Fabric Collective Accelerator (FCA)

User Manual

Version 2.5

Last Modified: December 23, 2014

www.mellanox.com

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 PRODUCTO(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

Mellanox Technologies, Ltd.
Beit Mellanox
PO Box 586 Yokneam 20692
Israel
www.mellanox.com
Tel: +972 (0)74 723 7200
Fax: +972 (0)4 959 3245

© Copyright 2014. Mellanox Technologies. All Rights Reserved.

Mellanox®, Mellanox logo, BridgeX®, ConnectX®, Connect-IB®, CoolBox®, CORE-Direct®, InfiniBridge®, InfiniHost®, InfiniScale®, MetroX®, MLNX-OS®, PhyX®, ScalableHPC®, SwitchX®, UFM®, Virtual Protocol Interconnect® and Voltaire® are registered trademarks of Mellanox Technologies, Ltd.

ExtendX™, FabricIT™, Mellanox Open Ethernet™, Mellanox Virtual Modular Switch™, MetroDX™, TestX™, Unbreakable-Link™ are trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

Contents

List of Figures

List of Tables

Revision History

Version 2.5 (Sep 30 2014)

- Updated section Building OpenMPI 1.6.x with FCA Support (on page 17)

Version 2.5 (Sep 30 2014)

- Removed the "`osm_type`" and "`ufm_url` " parameters from Table 4 (on page 18)

Version 2.5 (Dec 2012)

- Removed section Upgrading from FCA 2.0 or Later
- Updated the following sections:
  - FCA Installation Package Content (on page 12)
  - Downloading the FCA Software (on page 13)
  - Prerequisites (on page 14)
  - Building OpenMPI 1.6.x with FCA Support (on page 17)
  - Installing the FCA Manager on a Dedicated Node (on page 15)
  - Installing the FCA Manager from RPM (on page 15)

Version 2.2 (May 2012)

- Removed section Activating the Software License
- Updated the following sections:
  - Prerequisites (on page 14)
  - Installing the FCA Manager on a Dedicated Node (on page 15)
  - Configuring a Specific Rule (on page 23)
  - Upgrading from FCA 2.0 or Later
  - Starting the FCA Manager (on page 16)

Version 2.1.1 (December 2011)

- Updated the following sections to reflect offloading collective operations onto HCA:
  - Overview (text and graphics)
  - Supported Topologies
  - FCA Installation Package Content
- Updated Prerequisites and Installation sections for 2.1.1.
- Removed section on configuring Grid Director switches to enable FCA.
- Added note for OpenMPI 1.5.x in section "Building OpenMPI 1.4.x with FCA Support."

Preface

Audience

The intended audience for the Mellanox Fabric Collective Accelerator (FCA) User Manual is the MPI implementer and the network administrator responsible for managing FCA on Mellanox InfiniBand switches. It is assumed that the administrator is familiar with advanced concepts in network management.

Related Documentation

The following document is part of the library for network administrators and installers supporting the Mellanox FCA:

Document Name	Part Number
Mellanox Fabric Collective Accelerator Release Notes	DOC-00984

Typographical Conventions

Before you start using this guide, it is important to understand the terms and typographical conventions used in the documentation.

The following kinds of formatting in the text identify special information.

Formatting convention	Type of Information
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.
Emphasis	Use to emphasize the importance of a point or for variable expressions such as parameters.
CAPITALS	Names of keys on the keyboard. for example, SHIFT, CTRL, or ALT.
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another, for example, CTRL+P, or ALT+F4.

Document Conventions

NOTE: Identifies important information that contains helpful suggestions.

CAUTION: Alerts you to risk of personal injury, system damage, or loss of data.

WARNING: Warns you that failure to take or avoid a specific action might result in personal injury or a malfunction of the hardware or software. Be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents before you work on any equipment.

1 Introduction to Mellanox Fabric Collective Accelerator

1.1 Overview

The Mellanox Fabric Collective Accelerator (FCA) is a unique solution for offloading collective operations from the Message Passing Interface (MPI) process to the server CPUs. As a system-wide solution, FCA does not require any additional hardware. The FCA manager creates a topology based collective tree, and orchestrates an efficient collective operation using the CPUs in the servers that are part of the collective operation.

FCA accelerates MPI collective operation performance by up to 100 times providing a reduction in the overall job runtime. Implementation is simple and transparent during the job runtime.

