libmmc.so: A library for multithreaded MPI communication software offloading

This is an interposition library implementation of the software offloading concept [1] for MPI operations.

Compilation and Installation

Compiling: (requires CMake v3.12+)

```
mkdir -p build && cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
cmake --build . -j
cd ..
```

This will create the shared library *libmmc.so* in `build/lib`. If you want to install *libmmc.so*, you can copy the shared library file *libmmc.so* to your library path (e.g., `/usr/lib`).

Usage

Using the library requires either preloading *libmmc.so*, or linking *libmmc.so* to the application binary.

```
# Preload libmmc.so
export LD_PRELOAD={PATH TO LIBMMCSO}/libmmc.so
# Run MPI application
```

Thread affinity

It is recommended to pin the offloading thread to a core in the same NUMA domain as the threads of the multithreaded MPI process. This core should **not** be used by any of the application threads.

Controlling thread affinity with *libmmc.so* is possible using the environment variables `MMCSO_THREAD_AFFINITY` and `MMCSO_DISPLAY_AFFINITY`.

Syntax:

```
# Set thread affinity of offloading thread(s)
MMCSO_THREAD_AFFINITY={LIST}
LIST=rank:cpu[,LIST]
# Show thread affinity of offloading thread
MMCSO_DISPLAY_AFFINITY={TRUE|FALSE}
```

See `tools/run.sh` for an example usage with OpenMPI with 2 MPI processes and 3 OpenMP threads per MPI process.

The file `bench/mt_overlap` in the build folder contains a micro-benchmark that will be run using `run.sh`.

References

[1] 2015 Vaidyanathan et al.: Improving concurrency and asynchrony in multithreaded MPI applications using software offloading