How to use bullxMPI compiled with MXM and FCA tools

Author: Cyril Mazauric

Version: A1
Date: 29 May 2015

Table of Contents

1 Introduction

This document will summarize how to use the bullxMPI compiled with mellanox tools, MXM and FCA

2 How to source the environment

First of all, you should source your compiler. For example with Intel compiler

```
source /opt/intel/composer_xe_2015.2.164/bin/compilervars.sh intel64
```

To use MXM you have to load MXM environment with

```
export LD_PRELOAD=/opt/mellanox/mxm-3.3.3002/lib/libmxm.so
export HPCX_MXM_DIR=/opt/mellanox/mxm-3.3.3002
export MXM_HOME=$HPCX_MXM_DIR
export PATH=HPCX_MXM_DIR/bin:$PATH
export LD_LIBRARY_PATH=HPCX_MXM_DIR/lib:$LD_LIBRARY_PATH
```

or

```
module load mxm/3.3.3002
```

To use FCA, you have to load FCA environment with

```
module load fca/2.5.2379
```

then, you must load the bullxMPI_mlx, with

```
module load bullxmpi_mlx/bullxmpi_mlx-1.2.8.3
```

Please, you must respect the order: compiler, MXM/FCA and bullxMPI.

Remark: If you do not load MXM/FCA environment, you will use the bullxMPI **without** MXM and FCA tools

3 MXM

MXM (Mellanox Messaging), accelerates the underlying send/receive (or put/get) messages

To use MXM, we must specify the following variables:

```
export OMPI_MCA_pml=cm

export OMPI_MCA_mtl=mxm

export MXM_RDMA_PORTS=mlx5_0:1
```

To know all available variables to tune MXM, you can use the following command:

```
mxm_dump_config -t
```

*Generating MXM Environment Parameters*Variable	Valid Values	Description
MXM_LOG_LEVEL	•FATAL •ERROR •INFO •DEBUG •TRACE •REQ •DATA •ASYNC •FUNC •POLL •WARN (default)	MXM logging level. Messages with a level higher or equal to the selected will be printed.
MXM_LOG_FILE	String	If not empty, MXM will print log messages to the specified file instead of stdout. The following substitutions are performed on this string: •%p - Replaced with process ID •%h - Replaced with host name Value: String.
MXM_LOG_BUFFER	1024	Buffer size for a single log message. Value: memory units: <number>[b|kb|mb|gb]
MXM_LOG_DATA_SIZE	0	How much of the packet payload to print, at most, in data mode. Value: unsigned long.

MXM_HANDLE_ERRORS	•None: No error handling •Freeze: Freeze and wait for a debugger •Debug: attach debugger •bt: print backtrace	Error handling mode.
MXM_ERROR_SIGNALS	•ILL •SEGV •BUS •FPE •PIPE	Signals which are considered an error indication and trigger error handling. Value: comma-separated list of: system signal (number or SIGxxx)
MXM_GDB_COMMAND	gdb	If non-empty, attaches a gdb to the process in case of error, using the provided command. Value: string
MXM_DEBUG_SIGNO	HUP	Signal number which causes MXM to enter debug mode. Set to 0 to disable. Value: system signal (number or SIGxxx)
MXM_ASYNC_INTERVAL	50000.00us	Interval of asynchronous progress. Lower values may make the network more responsive, at the cost of higher CPU load. Value: time value: <number>[s|us|ms|ns]
MXM_ASYNC_SIGNO	ALRM	Signal number used for async signaling. Value: system signal (number or SIGxxx)
MXM_STATS_DEST	•udp:<host>[:<port>] - send over UDP to the given host:port. •stdout: print to standard output. •stderr: print to standard error. •file:<file-name>[:bin] - save to a file (%h: host, %p: pid, %c: cpu, %t: time, %e: exe)	Destination to send statistics to. If the value is empty, statistics are not reported.
MXM_STATS_TRIGGER	•timer:<interval>: dump in specified intervals. •exit: dump just before program exits (default)	Trigger to dump statistics Value: string