FCA is built on the following main principles:

- Topology-aware Orchestration

  The MPI collective logical tree is matched to the physical topology. The collective logical tree is constructed to assure:

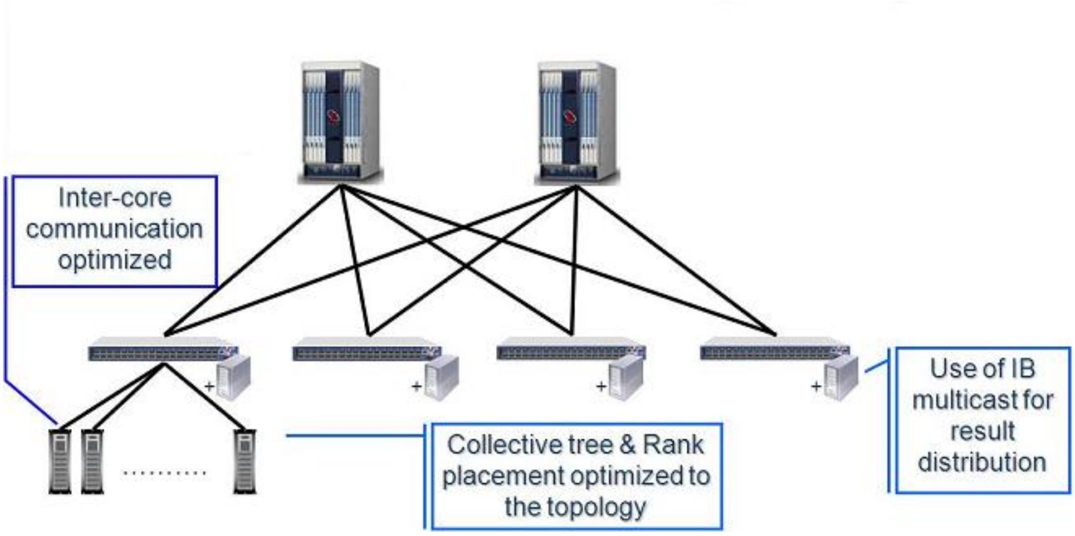
  - Maximum utilization of fast inter-core communication
  - Distribution of the results.

- Communication Isolation

  Collective communications are isolated from the rest of the traffic in the fabric using a private virtual network (VLane) eliminating contention with other types of traffic.

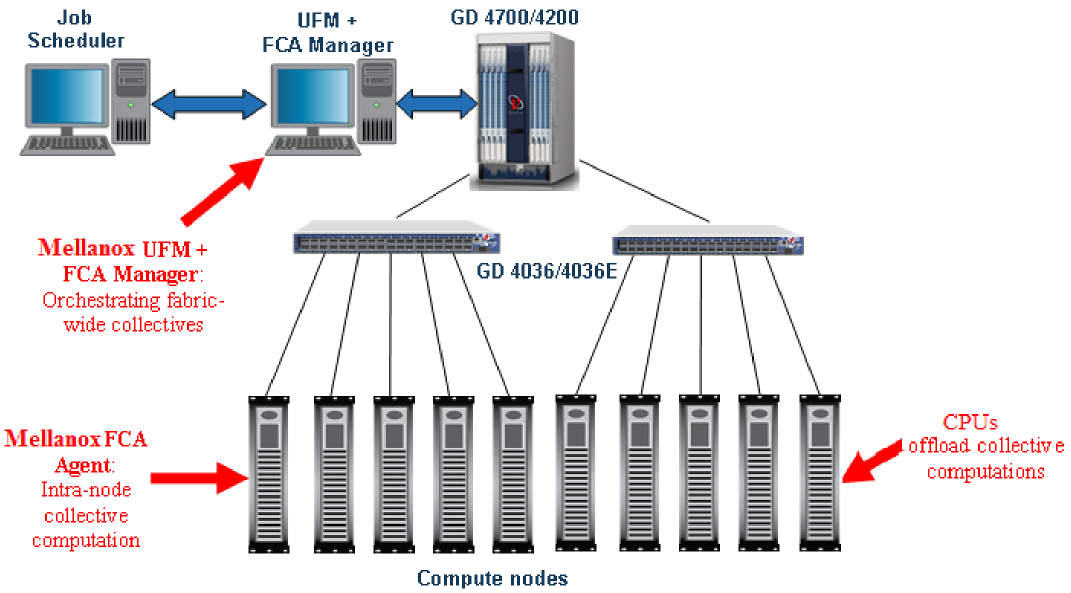
The following diagram summarizes the FCA architecture:

