



Connect. Accelerate. Outperform.™

Mellanox Unstructured Data Acceleration (UDA) Quick Start Guide

Rev 3.4.1

NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Mellanox Technologies
 350 Oakmead Parkway Suite 100
 Sunnyvale, CA 94085
 U.S.A.
www.mellanox.com
 Tel: (408) 970-3400
 Fax: (408) 970-3403

© Copyright 2015. Mellanox Technologies. All Rights Reserved.

Mellanox®, Mellanox logo, BridgeX®, CloudX logo, Connect-IB®, ConnectX®, CoolBox®, CORE-Direct®, GPUDirect®, InfiniHost®, InfiniScale®, Kotura®, Kotura logo, Mellanox Federal Systems®, Mellanox Open Ethernet®, Mellanox ScalableHPC®, Mellanox Connect Accelerate Outperform logo, Mellanox Virtual Modular Switch®, MetroDX®, MetroX®, MLNX-OS®, Open Ethernet logo, PhyX®, SwitchX®, TestX®, The Generation of Open Ethernet logo, UFM®, Virtual Protocol Interconnect®, Voltaire® and Voltaire logo are registered trademarks of Mellanox Technologies, Ltd.

Accelio™, CyPU™, FPGADirect™, HPC-X™, InfiniBridge™, LinkX™, Mellanox Care™, Mellanox CloudX™, Mellanox Multi-Host™, Mellanox NEO™, Mellanox PeerDirect™, Mellanox Socket Direct™, Mellanox Spectrum™, NVMeDirect™, StPU™, Spectrum logo, Switch-IB™, Unbreakable-Link™ are trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

Table of Contents

Table of Contents	3
List of Tables	4
Document Revision History	5
About This Manual	7
Intended Audience	7
Typographical Conventions	7
Common Abbreviations and Acronyms	8
Glossary	9
Related Documentation	11
Support and Updates Webpage	12
Chapter 1 Overview	13
1.1 Mellanox UDA Solution	13
1.2 Mellanox OFED for Linux	13
Chapter 2 Hardware Setup	14
2.1 Setting up the Adapter Cards	14
2.2 Setting up the Switch System	14
Chapter 3 Installing, Configuring and Running UDA Software	15
3.1 Supported Operating Systems	15
3.2 Installation Prerequisites	15
3.3 Installing UDA	16
3.4 UDA Configuration	18
3.4.1 RDMA Plug-in Parameters Basic Tuning Guidelines	21
3.5 UDA Log Setting	22
3.6 Running UDA	22
Appendix A Patching and Building Hadoop	24
Appendix B Configuring UDA on a CDH5+ Cluster via Cloudera Manager	25
Appendix C Configuring UDA on a HDP Cluster via Ambari	27

List of Tables

Table 1:Document Revision History	5
Table 2:Typographical Conventions	7
Table 3:Abbreviations and Acronyms	8
Table 4:Glossary	9
Table 5:Reference Documents	11

Document Revision History

Table 1 - Document Revision History

Revision	Date	Description
3.4.1	November 2015	Added the following section: <ul style="list-style-type: none"> • Appendix C: “Configuring UDA on a HDP Cluster via Ambari,” on page 27
	December 2014	Updated the following sections: <ul style="list-style-type: none"> • Section 3.2, “Installation Prerequisites,” on page 15 • Section 3.3, “Installing UDA,” on page 16 Added the following section: <ul style="list-style-type: none"> • Appendix B: “Configuring UDA on a CDH5+ Cluster via Cloudera Manager,” on page 25
3.4.0	September 2014	Updated the following sections: <ul style="list-style-type: none"> • Section 3.2, “Installation Prerequisites,” on page 15 • Section 3.3, “Installing UDA,” on page 16
3.3.3	December 2013	Updated the following sections: <ul style="list-style-type: none"> • Section 3.3, “Installing UDA,” on page 16 • Section 3.4, “UDA Configuration,” on page 18 • Appendix A: “Patching and Building Hadoop,” on page 24
3.2.2	September 2013	Updated the following sections: <ul style="list-style-type: none"> • Section 3.3, “Installing UDA,” on page 16 • Section 3.4, “UDA Configuration,” on page 18 • Appendix A: “Patching and Building Hadoop,” on page 24
3.1.11	June 2013	Updated the following sections: <ul style="list-style-type: none"> • Section 3.2, “Installation Prerequisites,” on page 15 • Section 3.3, “Installing UDA,” on page 16 • Section 3.4, “UDA Configuration,” on page 18 Added the following sections: <ul style="list-style-type: none"> • Section 3.5, “UDA Log Setting,” on page 22 • Appendix A: “Patching and Building Hadoop,” on page 24
3.0	August 2012	Major updates to the following chapters: <ul style="list-style-type: none"> • Chapter 1, “Overview” • Chapter 2, “Hardware Setup” • Chapter 3, “Installing, Configuring and Running UDA Software”

Table 1 - Document Revision History

Revision	Date	Description
2.1	April 2012	<ul style="list-style-type: none"> Renamed the document title (was Mellanox Web 2.0 Acceleration Kit Quick Start Guide) Reorganized the sections in Chapter 1, “Overview” and updated links to the software Consolidated all adapter cards HW and SW installation into Section 2.1, “Setting up the Adapter Cards”, on page 14 Consolidated all switch system HW and management SW installation into Section 2.2, “Setting up the Switch System”, on page 14 (the details of the HW installation have been removed; the reader is referred to the switch installation guide for the installation details) Added a prerequisite to increase the maximum number of memory translation table segments per HCA in Section 3.2, “Installation Prerequisites”, on page 15 Updated EULA path in Section 3.3, “Installing UDA”, on page 16 Updated Section 3.4, “UDA Configuration”, on page 18 Added Section 3.4.1, “RDMA Plug-in Parameters Basic Tuning Guidelines”, on page 21 Updated Section 3.6, “Killing Previous Hadoop Runs,” on page 39
1.1	October 2011	Updated section 3.3, “Mellanox UDA Installation” for UDA 2.0
1.0	June 2011	Initial release

About This Manual

This document describes the setup and configuration of Mellanox Unstructured Data Acceleration (UDA) software package for Hadoop Map Reduce frameworks.