MXM_MEMTRACK_DEST	•file:<filename>: save to a file (%h: host, %p: pid, %c: cpu, %t: time, %e: exe) •stdout: print to standard output •stderr: print to standard error	Memory tracking report output destination. If the value is empty, results are not reported.
MXM_INSTRUMENT	•%h: host •%p: pid •%c: cpu •%t: time •%e: exe.	File name to dump instrumentation records to. Value: string
MXM_INSTRUMENT_SIZE	1048576	Maximal size of instrumentation data. New records will replace old records. Value: memory units: <number>[b|kb|mb|gb]
MXM_PERF_STALL_LOOPS	0	Number of performance stall loops to be performed. Can be used to normalize profile measurements to packet rate Value: unsigned long
MXM_ASYNC_MODE	•Signal •none •Thread (default)	Asynchronous progress method Value: [none|signal|thread]
MXM_MEM_ALLOC	•cpages: Contiguous pages, provided by Mellanox-OFED. •hugetlb - Use System V shared memory API for getting pre-allocated huge pages. •mmap: Use private anonymous mmap() to get free pages. •libc: Use libc's memory allocation primitives. •sysv: Use system V's memory allocation.	Memory allocators priority. Value: comma-separated list of: [libc|hugetlb|cpages|mmap|sysv]

MXM_MEM_ON_DEMAND_MAP	•n: disable •y: enable	Enable on-demand memory mapping. USE WITH CARE! It requires calling mxm_mem_unmap() when any buffer used for communication is unmapped, otherwise data corruption could occur. Value: <y|n>
MXM_INIT_HOOK_SCRIPT	-	Path to the script to be executed at the very beginning of MXM initialization Value: string
MXM_SINGLE_THREAD	•y - single thread •n - not single thread	Mode of the thread usage. Value: <y|n>
MXM_SHM_KCOPY_MODE	•off: Don't use any kernel copy mode. •knem: Try to use knem. If it fails, default to 'off'. •autodetect: If knem is available, first try to use knem. If it fails, default to 'off' (default)	Modes for using to kernel copy for large messages.
MXM_IB_PORTS	*:*	Specifies which Infiniband ports to use. Value: comma-separated list of: IB port: <device>:<port_num>
MXM_EP_NAME	%h:%p	Endpoint options. Endpoint name used in log messages. Value: string
MXM_TLS	•self •shm •ud	Comma-separated list of transports to use. The order is not significant. Value: comma-separated list of: [self|shm|rc|dc|ud|oob]
MXM_ZCOPY_THRESH	2040	Threshold for using zero copy. Value: memory units: <number>[b|kb|mb|gb]
MXM_IB_CQ_MODERATION	64	Number of send WREs for which a CQE is generated. Value: unsigned
MXM_IB_CQ_WATERMARK	127	Consider ep congested if poll cq returns more than n wqes. Value: unsigned
MXM_IB_DRAIN_CQ	•n •y	Poll CQ till it is completely drained of completed work requests. Enabling this feature may cause starvation of other endpoints Value: <y|n>

MXM_IB_RESIZE_CQ	•n •y	Allow using resize_cq(). Value: <y|n>
MXM_IB_TX_BATCH	16	Number of send WREs to batch in one post-send list. Larger values reduce the CPU usage, but increase the latency because we might need to process lots of send completions at once. Value: unsigned
MXM_IB_RX_BATCH	64	Number of post-receives to be performed in a single batch. Value: unsigned
MXM_IB_MAP_MODE	•first: Map the first suitable HCA port to all processes (default). •affinity: Distribute evenly among processes based on CPU affinity. •nearest: Try finding nearest HCA port based on CPU affinity. •round-robin	HCA ports to processes mapping method. Ports not supporting process requirements (e.g. DC support) will be skipped. Selecting a specific device will override this setting.
MXM_IB_NUM_SLS	1: (default)	Number of InfiniBand Service Levels to use. Every InfiniBand endpoint will generate a random SL within the given range FIRST_SL..(FIRST_SL+NUM_SLS-1), and use it for outbound communication. applicable values are 1 through 16. Value: unsigned
MXM_IB_WC_MODE	•wqe: Use write combining to post full WQEs (default). •db: Use write combining to post doorbell. •flush: Force flushing CPU write combining buffers.wqe •flush (default)	Write combining mode flags for InfiniBand devices. Using write combining for 'wqe' improves latency performance due to one less wqe fetch. Avoiding 'flush' relaxes CPU store ordering, and reduces overhead. Write combining for 'db' is meaningful only when used without 'flush'. Value: comma-separated list of: [wqe|db|flush]