Figure 1: FCA Architecture


Inter-core communication optimized
Collective tree & Rank placement optimized to the topology
Use of IB multicast for result distribution

The following diagram shows the FCA components and the role that each plays in the acceleration process:

Figure 2: FCA Components


Job Scheduler
UFM + FCA Manager
GD 4700/4200
Mellanox UFM + FCA Manager: Orchestrating fabric-wide collectives
GD 4036/4036E
Mellanox FCA Agent: Intra-node collective computation
CPUs offload collective computations
Compute nodes

1.2 Supported MPI Collectives

FCA addresses a wide range of applications with out-of-the-box integration with leading MPI implementations such as Platform MPI and Open MPI, and requires no changes to the application.

The following MPI collectives are currently supported by FCA and accelerated:

- MPI_Reduce
- MPI_Allreduce
- MPI_Barrier
- MPI_Bcast
- MPI_AllGather
- MPI_AllGatherv

FCA supports an unlimited message size and advanced optimizations for Torus topologies. It can work with any InfiniBand Subnet Management based software (OpenSM, Embedded SM, Host SM).

FCA supports the following data types for Reduce and Allreduce operations:

- All data types for C language bindings, except MPI_LONG_DOUBLE.
- All data types for C reduction functions (C reduction types).
- The following data types for FORTRAN language bindings:
  - MPI_INTEGER
  - MPI_INTEGER2

  - MPI_INTEGER4
  - MPI_INTEGER8
  - MPI_REAL
  - MPI_REAL4
  - MPI_REAL8

FCA does *not* support data types for Fortran reduction functions (Fortran reduction types).

1.3 Supported Topologies

- FCA supports almost all fabric topologies:
  - Fat Tree
  - HyperScale
  - Torus
- FCA requires a Mellanox based Infiniband network.

1.4 Planning the Server Configuration

Following are points to consider when planning on which server to install the FCA Manager:

- The FCA Manager should be installed on a different server than the one where MPI jobs will run.
- If you *do not* have two servers running UFM for redundancy, you should install FCA Manager on the UFM server.
- If you *do* have two servers running UFM for redundancy, it is best to install FCA Manager on a non-UFM server and provide it with the virtual address of the UFM machine.
- Only a single instance of FCA Manager should be running in the fabric.

1.5 FCA Software Components

The FCA-related software components are listed in the following table:

Table 1: FCA-Related Packages

Package	Description
FCA Manager	The FCA Manager is server software that responds to requests from the MPI application to set up new communicators.
FCA MPI Runtime Libraries	The FCA MPI Runtime library is a user-level shared library which is integrated with specific MPI distributions (IBM PE, OpenMPI, Platforms MPI, MVAPICH2), that is responsible for offloading MPI collective operations into Fabric.

1.6 FCA Installation Package Content

The FCA installation package includes the following items:

- FCA - Mellanox Fabric Collector Accelerator Installation files
  - fca-<version>.x86_64.<OS>.rpm
  - fca-<version>.x86_64.<OS>.tar.gz

    where:

    <version> is the version of this release, and

    <OS> is one of the supported Linux distributions listed in Prerequisites (on page 14).
- Mellanox Fabric Collective Accelerator (FCA) Software: End-User License Agreement
- FCA Manager software
- FCA MPI runtime libraries
- Mellanox Fabric Collective Accelerator (FCA) User Manual
- Mellanox Fabric Collective Accelerator (FCA) Release Notes

2 Installation and Initial Configuration

2.1 Overview of Installation and Initial Configuration

FCA software includes the FCA Manager and the FCA MPI runtime support libraries. FCA Manager software should be installed on a central management node. For optimal performance, and to minimize interference with other applications, it is recommended to use a dedicated server for the FCA Manager installation.

The following sections provide step-by-step instructions for installing the FCA Server software, and installing the FCA Agent.

2.1.1 Downloading the FCA Software

This software download process applies to software updates as well as for first time installation.