Intended Audience

This manual is intended for system administrators responsible for the installation, configuration, management and maintenance of Mellanox UDA software. It is also intended for application developers.

Typographical Conventions

Table 2 - Typographical Conventions



Description	Convention
File names	<code>file.extension</code>
Directory names	<code>directory</code>
Commands and their parameters	<code>command param1</code>
Optional items	<code>[]</code>
Mutually exclusive parameters	<code>{ p1 p2 p3 }</code>
Optional mutually exclusive parameters	<code>[p1 p2 p3]</code>
Prompt of a <i>user</i> command under bash shell	<code>hostname\$</code>
Prompt of a <i>root</i> command under bash shell	<code>hostname#</code>
Prompt of a <i>user</i> command under tcsh shell	<code>tcsh\$</code>
Environment variables	VARIABLE
Code example	<code>if (a==b) { };</code>
Comment at the beginning of a code line	<code>!, #</code>
Characters to be typed by users as-is	bold font
Keywords	bold font
Variables for which users supply specific values	<i>Italic font</i>
Emphasized words	<i>Italic font</i>
Pop-up menu sequences	<code>menu1 --> menu2 -->... --> item</code>
Note	 <code><text></code>

Table 2 - Typographical Conventions

Description	Convention
Warning	 <text>

Common Abbreviations and Acronyms

Table 3 - Abbreviations and Acronyms (Sheet 1 of 2)

Abbreviation / Acronym	Whole Word / Description
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
b	(Small) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
FCoE	Fibre Channel over Ethernet
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
IB	InfiniBand
LSB	Least significant <i>byte</i>
lsb	Least significant <i>bit</i>
MSB	Most significant <i>byte</i>
msb	Most significant bit
NIC	Network Interface Card
SW	Software
VPI	Virtual Protocol Interconnect
IPoIB	IP over InfiniBand
PFC	Priority Flow Control
PR	Path Record
RDS	Reliable Datagram Sockets
RoCE	RDMA over Converged Ethernet
SDP	Sockets Direct Protocol
SL	Service Level

Table 3 - Abbreviations and Acronyms (Sheet 2 of 2)

Abbreviation / Acronym	Whole Word / Description
SRP	SCSI RDMA Protocol
MPI	Message Passing Interface
EoIB	Ethernet over InfiniBand
QoS	Quality of Service
ULP	Upper Level Protocol
VL	Virtual Lanes
vHBA	Virtual SCSI Host Bus adapter
uDAPL	User Direct Access Programming Library

Glossary

The following is a list of concepts and terms related to InfiniBand in general and to Subnet Managers in particular. It is included here for ease of reference, but the main reference remains the *InfiniBand Architecture Specification*.

Table 4 - Glossary

CA (Channel Adapter)	A device which terminates an InfiniBand link, and executes transport level functions
CLI	Command Line Interface. A user interface in which you type commands at the prompt
DMA (Direct Memory Access)	Allows hardware to move data blocks directly to the memory, bypassing the CPU
DNS	Domain Name System. A hierarchical naming system for devices in a computer network
Fabric Management	The use of a set of tools (APIs) to configure, discover, and manage a group of devices organized as a connected fabric.
Gateway	A network node that interfaces with another network using a different network protocol
GUID (Globally Unique Identifier)	A 64-bit number that uniquely identifies a device or component in a subnet
GID (Global Identifier)	A 128-bit number used to identify a Port on a network adapter (see below), a port on a Router, or a Multicast Group.
HA (High Availability)	A system design protocol that provides redundancy of system components, thus enables overcoming single or multiple failures and minimal downtime

Table 4 - Glossary

Host	A computer platform executing an Operating System which may control one or more network adapters
Hadoop	Open source, distributed, big data processing application. (an Apache project)
IB	InfiniBand
LID (Local Identifier)	A 16 bit address assigned to end nodes by the subnet manager Each LID is unique within its subnet.
MTU (Maximum Transfer Unit)	The maximum size of a packet payload (not including headers) that can be sent /received from a port
Network Adapter	A hardware device that allows for communication between computers in a network
QoS or Quality of Service	Quality of service is the ability to manage different applications or users by priority such that a required bit rate, delay, packet dropping probability, and/or other measures may be guaranteed.
RDMA (Remote Direct Memory Access)	Allows accessing memory on a remote side without involvement of the remote CPU
SA (Subnet Administrator)	The interface for querying and manipulating subnet management data
SSH	Secure Shell. A protocol (program) for securely logging in to and running programs on remote machines across a network. The program authenticates access to the remote machine and encrypts the transferred information through the connection.
Subnet Manager (SM)	An entity that configures and manages the subnet, discovers the network topology, assign LIDs, determines the routing schemes and sets the routing tables. There is only one master SM and possible several slaves (Standby mode) at a given time. The SM administers switch routing tables thereby establishing paths through the fabric
TCA (Target Channel Adapter)	A Channel Adapter that is not required to support verbs, usually used in I/O devices
UDA	Unstructured Data Acceleration
UDA Plugin	A software plugin that plugs into the Hadoop application
WebUI	Web User Interface. A user interface in which you select commands from drop down menus or by clicking on icons

