PXF
- MADlib

HAWQ

HAWQ is a parallel SQL query engine that combines the key technological advantages of the industry-leading Greenplum Database with the scalability and convenience of Hadoop. HAWQ reads data from and writes data to HDFS natively.

Using HAWQ functionality, you can interact with petabyte range data sets. HAWQ provides users with a complete, standards compliant SQL interface.

Leveraging Greenplum Database's parallel database technology, HAWQ consistently performs tens to hundreds of times faster than all Hadoop query engines in the market.

PXF

PXF enables SQL querying on data in the Hadoop components such as HBase, Hive, and any other distributed data file types. These queries execute in a single, zero materialization and fully-parallel workflow. PXF also uses the PADS advanced query optimizer and executor to run analytics on these external data sources, or transfers it to PADS to analyze locally. PXF connects Hadoop-based components to facilitate data joins, such as between HAWQ tables and HBase table. Additionally, the framework is designed for extensibility, so that user-defined connectors can provide parallel access to other data storage mechanisms and file types.

PXF Interoperability

PXF operates as an integral part of PADS, and as a light add-on to Pivotal HD. On the database side, PXF leverages the external table custom protocol system. Therefore, creating a PXF table provides the same interoperability as an external table in Greenplum Database. On the Pivotal HD side, the PXF component physically lives on the Namenode and each or some Datanodes. It operates mostly as a separate service and does not interfere with Hadoop components internals.

MADlib

MADlib is an open-source library for scalable in-database analytics. It provides data-parallel implementations of mathematical, statistical and machine learning methods for structured and unstructured data. MADlib combines the efforts used in commercial practice, academic research, and open-source development. You can find more information at http://madlib.net.

New Features

This section describes the new features in PADS 1.1.x

New Features in PADS 1.1.3

User Defined Functions

This release includes support for creating user defined functions (UDF) in C, SQL, PL/PGSQL using HAWQ.

PADS 1.1.3 does not support the following UDF functionality:

- Creating procedural languages.
- SECURITY DEFINER clause
- User defined aggregates
- Set returning functions
- TABLE functions
- Invoking Nested functions

pgcrypto support

You can install the pgcrypto module as a package. It provides PostgreSQL cryptographic functions for HAWQ. For more information see, http://www.postgresql.org/docs/8.3/static/pgcrypto.html.

GSSAPI Authentication Methods

RFC 2743 defines GSSAPI as the industry-standard protocol for secure authentication. HAWQ supports GSSAPI with Kerberos authentication based on RFC 1964. You can use GSSAPI to provide automatic authentication (single sign-on) for systems that support it.

Secure HDFS

HDFS security prevents malicious user impersonation. The HDFS daemons leverage Kerberos to perform user authentication on all remote procedure calls (RPCs).

HAWQ does not support the following HDFS security configuration:

- `hadoop.rpc.protection` other than authentication
- `dfs.encrypt.data.transfer` to `true`

New Features in PADS 1.1.2

HAWQInputFormat

HAWQInputFormat is a JAVA library that provides MapReduce jobs with direct access to HAWQ data stored on HDFS.

New Features in PADS 1.1.1

ORCA

ORCA is the new optimizer that extends the functionality of the existing planner to achieve better optimization results. Please see the *Pivotal ADS 1.1.2 Administrator Guide* for more information.

gpcheck enhancement

gpcheck is enhanced to check configuration settings such as disk usage, mount points, and HDFS-specific settings such as NameNode and DataNode configurations.

HDFS access layer fault tolerance enhancement

The HDFS access layer (libhdfs3) is enhanced to fail over to another DataNode when a read fails.

CSV Support

PADS 1.1.1 supports CSV formatted files, using `FORMAT 'CSV'` (standard HAWQ CSV formatting option).

New Accessor Classes

PADS 1.1.1 now supports two new Accessor classes:

- QuotedLineBreakAccessor
- LineReaderAccessor

Changes to Resolver Classes

Note the changes to the following Resolver classes:

- TextAccessor: This class has been deprecated in PADS 1.1.1. Use the class LinReaderAccessor.