➢ ***To download the FCA software***

1. Go to the Mellanox website.
2. Click the **Downloads** tab and select the relevant version of the software to download.
3. Save the file on your local drive.
4. Click **Close**.

3 Installing FCA

3.1 Prerequisites

Before you begin be certain that:

1. InfiniBand Subnet Management is installed and running on a dedicated node in the fabric.
2. Mellanox OFED 1.5.3-3.1.0 or later is installed.

   To download the latest MLNX_OFED version, go to:

   Mellanox OpenFabrics Enterprise Distribution for Linux (MLNX_OFED)
3. Mellanox ConnectX®-2 or ConnectX®-3 HCA with firmware version 2.9.1000 or later.

   To download the latest ConnectX® HCA firmware version, go to:

   Firmware Downloads

The minimum system requirements for installing and running FCA are listed in the following table.

NOTE: Mellanox OFED 1.5.3-3.1.0 includes FCA 2.2 and OpenMPI which is compiled with FCA v2.2.
Both packages should be removed prior to installing FCA v2.5. To remove them, run:

```
# rpm -e fca

# rpm -e openmpi
```

Table 2: System Requirements

Item	Requirement
FCA 2.5	
Supported switches	Mellanox IB QDR/FDR switches
Linux distributions (<OS>)	RHEL 6.2
Supported HCAs	Mellanox ConnectX®-2 HCA with firmware version 2.9.1000 or later Mellanox ConnectX®-3 HCA with firmware version 2.10.0000 or later
Open Message Passing Interface (MPI) Project	Open MPI 1.6.3 or later.*
Open Fabrics Enterprise Distribution (OFED™)	1.5.3-3.1.0 or later
Root permission	The installer should have root permissions for post-installation tasks.
InfiniBand Subnet Management	All InfiniBand Subnet Management based software is supported in FCA version 2.5.

3.2 Installing the FCA Manager on a Dedicated Node

FCA Manager must be installed on a dedicated machine, which is not a part of the cluster nodes, with only a single instance of the FCA Manager in operation per fabric.

NOTE: We recommend that the FCA Manager be installed on the same node as OpenSM.

There are two options for installing the FCA Manager when using MLNX_OFED v1.5.3-3.1.0:

- From an RPM (on page 15) - Select this option if you want to install the FCA Manager on the machine`s local disk and let the RPM package handle all post-install tasks
- From Tarball (on page 15) - Select this option if you wish to install FCA Manager in any location (user`s home directory, NFS shared folder, etc.) There are number of post-install tasks that need to be applied as root on every cluster node after you install FCA from a tarball.

Select one of the installation options according to your site's installation policy.

3.2.1 Installing the FCA Manager from RPM

➢ ***To install the FCA Manager on all cluster nodes from an RPM (as root) when using MLNX_OFED v1.5.3-3.1.0***

1. Enter the following command:

```
# rpm -e fca
# rpm -e openmpi
# rpm -ihv fca-2.5.x86_64.rpm
```

2. Set the environment variable pointing to the installed location of FCA in the user login profile.

```
# export FCA_MGR_HOME=/opt/mellanox/fca
```

3. (Optional) Configure the FCA Manager to start automatically after boot.

```
# /etc/init.d/fca_managerd install_service
```

3.2.2 Installing the FCA Manager from Tarball

➢ ***To install the FCA Manager from Tarball in the shared NFS location when using MLNX_OFED v1.5.3-3.1.0***

1. Enter the following commands:

```
# mkdir -p /usr/local/mellanox
# cd /usr/local/mellanox
# tar zxvf fca-2.5.xxxx.x86_64.tar.gz
```

2. Run the following post-install scrip on all hosts.

```
# cd fca-2.5.xxxx.x86_64
#./scripts/udev-update.sh
```

3. Set the environment variable pointing to the extracted location of FCA in the user login profile.

```
# export FCA_MGR_HOME=/usr/local/mellanox/fca-2.5.xxxx.x86_64
```

3.2.3 Starting the FCA Manager

➢ ***To start the FCA Manager***

- Enter the following command

```
# $FCA_MGR_HOME/scripts/fca_managerd start
```

- For RPM setup only enter:

```
/etc/init.d/fca_managerd start
```

- To configure FCA manager to start automatically after boot, run:

```
/etc/init.d/fca_manager install_service
```

4 Installing FCA MPI Support Libraries

You can install the FCA MPI support libraries from either an RPM or from Tarball, on all cluster nodes, or, to the shared NFS location using Tarball.

