

InfiniDB[®]

Apache Hadoop

Configuration Guide

Release: 4.6
Document Version: 4.6-1

InfiniDB Apache Hadoop™ Configuration Guide
July 2014
Copyright © 2014 InfiniDB Corporation. All Rights Reserved.

InfiniDB, the InfiniDB logo and any other product or service names or slogans contained in this document are trademarks of InfiniDB and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of InfiniDB or the applicable trademark holder.

Hadoop, Sqoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation.
All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of InfiniDB.

InfiniDB may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from InfiniDB, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. The information in this document is subject to change without notice. InfiniDB shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Contents

1	Introduction	4
1.1	Audience	4
1.2	List of documentation	4
1.3	Obtaining documentation	4
1.4	Documentation feedback	4
1.5	Additional resources	4
2	Overview	5
3	Key Differences from standard InfiniDB	5
4	Installation of InfiniDB for Apache Hadoop™	6
4.1	Preparing for Installation	6
4.1.1	OS Information	6
4.1.2	Prerequisites	6
4.2	Installation Overview	6
4.2.1	Disable Firewall	6
4.2.2	SSH Keys	7
4.2.3	PDSH Setup	7
4.2.4	NTPD Setup	7
4.2.5	Additional Hadoop Configuration	8
4.2.6	Download and Installation	8
4.2.7	InfiniDB Configuration	8
5	Administrative	10
5.1	Starting/Stopping InfiniDB for Apache Hadoop	10
5.2	Logging into InfiniDB for Apache Hadoop	10
5.3	Using InfiniDB for Apache Hadoop	10
6	Importing Data into InfiniDB for Apache Hadoop	11
6.1	cpimport	11
6.2	Apache Sqoop™ – InfiniDB	12
6.2.1	Assumptions	12
6.2.2	Installation	12
6.2.3	Sqoop Export	14
7	Moving Data from InfiniDB for Apache Hadoop	17
7.1	pdsh	17
7.2	Simple Table import (tab delimited)	17
7.3	Other delimited file	17
7.3.1	Intermediate File	18
7.3.2	Pipe Through Filters	18
7.4	Using a Join	18
8	Use of the Linux Control Groups	19
8.1	Creation of cgroup	19
8.1.1	Prerequisites	19
8.1.2	Create cgroup for InfiniDB	19
8.1.3	Associate the InfiniDB Service to the cgroup	20
8.1.4	Associating cpimport and cpimport.bin to cgroup	20
8.2	Use of cgroup for InfiniDB	20
9	OAM/Support Commands	22
9.1	getstorageconfig	22
9.2	calpontSupport	22

1 Introduction

This guide contains information needed to perform installation of InfiniDB for Apache Hadoop™ and the subsequent administration.

1.1 Audience

This guide is written for IT administrators who are responsible for implementing the InfiniDB System on an Apache Hadoop Distributed File System (HDFS).

1.2 List of documentation

The InfiniDB Database Platform documentation consists of several guides intended for different audiences. The documentation is described in the following table:

Document	Description
InfiniDB Administrator's Guide	Provides detailed steps for maintaining InfiniDB.
InfiniDB Concepts Guide	Introduction to the InfiniDB analytic database.
InfiniDB Minimum Recommended Technical Specifications	Lists the minimum recommended hardware and software specifications for implementing InfiniDB.
InfiniDB Installation Guide	Contains a summary of steps needed to perform an install of InfiniDB.
InfiniDB Multiple UM Configuration Guide	Provides information for configuring multiple User Modules.
InfiniDB SQL Syntax Guide	Provides syntax native to InfiniDB.
Performance Tuning for the InfiniDB Analytics Database	Provides help for tuning the InfiniDB analytic database for parallelization and scalability.
InfiniDB Windows Installation and Administrator's Guide	Provides information for installing and maintaining InfiniDB for Windows.

1.3 Obtaining documentation

These guides reside on our <http://www.infinidb.co> website. Contact support@infinidb.co for any additional assistance.

1.4 Documentation feedback

We encourage feedback, comments, and suggestions so that we can improve our documentation. Send comments to support@infinidb.co along with the document name, version, comments, and page numbers.

1.5 Additional resources

If you need help installing, tuning, or querying your data with InfiniDB, you can contact support@infinidb.co.