- TextResolver: This class has been deprecated in PADS 1.1.1. Use the class StringPassResolver.

New Data Type

PADS 1.1.1 supports the new Hive 11 data type, `DECIMAL`.

New Feature in PADS 1.1

JBOD Support

JBOD is an array of drives, which allow you to access each drive independently. If HAWQ segment hosts are installed on JBOD, HAWQ can use the disks to store large amounts of data such as intermediate results on workfiles. Storing intermediate results on the workfiles is especially useful during query execution. HAWQ with JBOD support uses multiple disks for multiple sessions. Disks are cycled at each session.With JBOD support, the workfile IO will be balanced across disks.

Enabling JBOD support in HAWQ is simple. You can create directories during HAWQ initialization and add a new configuration parameter for `gpinitsystem`. See the Pivotal HAWQ Installation Guide for more details on enabling this feature.

About the ADS 1.1.3 Release

Please refer to the following sections for more information about this release.

- Supported Platforms
- Installation options
- Resolved Issues in PADS 1.1.3
- Known Issues in PADS 1.1.x
- Pivotal and Greenplum Interoperability
- PADS 1.1.3 and Pivotal HD Documentation
- Use of Open Source
- Upgrading to HAWQ 1.1.x
- Troubleshooting a Failed Upgrade

Supported Platforms

PADS 1.1.3 supports the following platforms:

- Red Hat Enterprise 6.4-64 bit and 6.2-64 bit
- CentOS 6.4-64 bit and 6.2-64 bit

Installation options

There are two ways to install HAWQ.

- Stand alone install – Please see *HAWQ 1.1.3.0 Installation Guide*
- ICM install – Please see *Pivotal HD Enterprise 1.0 Installation and Administrator Guide.*

Resolved Issues in PADS 1.1.3

The table below lists issues that are now resolved in PADS 1.1.3.

For issues resolved in prior releases, refer to the corresponding release notes available from Support Zone.

Table 1 Resolved Issues in PADS 1.1.3

Issue	Category	Resolved in	Description
HAWQ-216	Management Tools	PADS 1.1	`gpstart` failed if the cluster is busy. This issue did not happen frequently, since the system is idle at start time. However, it is important to remember that it could happen because the system may have other non-HAWQ workload on the same cluster.
HAWQ-245	PXF	PADS 1.1	Could not add PXF-specific logic to `gp_external_max_segments`. Modifying this parameters allows you to achieve the maximum possible distribution on hosts. You can also limit the number of segments that are running.
HAWQ-246	PXF	PADS 1.1	A `segmentdb` could not read a block located on the same machine. The algorithm was enhanced to account for cases where PXF services are located on a remote region server.
HAWQ-282	PXF	PADS 1.1	The ANALYZE command silently ignored an incorrect Analyzer name when analyzing a PXF table. A WARNING is now issued.
HD-2362	PXF	PADS 1.1	The select query fails for an external table created on views for any HIVE table. A more comprehensive error message is now given.
HAWQ-990	AO tables and Column Store	PADS 1.1.3	**AOInputFormat: bytesTo DecimalStr throws the ArrayIndexOutOf BoundsException**. Occurs when a table has columns with datatypes that are greater and less than 8 characters long. For example, the table may contain timestamps and int4. If all the 8-length columns in a tuple have a null value, HAWQInputFormat throws the ArrayIndexOutOfBoundsException exception.
HAWQ-1023	HDFS Access Layer	PADS 1.1.3	HAWQInputFormat throws the ClassCastException if it reads an invalid metadata file.
HAWQ-1031	AO tables and Column Store	PADS 1.1.3	You will see the NullPointerException if you try to access a column using the wrong column name.
HAWQ-1032	AO tables and Column Store	PADS 1.1.3	If you inserted a large block of data, for example about 2M, in a single query, HAWQInputFormat will not be able to access it. **Workaround**: insert the data in multiple queries
ARD-132	PXF	PADS 1.1.3	During a join between two PXF tables, the optimizer may periodically hold the HBase side of the scan. In such a situation, the PXF HBase scanner timeout exception is not caught. **Workaround**: Increase the HBase scanner timeout.