MXM_IB_LID_PATH_BITS	0: (default) 0 <= value< 2^(LMC) - 1	InfiniBand path in bits which will be the low portion of the LID, according to the LMC in the fabric. Value: unsigned
MXM_IB_FIRST_SL	0-15	The first Infiniband Service Level number to use. Value: unsigned
MXM_IB_CQ_STALL	100	CQ stall loops for SandyBridge far socket. Value: unsigned
MXM_UD_ACK_TIMEOUT	300000.00us	Timeout for getting an acknowledgment for sent packet. Value: time value: <number>[s|us|ms|ns]
MXM_UD_FAST_ACK_TIMEOUT	1024.00us	Timeout for getting an acknowledgment for sent packet. Value: time value: <number>[s|us|ms|ns]
MXM_FAST_TIMER_RESOLUTION	64.00us	Resolution of ud fast timer.The value is treated as a recommendation only. Real resolution may differ as mxm rounds up to power of two. Value: time value: <number>[s|us|ms|ns]
MXM_UD_INT_MODE	rx	Traffic types to enable interrupt for. Value: comma-separated list of: [rx|tx]
MXM_UD_INT_THRESH	20000.00us	The maximum amount of time that may pass following an mxm call, after which interrupts will be enabled. Value: time value: <number>[s|us|ms|ns]
MXM_UD_WINDOW_SIZE	1024	The maximum number of unacknowledged packets that may be in transit. Value: unsigned
MXM_UD_MTU	65536	Maximal UD packet size. The actual MTU is the minimum of this value and the fabric MTU. Value: memory units: <number>[b|kb|mb|gb]
MXM_UD_CA_ALGO	•none: no congestion avoidance •bic: binary increase (default)	Use congestion avoidance algorithm to dynamically adjust send window size. Value: [none|bic]

MXM_UD_CA_LOW_WIN	0	Use additive increase multiplicative decrease congestion avoidance when current window is below this threshold. Value: unsigned
MXM_UD_RX_QUEUE_LEN	4096	Length of receive queue for UD QPs. Value: unsigned
MXM_UD_RX_MAX_BUFS	-1	Maximal number of receive buffers for one endpoint. -1 is infinite. Value: integer
MXM_UD_RX_MAX_INLINE	0	Maximal size of data to receive as inline. Value: memory units: <number>[b|kb|mb|gb]
MXM_UD_RX_DROP_RATE	0	If nonzero, network packet loss will be simulated by randomly ignoring one of every X received UD packets. Value: unsigned
MXM_UD_RX_OOO	•n •y	If enabled, keep packets received out of order instead of discarding them. Must be enabled if network allows out of order packet delivery, for example, if Adaptive Routing is enabled Value: <y|n>
MXM_UD_TX_QUEUE_LEN	128	Length of send queue for UD QPs. Value: unsigned
MXM_UD_TX_MAX_BUFS	32768	Maximal number of send buffers for one endpoint. -1 is infinite. Value: integer
MXM_UD_TX_MAX_INLINE	128	Bytes to reserve in TX WQE for inline data. Messages which are small enough will be sent inline. Value: memory units: <number>[b|kb|mb|gb]
MXM_CIB_PATH_MTU	default	Path MTU for CIB QPs. Possible values are: default, 512, 1024, 2048, 4096. Setting “default” will select the best MTU for the device. Value: [default|512|1024|2048|4096]
MXM_CIB_HARD_ZCOPY_THRESH	16384	Threshold for using zero copy. Value: memory units: <number>[b|kb|mb|gb]