2 Overview

InfiniDB for Apache Hadoop™ enables you to take advantage of the scalability, flexibility, and cost-savings of Apache Hadoop using an interactive SQL interface. Data is more accessible to more users, and existing SQL tools can be leveraged.

3 Key Differences from standard InfiniDB

InfiniDB for Apache Hadoop has the following differences from the non-HDFS based InfiniDB:

- Compression – Compression will always be enabled.
- Transactions – There is no transaction processing. Once data has been successfully written, no rollback can occur.
- While InfiniDB for Hadoop supports DML operations, the performance of DML is substantially slower than standard InfiniDB due to the write-once aspect of HDFS. Use DML sparingly and only in areas where the performance can be tolerated. InfiniDB for Hadoop is optimized for bulk load via cpimport or InfiniDB sqoop and all queries perform similarly to standard InfiniDB.

4 Installation of InfiniDB for Apache Hadoop™

4.1 Preparing for Installation

Similar to the InfiniDB Standard, you should determine how much disk space your system will need before loading InfiniDB to ensure the required space is available.

Note: There is no upgrade path for an existing Apache Hadoop system or an existing InfiniDB system. You must use these procedures for installing a new system and import the data.

4.1.1 OS Information

InfiniDB for Apache Hadoop is supported on the following:

- CentOS 6
- Ubuntu 12.04

Note that Hadoop distributions themselves have OS requirements.

4.1.2 Prerequisites

The Apache Hadoop™ environment must be set up on all nodes. Please refer to <http://wiki.apache.org/hadoop/QuickStart> for a quick start.

Versions of Apache Hadoop supported by InfiniDB:

- [Cloudera 4.x](#) (Both Package and Parcel)
- [HortonWorks 1.3](#)

Additional packages to be installed on all servers:

- expect
- pdsh-2.27-1.el6.rf
- Oracle JDK Installation

InfiniDB for Hadoop requires JDK. The JDK in use for the Hadoop installation will be auto-detected and used by InfiniDB.

4.2 Installation Overview

Once Apache Hadoop has been installed on all nodes to be used, InfiniDB can now be installed. This is done by using the normal installation procedures as documented in the “Installing and Configuring Calpont InfiniDB” section of the InfiniDB Installation Guide with a couple additional steps needed when installing InfiniDB with Apache Hadoop. These steps will be documented here.

4.2.1 Disable Firewall

- Ensure the local firewall settings are off and disabled on all servers where InfiniDB is being installed:

```
service iptables stop
chkconfig iptables off
```

- Ensure SeLinux is disabled. For more information, visit the following website:
http://www.crypt.gen.nz/selinux/disable_selinux.html

4.2.2 SSH Keys

Ensure SSH Keys are set up between all InfiniDB Servers.

- Example for 3 PM, 1UM system, these are the steps required to configure PM-1 for passwordless ssh. The equivalent steps must be repeated on every PM in the system.

```
[infinidb@idb-pm-1 ~] $ ssh-keygen
[infinidb@idb-pm-1 ~] $ ssh-copy-id -i ~/.ssh/id_rsa.pub idb-pm-1
[infinidb@idb-pm-1 ~] $ ssh-copy-id -i ~/.ssh/id_rsa.pub idb-pm-2
[infinidb@idb-pm-1 ~] $ ssh-copy-id -i ~/.ssh/id_rsa.pub idb-pm-3
[infinidb@idb-pm-1 ~] $ ssh-copy-id -i ~/.ssh/id_rsa.pub idb-um-1
```

4.2.3 PDSH Setup

Ensure the package of 'pdsh-2.27-1' is installed on all servers and that the command 'pdsh -a <command>' successfully executes on every node in the cluster. To test, simply run 'pdsh -a date' and verify a return value appears for every machine.

If your pdsh version includes the 'machines' module (check with `pdsh -v`), then the file `/etc/pdsh/machines` should contain the host-names of the all the servers in the InfiniDB system. This can be setup and distributed from the Hadoop NameNode server:

```
# vi /etc/pdsh/machines
idb_pm_1.mycompany.com
idb_pm_2.mycompany.com
# pdcp -a /etc/pdsh/machines /etc/pdsh/
```

If your pdsh version instead includes the 'genders' module (this is typical on Ubuntu 12.04), then configure `/etc/genders` with one row for each node in the InfiniDB system along with the optional 'pdsh_rcmd_type=ssh' specification. A similar example would be:

```
# vi /etc/genders
idb_pm_1 pdsh_rcmd_type=ssh
idb_pm_2 pdsh_rcmd_type=ssh
# pdcp -a /etc/genders /etc/genders
```

4.2.4 NTPD Setup

Ensure that the Network Time Protocol Daemon is running on each server in the cluster.