Known Issues in PADS 1.1.x

This section lists all the known issues in PADS 1.1.x. A workaround is provided where applicable:

Important: Pivotal Hadoop (PHD) 1.1. has a new High Availability (HA)feature. This feature has the following known issue with HAWQ: If NameNode HA is configured in a cluster, HAWQ is unable to take advantage of HA capability when the primary

NameNode fails. Pivotal recommends that if you want to use HAWQ, you should not configure HA for HDFS. In the cas of the primary NameNose failing, HAWQ will not be able to failover to the secondary NameNode. Manual re-direction to secondary NameNode is not currently supported in this release. Therefore we highly recommend that HA should be disabled when used with HAWQ.

Known Issues in HAWQ 1.1.3

Table 2 All Known Issues in HAWQ 1.1.3

Issue	Category	Description
HAWQ-1167	Performance	Enabling Kerberos shows a 10% downgrade in HAWQ performance.
HAWQ-1099	Connectivity	If you enable kerberos authentication, the ODBC function SQL GetInfo returns an incorrect version of HAWQ.
HAWQ-1078	Query execution	Continuously issue deepslice queries cause error in HDFS with kerberos.
HAWQ-1056	DML	Inserting data into a temp table generates an Append-only Storage Write error.
HAWQ-1022	Utility commands	Default value for dfs.client.socket-timeout in HDFS setting in gpcheck need to be corrected.

Known Issues in HAWQ 1.1.0.3

Table 3 All Known Issues in HAWQ 1.1.0.3

Issue	Category	Description
HAWQ-859	Query Optimzer	**pg_dumpall test suite runs slowly**. The overhead is due to the command pg_dumpall. pg_dumpall generates multiple queries over the catalog tables. Since ORCA optimizes these queries. Although these are simple queries, ORCA adds the overhead. **Workaround**: Turn ORCA off.

Known Issues in HAWQ 1.1.0.1

Table 4 All Known Issues in HAWQ 1.1.0.1

Issue	Category	Description
HAWQ-256	Storage	HAWQ does not support kerberos authentication on HDFS.
HAWQ-255	Network	HAWQ does not support the IPv6 protocol.
HAWQ-225	Storage	When the number of partitions or columns of a column oriented table is large or write concurrency is high, HAWQ encounters an HDFS concurrency write limitation. Data loading performance may degrade and fail. **Workaround:** for partitioned tables, load data partitions one by one, instead of loading all the data randomly to all the partitions.
HAWQ-224	Backup and Restore	Only non-parallel logical backup and restore is supported. Pivotal recommends that you use physical backup and restore.
HAWQ-26	DDL	**`duplicate key violates unique constraint pg_type_typname_nsp_index`** When two sessions attempt to create a table with the same name and in the same namespace, one of the sessions will error out with a less user-friendly error message of the form "duplicate key violates unique constraint".

Known Issues in PXF 2.x.x

This section lists the known issues in PXF 2.x.x. A workaround is provided where applicable:

Table 5 All Known Issues in PXF 2.0.x

Issue	Component and Version	Description
HAWQ-1142	PXF 2.0.3	If you write the ANALYZE command on PXF writable tables it fails with an error. **Workaround**: Add FRAGMENTER or PROFILE parameters to the location of the PXF writable table. The ANALYZE command will issue a warning, but will not fail.
HAWQ-1111	PXF 2.0.3	HAWQ does not report the details of a BadRecordException error message generated in PXF. **Workaround**: You must look for the details in the PXF log on HDFS.
HAWQ-1133	PXF 2.0.3	A query might fail if the same DN has to read several BZip2 jobs at the same time. This is caused because BZip2 codec is not thread safe. **Workaround**: Pivotal recommends that you do not read more than one BZip2 files at the same time.
HAWQ-1141	PXF 2.0.2	If the rows in a Hive array are in different sizes, HiveResolver returns incorrect data.
HD-6122	PXF 2.0.2	PXF only works with HiveServer1, not HiveServer2.
HAWQ-291	PXF 2.0.1	HDFS does not work properly when accessing data files that contain header rows. **Workaround**: Specify an error table with the “no HEADER” flag.

