OpenSHMEM over Portable MPI RMA with Asynchronous Progress Support

Min Si, Pavan Balaji

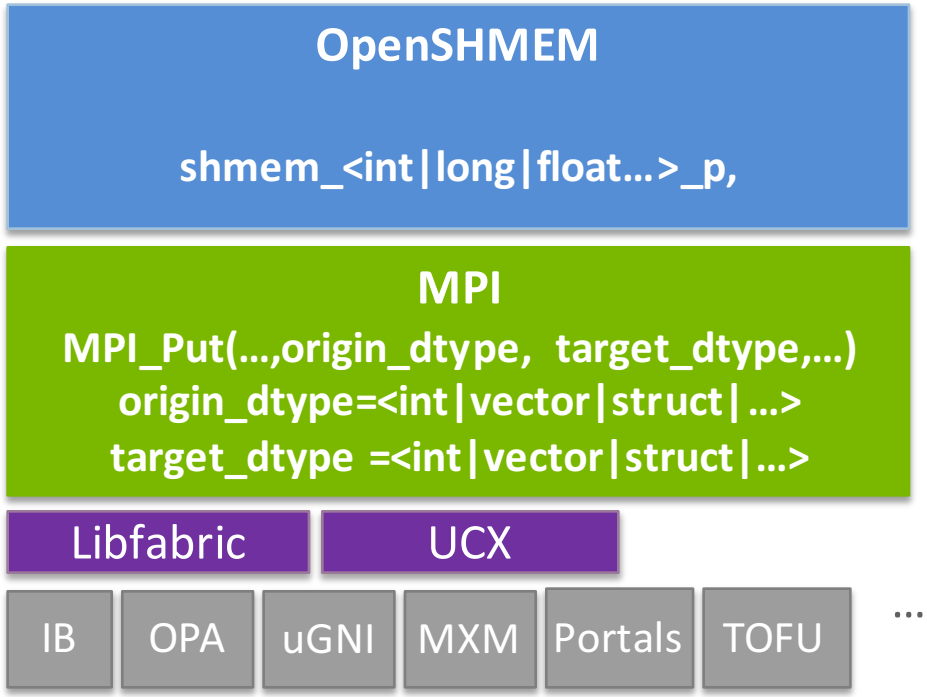
Programming Models and Runtime Systems Group

Argonne National Laboratory, USA

OpenSHMEM over MPI and Challenges

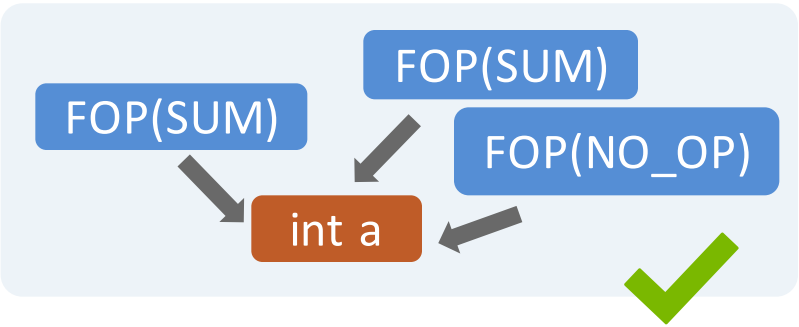
- **OpenSHMEM**
  - Focusing on one-sided and collective abstraction
  - Specialized API allows direct and highly efficient optimization opportunities
- **MPI**
  - Low level library focusing on completeness of feature (e.g., two-sided, one-sided, collectives, various operation types)
  - Explicit user-control of communication

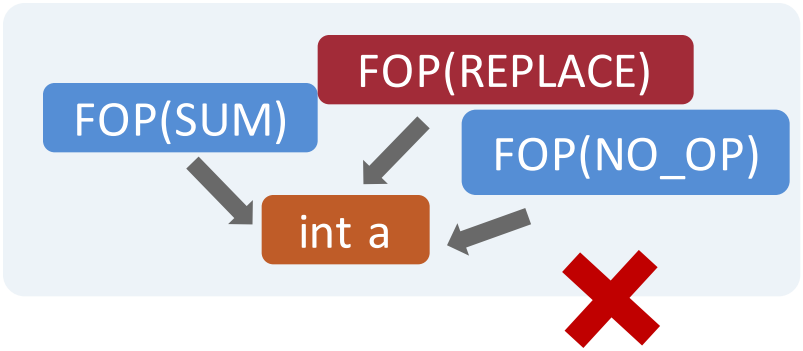
- **OpenSHMEM over MPI ?**
  - Improve portability but raise **over-generalization** issues
  - Can we resolve and how far ?



Limitations in Atomic Operation Support (Semantics)

- Semantics mismatching
  - **OpenSHMEM AMO:** Ensure atomicity for two different AMOs to access the same memory region simultaneously.
  - **MPI Accumulate Ops:** Support atomicity between operations only when they are "**same_op_no_op**" or "**same_op**".
- To ensure correctness
  - Propose new value "**none**" for "**accumulate_ops**" in MPI-4 standard
  - Allow runtime to fully support atomicity for any different operations.