For further information, see Installing the FCA Manager from RPM (on page 15) and Installing the FCA Manager from Tarball (on page 15).

4.1 Building OpenMPI 1.6.x with FCA Support

NOTE: If you use OpenMPI 1.6.x or later, no patch is required. OpenMPI 1.6.x supports FCA natively.

➢ ***To build OpenMPI 1.6.x with FCA support***

1. Download OpenMPI 1.6.x from the OpenMPI site.

   Enter the following commands:

```
$ mkdir -p $HOME/openmpi
$ cd $HOME/openmpi
$ wget
http://www.open-mpi.org/software/ompi/v1.6.x/downloads/openmpi-1.6.3.tar.
gz
$ tar zxvf openmpi-1.6.3.tar.gz
$ cd openmpi-1.6.3
```

2. Install OpenMPI 1.6.x with FCA support.

```
./autogen.sh && ./configure –with-fca
```

4.2 Verifying the FCA Installation

➢ ***To verify that OpenMPI is working with the FCA installation***

- Enter the following command:

```
$ $MPI_HOME/bin/ompi_info --param coll fca|grep fca_enable
```

  The list of FCA parameters should be displayed as a command output.

4.3 Running MPI Jobs with FCA

Make sure that the FCA manager tarball is unpacked and available from all cluster nodes. Its opened location is referenced below as `$FCA_HOME`.

1. Use the following script examples with the information provided on how to run MPI jobs with FCA for different MPI vendors:
   - For OpenMPI MPI: `$FCA_HOME/scripts/run-ompi-fca.sh`
   - For Platforms MPI: `$FCA_HOME/scripts/run-pmpi-fca.sh`
   - For MVAPICH2: `$FCA_HOME/scripts/run-mvapich2-fca.sh`
2. Check the `$FCA_HOME/etc/fca_mpi_spec.ini` file for various FCA tuning options.

5 Configuring FCA

5.1 FCA Manager Configuration Parameters

The `fca_manager_spec.ini` file is a configuration file containing FCA-related parameters which you can change or overwrite using the command line during runtime.

The FCA Manager process reads its configuration on startup from the *$FCA_MGR_HOME/etc/fca_manager_spec.ini* file.

Depending on the method used to install FCA, the Mellanox-provided parameter file, *fca_manager_spec.ini*, will be located in the path described in the following table:

Table 3: Paths for FCA Manager INI File

Installation Method	Path to fca_manager_spec.ini
From RPM	/opt/mellanox/fca/etc/
From Tarball	$FCA_HOME/etc/

The FCA Manager configuration file is in INI format and contains two sections: "fmm" and "ib".

To set parameter values in the *fca_manager_spec.ini* file, edit the file as necessary using the following format:

```
variable = value
```

Example:

```
[fmm]
debug_level = 5
log_file = fmm.log
```

Table 4: FCA Manager INI File Parameters

Parameter	Description	Values
The following parameters may be changed in the INI file under the [fmm] section:		
debug_level	Verbosity level of fca_manager for debugging. The debug levels are: • 0 (fatal) • 1 (error) • 2 (warn) • 3 (info) • 4 (debug) • 5-7 (detailed debug info)	integer between 0-7 The default is 3.
log_file	FCA Manager log filename. The logfile name can contain printf like tokens, which are substituted during log file creation: • %H - hostname where FCA Manager is running • %D - current date in format	string representing log file name The default is: fmm_%H_%D.log

Parameter	Description	Values
	DDMMYYYY - %T - current thread ID.	
log_file_max_size	This is a critical size in MB of the log_ file to which the file will be rolled. If set to zero - rolling is disabled and file size is unlimited.	any positive valid number The default is 10.
log_file_max_backup_files	Denotes the number of backup files to be created. Effective only when log_file_max_size parameter has value greater than zero.	integer The default is 20.
enable_stdout	Determines whether the log should also be written to the standard output. Valid values: - y or 1 = enable - n or 0 = disable	character The default is enable.
The following parameters may be changed in the INI file under section [ib]:		
dev_name	If set, the specified IB device will be used for communication. The name as appears in */sys/class/infiniband/* directory. If not set, the first device with ACTIVE port is used.	string representing active IB device name The default is set.
port_num	If set, the selected port number is used on the matching device. If not set or zero, the first active port is used.	positive integer The default is unset (use auto-discover).
service_level	Quality of Service (QoS) is offered in IB as a means to offer some guarantees/minimum requirements for certain applications on the fabric. SL2VL mapping should be configured in OpenSM. Valid values: 0-15 Note that OpenSM works by default with QoS values of 0-7.	integer The default is 0.