Run the following commands to start the Daemon and to set it up so it will be restarted on server reboot.

These need to run as root use or use 'sudo' if running as non-root user.

```
# pdsh -a /etc/init.d/ntpd restart
# pdsh -a chkconfig --add ntpd
```

4.2.5 Additional Hadoop Configuration

There are a few Hadoop settings that should be made to optimize InfiniDB performance:

- The setting for `DataNode Local Path Access Users` (`dfs.block.local-path-access.user`) needs to include user that will run InfiniDB - most likely `root`.
- Verify that Enable HDFS Short Circuit Read (`dfs.client.read.shortcircuit`) is enabled and update as necessary.
- Follow the procedure for your Hadoop distribution to distribute the configuration change and restart services.

4.2.6 Download and Installation

Perform the download and installation steps from the Installation Guide for the appropriate package you have downloaded. This example is for an RPM install:

1. After installing the rpms, deps or binary package (and running post-install) script, it will provide you the commands to launch the postConfigure script to complete the installation of InfiniDB onto the system:

An example from an RPM install:

```
rpm -ivh infinidb-libs-release#.x86_64.rpm infinidb-platform-release#.x86_64.rpm
Preparing... ##### [100%]
 1:infinidb-libs ##### [ 33%]
Calpont RPM install completed
 2:infinidb-platform ##### [ 67%]
The next steps are:

. /usr/local/Calpont/bin/setenv-hdfs-20
/usr/local/Calpont/bin/postConfigure
Calpont RPM install completed
```

Note: For Enterprise Customers, the `infinidb-enterprise-release#.x86_64.rpm` would be included.

4.2.7 InfiniDB Configuration

The InfiniDB for Apache Hadoop configuration is completed by using the postConfigure script:

1. Run the steps that are provided from the install process above:

Example steps of a Cloudera package install:

```
. /usr/local/Calpont/bin/setenv-hdfs-20
Note: For Hortonworks, you may see this command:
. /usr/local/Calpont/bin/setenv-hdfs-12
```

```
/usr/local/Calpont/bin/postConfigure
```

2. The configuration script will follow the steps as outlined in the InfiniDB Configuration section of the InfiniDB Installation Guide. Only the "Setup Storage Configuration" prompt will differ. The

script will recognize the InfiniDB is being installed on an Apache Hadoop system and will display the following. You will now see a 'hdfs' option for storage type:

```
===== Setup Storage Configuration =====
```

```
----- Setup High Availability Data Storage Mount Configuration -----
```

There are 3 options when configuring the storage: internal, external, or hdfs

- 'internal' - This is specified when a local disk is used for the dbroot storage or the dbroot storage directories are manually mounted externally but no High Availability Support is required.
- 'external' - This is specified when the dbroot directories are externally mounted and High Availability Failover Support is required.
- 'hdfs' - This is specified when hadoop is installed and you want the dbroot directories to be controlled by the Hadoop Distributed File System (HDFS).

Select the type of Data Storage [1=internal, 2=external, 4=hdfs] (4) > **<Enter>**

NOTE: Choose option 4 (or Enter for the default) for HDFS.

Enter the Hadoop Datafile Plugin File (/usr/local/Calpont/lib/hdfs-20.so) > **<Enter>**

NOTE: For Hortonworks, you must enter: /usr/local/Calpont/lib/hdfs-12.so

Running HDFS Sanity Test (Please Wait): PASSED

3. Continue with the rest of the configuration script as documented in the Installation Guide for completion of the InfiniDB configuration.

5 Administrative

Please reference the **InfiniDB Administrator's Guide** for detailed administrative tasks/options for setting up and maintaining InfiniDB. Some items below have been included in this document that are specific to InfiniDB for Apache Hadoop.

