MPI One-Sided Communication API extensions test cases

Piotr Dorożyński and Paweł Czarnul

April 13, 2015

Abstract

This document describes test cases for implemented MPI One-Sided Communication API extensions.

	Project	**Optimized MPI API for persistent memory**
1	Author	Paweł Czarnul
	Version	0.1
	Modification date	December 6, 2014
	Description	Template
2	Author	Piotr Dorożyński
	Version	0.2
	Modification date	April 12, 2015
	Description	Initial description.
3	Author	Paweł Czarnul
	Version	0.3
	Modification date	April 13, 2015
	Description	Corrections

1 Unit tests

1.1 Helper functions

1.1.1 `parse_mpi_info_bool`

Name	Parse `MPI_Info` boolean value set to true
Description	Parse `MPI_Info` with some boolean key set to `true`.
Expected result	Result should be `true`.

Name	Parse `MPI_Info` boolean value set to false
Description	Parse `MPI_Info` with some boolean key set to `false`.
Expected result	Result should be `false`.

Name	Parse `MPI_Info` boolean value not set
Description	Parse `MPI_Info` with some boolean key not set.
Expected result	Result should be `false`.

Name	Parse `MPI_Info` boolean value set to a wrong value
Description	Parse `MPI_Info` with some boolean key set to a wrong value i.e. not `true` or `false`.
Expected result	Result should be `false`.

1.1.2 `parse_mpi_info_name`

Name	Parse `MPI_Info` with a value for a key `pmem_name` set to a proper name
Description	Parse `MPI_Info` with a value for a key `pmem_name` set to a proper name.
Expected result	Result should be the same as the value provided.

Name	Parse `MPI_Info` with a value for a key `pmem_name` set to too long a name
Description	Parse `MPI_Info` with a value for a key `pmem_name` set to too long a name.
Expected result	Error `MPI_ERR_PMEM_NAME`

Name	Parse `MPI_Info` with a value for a key `pmem_name` not set
Description	Parse `MPI_Info` with a value for a key `pmem_name` not set.
Expected result	Error `MPI_ERR_PMEM_NAME`

1.1.3 `parse_mpi_info_mode`

Name	Parse `MPI_Info` with a value for a key `pmem_mode` set to `expand`
Description	Parse `MPI_Info` with a value for a key `pmem_mode` set to `expand`.
Expected result	Result should be `MPI_PMEM_MODE_EXPAND`

Name	Parse `MPI_Info` with a value for a key `pmem_mode` set to `checkpoint`
Description	Parse `MPI_Info` with a value for a key `pmem_mode` set to `checkpoint`.
Expected result	Result should be `MPI_PMEM_MODE_CHECKPOINT`

Name	Parse `MPI_Info` with a value for a key `pmem_mode` set to a wrong value
Description	Parse `MPI_Info` with a value for a key `pmem_mode` set to a wrong value.
Expected result	Error `MPI_ERR_PMEM_MODE`

Name	Parse `MPI_Info` with a value for a key `pmem_mode` not set
Description	Parse `MPI_Info` with a value for a key `pmem_mode` not set.
Expected result	Error `MPI_ERR_PMEM_MODE`

1.1.4 `parse_mpi_info_checkpoint_version`

Name	Parse `MPI_Info` with a value for a key `pmem_checkpoint_version` set to a proper value
Description	Parse `MPI_Info` with a value for a key `pmem_checkpoint_version` set to a proper value.
Expected result	Result should be the same as provided value.

Name	Parse `MPI_Info` with a value for a key `pmem_checkpoint_version` not set
Description	Parse `MPI_Info` with a value for a key `pmem_checkpoint_version` not set.
Expected result	Result should be -1.

1.1.5 `check_if_window_exists_and_its_size`

Name	Check if the window exists for the first existing window
Description	Create 3 windows, delete all windows except the first one and check if the first one exists.
Expected result	Result should be `true`.

Name	Check if the window exists for the middle existing window
Description	Create 3 windows, delete all windows except the second one and check if the second one exists.
Expected result	Result should be `true`.

Name	Check if the window exists for the last existing window
Description	Create 3 windows, delete all windows except the third one and check if the third one exists.
Expected result	Result should be `true`.

Name	Check if the window exists for the non existing window that is first in global metadata file
Description	Create 3 windows, delete the first window and check if the first window exists.
Expected result	Result should be `false`.

Name	Check if the window exists for the non existing window that is in the middle in global metadata file
Description	Create 3 windows, delete the second window and check if the second window exists.
Expected result	Result should be `false`.