MXM_CIB_MIN_RNR_TIMER	25	InfiniBand minimum receiver not ready timer, in seconds (must be >= 0 and <= 31)
MXM_CIB_TIMEOUT	20	InfiniBand transmit timeout, plugged into formula: 4.096 microseconds * (2 ^ timeout) (must be >= 0 and <= 31) Value: unsigned
MXM_CIB_MAX_RDMA_DST_OPS	4	InfiniBand maximum pending RDMA destination operations (must be >= 0) Value: unsigned
MXM_CIB_RNR_RETRY	7	InfiniBand “receiver not ready” retry count, applies ONLY for SRQ/XRC queues. (must be >= 0 and <= 7: 7 = “infinite”) Value: unsigned
MXM_UD_RNDV_THRESH	262144	UD threshold for using rendezvous protocol. Smaller value may harm performance but excessively large value can cause a deadlock in the application. Value: memory units: <number>[b|kb|mb|gb]
MXM_UD_TX_NUM_SGE	3	Number of SG entries in the UD send QP. Value: unsigned
MXM_UD_HARD_ZCOPY_THRESH	65536	Threshold for using zero copy. Value: memory units: <number>[b|kb|mb|gb]
MXM_CIB_RETRY_COUNT	7	InfiniBand transmit retry count (must be >= 0 and <= 7) Value: unsigned
MXM_CIB_MSS	4224	Size of the send buffer. Value: memory units: <number>[b|kb|mb|gb]
MXM_CIB_TX_QUEUE_LEN	256	Length of send queue for RC QPs. Value: unsigned
MXM_CIB_TX_MAX_BUFS	-1	Maximal number of send buffers for one endpoint. -1 is infinite. Warning: Setting this param with value != -1 is a dangerous thing in RC and could cause deadlock or performance degradation Value: integer
MXM_CIB_TX_CQ_SIZE	16384	Send CQ length. Value: unsigned

MXM_CIB_TX_MAX_INLINE	128	Bytes to reserver in TX WQE for inline data. Messages which are small enough will be sent inline. Value: memory units: <number>[b|kb|mb|gb]
MXM_CIB_TX_NUM_SGE	3	Number of SG entries in the RC QP. Value: unsigned
MXM_CIB_USE_EAGER_RDMA	y	Use RDMA WRITE for small messages. Value: <y|n>
MXM_CIB_EAGER_RDMA_THRESHOLD	16	Use RDMA for short messages after this number of messages are received from a given peer, must be >= 1 Value: unsigned long
MXM_CIB_MAX_RDMA_CHANNELS	8	Maximum number of peers allowed to use RDMA for short messages, must be >= 0 Value: unsigned
MXM_CIB_EAGER_RDMA_BUFFS_NUM	32	Number of RDMA buffers to allocate per rdma channel, must be >= 1 Value: unsigned
MXM_CIB_EAGER_RDMA_BUFF_LEN	4224	Maximum size (in bytes) of eager RDMA messages, must be >= 1 Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_RX_MAX_BUFFERS	-1	Maximal number of receive buffers for endpoint. -1 is infinite. Value: integer
MXM_SHM_RX_MAX_MEDIUM_BUFFERS	-1	Maximal number of medium sized receive buffers for one endpoint. -1 is infinite. Value: integer
MXM_SHM_FIFO_SIZE	64	Size of the shmem tl's fifo. Value: unsigned
MXM_SHM_WRITE_RETRY_COUNT	64	Number of retries in case where cannot write to the remote process. Value: unsigned
MXM_SHM_READ_RETRY_COUNT	64	Number of retries in case where cannot read from the shmem FIFO (for multi-thread support). Value: unsigned

MXM_SHM_HUGETLB_MODE	•y: Allocate memory using huge pages only. •n: Allocate memory using regular pages only. •try: Try to allocate memory using huge pages and if it fails, allocate regular pages (default).	Enable using huge pages for internal shared memory buffers Values: <yes|no|try>
MXM_SHM_RX_BUFF_LEN	8192	Maximum size (in bytes) of medium sized messages, must be >= 1 Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_HARD_ZCOPY_THRESH	2048	Threshold for using zero copy. Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_RNDV_THRESH	65536	SHM threshold for using rendezvous protocol. Smaller value may harm performance but too large value can cause a deadlock in the application. Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_KNEM_MAX_SIMULTANEOUS	0	Maximum number of simultaneous ongoing knem operations to support in shmem tl. Value: unsigned
MXM_SHM_KNEM_DMA_CHUNK_SIZE	67108864	Size of a single chunk to be transferred in one dma operation. Larger values may not work since they are not supported by the dma engine Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_RELEASE_FIFO_FACTOR	0.500	Frequency of resource releasing on the receiver's side in shmem tl. This value refers to the percentage of the fifo size. (must be >= 0 and < 1) Value: floating point number
MXM_TM_UPDATE_THRESHOLD_MASK	8	Update bit-mask length for connections traffic counters.(must be >= 0 and <= 32: 0 - always update, 32 - never update.)# Value: bit count