5.1 *Starting/Stopping InfiniDB for Apache Hadoop*

- Before starting the InfiniDB service, the Hadoop/HDFS service must have been started.
- At any point on a running InfiniDB for Apache Hadoop system, InfiniDB should be stopped before the HDFS service is stopped.

5.2 *Logging into InfiniDB for Apache Hadoop*

Logging into InfiniDB is simply typing the following at a command prompt:

```
idbmysql
```

Note: If you have a previously created database, you may specify the database name when logging in:

```
idbmysql mydb
```

5.3 *Using InfiniDB for Apache Hadoop*

Once in, standard procedures for syntax may be followed. Please reference the **InfiniDB SQL Syntax Guide** for further information.

6 Importing Data into InfiniDB for Apache Hadoop

The recommended method for getting data into InfiniDB is thru Importing data. There are two methods for importing data into an InfiniDB for Apache Hadoop system:

1. Original cpimport
2. Apache Sqoop™ - InfiniDB

6.1 *cpimport*

Executing cpimport on an InfiniDB for Apache Hadoop system is similar to loading data into InfiniDB Standard. Please see “Importing Data” in the **InfiniDB Administrator’s Guide** for more details.

Some example uses:

Example colxml command:

This command is to have a different job id of today’s date (other than the default of 299):

```
/usr/local/Calpont/bin/colxml mydb -j20130331
```

Example cpimport command:

These commands are used to read the source file from a different directory:

```
/usr/local/Calpont/bin/cpimport -f/source/mydb mydb mytable mytable.tbl
```

```
/usr/local/Calpont/bin/cpimport mydb mytable /source/mydb/mytable.tbl
```

Example cpimport command with multiple files used as source data:

Use a comma with no spaces when importing from multiple files. If spaces exist in the file names, then use double quotation marks around the file name :

```
/usr/local/Calpont/bin/cpimport mydb mytable mytable mytable1.tbl,mytable2.tbl
```

```
/usr/local/Calpont/bin/cpimport mydb mytable mytable mytable1.tbl,"mytable two.tbl"
```

6.2 Apache Sqoop™ – InfiniDB

Apache Sqoop is a tool designed to transfer data between Hadoop and relational databases. The InfiniDB extension to the Apache Sqoop tool utilizes the import and export function of Sqoop to load data from HDFS to InfiniDB. The HDFS side of the work is distributed, but the database work is performed on a single machine. Since InfiniDB is a distributed database, it makes more sense to attach to individual instances of the database, one on each Performance Module, and run SQL statements affecting just the data on that machine. Thus the work is distributed for both the HDFS side and the database side.

InfiniDB has made enhancements to optimize Sqoop's behavior for the InfiniDB database. Sqoop for InfiniDB attempts to build splits such that the hdfs data is machine local as well as the database. There are instances and configurations that may preclude this, but for the most part, all work is local.

The following describe the installation and use of this tool.

6.2.1 Assumptions

1. You should be familiar with the basic use of Apache Sqoop export before using this tool:
<https://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html>
2. These instructions are for a UM-NameNode, PMs-DataNodes setup
3. Data to import already exist on HDFS.
4. cpimport is available on each data node (PM).
5. The data format is supported by Sqoop export.

6.2.2 Installation

1. On the UM, download and unpack the `infinidb-sqoop-release#.tar.gz` file (obtained from the www.infinidb.co website within any of the Hadoop Details pages) into a scoop directory, referred to as the `$SQOOP_HOME` in this guide. In most cases, you want this to be `/usr/lib/infinidb-sqoop-1.4.2`:

```
# tar -xvf infinidb-sqoop-release#.tar.gz
```

2. Download mysql jdbc driver '`mysql-connector-java-5.x.xx-bin.jar`' and place it in `$SQOOP_HOME/lib`:

<http://dev.mysql.com/downloads/connector/j/#downloads>

3. On each PM:
 - Copy `$SQOOP_HOME/bin/cpimport_sqoop` from UM to all the PMs `/usr/local/Calpont/bin` directory
 - Run this command on all PMs:

```
chmod u+s /usr/local/Calpont/bin/cpimport.bin
```

This causes `cpimport.bin` to run with root permissions regardless of which user actually invokes it, because sqoop map-reduce jobs are invoked from user `mapred`.