Name	Check if the window exists for the non existing window that is the last one in global metadata file
Description	Create 3 windows, delete the third window and check if the third window exists.
Expected result	Result should be `false`.

Name	Check if the window exists for the non existing window that never existed
Description	Create 3 windows and check if some other window exists.
Expected result	Result should be `false`.

Name	Check window size for a window with proper size
Description	Create a window and check if this window exists and has a proper size (in a mode set to `checkpoint`).
Expected result	Result should be `true`.

Name	Check window size for a window with improper size
Description	Create a window and check if this window exists and has a proper size (in a mode set to `checkpoint`), but specifying an improper size.
Expected result	Error `MPI_ERR_SIZE`.

Name	Check window size for a non existing window with improper size
Description	Create a window, delete this window and check if this window exists and has a proper size (in a mode set to `checkpoint`), but specifying an improper size.
Expected result	Result should be `false`.

1.1.6 `create_window_metadata_file`

Name	Create metadata file of a window that never existed before
Description	Create 3 windows and check metadata after each creation.
Expected result	Each window should have an associated empty metadata file and there should be a record for this file in the global metadata file.

Name	Create metadata file of a window that existed before that is the first in global metadata file
Description	Create 3 windows, delete first window, recreate the window and check metadata.
Expected result	Window should have an associated empty metadata file and there should be a record for this file in global metadata file.

Name	Create metadata file of a window that existed before that is in the middle in global metadata file
Description	Create 3 windows, delete second window, recreate the window and check metadata.
Expected result	Window should have an associated empty metadata file and there should be a record for this file in global metadata file.

Name	Create metadata file of a window that existed before that is the last in global metadata file
Description	Create 3 windows, delete third window, recreate the window and check metadata.
Expected result	Window should have an associated empty metadata file and there should be a record for this file in global metadata file.

1.1.7 update_window_size_in_metadata_file

Name	Update window size of an existing window that is the first window in global metadata file
Description	Create 3 windows, change the size of the first window and check metadata.
Expected result	The window size should be updated to a new window size.

Name	Update window size of an existing window that is the middle window in global metadata file
Description	Create 3 windows, change the size of the second window and check metadata.
Expected result	The window size should be updated to a new window size.

Name	Update window size of an existing window that is the last window in global metadata file
Description	Create 3 windows, change the size of the last window and check metadata.
Expected result	The window size should be updated to a new window size.

Name	Update window size of a window that never existed
Description	Try to change window size of a window that never existed.
Expected result	Error `MPI_ERR_PMEM_NAME`

Name	Update window size of a deleted window
Description	Create a window, delete it and try to change its size.
Expected result	Error `MPI_ERR_PMEM_NAME`

1.1.8 set_checkpoint_versions

Name	Set checkpoint versions in the `expand` mode
Description	Set checkpoint versions with mode set to `expand`.
Expected result	`next_checkpoint_version` $== 0$, `last_checkpoint_version` $== -1$, `highest_checkpoint_version` $== -1$

Name	Set checkpoint versions in `checkpoint` mode with search of the last checkpoint
Description	Create a window and synchronize 3 times in order to make 3 checkpoints. Set checkpoint versions with `last_checkpoint_version` set to -1 in order to search for the last checkpoint.
Expected result	`next_checkpoint_version` $== 3$, `last_checkpoint_version` $== 2$, `highest_checkpoint_version` $== 2$

Name	Set checkpoint versions in `checkpoint` mode with search of the last checkpoint and the last checkpoint version is not the highest
Description	Create a window, synchronize 3 times to make 3 checkpoints and delete the first and the last checkpoint. Set checkpoint versions with `last_checkpoint_version` set to -1 in order to search for last checkpoint.
Expected result	`next_checkpoint_version` $== 2$, `last_checkpoint_version` $== 1$, `highest_checkpoint_version` $== 2$

Name	Set checkpoint versions in `checkpoint` mode with search of the last checkpoint on all processes
Description	Create window and synchronize 3 times to make 3 checkpoints, delete the last checkpoint version on one of the processes. Set checkpoint versions with `last_checkpoint_version` set to -1 in order to search for last checkpoint.
Expected result	`next_checkpoint_version` $== 2$, `last_checkpoint_version` $== 1$, `highest_checkpoint_version` $== 2$

Name	Set checkpoint versions in `checkpoint` mode with search of the last checkpoint on all processes with one process not having necessary checkpoint
Description	Create a window and synchronize 3 times to make 3 checkpoints, delete the last checkpoint version on one of the processes and second version on another process. Set checkpoint versions with `last_checkpoint_version` set to -1 in order to search for the last checkpoint.
Expected result	Error `MPI_ERR_PMEM`