6 FCA MPI Runtime Library Configuration Parameters

6.1 Specifying FCA Parameters as "mpirun" Command Line Arguments

The FCA runtime library is used by MPI to offload collective operations into IB switches. You can supply configuration parameters to the FCA runtime library. The configuration parameters may be passed to the FCA library by either loading the parameters from a configuration INI file, or entering the parameters in a command line to the MPI job, or setting them from the shell environment.

- The default configuration file for the FCA MPI runtime parameter is located at `$FCA_HOME/etc/fca_mpi_spec.ini`.
- The FCA parameters can be entered as command line parameters as part of the `mpirun` command.

➢ ***To pass FCA parameter from shell environment:***

- Enter the following command;

```
export fca_<ini_section_name>_<ini_section_param_name>=value
```

Example:

```
export fca_mpi_debug_level=5
```

or provide the parameter to OpenMPI as command line argument:

```
mpirun -x fca_mpi_debug_level=5 <... other mpirun parameters ...>
```

6.2 Specifying FCA Parameters in the INI File

At runtime, use `mpirun`'s '-x' command switch to *overwrite* FCA parameters set in the `fca_mpi_spec.ini` file.

Set the FCA parameters in an MPI job at runtime with the following syntax:

```
$MPI_HOME/bin/mpirun -np 32 -machinefile hostfile -mca
coll_oma_fca_library_path $FCA_HOME/lib/libfca.so -mca btl sm,self,openib -x
<param>=value <other mpi options> mpi_hello_world
```

where:

- **$MPI_HOME** represents the path to the MPI installation directory.
- **$FCA_HOME** represents the path to the FCA software directory.

NOTE: The -x command switch is used as follows: -x <param>=<value>

Example: `-x fca_mpi_collect_stats=y`

Table 5: FCA Parameters in Open MPI at Run Time

INI File Parameter	Description	Values
The following parameters may be changed in the INI file under section [mpi].		

INI File Parameter	Description	Values
fca_mpi_debug_level	Verbosity level for MPI FCA debugging. The debug levels are: • 0 (fatal) • 1 (error) • 2 (warn) • 3 (info) • 4 (debug) • 5-7 (detailed debug info)	integer between 0-7 The default is 2.
fca_mpi_log_file	FCA log filename. The logfile name can contain printf like tokens, which are substituted during log file creation: • %H - hostname of the process • %u - current time in ms. • %T - current thread ID • %s - time in sec. • %t - time in ticks.	String representing log file name or empty for none. The default is none.
fca_mpi_enable_stdout	Determines whether the FCA log should also be written to the standard output. Valid values: <y|n>	character The default is y.
fca_mpi_fp_sum_fixedpoint	Use fixed-point math when performing floating point summation, to keep a consistent result regardless of the order of operations. Valid values: <y|n>	character The default is n.
fca_mpi_collect_stat	Collect MPI application performance statistics. Valid values: <y|n>	character The default is n.
fca_mpi_stats_max_ops	Max number of different MPI collective operations for which to collect statistics This option is effective when "collect_stat = y".	positive integer The default is 1000.
fca_mpi_stats_file_name	File name in which to keep collected statistics (collect_stats must be enabled for this parameter to take effect).	string The default is: fca_stats.xml
The following parameters may be changed in the INI file under section [ib].		
fca_ib_dev_name	If set, the specified IB device will be used for communication. The name as appears in */sys/class/infiniband/ directory*. If not set, the first device with an ACTIVE port is used.	string representing active IB device name [default] Leave empty or commented, then auto-discovery will be used.

