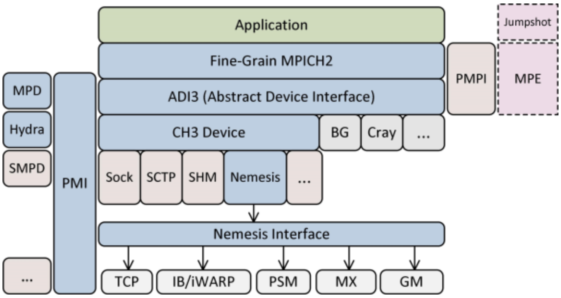
FG-MPI: Fine-Grain MPI

Humaira Kamal and Alan Wagner <kamal,wagner>@cs.ubc.ca

NSS Lab, University of British Columbia.

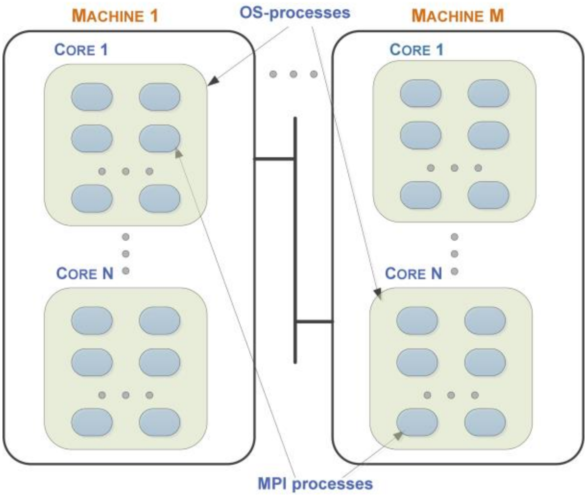
FG-MPI extends the execution model of the Message Passing Interface (MPI) to expose large-scale, fine-grain concurrency. **FG-MPI** is integrated into the **MPICH2** middleware, which is an **award winning**, **production quality** implementation of the MPI standard from the Argonne National Laboratory.



FG-MPI Architecture. Blue regions show the layers of MPICH2 that were augmented in the FG-MPI implementation. (*Figure adapted from MPICH2 SC07 flyer*).

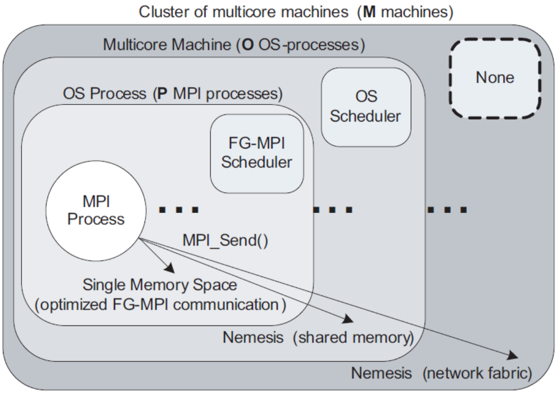
Decouples MPI processes from the Hardware

- FG-MPI supports interleaved execution of multiple MPI processes inside an OS-process.
- Allows the user to run millions of MPI processes without needing the corresponding number of processor cores.
- Makes it possible for the user to design algorithms and vary the number of MPI processes to match the problem rather than the hardware.
- The granularity of existing MPI programs can be adjusted through the command-line to better fit the cache leading to improved performance.



Location-aware Communication

- Exploits locality of MPI processes for optimized communication within the same OS-process. Communication on the same machine leverages MPICH2's optimized shared memory Nemesis intra-node subsystem.
- For further optimization, FG-MPI defines additional zero-copy communication routines to allow the user to pass reference to the data buffer and avoid message copying among co-located MPI processes inside an OS-process.


Cluster of multicore machines (M machines)
Multicore Machine (O OS-processes)
None
OS Process (P MPI processes)
OS Scheduler
FG-MPI Scheduler
MPI Process
. . .
. . .
. . .
MPI_Send()
Single Memory Space (optimized FG-MPI communication)
Nemesis (shared memory)
Nemesis (network fabric)

FG-MPI: Fine-Grain MPI

Function-level Parallelism

- FG-MPI has a light-weight and scalable design to support function-level parallelism.
- Each of the MPI processes inside an OS-process execute **functions instead of main programs**.
- Provides a simple API for the user to specify binding of MPI processes to functions.

Flexibility of Mapping

```
mpiexec -nfg 1000 -n 4 myprog
```

```
mpiexec -nfg 500 -n 8 myprog
```

```
mpiexec -nfg 500 -n 3 myprog:-nfg 250 -n 2 myprog:
          -nfg 1000 -n 2 myprog
```

- FG-MPI adds a **nfg** flag to the mpiexec command to specify the number of fine-grain processes inside an OS-process.
- Allows MPI processes to be **flexibly mapped** to OS-processes, cores and machines.
- Different mappings can be executed without recompilation.
- Allows **seamless execution of hundreds and thousands of MPI processes** on a laptop or a cluster.
- The mpiexec command is backward compatible and it is also possible to mix an OS-process with a single MPI process with those with multiple MPI processes.
- FG-MPI runs on commodity systems.

Development and Testing

- Programmers can **develop MPI programs that scale** to hundreds and thousands on their laptops or desktops.
- FG-MPI implements an integrated runtime scheduler that is reactive to the events occurring inside the MPI middleware. It provides the flexibility to **select different runtime schedulers** on the command line.
- Use of deterministic scheduler like round-robin can **help in debugging** and testing of programs.
- FG-MPI provides the ability to run all MPI processes inside a single OS-process. This can help detect program safety issues like deadlock.
- Allows for easy porting of existing MPI programs through addition of a small boiler-plate code.

Over a 100 million MPI processes

- In January 2013, we ran a series of experiments on the Western Canada Research Grid (**WestGrid**) computing facility, and successfully scaled to execute over a **100 million MPI processes** on 6,480 processor cores with 16,000 co-located MPI processes in each OS-process.
- FG-MPI's experiments demonstrate the scalability of the MPICH2 middleware and the basic routines in MPI.