Name	Set checkpoint versions in `checkpoint` mode with a proper version specified
Description	Create a window and synchronize 3 times to make 3 checkpoints. Set checkpoint versions with `last_checkpoint_version` set to 1.
Expected result	`next_checkpoint_version` == 2, `last_checkpoint_version` == 1, `highest_checkpoint_version` == 2

Name	Set checkpoint versions in `checkpoint` mode with proper version specified in append mode
Description	Create a window and synchronize 3 times to make 3 checkpoints. Set checkpoint versions with `last_checkpoint_version` set to 1 and `append_checkpoints` set to `true`.
Expected result	`next_checkpoint_version` == 3, `last_checkpoint_version` == 1, `highest_checkpoint_version` == 2

Name	Set checkpoint versions in `checkpoint` mode with negative version specified
Description	Create a window and synchronize 3 times to make 3 checkpoints. Set checkpoint versions with `last_checkpoint_version` set to -2.
Expected result	Error `MPI_ERR_PMEM_CKPT_VER`

Name	Set checkpoint versions in `checkpoint` mode with too high a version specified
Description	Create a window and synchronize 3 times to make 3 checkpoints. Set checkpoint versions with `last_checkpoint_version` set to 3.
Expected result	Error `MPI_ERR_PMEM_CKPT_VER`

Name	Set checkpoint versions in `checkpoint` mode with a deleted version specified
Description	Create a window and synchronize 3 times to make 3 checkpoints and delete the second checkpoint. Then set checkpoint versions with `last_checkpoint_version` set to 1.
Expected result	Error `MPI_ERR_PMEM_CKPT_VER`

1.1.9 `copy_data_from_checkpoint`

Name	Copy data from existing checkpoint
Description	Create a window and synchronize 3 times to make 3 checkpoints. Copy data from the second checkpoint.
Expected result	Data copied to `destination` should be identical to data used to create the second checkpoint.

Name	Copy data from not existing checkpoint
Description	Create a window and try to copy data from the second checkpoint (which does not exist).
Expected result	Error `MPI_ERR_PMEM`

Name	Copy data from deleted checkpoint
Description	Create a window and synchronize 3 times to make 3 checkpoints, delete the second checkpoint and try to copy data from the second checkpoint.
Expected result	Error `MPI_ERR_PMEM`

1.1.10 `delete_old_checkpoints`

Name	Delete all checkpoints from a window with all previous checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints and delete all old checkpoints.
Expected result	All old checkpoint files should not exist and metadata file should be set to an empty file.

Name	Delete all checkpoints from window without first checkpoint
Description	Create window, synchronize 3 times to create 3 checkpoints, delete the first checkpoint version and delete all old checkpoints.
Expected result	All old checkpoint files should not exist and metadata file should be set to an empty file.

Name	Delete all checkpoints from window without middle checkpoint
Description	Create window, synchronize 3 times to create 3 checkpoints, delete the second checkpoint version and delete all old checkpoints.
Expected result	All old checkpoint files should not exist and metadata file should be set to an empty file.

Name	Delete all checkpoints from window without last checkpoint
Description	Create a window, synchronize 3 times to create 3 checkpoints, delete the last checkpoint version and delete all old checkpoints.
Expected result	All old checkpoint files should not exist and metadata file should be set to an empty file.

1.2 Initialization functions

1.2.1 `MPI_Win_create_pmem`

Name	Create a window with `pmem_is_pmem` set to `true`
Description	Create a window with `pmem_is_pmem` set to `true`.
Expected result	The window should be created and metadata should be set properly.

Name	Create a window with `MPI_Info` set to an empty object.
Description	Create a window with `MPI_Info` set to an empty object.
Expected result	The window should be created and metadata should be set properly.

Name	Create a window with `MPI_Info` set to `MPI_INFO_NULL`
Description	Create a window with `MPI_Info` set to `MPI_INFO_NULL`.
Expected result	The window should be created and metadata should be set properly.

1.2.2 `MPI_Win_allocate_pmem`

Name	Expand mode with a non existing window
Description	Try to allocate a window in the expand mode.
Expected result	The window should be created and metadata should be set properly.

Name	Checkpoint mode with a non existing window
Description	Try to allocate a window in checkpoint mode when a window with such name does not exist.
Expected result	Error `MPI_ERR_PMEM_NAME`.