6.2.2.1 Cloudera Hadoop Parcel Install Considerations

When using the Parcel install method for the Cloudera Hadoop Manager, the following commands must also be run (as root).

1. On the UM (wherever SQOOP is to be run):

```
# export HADOOP_HOME=$HADOOP_PREFIX
```

`$HADOOP_HOME` is deprecated, but still needed by InfiniDB. `$HADOOP_PREFIX` is always set by the Hadoop install for both Parcel or Package.

```
# echo $HADOOP_PREFIX
```

It should look something like this:

```
/opt/cloudera/parcels/CDH-ClouderaRelease#/lib/Hadoop
```

2. For each PM, find `libhdfs.so.0.0.0` under

```
/opt/cloudera/parcels/CDH-ClouderaRelease#/lib64
```

and create a symbolic link in `/usr/local/Calpont/lib` (or wherever InfiniDB is installed):

```
cd /usr/local/Calpont/lib
```

```
ln -s $HADOOP_PREFIX/../../lib64/libhdfs.so.0.0.0 libhdfs.so.0.0.0
```

-or-

```
ln -s
```

```
/opt/cloudera/parcels/CDH-ClouderaRelease#/lib64/libhdfs.so.0.0.0  
libhdfs.so.0.0.0
```

Note that `libhdfs.so.0.0.0` is not directly under `$HADOOP_PREFIX`, but in `lib64` of the upper directory.

6.2.3 Sqoop Export

sqoop export moves data from hdfs to InfiniDB. It uses InfiniDB's cpimport facility to quickly move data into InfiniDB. By creating exactly one split on each InfiniDB PM, the data is moved in the most efficient way possible. Sqoop export for InfiniDB attempts to create splits in a way that the hdfs data is local to each instance of cpimport while maintaining load balance.

6.2.3.1 Invocation

To use sqoop export, the following preconditions must be met:

- InfiniDB is running
- The destination schema and table exists in InfiniDB.
- The source files exist in hdfs.

Run sqoop from \$SQOOP_HOME/bin. This should reflect the location of the InfiniDB installation of sqoop. You may have another sqoop installation available, so be sure you use the right one.

All sqoop export command line options will be accepted. The following shows the differences in sqoop export for InfiniDB.

Common arguments

Argument	Description
--connect <jdbc-uri>	Specify JDBC connect string. The connections string must indicate infinidb as the database.
--connection-manager <class-name>	This will be ignored.
--driver <class-name>	Manually specify JDBC driver class to use
--hadoop-home <dir>	Override \$HADOOP_HOME
--help	Print usage instructions
-P	Read password from console
--password <password>	Set authentication password
--username <username>	Set authentication username
--verbose	Print more information while working
--connection-param-file <filename>	Optional properties file that provides connection parameters
-D <property=value>	Assign a value to a property, similar to what might be found in conf files.

Export control arguments

Argument	Description
--direct	Tells sqoop to use cpimport for fast path. Required.
--export-dir <dir>	HDFS source path for the export. Required.
-m, --num-mappers <n>	Ignored
--table <table-name>	Table to populate. Required.
--update-key <col-name>	Ignored
--update-mode <mode>	Ignored
--input-null-string <null-string>	The string to be interpreted as null for string columns.
--input-null-non-string <null-string>	The string to be interpreted as null for non-string columns.
--staging-table <staging-table-name>	The table in which data will be staged before being inserted into the destination table.
--clear-staging-table	Indicates that any data present in the staging table can be deleted.
--batch	Ignored

6.2.3.2 Sqoop Export Example

Server name of the InfiniDB User Module - infinidb.um.com

mysql user – root

password – pwd

schema – myDatabase

table –myTable

data file source - /usr/local/Calpont/data1/myTable.tbl

field delimiter – ‘|’

```
./sqoop export -D mapred.task.timeout=0 --direct --connect
jdbc:infinidb://infinidb.um.com/myDatabase --username root --password pwd --table
myTable --export-dir /usr/local/Calpont/data1/myTable.tbl --input-fields-terminated-by
'|'
```

6.2.3.3 Notes

The -D mapred.task.timeout=0 prevents Hadoop from timing out when the map task takes a long time. We're not running multiple tasks per server, and timers might pop. According to the sqoop documentation, -D and some other arguments must follow 'export' and precede the tool specific arguments.

The last argument --input-fields-terminated-by '|' is not a sqoop argument. Sqoop assumes that, since it doesn't recognize it, it must be something the underlying tool needs and passes it to cpimport. In this case, our data to be exported has '|' as the field delimiter.

Cpimport, the underlying tool that's moving data into InfiniDB, has a number of other options that may be useful:

- --input-fields-terminated-by <char> Sets the input field delimiter

- `--enclosed-by <char>` Sets the input field enclosing character
- `--escaped-by <char>` Sets the escape character
- `--input-null-string <null-string>` The string to be interpreted as null for string columns. (a standard sqoop argument that is passed to `cpimport`)
- `--input-null-non-string <null-string>` The string to be interpreted as null for non-string columns. (a standard sqoop argument that is passed to `cpimport`)

6.2.3.4 Troubleshooting

While the JDBC connection string is required to have 'infinidb' as the database portion, the underlying code still uses the MySQL connection. Therefore, `mysqld` must be running on the declared server. While `cpimport` is used for moving the data, `sqoop` still needs access to the MySQL connection to gather meta-data.

6.2.3.5 Missing jdbc driver

ERROR sqoop.Sqoop: Got exception running Sqoop: java.lang.RuntimeException: Could not load db driver class: com.mysql.jdbc.Driver

Cause: MySQL jdbc driver is not in `$SQOOP_HOME/lib` directory.

6.2.3.6 Access denied

ERROR java.sql.SQLException: Access denied for user 'root'@'qa4pm-head.calpont.com' (using password: YES)

Cause: The connection string is not correctly formed. Most likely the wrong password is provided.

6.2.3.7 Missing User directory

ERROR security.UserGroupInformation: PrivilegedActionException as:user1 (auth:SIMPLE)
cause:org.apache.hadoop.security.AccessControlException: Permission denied: user=root,
access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x

Cause: The "user1" user lacks a home directory that sqoop needs to use under `/user`

To fix:

```
sudo -u hdfs hadoop fs -mkdir /user/user1  
sudo -u hdfs hadoop fs -chown user1:user1 /user/user1
```


7 Moving Data from InfiniDB for Apache Hadoop

The recommended method for getting data from InfiniDB to HDFS is thru the use of the `pdsh` command.

7.1 `pdsh`

`pdsh` is a program that runs remote commands in parallel on different machines. The host name file should already be set up if InfiniDB is installed. `pdsh` documentation can be found at

<http://linux.die.net/man/1/pdsh>

7.2 Simple Table import (tab delimited)

For many use cases, all we want to do is get a table's data into hdfs as a delimited file. For tab delimited, this is fairly straight forward. To select a table into hdfs using local query:

```
pdsh -a -x my_server_um1 "/usr/local/Calpont/mysql/bin/mysql --defaults-  
file=/usr/local/Calpont/mysql/my.cnf -u root -N -e 'set infinidb_local_query=1;  
select * from table1' | hdfs dfs -put - /users/root/t1/t1%n.tbl"
```

Explanation:

`-a` : send to all hosts.

`-x` : exclude host, in this case `<UM>`. We don't run local queries on the UM. This can be multiple server names.

The command inside the double quotes is what is sent to the remote hosts:

- The `"/usr/local/Calpont/mysql/bin/mysql --defaults-file=/usr/local/Calpont/mysql/my.cnf"` is the alias for `idbmysql` and doesn't get set up in all cases. You can try `idbmysql` instead and if it works, is much simpler.
- In this example, we send two SQL statements, the first to set the `infinidb_local_query` switch, and the second to perform the query.
- Finally, we pipe the output to our destination. Notice the `%n` in the filename. It's important that each instance of this command pipe to a different filename. `pdsh` has a very limited substitution capability. `%n` appends an incrementing number to the file name. You're free to use the other substitution variables from `pdsh`, so long as the result is a unique file name. Placing all the files in a single directory is normal, and most Hadoop jobs are used to handling sets of files this way. For example, Sqoop Export could directly export these back to InfiniDB.

7.3 Other delimited file

Sometimes a tab delimited file just won't do. For example, InfiniDB usually uses the pipe `"|"` as a delimiter. In addition, you may want to put quotes around your fields.