Related Documentation

Table 5 - Reference Documents

Document Name	Description
InfiniBand Architecture Specification, Vol. 1, Release 1.2.1	The InfiniBand Architecture Specification that is provided by IBTA
Mellanox OFED for Linux	Software and documentation can be found at http://www.mellanox.com/content/pages.php?pg=products_dyn&product_family=26&menu_section=34
Mellanox MLNX-OS™ Switch Management Software documents	Documentation collateral for MLNX-OS™ CLI, configuration and HowTOs. See http://www.mellanox.com/content/pages.php?pg=mlnx_os&menu_section=55
Firmware Release Notes for Mellanox adapter devices	See the Release Notes PDF file relevant to your adapter device under docs/ folder of installed package.
ConnectX®-3 Dual Port FDR 56Gb/s InfiniBand Adapter Card User Manual	This manual provides details of the interfaces of ConnectX-3 FDR InfiniBand adapter cards, specifications, required software and firmware for operating the boards, and relevant documentation. http://www.mellanox.com/related-docs/user_manuals/ConnectX-3_VPI_Single_and_Dual_QSFP_Port_Adapter_Card_User_Manual.pdf
ConnectX®-3 Dual Port 40GbE Adapter Card User Manual	This manual provides details of the interfaces of ConnectX-3 EN 40 Gb/s Ethernet adapter cards, specifications, required software and firmware for operating the boards, and relevant documentation. See http://www.mellanox.com/related-docs/user_manuals/ConnectX-3_Ethernet_Single_and_Dual_QSFP_Port_Adapter_Card_User_Manual.pdf
SX6036 SwitchX® 1U 36 Port FDR 56Gb/s InfiniBand Switch Installation Guide Document No. 3489	This manual provides installation and set-up instructions for the SX6036 FDR top of rack InfiniBand Switch platforms. See http://www.mellanox.com/related-docs/user_manuals/SX60XX_Installation_Guide.pdf
SX1036 SwitchX® 1U 36 Port QSFP 40Gb/E Switch Installation Guide Document No. 3468	This manual provides installation and set-up instructions for the SX1036 40Gb/s Ethernet top of rack Switch platforms. See http://www.mellanox.com/related-docs/user_manuals/SX10XX_Installation_Guide.pdf

Support and Updates Webpage

Please visit the following Web site for downloads, FAQ, troubleshooting, future updates to this manual, etc: http://support.mellanox.com/SupportWeb/software_products/hostacceler_products/UDA.

1 Overview

1.1 Mellanox UDA Solution

Mellanox UDA (Unstructured Data Accelerator) is a software plugin that accelerates Hadoop and improves the scaling of Hadoop clusters executing data-analytics intensive applications. A novel data shuffling protocol is provided for Hadoop to take advantage of RDMA in the network technologies InfiniBand and RoCE (RDMA over Converged Ethernet). Mellanox UDA is an RDMA based software plugin which combined with MLNX Linux (MLNX OFED) inbox driver and ConnectX® based adapter cards will accelerate tasks associated with Map/Reduce file transfer. UDA more than doubles the data processing throughput and reduces CPU utilization by half of Hadoop nodes. Mellanox UDA is developed in collaboration with Auburn University, Alabama.

1.2 Mellanox OFED for Linux

Mellanox OFED for Linux (MLNX_OFED_LINUX) is provided as ISO images, one per a supported Linux distribution, that includes *source code* and *binary* RPMs, firmware, utilities, and documentation. The ISO image contains an installation script (called `mlnxofedinstall`) that performs the necessary steps to accomplish the following:

- Discover the currently installed kernel
- Uninstall any InfiniBand stacks that are part of the standard operating system distribution or another vendor's commercial stack
- Install the MLNX_OFED_LINUX binary RPMs (if they are available for the current kernel)
- Identify the currently installed InfiniBand HCAs and perform the required firmware updates

2 Hardware Setup

2.1 Setting up the Adapter Cards

This manual assumes one or more of the Mellanox ConnectX® family adapter cards is installed in your host machine. Mellanox UDA package takes advantage of the silicon architectures of ConnectX®-3, ConnectX®-2 and ConnectX® based InfiniBand and Ethernet adapter cards. For details, please refer to the relevant adapter card user manual available under www.mellanox.com -> Products -> Adapters.

When using an OEM pre-installed card please refer to the OEM server user manual.

Mellanox UDA requires the installation of Mellanox OFED for Linux driver, version 2.2 or later. Mellanox UDA is currently supported on Linux based machines only. Visit the driver Web page below to access software and documents. The supported Linux distributions and kernels are listed in the release notes file; the installation instructions are provided in the user manual.

See www.mellanox.com -> Products -> Software > InfiniBand/VPI Drivers -> Linux SW/Drivers

2.2 Setting up the Switch System

Mellanox UDA benefits from lossless fabric characteristics and requires an RDMA based network. The RDMA capability is available on InfiniBand and RoCE (RDMA over Converged Ethernet) based networks. For the best performance of Mellanox UDA, it is recommended to use Mellanox Ethernet and InfiniBand switches as the software utilizes their architectures.

Visit www.mellanox.com -> Products -> Switches for the state-of-the-art switch portfolio Mellanox offers for Big Data clusters.

3 Installing, Configuring and Running UDA Software

3.1 Supported Operating Systems

Please refer to the product release notes.

3.2 Installation Prerequisites

Prior to installing UDA on a cluster node:

1. Make sure you have a Hadoop environment installed and running on the node.
2. Make sure `ulimit -l` is set to unlimited in all slaves and master nodes.

If it is not set:

- a. Add the following line to your `~/.bashrc` file.

```
ulimit -l unlimited
```

- b. Set the parameters below as follow in the `/etc/security/limits.conf` file.

```
* soft memlock unlimited
* hard memlock unlimited
```

3. Increase the maximum number of memory translation table segments per HCA¹.

```
# echo "options mlx4_core log_num_mtt=24 log_mtts_per_seg=0" > /etc/modprobe.d/mofed.conf
```

- a. Reboot the server or restart the `openibd`.