Pivotal and Greenplum Interoperability

Pivotal releases a number of client tool packages on various platforms that can be used to connect to PADS. The following table describes the client tool package compatibility with PADS. Client tool packages are available at the EMC Download Center.

Table 6 Interoperability matrix

Client Package	Description of Content	Operating system	Client version	HAWQ version
Connectivity	Standard PostgreSQL Database Drivers (ODBC, JDBC)	Windows 2008 RedHat 6.4 and 6.2, 64 bit	4.2.6 SP	1.1.3
HAWQ Client	Command-line interface	RedHat 6.4 and 6.2, 64 bit	4.2.6 SP	1.1.3
Pivotal Command Center	A web-based tool for managing and monitoring your Pivotal HD cluster. Note: Pivotal Command Center 2.0.x does not support DCA V1, DCA V2 or Greenplum Database.	RedHat 6.4 and 6.2, 64 bit CentOS 6.4 and 6.2, 64 bit	2.1	1.1.3

Table 6 Interoperability matrix

Client Package	Description of Content	Operating system	Client version	HAWQ version
PXF	Extensibility layer to provide support for external data formats such as HBase and Hive.	RedHat 6.4 and 6.2, 64 bit CentOS 6.4 and 6.2, 64 bit	2.1.0	1.1.3
Pivotal HD	Pivotal Hadoop	RedHat 6.4 and 6.2, 64 bit CentOS 6.4 and 6.2, 64 bit	1.1	1.1.3
Pivotal Data Loader	Data Loader is a management tool that loads data to distributed data analytics platforms such as Hadoop, and Greenplum database.	RedHat 6.4 and 6.2, 64 bit CentOS 6.4 and 6.2, 64 bit	2.0.3	1.1.3

PADS 1.1.3 and Pivotal HD Documentation

The following PADS and related documentation is available in PDF format on our website at www.gopivotal.com. Additionally, you can still access product documentation from EMC's Support Zone:

Make sure the hypertext link for gopivotal is to this path:

http://gopivotal.com/pivotal-products/pivotal-data-fabric/pivotal-hd.

Table 7 HAWQ documentation

Title	Revision
Pivotal PADS 1.1.3 Release Notes (This document)	A01
Pivotal HAWQ 1.1 Installation Guide	A08
Pivotal ADS 1.1 Administrator Guide	A06
Pivotal HD Enterprise 1.1 Installation and Administrator Guide	A01
Pivotal HD DataLoader 2.0 Installation and User Guide	A05
Pivotal HD 1.1 Stack and Tool Reference Guide	A01
Pivotal Command Center 2.1 User Guide	A01
Pivotal Extension Framework Installation and User Guide	A01

Upgrading to HAWQ 1.1.x

The upgrade path supported for this release is HAWQ 1.0.x to HAWQ 1.1.x.

Note: Pivotal recommends that you back up any existing data before upgrading to HAWQ1.1.x.

For detailed upgrade procedures and information, see the following sections:

- Upgrading from HAWQ 1.1.x to HAWQ 1.1.y
- Upgrading from 1.0.x to HAWQ 1.1.x

Note: Follow these instructions if you installed HAWQ manually. To upgrade PHD Manager, see the Pivotal HD Enterprise 1.0 Installation and Administration Guide.

Upgrading from HAWQ 1.1.x to HAWQ 1.1.y

An upgrade from HAWQ 1.1.x to HAWQ 1.1.y involves stopping HAWQ, updating the HAWQ software binaries, and restarting HAWQ.