Name	Expand mode with an existing window
Description	Create a window, synchronize 3 times to create 3 checkpoints and reopen the window in the expand mode.
Expected result	All checkpoints should be deleted after reopening the window and all metadata should be set appropriately.

Name	Checkpoint mode with an existing window
Description	Create a window, synchronize 3 times to create 3 checkpoints and reopen the window in checkpoint mode loading the last window version.
Expected result	After reopening the window contents should be the same as contents of the last checkpoint and all metadata should be set appropriately.

1.2.3 `MPI_Win_create_dynamic_pmem`

Name	Create a window with `pmem_is_pmem` set to `true`
Description	Create a window with `pmem_is_pmem` set to `true`.
Expected result	The window should be created and metadata should be set properly.

Name	Create window with `MPI_Info` set to empty object
Description	Create a window with `MPI_Info` set to empty object.
Expected result	Window should be created and metadata should be set properly.

Name	Create a window with `MPI_Info` set to `MPI_INFO_NULL`
Description	Create a window with `MPI_Info` set to `MPI_INFO_NULL`.
Expected result	Window should be created and metadata should be set properly.

1.2.4 `MPI_Win_attach_pmem`

Name	Attach memory areas to window
Description	Attach 3 memory areas to window.
Expected result	Memory areas list should contain 3 memory areas.

1.2.5 `MPI_Win_detach_pmem`

Name	Detach first memory area
Description	Attach 3 memory areas to window and detach the first one.
Expected result	Memory areas list should contain 2 non detached memory areas.

Name	Detach middle memory area
Description	Attach 3 memory areas to window and detach the second one.
Expected result	Memory areas list should contain 2 non detached memory areas.

Name	Detach the last memory area
Description	Attach 3 memory areas to window and detach the third one.
Expected result	Memory areas list should contain 2 non detached memory areas.

1.2.6 `MPI_Win_set_info_pmem`

Name	Set new `MPI_Info` object containing negation of all possible fields specified during creation using `MPI_Win_create_pmem`
Description	Set new `MPI_Info` object containing negation of all possible fields specified during creation using `MPI_Win_create_pmem`.
Expected result	All window metadata should not change.

Name	Set new `MPI_Info` object containing negation of all possible fields specified during creation using `MPI_Win_allocate_pmem`
Description	Set new `MPI_Info` object containing negation of all possible fields specified during creation using `MPI_Win_allocate_pmem`.
Expected result	`pmem_dont_use_transactions` and `pmem_keep_all_checkpoints` have changed in window metadata. Other keys are ignored.

1.3 Synchronization functions

Since all synchronization functions works in almost the same way i.e. they call the same helper functions, they all may be tested by testing helper function `create_checkpoint`. Test cases for this function are described below.

Name	Creation of checkpoints with consecutive numbers with keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints, but each time with different data.
Expected result	Checkpoints with consecutive numbers are created and saved checkpoint data are the same as the source data used for creating checkpoint. `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

Name	Creation of checkpoints with consecutive numbers overwriting old checkpoints with keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints, but each time with different data. Then open window again in checkpoint mode and load from the second checkpoint version. Create 3 more checkpoints.
Expected result	Checkpoints with consecutive numbers are created and saved checkpoint data are the same as the source data used for creating checkpoint. After loading window data from the checkpoint, new checkpoint data should overwrite the old checkpoint data. The variables `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

Name	Creation of checkpoints with non consecutive numbers (append mode) with keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints, but each time with different data. Then open window again in checkpoint mode and load from the second checkpoint version. Create 2 more checkpoints.
Expected result	Checkpoints with consecutive numbers are created and saved checkpoint data are the same as the source data used for creating checkpoint. `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

Name	Creation of checkpoints with consecutive numbers without keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints.
Expected result	Only one checkpoint (the last one) exists. Metadata are set properly. `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

Name	Creation of checkpoints with consecutive numbers overwriting old checkpoints without keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints. Then open window again in checkpoint mode and load from first checkpoint version. Create 3 more checkpoints.
Expected result	Only one checkpoint (the last one) exists. Metadata are set properly. `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

Name	Creation of checkpoints with non consecutive numbers (append mode) without keeping old checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints. Then open window again in checkpoint mode and load from first checkpoint version. Create 3 more checkpoints.
Expected result	Only checkpoints 2, 3 and the last should exist. Metadata are set properly. `last_checkpoint_version`, `next_checkpoint_version` and `highest_checkpoint_version` are properly modified after each synchronization.