To restart the `openibd`:

```
# sudo service openibd restart
```

- b. Verify the changes took effect.

```
# cat /sys/module/mlx4_core/parameters/log_num_mtt
# cat /sys/module/mlx4_core/parameters/log_mtts_per_seg
```

4. Disable swap on all the nodes in the cluster.

The swap option can be disabled as follow:

- Edit the `/etc/fstab` file. Remove the swap file system and run once the command below.²

```
# swapoff -a
```

5. [Optional] Run Open MPI to verify RDMA connectivity in your cluster:

- a. Set `HOSTS` variable to a comma delimited list of your cluster's host names.
- b. Set `NUM_OF_HOSTS` to the number of hosts in your cluster.
- c. Set `OPENMPI_VER` variable to your Open MPI version.
- d. Set `MLX_PORT` to the ConnectX port your cluster is using (i.e: for port 1, set it to `mlx4_0:1`)

1. If you need more than 64GB, you can increase the maximum amount of available RDMA memory by increasing the value of `log_mtts_per_seg`.

2. We recommend using this option as it is a one time operation.

- e. Run the following as root user:

```
/usr/mpi/gcc/openmpi- $\{OPENMPI\_VER\}$ /bin/mpirun --allow-run-as-root --display-map -H  $\{HOSTS\}$ 
-np  $\{NUM\_OF\_HOSTS\}$  -mca btl_openib_ib_rnr_retry 0 -mca btl_openib_ib_retry_count 0 -mca
coll_fca_enable 0 --bind-to core --map-by node --display-map -mca pml obl -mca btl
self,sm,openib -mca btl_openib_cpc_include rdmacm -mca btl_openib_if_include  $\{MLX\_PORT\}$  /
usr/mpi/gcc/openmpi- $\{OPENMPI\_VER\}$ /tests/IMB-3.2.4/IMB-MPI1 alltoall
```

3.3 Installing UDA

The following steps describe how to install the UDA distribution.

In case you are using Cloudera Hadoop 5 and above with Cloudera Manager, please refer to [Appendix B: “Configuring UDA on a CDH5+ Cluster via Cloudera Manager,”](#) on page 25.

In case you are using HDP2 and above with Amabri, please refer to [Appendix C: “Configuring UDA on a HDP Cluster via Ambari,”](#) on page 27

Step 1. Install Apache Hadoop 1.x.y or Hadoop 2.x.y. The installation guide and configurations of Apache Hadoop are available at hadoop.apache.org.

Step 2. Test your vanilla Hadoop installation to make sure you have a successful and tuned installation. For tuning and configuration details, see http://hadoop.apache.org/common/docs/<Hadoop Version>/cluster_setup.html.

Step 3. [Optional] Patch Hadoop with Mellanox plugin ability patch.

In case you are using one of Hadoop versions below, please patch your Hadoop with Mellanox plugin-ability patch.

- CDH 4.x.y MRv1 prior to CDH 4.4.0
- HDP 1.1
- community hadoop-1.x.y prior to hadoop-1.3.0
- community hadoop-2.x.y prior to hadoop-2.3.0

1. Download the appropriate patch from the UDA's Github repository at <http://github.com/Mellanox/UDA/wiki/Downloads>.

2. Apply the patch as follows¹:

```
# cd <hadoop extraction directory>
# patch -p0 <patch_name>
# echo $?
```

3. Build according to the instruction in [Appendix A: “Patching and Building Hadoop,”](#) on page 24.

4. [Optional] Run a Terasort job.

Step 4. Install the UDA RPM on all cluster nodes.

1. Use the following install command:

```
# sudo rpm -Uvh <rpm location>
```

1. Please verify the output of this command is 0

2. Make sure that all the files were successfully installed by running the following query. Expected output is listed as command output below.

```
# rpm -ql libuda
/usr/lib64/uda/LICENSE.txt
/usr/lib64/uda/README
/usr/lib64/uda/journal.txt
/usr/lib64/uda/libuda.so
/usr/lib64/uda/source.tgz
/usr/lib64/uda/uda-hadoop-1.x-old.jar
/usr/lib64/uda/uda-hadoop-1.x.jar
/usr/lib64/uda/uda-hadoop-2.x.jar
/usr/lib64/uda/uda-hadoop-3.x.jar
/usr/lib64/uda/utils.tgz
```

3. Add at the end of your `hadoop-env.sh` a line containing the jar name matching your hadoop version.

- For Hadoop 1.x.y, HDP 1.1 add to your `hadoop-env.sh` `add`¹:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/lib64/uda/uda-hadoop-1.x.jar
```

- For Hadoop 2.x.y:

- Perform the following command on each of your slaves.:

```
ln -s /usr/lib64/uda/uda-hadoop-2.x.jar <YOUR-HADOOP-HOME>/share/hadoop/common/lib/
```

Note: If you are using CDH, perform:

```
ln -s /usr/lib64/uda/uda-hadoop-2.x.jar <YOUR-HADOOP-HOME>/lib/hadoop/
```

- For CDH4.x.y MRv1:

- Run:

```
hadoop version
```

Example output:

```
$ hadoop version
Hadoop 2.0.0-cdh4.4.0
Subversion file:///data/1/jenkins/workspace/generic-package-rhel64-6-0/topdir/BUILD/hadoop-2.0.0-cdh4.4.0/src/hadoop-common-project/hadoop-common -r c0eba6cd38c984557e96a16ccd7356b7de835e79
Compiled by jenkins on Tue Sep 3 19:33:17 PDT 2013
From source with checksum ac7e170aa709b3ace13dc5f775487180
This command was run using /opt/cloudera/parcels/CDH-4.4.0-1.cdh4.4.0.p0.39/lib/hadoop/hadoop-common-2.0.0-cdh4.4.0.jar
```

- Create a symbolic link to UDA **on each slave** using the path shown in the last line of the output.

```
sudo ln -s /usr/lib64/uda/uda-hadoop-1.x.jar /opt/cloudera/parcels/CDH-4.4.0-1.cdh4.4.0.p0.39/lib/hadoop/
```

Step 5. Restart MapReduce and/or YARN.

1. When using the old v1 patch and plugin, add: `export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/lib64/uda/uda-hadoop-1.x-v1.jar`

3.4 UDA Configuration

Assume a cluster with 16 nodes, eagle1 through eagle16, where you wish to set eagle1 as the master of the InfiniBand cluster and the rest as slaves. Similar settings are needed for RoCE based deployments, replacing the InfiniBand host name with the corresponding Ethernet host name.

Step 1. For a single homed machines, skip to the next step.

For multi-homed machines, you first need to configure hadoop to use the right interface by setting the “slave.host.name” property. Note, this is a special property and requires each node host to have a unique property value along with the appropriate interface. The host name can be configured as follow:

On all slaves and master, add to the file /etc/hosts the hadoop addresses of all hosts in the cluster (in the format: 40.0.0.1 eagle1.ib.cluster). In this case, configure as follow:

The actions performed below, are supported in Hadoop v1.x.y only.

- hadoop-env.sh:

```
export HADOOP_OPTS="-Djava.net.preferIPv4Stack=true -DHADOOPHOSTNAME=`hostname`.ib.cluster ${HADOOP_OPTS}"
```

- core-site.xml:

```
<property>
  <name>slave.host.name </name>
  <value>${HADOOPHOSTNAME}</value>
</property>
```

Step 2. XML Configuration:

1. HDFS settings:

Merge the following lines into your hdfs-site.xml:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.datanode.dns.interface</name>
    <value>ib0</value>
    <description>The name of the Network Interface from
    which a data node should report its IP address.
    </description>
  </property>
</configuration>
```

2. TaskTracker level settings:

Merge the following lines into your mapred-site.xml:



These lines must be in `mapred-*.xml` to be considered during TaskTracker initialization. Therefore, this step cannot be performed per job only.

```

<?xml version="1.0"?>
  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  <configuration>
    <property>
      <name>mapreduce.shuffle.provider.plugin.classes</name>
      <value>com.mellanox.hadoop.mapred.UdaShuffleProviderPlugin,
org.apache.hadoop.mapred.TaskTracker$DefaultShuffleProvider</value>
      <description>1A comma-separated list of classes that should be loaded as
      ShuffleProviderPlugin(s).
      A ShuffleProviderPlugin can serve shuffle requests from reducetasks.
      Each class in the list must be an instance of
      org.apache.hadoop.mapred.ShuffleProviderPlugin.
      </description>
    </property>

    <property>
      <name>mapred.tasktracker.dns.interface</name>
      <value>ib0</value>
    </property>
  </configuration>

```

1. The example below is used for the old v1 version of Mellanox plugin (uda-hadoop-1.x-v1.jar).

```

<property>
  <name>mapred.tasktracker.shuffle.provider.plugin</name>
  <value>com.mellanox.hadoop.mapred.UdaShuffleProviderPlugin</value>
  <description>Represents plugin for shuffle at TaskTracker side.
  Default value is: (empty string)
  You can also try: com.mellanox.hadoop.mapred.UdaShuffleProviderPlugin
</description>
</property>

```

3. Optional settings:

The following are optional default parameter settings for UDA.

```
<?xml version="1.0"?>
  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
  <configuration>
    <property>
      <name>mapred.rdma.compression.buffer.ratio</name>
      <description>The ratio in which memory is divided between RDMA buffer and
decompression buffer (used only with intermediate data compression)
      </description>
      <value>0.20</value>
    </property>
    <property>
      <name>mapred.rdma.cma.port</name>
      <description>Port number to be used for the RDMA connection
      </description>
      <value>9011</value>
    </property>
    <property>
      <name>mapred.rdma.wqe.per.conn</name>
      <description>Number of allocated Work Queue Elements (WQEs)
for Receive Queue per connection.
      </description>
      <value>256</value>
    </property>
    <property>
      <name>mapred.rdma.buf.size</name>
      <value>1024</value>
      <description>Used by both UdaShuffleProvider and
UdaShuffleConsumer:
      <ul style="list-style-type: none; padding-left: 0;">
        <li>- UdaShuffleProvider (TaskTracker): determines the RDMA&AIO
          Buffers size to satisfy Map Output's RDMA fetch requests</li>
        <li>- UdaShuffleConsumer (Reducer): user preferred RDMA buffer
          size for fetching map outputs. Size is in KB and must be
          aligned to page size.</li>
      </ul>
      </description>
    </property>
```

```

<property>
  <name>mapred.rdma.compression.buffer.ratio</name>
  <description>The ratio in which memory is divided between RDMA buffer and
  decompression buffer (used only with intermediate data compression)
</description>
  <value>0.20</value>
</property>
<property>
  <name>mapred.rdma.buf.size.min</name>
  <value>32</value>
  <description>UDA reducer allocates RDMA buffers according to
  'mapred.rdma.buf.size'. If the buffer size is too big then a
  smaller buffer will be used while 'mapred.rdma.buf.size.min'
  is the limit.
  Bigger RDMA buffers improve the shuffle performance.
  Too small buffer sizes can significantly reduce performance.
  The task will fail if the reducer needs to use a buffer size
  smaller than 'mapred.rdma.buf.size.min'.
</description>
</property>
</configuration>

```

4. YARN configuration, for Hadoop-2.x.y. It requires modifying the yarn-site.xml^{1,2}.

```

<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle,uda_shuffle</value>
</property>

<property>
  <name>yarn.nodemanager.aux-services.uda_shuffle.class</name>
  <value>com.mellanox.hadoop.mapred.UdaShuffleHandler</value>
</property>
<property>
  <name>mapreduce.job.shuffle.provider.services</name>
  <value>uda_shuffle</value>
  <description>A comma-separated list of classes that should be loaded as addi
  tional ShuffleProviderPlugin(s). A ShuffleProviderPlugin can serve shuffle
  requests from reducetasks.
</description>
</property>

```

5. Verify mapreduce.application.classpath property value in mapred-site.xml contains \$HADOOP_MAPRED_HOME. If not, add UDA JAR path to mapreduce.application.classpath.

3.4.1 RDMA Plug-in Parameters Basic Tuning Guidelines

- UdaShuffleProviderPlugin allocates buffers for reading MOFs from the disk and for writing them using RDMA to satisfy reduce task shuffle requests. Therefore, UdaShuf-

1. If you are using Cloudera Manager, make sure you insert the above snippet in both Service-Wide and Gateway categories.
 2. If you already have entry with the name "yarn.nodemanager.aux-services.mapreduce_shuffle.class", please add this entry in addition to it, do not replace it!

fileProviderPlugin's buffer size determines the max buffer size to be used also by reduce tasks.

- When TaskTracker is spawned and the UdaShuffleProviderPlugin is initialized, it is essential that the `mapred.rdma.buf.size` parameter is properly configured to satisfy reducers. RDMA buffers from each reducer are allocated from `mapred.child.java.opts * mapred.job.shuffle.input.buffer.percent`.

When UDA is enabled, each reducer must allocate $2 * \#MOFs = 2 * \text{Dataset/Blocksize}$. Unless you have memory issues we recommend that each RDMA buffer will be of size 1024 KB for optimum performance.

For example, when running a job with a 100GB input size and a 256MB split size, 600 MOFs are created. This requires configuring at least 1200 buffers. Continuing with the above example, a configuration that runs 4 slots of reducers per node requires the allocation of $4 \times 1200 = 4800$ buffers for the job. By using `mapred.rdma.buf.size=1024`, a total of 4800MB is allocated per node.

3.5 UDA Log Setting

UDA Consumer and Provider logs are now integrated with the Hadoop log system and their properties are configured via the Hadoop's `log4j.properties` file (in your `<hadoop-conf-dir>`).

The Consumer logs are integrated into the ReduceTask whereas the Provider logs are integrated into the TaskTracker.

To configure these modules, add the lines below to the `log4j.properties` files:

- `log4j.logger.org.apache.hadoop.mapred.ShuffleConsumerPlugin=<log_level>`
- `log4j.logger.org.apache.hadoop.mapred.ShuffleProviderPlugin=<log_level>`

The ShuffleProviderPlugin logging level can be changed at runtime. To do so, type:

```
#> bin/hadoop daemonlog -setlevel <hostname>:50060 org.apache.hadoop.mapred.ShuffleProvider-
Plugin <log level>
```



In the example above, 50060 is the default port value of `mapreduce.task-tracker.http.address`

If logs are not modified, UDA log level will set to the default setting of the distribution (default is INFO).

3.6 Running UDA

➤ *To verify UDA was installed properly, run a simple MapReduce job:*

Step 1. Export the following variable in bash.

```
export UDA_ENABLE="-Dmapred.reduceetask.shuffle.consumer.plugin=com.mellanox.
hadoop.mapred.UdaShuffleConsumerPlugin -Dmapreduce.job.reduce.shuffle.consumer.plugin.
class=com.mellanox.hadoop.mapred.UdaShuffleConsumerPlugin"
```



To run MapReduce jobs with UDA, the above properties must be passed via the command line or alternatively, set in `mapred-site.xml`.

Step 2. Run Pi.

```
hadoop jar <mapreduce-lib-path>/hadoop-mapreduce-examples.jar pi $UDA_ENABLE 8 2000
```

Step 3. Verify UDA ran successfully. Make sure the following line exists in the job's reducer log.

```
====XXX Successfully closed UdaShuffleConsumerPlugin XXX====
```

If you are using YARN with log aggregation, use the following command to retrieve MapReduce's logs.

```
yarn logs -applicationId <applicationId>
```

Appendix A: Patching and Building Hadoop

This section provides the procedure to add to the supported Hadoop distributions the plug-in ability if the ability is lacking, enabling the application to utilize or disable UDA.

If you are using one of Hadoop versions below, please patch your Hadoop with Mellanox plugin-ability patch.

- CDH 4.x.y MRv1 prior to CDH 4.4.0
- HDP 1.1
- community hadoop-1.x.y prior to hadoop-1.3.0
- community hadoop-2.x.y prior to hadoop-2.3.0

➤ **To patch and build Hadoop:**

Step 1. Download Hadoop from <http://hadoop.apache.org/releases.html>.

Step 2. Extract the tarball on all the nodes and test your installation.

Step 3. Download Mellanox's patch from <http://github.com/Mellanox/UDA>.

The URL is: <http://www.mellanox.com/downloads/UDA/HADOOP-1.x.y-v2.patch>

Step 4. Patch hadoop.

1. Extract hadoop-x.y.z.tar.gz in a temp directory.
2. Change directory into the extraction directory.
3. Run the Mellanox patch.

```
# patch -p0 < HADOOP-1.x.y.patch
```

4. Verify the previous operation was successful.

The expected result should be 0.

```
# echo $?
```

Step 5. Build your patched Hadoop.