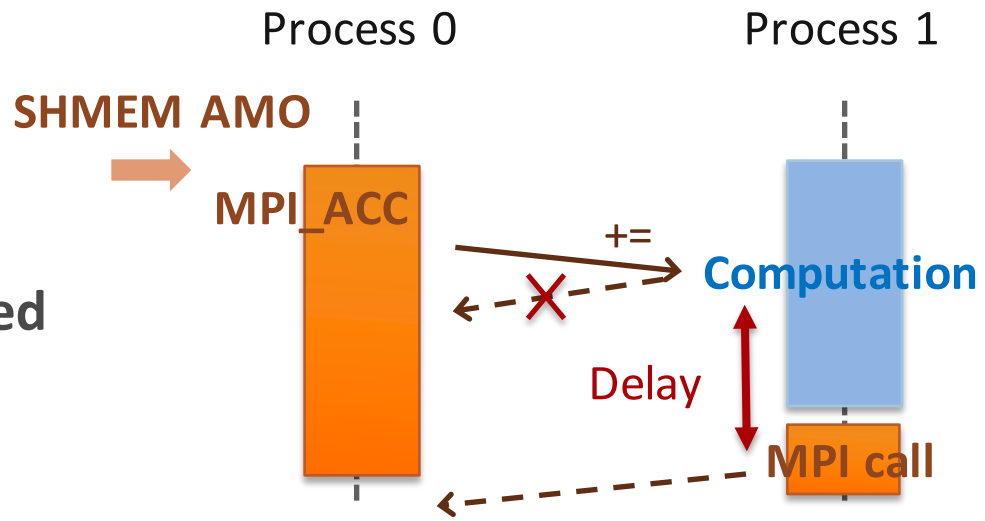




Limitations in Atomic Operation Support (Performance)

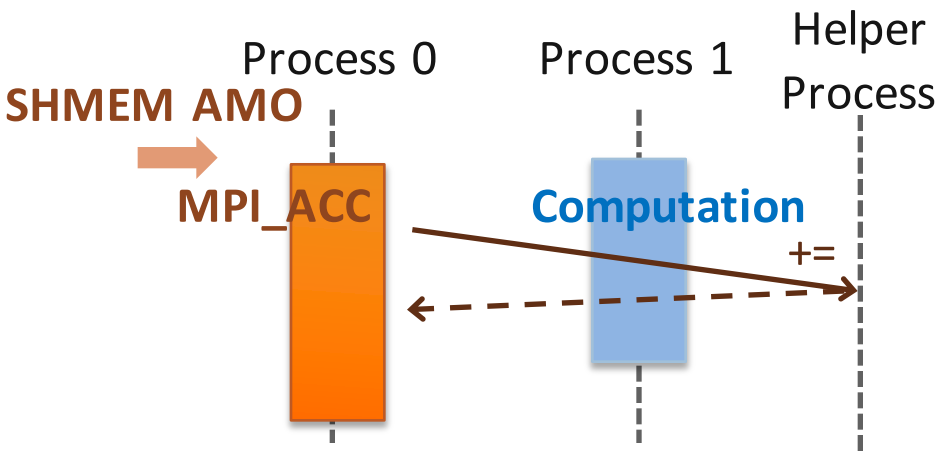
- Various OP types in MPI accumulates
  - SUM|REPLACE|MAX|MIN|…
- With "accumulate_ops = none"
  - Hardware might not support all the atomic operations
  - Type of concurrent operations is unknown
  - **All operations have to be handled in MPI software (by calling MPI on target) to ensure atomicity**

- Performance Limitation
  - **Lack of asynchronous progress** in SW-handled MPI OPs
  - Long delay happens if target is busy in computation



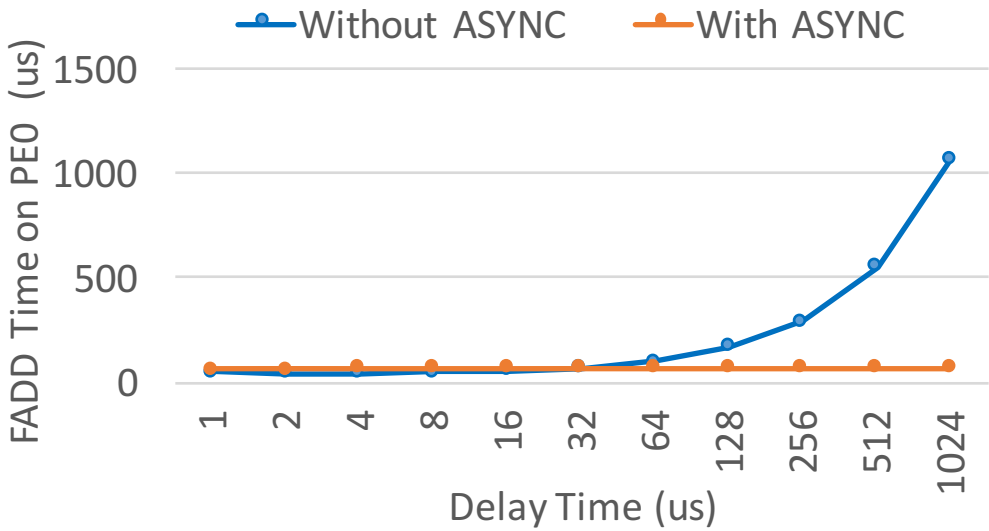
With Asynchronous Progress Support

- Casper
  - Dedicating arbitrary number of cores to **helper processes**
  - Helper process intercepts all RMA operations to the user processes



- Performance Showcase

```
For each iteration {
        if (me == 0) {
            For 10 operations:
                shmem_int_fadd(dest,  1, pe=1)
        } else  {
             while (time <= Delay_Time);
        }
}
```



OSHMPI over IntelMPI 2017 on two nodes interconnected by Omni-Path