1.4 Managing functions

1.4.1 `MPI_Win_pmem_set_root_path`

Name	Too long root path
Description	Provide path parameter that is longer than `MPI_PMEM_MAX_ROOT_PATH`.
Expected result	Error `MPI_ERR_PMEM_ROOT_PATH`

Name	Not existing root path
Description	Provide path parameter that points to non existing directory.
Expected result	Error `MPI_ERR_PMEM_ROOT_PATH`

Name	Root path pointing to regular file
Description	Provide path parameter that points to a regular file.
Expected result	Error `MPI_ERR_PMEM_ROOT_PATH`

Name	Root path with non existing `.windows` file
Description	Provide path parameter that points to an existing directory without `.windows` file.
Expected result	Empty `.windows` file is created.

Name	Root path with existing `.windows` file
Description	Set root path to a valid path, create 2 windows, delete one of them, set root path once again and check the result.
Expected result	`.windows` file reflects status before second root path setting.

1.4.2 `MPI_Win_pmem_list`

Name	List windows
Description	Create 3 windows, delete one of them and check whether appropriate metadata are returned.
Expected result	Appropriate metadata saved in windows structure.

1.4.3 `MPI_Win_pmem_get_versions`

Name	List windows and their versions
Description	Create 3 windows, delete one of them, create 3 checkpoints in the remaining 2 windows (by synchronizing), delete first checkpoint version from one of the windows and the second checkpoint version in the other window and check whether appropriate metadata are returned.
Expected result	Appropriate metadata saved in versions structure.

1.4.4 `MPI_Win_pmem_delete`

Name	Delete window with all previous checkpoints
Description	Create a window, synchronize 3 times to create 3 checkpoints and delete the window.
Expected result	All old checkpoints as well as data and metadata file should be deleted (files with appropriate names should not exist) and global metadata file should indicate that file is deleted.

Name	Delete window that was already deleted
Description	Create a window and delete the window twice.
Expected result	The second attempt to delete the window should complete without errors and all old checkpoints as well as data and metadata file should be deleted (files with appropriate names should not exist) and global metadata file should indicate that file is deleted.

Name	Delete window that never existed
Description	Try to delete a window, which does not exist.
Expected result	Error `MPI_ERR_PMEM_NAME`

Name	Delete window without first checkpoint
Description	Create a window, synchronize 3 times to create 3 checkpoints, delete first checkpoint and delete the window.
Expected result	All old checkpoints as well as data and metadata file should be deleted (files with appropriate names should not exist) and global metadata file should indicate that file is deleted.

Name	Delete window without middle checkpoint
Description	Create a window, synchronize 3 times to create 3 checkpoints, delete the second checkpoint and delete the window.
Expected result	All old checkpoints as well as data and metadata file should be deleted (files with appropriate names should not exist) and global metadata file should indicate that file is deleted.

Name	Delete window without the last checkpoint
Description	Create a window, synchronize 3 times to create 3 checkpoints, delete the last checkpoint and delete the window.
Expected result	All old checkpoints as well as data and metadata file should be deleted (files with appropriate names should not exist) and global metadata file should indicate that file is deleted.

1.4.5 `MPI_Win_pmem_delete_version`

Name	Delete window version of never existing window
Description	Try to delete any window version e.g. version 0 of window that never existed.
Expected result	Error `MPI_ERR_PMEM_NAME`

Name	Delete window version of deleted window
Description	Create a window, delete window and then try to delete any window version e.g. version 0.
Expected result	Error `MPI_ERR_PMEM_NAME`

Name	Delete window version that never existed
Description	Create a window, synchronize 3 times and try to delete window with version higher than 2.
Expected result	Error `MPI_ERR_PMEM_CKPT_VER`

Name	Delete window version that is already deleted
Description	Create a window, synchronize 3 times and delete window version 1 twice.
Expected result	The second deletion will succeed, 2 checkpoint data files will exist and all metadata will be set properly.

Name	Delete first window version.
Description	Create a window, synchronize 3 times and delete first window version
Expected result	The first checkpoint data file will not exist and this version will be marked as deleted in metadata. All other window versions will be untouched.

Name	Delete middle window version.
Description	Create a window, synchronize 3 times and delete the second window version
Expected result	The second checkpoint data file will not exist and this version will be marked as deleted in metadata. All other window versions will be untouched.

Name	Delete the last window version.
Description	Create a window, synchronize 3 times and delete the last window version
Expected result	The last checkpoint data file will not exist and this version will be marked as deleted in metadata. All other window versions will be untouched.