Hadoop-1.x.y example:

```
ant -Djava5.home=/usr/lib64/java/jdk1.6.0_25 clean tar
```

This will create you a new tar.gz file under the ./build/ dir. (notice that the result will be called hadoop-1.1.3-SNAPSHOT since it is not a default 1.1.2 version).

Hadoop-2.x.y example:

```
mvn package -Pdist -DskipTests -Dtar
```


Appendix B: Configuring UDA on a CDH5+ Cluster via Cloudera Manager

This section was written under the following assumptions:

- Your Cloudera cluster is configured to use an RDMA-supported interface.
- Each of your cluster's nodes has MLNX_OFED 2.2 or later.

If you are using an earlier version, you may have to build UDA manually.

For further information and for the source code, please refer to our Github repository at <http://github.com/Mellanox/UDA>.

Step 1. Download UDA's RPM from <http://github.com/Mellanox/UDA> and install it on all nodes:

```
rpm -i /path/to/libuda-3.4.1-0.1034.el6.x86_64.rpm
```

Step 2. Create a symbolic link to UDA's JAR in `#{HADOOP_HOME}/lib/hadoop` on all nodes:

```
ln -s /usr/lib64/uda/uda-hadoop-2.x.jar #{HADOOP_HOME}/lib/hadoop
```

Step 3. Go to the Cloudera Manager:

Step a. Enter **Clusters -> Services -> YARN -> Configuration**.

Step b. Insert the following property in the advanced code snippet for **mapred-site.xml** of the **Service-Wide** category (see [Figure 1](#)):

```
<property>
<name>mapreduce.shuffle.provider.plugin.classes</name>
<value>com.mellanox.hadoop.mapred.UdaShuffleProvider-
Plugin,org.apache.hadoop.mapred.TaskTracker$DefaultShuffleProvider</value>
</property>
```

Step c. Insert the following properties in the advanced code snippet for **yarn-site.xml** in both **Service-Wide** and **Gateway** categories (see [Figure 2](#)).
Make sure to **Save Changes** afterwards:

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle,uda_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.uda_shuffle.class</name>
<value>com.mellanox.hadoop.mapred.UdaShuffleHandler</value>
</property>
<property>
<name>mapreduce.job.shuffle.provider.services</name>
<value>uda_shuffle</value>
</property>
```

Step d. Verify the `mapreduce.application.classpath` property value in contains `#{HADOOP_MAPRED_HOME}`. If not, add UDA JAR path to `mapreduce.application.classpath`.

Step 4. Restart YARN.

Go to the Cloudera Manager -> **Clusters -> Services -> YARN -> Actions -> Restart**

Step 5. [Optional] In BASH, export UDA_ENABLE environment variable.

```
export UDA_ENABLE="-D mapreduce.job.reduce.shuffle.consumer.plugin.class=com.mellanox.hadoop.mapred.UdaShuffleConsumerPlugin"
```

Step 6. Run a small MapReduce job with UDA.

```
hadoop jar /path/to/hadoop-mapreduce-examples.jar pi ${UDA_ENABLE} 8 200
```

Step 7. Verify that UDA was used by checking the reducer's log. Look for the following:

```
====XXX Successfully closed UdaShuffleConsumerPlugin XXX====
```



In case you have YARN's log aggregation enabled, the application's logs can be fetched using:

```
yarn logs -applicationId <applicationId>
```

Figure 1: Configuring mapred-site.xml via Cloudera Manager

The screenshot shows the Cloudera Manager interface for configuring the `mapred-site.xml` file. The search bar contains "mapred-site". The configuration table is as follows:

Category	Property	Value	Description
Service-Wide / Advanced	YARN Service MapReduce Advanced Configuration Snippet (Safety Valve)	<pre><property> <name>mapreduce.shuffle.provider.plugin.classes</name> <value>com.mellanox.hadoop.mapred.UdaShuffleProviderPlugin,org.apache.hadoop.mapred.TaskTracker\$DefaultShuffleProvider</value> </property></pre>	For advanced use only, a string to be inserted into <code>mapred-site.xml</code> . Applies to configurations of all roles in this service except client configuration.

Figure 2: Configuring yarn-site.xml via Cloudera Manager

The screenshot shows the Cloudera Manager interface for configuring the `yarn-site.xml` file. The search bar contains "yarn-site". The configuration table is as follows:

Category	Property	Value	Description
Service-Wide / Advanced	YARN Service Advanced Configuration Snippet (Safety Valve) for yarn-site.xml	<pre><property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle,uda_shuffle</value> </property> <property> <name>yarn.nodemanager.aux-services.uda_shuffle.class</name> <value>com.mellanox.hadoop.mapred.UdaShuffleHandler</value> </property></pre>	For advanced use only, a string to be inserted into <code>yarn-site.xml</code> . Applies to configurations of all roles in this service except client configuration.
Gateway Default Group / Advanced	YARN Client Advanced Configuration Snippet (Safety Valve) for yarn-site.xml	<pre><property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle,uda_shuffle</value> </property> <property> <name>yarn.nodemanager.aux-services.uda_shuffle.class</name> <value>com.mellanox.hadoop.mapred.UdaShuffleHandler</value> </property></pre>	For advanced use only, a string to be inserted into the client configuration for <code>yarn-site.xml</code> .

Appendix C: Configuring UDA on a HDP Cluster via Ambari

This section was written under the following assumptions:

- Your Cloudera cluster is configured to use an RDMA-supported interface.
- Each of your cluster's nodes has MLNX_OFED 2.2 or later.

If you are using an earlier version, you may have to build UDA manually.

For further information and for the source code, please refer to our Github repository at <http://github.com/Mellanox/UDA>.

Step 1. Download UDA's RPM from <http://github.com/Mellanox/UDA> and install it on all nodes:

```
rpm -i /path/to/libuda-3.4.1-0.1034.el6.x86_64.rpm
```

Step 2. Create a symbolic link to UDA's JAR in `/usr/hdp/current/hadoop-mapreduce-client/lib/` on all nodes:

```
ln -s /usr/lib64/uda/uda-hadoop-2.x.jar /usr/hdp/current/Hadoop-mapreduce-client/lib/
```

Step 3. In the Amabri, enter **MapReduce2 -> Configs -> Advanced**.

Step a. In the **Custom mapred-site** insert the following properties (see [Figure 3](#)):

```
<property>
<name>mapreduce.shuffle.provider.plugin.classes</name>
<value>com.mellanox.hadoop.mapred.UdaShuffleProvider-
Plugin,org.apache.hadoop.mapred.Task-Tracker$DefaultShuffleProvider</value>
</property>
<property>
<name>mapreduce.job.shuffle.provider.services</name>
<value>uda_shuffle</value>
</property>
```

Step b. In the **Advanced mapred-site** verify that the `mapreduce.application.classpath` property value contains `$HADOOP_MAPRED_HOME`. If not, add UDA JAR path to `mapreduce.application.classpath`.

Step 4. In the Amabri enter **Yarn -> Configs -> Advanced**:

Step a. In the **Custom yarn-site** insert the following property (see [Figure 4](#)):

```
<property>
<name>yarn.nodemanager.aux-services.uda_shuffle.class</name>
<value>com.mellanox.hadoop.mapred.UdaShuffleHandler</value>
</property>
```

Step b. In the **Node Manager** insert the following property (see [Figure 5](#)):

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle,uda_shuffle</value>
</property>
```

Step 5. Restart YARN.

In the Ambari -> **Yarn- > Service Actions- > Restart all**

Step 6. Restart MapReduce2.

In the Ambari -> **MapReduce2 -> Service Actions- > Restart all**

Step 7. [Optional] In BASH, export UDA_ENABLE environment variable.

```
export UDA_ENABLE="-D mapreduce.job.reduce.shuffle.consumer.plugin.class=com.mellanox.hadoop.mapred.UdaShuffleConsumerPlugin"
```

Step 8. Run a small MapReduce job with UDA.

```
hadoop jar /path/to/hadoop-mapreduce-examples.jar pi ${UDA_ENABLE} 8 20
```

Step 9. Verify that UDA was used by checking the reducer's log. Look for the following:

```
====XXX Successfully closed UdaShuffleConsumerPlugin XXX====
```



In case you have YARN's log aggregation enabled, the application's logs can be fetched using:

```
yarn logs -applicationId <applicationId>
```

Figure 3: Configuring mapred-site.xml via Ambari

The screenshot shows the Ambari web interface for configuring the MapReduce2 service. The left sidebar lists various services, with MapReduce2 selected. The main area shows the configuration for the 'MapReduce2 Default (4)' group. Under the 'Custom mapred-site' section, two configuration entries are visible:

- `mapreduce.job.shuffle.provider.services` is set to `uda_shuffle`.
- `mapreduce.shuffle.provider.plugin.classes` is set to `com.mellanox.hadoop.mapred.UdaShuffleProviderPlugin,org.apache.hadoop.mapred.Tat`.

Figure 4: Configuring yarn-site.xml (Cstom yarn-site) via Ambari

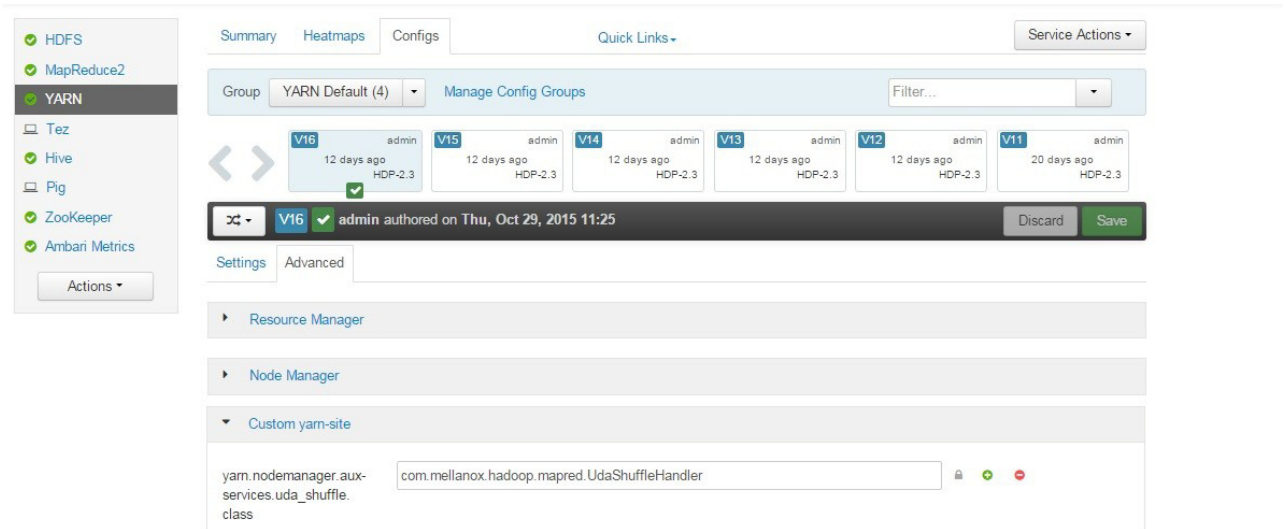


Figure 5: Configuring yarn-site.xml (Node Manager) via Ambari