MXM_TM_PROMOTE_THRESHOLD	5	Relative threshold (percentage) of traffic for promoting connections, Must be >= 0. Value: unsigned
MXM_TM_PROMOTE_BACKOFF	1	Exponential backoff degree (bits) for all counters upon a promotion, Must be >= 0 and <=32. Value: unsigned
MXM_RC_QP_LIMIT	64	Maximal amount of RC QPs allowed (Negative for unlimited). Value: integer
MXM_DC_QP_LIMIT	64	Maximal amount of DC QPs allowed (Negative for unlimited). Value: integer
MXM_DC_RECV_INLINE	•128 •512 •1024 •2048 •4096	Bytes to reserve in CQE for inline data. In order to allow for inline data, -x MLX5_CQE_SIZE=128 must also be specified. Value: memory units: <number>[b|kb|mb|gb]
MXM_CIB_RNDV_THRESH	16384	CIB threshold for using rendezvous protocol. Smaller value may harm performance but excessively large value can cause a deadlock in the application. Value: memory units: <number>[b|kb|mb|gb]
MXM_SHM_USE_KNEM_DMA	•n •y	Whether or not to offload to the DMA engine when using KNEM Value: <y|n>

4 FCA

FCA (Fabric Collectives Accelerations) accelerates the underlying collective operations used by the MPI/PGAS languages.

To use FCA, we must specify the following variables:

```
export OMPI_MCA_coll=^ghc
export OMPI_MCA_coll_fca_priority=95
export OMPI_MCA_coll_fca_enable=1
```

To know all available FCA variables, you can use the following command:

```
ompi_info -all
```

For example, you can use the following variables to enable/disable FCA for some mpi functions:

```
export OMPI_MCA_coll_fca_enable_barrier=0
export OMPI_MCA_coll_fca_enable_bcast=0
export OMPI_MCA_coll_fca_enable_reduce=0
export OMPI_MCA_coll_fca_enable_reduce_scatter=0
export OMPI_MCA_coll_fca_enable_allreduce=0
export OMPI_MCA_coll_fca_enable_allgather=0
export OMPI_MCA_coll_fca_enable_allgatherv=0
export OMPI_MCA_coll_fca_enable_gather=0
export OMPI_MCA_coll_fca_enable_gatherv=0
```

5 A default environment

From ATOS point of view, a good environment will be to use bullxMPI_mlx with MXM.

Consequently, by default, you can use the following setting:

Module load compiler intel

Module load mxm/<version>

Module load fca/<version>

Module load bullxmpi_mlx/<version>

export OMPI_MCA_pml=cm

export OMPI_MCA_mtl=mxm

export OMPI_MCA_coll=^ghc

export MXM_RDMA_PORTS=mlx5_0:1

With this setting, you will use bullxMPI and MXM. FCA is not used by default but could be used thanks to variables defined in the FCA section.

6 An example with IMB

The following script shows how to submit an IMB barrier with mxm and fca on 500 nodes (HSW E5-2680 v3).

```
#SBATCH --nodes=500
#SBATCH --ntasks=12000
#SBATCH --ntasks-per-node=24
#SBATCH --threads-per-core=1
#SBATCH -J IMB
#SBATCH -p compute
#SBATCH --time=00:30:00
#SBATCH --exclusive

#LOAD ENV
module load intel/2015.2.164
module load mxm/3.3.3002
module load fca/2.5.2379
module load bullxmpi_mlx/bullxmpi_mlx-1.2.8.3
# TO USE MXM
```

```
export OMPI_MCA_pml=cm
export OMPI_MCA_mtl=mxm
export MXM_RDMA_PORTS=mlx5_0:1
#TO USE FCA
export OMPI_MCA_coll=^ghc
export OMPI_MCA_coll_fca_priority=95
export OMPI_MCA_coll_fca_enable=1

#SLURM SETTNG
export OMPI_MCA_ess=^pmi
export OMPI_MCA_pubsub=^pmi

#EXEC WITH SRUN
time srun  -K1 --resv-ports  --cpu_bind=map_cpu:0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
--cpu-freq=2500000  -n $SLURM_NTASKS   ./IMB-MPI1  -off_cache 30,64  -npmin $SLURM_NTASKS  barrier
```

The following graphics shows the performance gain due to FCA module for Allreduce and Barrier operations: