# AI400X2 24x3.5TiB nvme 4xHDR200

System description (Multi-host)

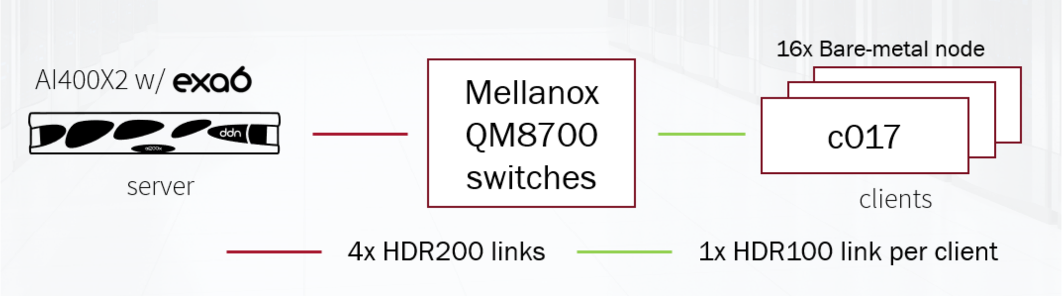
DDN

August 4, 2023

## Contents

# 1 Hardware

The hardware is composed of one client and one server connected to the same switch.



## 1.1 Client

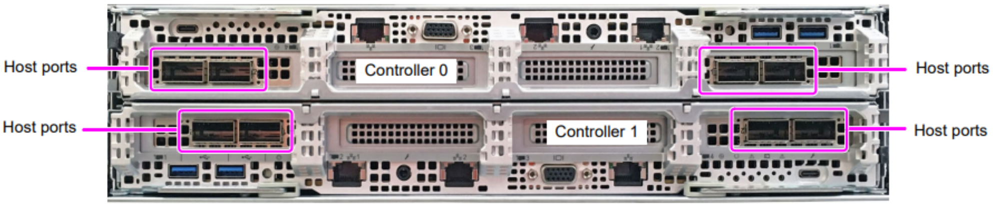
The client are **bare-metal** node (compute) with 1 CPU socket and 96GiB of RAM based on Intel S2600BPB motherboard. All clients are configured the same (hw & sw). The clients communicate with the server through IB using one dedicated link.

- 16x HDR100 links total (one per client)

## 1.2 Server

The server is **a commercial DDN AI400X2** appliance (storage) providing a single EXAScaler filesystem to the clients (one filesystem total). The configuration is the default one with 24x SAMSUMG NVME drives with 3.5TiB per drive. The server communicate with the client through IB:

- 4x dual ports HDR200 adapters for a total of 4x HDR200 links (only one link per HCA)



## 1.3 Switch

Mellanox QM8700 switches. The server and clients are connected on it.

# 2 Software

## 2.1 Client

The clients distribution is Rocky Linux 8.7 (Green Obsidian). The Mellanox OFED version is `MLNX_OFED_LINUX-5.9-0.5.6.0` (OFED-5.9-0.5.6)

## 2.2 Server

The AI400X2 uses EXAScaler SFA Rocky version 6.2.0-r9 on top of the SFAOS stack (SFAOS version 12.2.0.3 ) The EXAScaler driver uses on the server is `2.14.0_ddn85`. This is the GA version of the product with no modification applied. The OFED version is OFED-internal-5.8-2.0.3.

## 2.3 Switch

NA

# 3 Settings

## 3.1 Client

The standard EXAScaler 6.2.0 driver (GA) is installed on the client using the default installation procedure. This driver is used for the client to access the remote EXAScaler filesystem.

### 3.1.1 EXAScaler driver configuration

The EXAScaler configuration file has been generated automatically following the default installation procedure. No tuning is applied. Therefore checksums are enabled, which is known to decrease the performance. This configuration file is necessary to mount the filesystem and is generated automatically. It is not part of an optimization or customization. It is the default configuration file.