1. Log in to your HAWQ master host as the HAWQ administrative user:

   ```
   $ su - gpadmin
   ```

2. Perform a smart shutdown of your current HAWQ 1.1.x system (shut down all active connections to the database):

   ```
   $ gpstop
   ```

3. Run the installer for 1.1.y on the HAWQ master host using rpm. This installs HAWQ to `/usr/local/hawq-1.1.y` alongside any older versions, and it will point a soft link from `/usr/local/hawq to /usr/local/hawq-1.1.y`.

   ```
   $ su - root
   # rpm -ivh hawq-1.1.y.x86_64.rpm --force
   ```

4. Run the following command to install the HAWQ 1.1.y binaries on all the hosts specified in the *hostfile*:

   ```
   # gpssh -f hostfile -e "rpm -ivh hawq-1.1.y.x86_64.rpm
   --force"
   ```

5. After all segment hosts have been upgraded, you can log in as `gpadmin` user and restart your HAWQ system:

   ```
   $ su - gpadmin
   $ gpstart
   ```

Upgrading from 1.0.x to HAWQ 1.1.x

This section describes how you can upgrade from HAWQ 1.0.x or later to HAWQ 1.1.x.

This section divides the upgrade into the following phases: pre-upgrade preparation, software installation, upgrade execution, and post-upgrade tasks.

Important: Carefully evaluate each section and perform all required and conditional steps. Failing to perform any of these steps can result in an aborted upgrade, placing your system in an unusable or even unrecoverable state.

Pre-Upgrade Preparation

Perform these steps on your current HAWQ system. This procedure is performed from your HAWQ master host and should be executed by the HAWQ superuser (`gpadmin`).

1. Log in to the HAWQ master as the `gpadmin` user:

   ```
   $ su - gpadmin
   ```

2. (*optional*) Vacuum all databases prior to upgrade. For example:

   ```
   $ vacuumdb database_name
   ```

3. (*optional*) Clean out old server log files from your master and segment data directories. For example, to remove log files from 2011 from your segment hosts:

   ```
   $ gpssh -f seg_host_file -e 'rm
   /gpdata/*/gp*/pg_log/gpdb-2011-*.csv'
   ```

   Note: Running Vacuum and cleaning out old logs files is not required, but it will reduce the size of HAWQ files to be backed up and migrated.

4. Run `gpstate` to check for failed segments.

   ```
   $ gpstate
   ```

5. If you have failed segments, you must recover them using `gprecoverseg` before you can upgrade.

   ```
   $ gprecoverseg
   ```

6. Copy or preserve any additional folders or files (such as backup folders) that you have added in the HAWQ data directories or `$GPHOME` directory. Only files or folders strictly related to HAWQ operations are preserved by the migration utility.

Install the HAWQ Software Binaries

1. Run the installer for 1.1.x on the HAWQ master host using rpm. This installs HAWQ to `/usr/local/hawq-1.1.x` alongside any older versions, and it will point a soft link from `/usr/local/hawq to /usr/local/hawq-1.1.x`.

   ```
   $ su - root
   # rpm -ivh hawq-1.1.x.x86_64.rpm --force
   ```

2. Run the following command to then install the HAWQ 1.1.x binaries on all the hosts specified in the *hostfile*.

   ```
   # gpssh -f hostfile -e "rpm -ivh hawq-1.1.x.x86_64.rpm
   --force"
   ```

Upgrade Execution

During upgrade, all client connections to the master are locked out. Before performing this procedure, inform all database users of the upgrade and lockout time frame. From this point onward, be sure that no one is on the system until the upgrade is complete.