INI File Parameter	Description	Values
fca_ib_port_num	If set, the selected port number is used on the matching device. If not set or zero, the first active port is used.	positive integer The default is: unset (use auto-discover)

Example:

```
/opt/openmpi/1.6.3/bin/mpirun -np 32 -machinefile hostfile -mca btl
sm,self,openib -x fca_mpi_debug_level=4 mpi_hello_world
```

7 Configuring Rules for Offloading

The FCA system is provided with user-defined rules to select the most suitable offloading method for MPI communication. The used-defined rules consider the following MPI Communicator parameters:

- message size range (in bytes)
- communicator size (in ranks)
- offloading method (CD - CoreDirect, UD, MPI native)
- Operation (for Reduce/AllReduce)
- Data type (for Reduce/AllReduce)

7.1 Enabling Dynamic Rules Mechanism

➢ ***To enable dynamic rules mechanism:***

- Enter the following command in the `fca_mpi_spec.ini` file section called "rules";

```
enable = <0|1>
```

Example:

```
enable = 1
```

7.2 Configuring a Specific Rule

User-defined offloading rules are added and enumerated in the `fca_mpi_spec.ini` file. Every user-defined rule is represented by a new INI file section, named in the following format:

```
[rule-<coll_name>-<SN>]
```

- `coll_name` can be one of the following values:
  - reduce
  - allreduce
  - bcast
  - barrier
  - allgather
  - allgatherv
- `SN` is a rule serial number for given `coll_name`

The default value for min/max params is -1, "no limit".

Valid offload types are:

- ud - use FCA in UD mode
- cd (default) - use FCA in COREDirect mode
- none - do not use FCA

Rules are applied by the first match. If none of the rules match, the default is to use FCA with COREDirect mode.

The following is a list of valid rules parameters for FCA:

Parameter	Description	Default
msg_size_min = <int>	Minimum message size.	-1 - No limit
msg_size_max = <int>	Maximum message size.	-1 - No limit
comm_size_min = <int>	Minimum communicator size.	-1 - No limit
comm_size_max = <int>	Maximum communicator size.	-1 - No limit
offload_type = <string>	FCA offload type.	cd (COREDirect mode)
data_type = <string>	Data type given as a parameter.	Applicable for reduce/allreduce
reduce_op = <string>	Reduce operation type requested.	Applicable for reduce/allreduce

Examples of reduce rules

```
[rules]
enable         = 1

[rule-reduce-1]
msg_size_min   = 256
msg_size_max   = 1024
comm_size_min  = 30
comm_size_max  = 35
offload_type   = ud
data_type      = MPI_CHAR
reduce_op      = MPI_LXOR

[rule-reduce-2]
msg_size_min   = 1
msg_size_max   = 2024
comm_size_min  = 1
comm_size_max  = 10
offload_type   = none
```

8 OpenMPI MCA Parameters to Control FCA Offload

The complete list of OpenMPI FCA related parameters can be extracted using the `ompi_info` command.

- ***To extract the complete list of OpenMPI FCA related parameters:***

```
$MPI_HOME/bin/ompi_info --param coll fca
```

- ***To provide MCA parameters to the OpenMPI mpirun command, use the following format:***

```
$MPI_HOME/bin/mpirun -mca <param> <value>
```

Example:

```
$MPI_HOME/bin/mpirun -mca coll_fca_verbose 1 <... other mpirun args ...>
```

The following is a list of MCA parameters for FCA:

Parameter	Description	Default
coll_fca_priority = <int>	Priority of the fca coll component	80
coll_fca_verbose = <int>	Verbose level of the fca coll component	0
coll_fca_enable = <0|1>	Enable/Disable Fabric Collective Accelerator	1
coll_fca_spec_file = <string>	Path to the FCA configuration file fca_mpi_spec.ini	$FCA_HOME/etc/fca_mpi_spec.ini
coll_fca_library_path = <string>	Path to FCA runtime library	$FCA_HOME/lib/libfca.so
coll_fca_np = <int>	Minimal allowed job's NP to activate FCA	64
coll_fca_enable_barrier = <0|1>	Enable/Disable FCA Barrier support	1
coll_fca_enable_bcast = <0|1>	Enable/Disable FCA Bcast support	1
coll_fca_enable_reduce = <0|1>	Enable/Disable FCA Reduce support	1
coll_fca_enable_allreduce = <0|1>	Enable/Disable FCA Allreduce support	1
coll_fca_enable_allgather = <0|1>	Enable/Disable FCA Allgather support	1
coll_fca_enable_allgatherv = <0|1>	Enable/Disable FCA Allgatherv support	1