7.3.1 Intermediate File

One option is to output to a local file using select into outfile and then –put it into hdfs. This can be done in one line. Select into outfile has a set of options controlling the format of the output, so this may get exactly what you want. Unfortunately, it takes an extra copy.

For this example, these three steps must be performed on each node:

- Perform the query into a tmp file
- Copy the data into hdfs
- Remove the tmp file

```
pdsh -a -x my_server_um1 "/usr/local/Calpont/mysql/bin/mysql --defaults-
file=/usr/local/Calpont/mysql/my.cnf -u root -N -e 'set infinidb_local_query=1;
select o_orderkey, o_orderstatus, o_orderdate from my_db.my_orders into outfile
\"/tmp/orders.tbl\" fields terminated by \"|\" enclosed by '\"\\\"\\\"' && hdfs dfs
-put /tmp/orders.tbl /users/root/orders/orders%n.tbl ; rm /tmp/orders.tbl"
```

Notes

- All the escape sequences are needed. pdsh takes everything between the outside double quotes. InfiniDB takes everything between the inside single quotes. without the escape sequences, the parser gets totally confused.
- The destination hdfs directory must already exist and you must have write permissions.
- You must have write permissions in the /tmp directory on each node.

7.3.2 Pipe Through Filters

Another option is to Pipe the output through various filters to get the desired result. A common practice is to pipe through tr to convert the <tab> to “|” and then through sed to remove the word NULL from the output.

```
pdsh -a -x my_server_um1 "/usr/local/Calpont/mysql/bin/mysql --defaults-
file=/usr/local/Calpont/mysql/my.cnf -u root -N -e 'set infinidb_local_query=1;
select o_orderkey, o_orderstatus, o_orderdate from my_db.my_orders' | tr '\t'
'|' | sed -e 's/\([^|]\)$/\1|/' -e 's/|NULL|/|/' | hdfs dfs -put -
/users/root/orders/orders%n.tbl"
```

This may put a “|” at the end of each row.

7.4 Using a Join

To use a join, set the infinidb_local_query switch to 0 and use the idbLocalPM() function similar to:

```
"SELECT a.ID, c2, c3, c4 FROM factTable AS a JOIN dimTable AS b ON a.ID=b.ID
WHERE idbPm(a.ID)=idbLocalPM()"
```

Of course, much more can be done with pdsh and scripts or even programs.

8 Use of the Linux Control Groups

When InfiniDB for Apache Hadoop is installed into a customer environment, it most likely will be in an already running Hadoop system with other Hadoop applications running. In such cases, It cannot be assumed that all the system resources (memory, cpuset) are exclusively available to InfiniDB, as these will be also used by the other Hadoop applications running on the same cluster.

Resource Management of InfiniDB, when deployed along with other Hadoop application, should be such that InfiniDB perform optimally while not negatively impacting other Hadoop applications and vice-versa. For the purpose of resource management, Control Groups (cgroups) can be used. Cgroup is a Linux kernel feature to limit, police, account and isolate resource usage (CPU, memory, disk I/O, etc.) of process groups. The Linux Kernel needs to be 2.6.24 or higher version. RedHat documentation for cgroup can be found at: https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch-Using_Control_Groups.html#The_cgconfig_Service

8.1 Creation of cgroup

The following instructions are for creating a cgroup for InfiniDB and associating this cgroup with the InfiniDB service:

8.1.1 Prerequisites

Linux 2.6.24 or higher
libcgroup is installed

8.1.2 Create cgroup for InfiniDB

1. Determine following resources needed by InfiniDB application on the server. See InfiniDB Hardware Sizing Guide:
 - a. CPU Cores – number of cpu cores to be allocated to InfiniDB application. Then decide which cpu cores will be allocated to InfiniDB application. E.g if InfiniDB is to be assigned 4 cpu- For CPU 0 to 3 then the list would be 0-3, or For CPUs 2 to 5 list would be 2-5.
 - b. CPU - % of CPUs to be allocated to InfiniDB application
 - c. Memory – Amount of memory in bytes to be allocated to InfiniDB
2. Define cgroup for InfiniDB in /etc/cgconfig.conf:
 - a. If /etc/cgconfig.conf does not exist, then create a cgconfig.conf file in the /etc directory with the following lines at the top, otherwise go directly to step b:

```
mount {
    cpuset = /sys/fs/cgroups/cpuset;
    memory = /sys/fs/cgroups/memory;
}
```

Note: If you are already using cgroups, you may already have a location. If so, you will need to either mount that location to /sys/fs/cgroups or create a link to where it is mounted.