1. Source the path file from your old 1.0.x installation. For example:

   ```
   $ source /usr/local/hawq-1.0.x/greenplum_path.sh
   ```

2. If your system has a standby master host configured, remove the standby master from your system configuration. For example:

   ```
   $ gpinitstandby -r
   ```

3. Perform a clean shutdown of your current ADS system. For example:

   ```
   $ gpstop
   ```

4. Source the path file from your new HAWQ 1.1.x installation. For example:

   ```
   $ source /usr/local/hawq-1.1.x/greenplum_path.sh
   ```

5. As `gpadmin`, run the 1.1.x version of the migration utility specifying your old and new `GPHOME` locations. If your system does not have mirrors, use `gpmigrator`. For example on a system with mirrors:

   ```
   $ su - gpadmin
   $ gpmigrator /usr/local/hawq-1.0.x /usr/local/hawq-1.1.x
   ```

   `Note: If the migration does not complete successfully, contact Customer Support (see` "Troubleshooting a Failed Upgrade" `on page` 14`).`

6. The migration can take a while to complete. After the migration utility has completed successfully, the HAWQ 1.1.x system will be running and accepting connections.

Post-Upgrade (on your HAWQ 1.1.x system)

1. If your system had a standby master host configured, reinitialize your standby master using `gpinitstandby`:

   ```
   $ gpinitstandby -s standby_hostname
   ```

2. If your system uses external tables with `gpfdist`, stop all `gpfdist` processes on your ETL servers and reinstall `gpfdist` using the compatible HAWQ 1.1.x Load Tools package. Application Packages are available at the EMC Download Center.

3. If you want to use the Pivotal Command Center management tool, install the latest Command Center Console. To update your environment variable to point to the latest Command Center binaries, source the `gpperfmon_path.sh` file from your new installation.

   Note: The Pivotal Command Center management tool replaces Greenplum command Center and Greenplum Performance Monitor. Command Center Console packages are available from the EMC Download Center.

4. If you had created tables using GPXF, you need to `DROP` these tables and `CREATE` them again using PXF 2.0.1.

   See Appendix G, Pivotal Extension Framework, in the Pivotal ADS 1.1 Administrator Guide, A02, for instructions about how to install PXF 2.0.1.

   Note: When you `CREATE` the tables for PXF2.0.1, remember to perform the following:

   a. Change the protocol name in the `LOCATION` clause from `gpxf` to `pxf`.

b. Ensure that Fragmenter, Accessor, and Resolver are *always* specified for the table.

c. Manually recreate the two symbolic links that are removed during the update:

```
cd /usr/lib/gphd # base directory for PHD
mkdir publicstage
chown hdfs /usr/lib/gphd/publicstage
chmod 777 /usr/lib/gphd/publicstage
ln -s gpxf-x.x.x gpxf
```

d. Check that you have the new names for the Fragmenter, Accessor, and Resolver classes.

See Appendix G, Pivotal Extension Framework, in the *PADS 1.1 Administrator Guide*, A02, for information about the Java classes.

e. Check that you are using the correct gucs for PXF:

- `gpxf_enable_filter_pushdown -> pxf_enable_filter_pushdown`
- `gpxf_enable_stat_collection -> pxf_enable_stat_collection`
- `gpxf_enable_locality_optimizations -> pxf_enable_locality_optimizations`

5. Inform all database users of the completed upgrade. Tell users to update their environment to source the HAWQ 1.1.x installation (if necessary).

Troubleshooting a Failed Upgrade

If you experience issues during the migration process, go to the Support page at Support Zone or contact Greenplum customer support at one of the following numbers:

United States: 800-782-4362 (1-800-SVC-4EMC)

Canada: 800-543-4782

Worldwide: +1-508-497-7901

Be prepared to provide the following information:

- A detailed list of upgrade procedures completed.
- Log output from `gpmigrator` (located in `~/gpAdminLogs`).

Use of Open Source

This product may be distributed with open source code, licensed to you in accordance with the applicable open source license. If you would like a copy of any such source code, EMC will provide a copy of the source code that is required to be made available in accordance with the applicable open source license. EMC may charge reasonable shipping and handling charges for such distribution. Please direct requests in writing to EMC Legal, 176 South St., Hopkinton, MA 01748, ATTN: Open Source Program Office.