Content of /etc/modprobe.d/lustre.conf on each client:

```
# This file has been generated by exa-client-deploy
#
# Do not edit unless exa-client-deploy service is stopped & disabled
# e.g: 'systemctl status exa-client-deploy'
#

options lnet networks="o2ib(ib0)"
options lnet lnet_transaction_timeout=100
options lnet lnet_retry_count=2
options ko2iblnd peer_credits=32
options ko2iblnd peer_credits_hiw=16
options ko2iblnd concurrent_sends=64
options ksocklnd conns_per_peer=0
```

## 3.2 Server

No tuning applied on the DDN AI400X2 after the initial installation. The inital installation was automated using the latest version of our manufacturing script (mfg-e-dcr-20230627-1). The filesystem was mounted on the client after the end of this script.

## 3.3 Switch

No tuning applied.

# 4 Misc

## 4.1 Drop cache

The cache of both the clients and the server is dropped between two iterations using the following function for all tests. The function return an error if a command fails. Clustershell ’-S’ option is use to ensure proper exit code is retrieved.

```
clear_caches()
{
    # Clear cache on client
    clush -abS "sync; echo 3 > /proc/sys/vm/drop_caches"  || return 1
    # Clear cache on EXAScaler VMs
    ssh root@${exafsip}  "clush -abS ’sync; echo 3 > /proc/sys/vm/drop_caches’" || return 1
}
```

## 4.2 No tunings

To ensure no tunings are applied on the clients, the EXAScaler driver is reconfigured from scratch on each client before the benchmark run.

```
umount_mount_on_all_nodes()
{
    # If the filesystem is mounted: umount
    clush -abS "if mount -t lustre |grep "." ; then \
    umount -t lustre -a && lustre_rmmod ; fi" || return 1
    # If the kernel module is loaded: unload
    clush -abS "if lsmod |grep lustre ; then lustre_rmmod ; fi" || return 1

    # Remove any previous deployment script
    clush -abS "if [ -d /root/exa-client ] ; then rm -rf /root/exa-client ; fi" || return 1
    # Remove previous config file
    clush -abS "if [ -f /etc/modprobe.d/lustre.conf ] ; then \
    rm /etc/modprobe.d/lustre.conf ; fi" || return 1

    # Download the deployment script
    scp <ip>:/scratch/EXA*/exa-client-6.2.0.tar.gz /root/ || return 1
    (cd /root && tar xvf exa-client-6.2.0.tar.gz ) || return 1
    clush -abcS /root/exa-client/ || return 1

    # Apply the deployment script on all nodes
    clush -abS ’(cd /root/exa-client && \
    ./exa_client_deploy.py -c -y --lnets "o2ib(ib0)")’ || return 1

    # Mount the exascaler filesystem on all nodes
    clush -abS "mkdir -p /lustre/ai400x2/client" || return 1
    sleep 5
    if ! clush -abS "mount -t lustre (...)" ; then
  sleep 5
  clush -abS "mount -t lustre (...)" || return 1
    fi
}
```

### 4.3 Logs

All the commands were logged and checked for any error. The following loop was used with bert and unet3d.

```
for cmd in check_number_of_nodes \
         umount_mount_on_all_nodes \
         check_nodes_are_the_same \
         find_exa_fs \
         sync_run_directory \
         create_result_dir \
         find_total_mem \
         find_nb_files \
         cleanup_dirs_on_each_node \
         create_files_from_each_node \
         run_bert ; do
    printf "%-60s" "$(date +’%m/%d/%Y %H:%M’): \
    $(echo $cmd|sed ’s@_@ @g’|sed -E ’s@(^.)@\u\1@g’)  "
    if ! $cmd > ${LOGDIR}/cmds/${cmd}.log 2>&1 ; then
        printf "%s\n" "x"
        printf "See ${LOGDIR}/cmds/${cmd}.log\n"
        exit 1
    else
        printf "%s\n" ""
    fi
done
```