- b. Modify /etc/fstab to add the following entry:

```
none /sys/fs tmpfs defaults 0 0
```

and run the mount command:

```
mount -a
```

c. Add the following lines at the end of `/etc/cgconfig.conf` file:

```
group idbgroup {
memory {
    memory.limit_in_bytes = <bytes to be dedicated to InfiniDB -
                                determined in step (1)>;
}
cpuset {
    cpuset.mems = 0;
    cpuset.cpus = <list of cpu cores assigned to InfiniDB -
                    determined in step (1)>;
}
}
```

8.1.3 Associate the InfiniDB Service to the cgroup

1. Create the `infinidb` file in the `/etc/sysconfig` directory and add the following lines:

```
CGROUP_DAEMON="memory:daemons/idbgroup"
CGROUP_DAEMON="cpu:daemons/idbgroup"
CGROUP_DAEMON="cpuset:daemons/idbgroup"
```

2. Restart the `cgconfig` service:

```
service cgconfig restart
```

3. Restart the server.

8.1.4 Associating `cpimport` and `cpimport.bin` to cgroup

1. In order for `cpimport` and `cpimport.bin` to start in the same cgroup as the `infinidb` cgroup, run using `cgexec`:

```
cgexec -g cpuset,memory:/idbgroup cpimport
cgexec -g cpuset,memory:/idbgroup cpimport.bin
```

8.2 Use of cgroup for InfiniDB

The cgroup should be assigned with number the `cpuset` and `memory` needed to support the expected query work load, active data set and data buffer cache following standard hardware sizing guidelines for InfiniDB. At the same time standard hardware sizing guidelines for HDFS and the rest of the Hadoop applications should be followed to determine the amount of `cpu` and `memory` resources needed by the Hadoop ecosystem. The physical size of the hardware should be the total of the size determined for InfiniDB and size determined/needed for the Hadoop ecosystem.

To use the pre-configured cgroup in InfiniDB, add the following entry under the `<SystemConfig>` element of the `Calpont.xml` configuration file:

```
<CGroup>idbgroup</CGroup>
```

Once modifications have been made, a restartSystem is required for the changes to take effect.

Notes:

TotalUMMemory configuration parameter in Calpont.xml can now be a percentage so that it doesn't have to change every time they modify the cgroup or the membership:

- If the InfiniDB processes are in a cgroup, the memory limit will be a percentage of the memory assigned to the cgroup.
- If the InfiniDB processes are not in a cgroup, the memory limit will be a percentage of the system's memory.

The TotalUmMemory and NumBlocksPct settings in Calpont.xml are defaulted lower for Hadoop installations compared to non-Hadoop installations. These settings are defaulted as a percentage of total server memory. If using a control group, you may want to increase these settings proportionally. For example, if NumBlocksPct is set to 35 and your control group is set up to use half of the available memory, change NumBlocksPct to 70.

9 OAM/Support Commands

9.1 *getstorageconfig*

The `getstorageconfig` command will now show a storage type of 'hdfs' for an InfiniDB for Apache Hadoop configured system:

`getstorageconfig` will show 'hdfs' and 'gluster' storage types

```
InfiniDB> getstorageconfig  
getstorageconfig  Fri Oct  4 19:11:25 2013
```

System Storage Configuration

```
Performance Module (DBRoot) Storage Type = hdfs  
System Assigned DBRoot Count = 4  
DBRoot IDs assigned to 'pm1' = 1  
DBRoot IDs assigned to 'pm2' = 2  
DBRoot IDs assigned to 'pm3' = 3  
DBRoot IDs assigned to 'pm4' = 4
```

9.2 *calpontSupport*

Note: This feature is only available with InfiniDB Enterprise.

The script 'calpontSupport' generates a number of text reports of reports about the system and tars up the InfiniDB System Logs and puts them into a single tar file. For more information, see the InfiniDB Administrator's Guide.

There is a new option that will display HDFS specific information:

```
-hd Output Apache Hadoop™ reports only (if applicable)
```