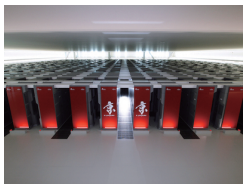
# Performance improvement of MODYLAS using Remote Direct Memory Access on the K computer

Masahiro Nakao,[1] Hitoshi Murai,[1] Mitsuhisa Sato,[1] Yoshimichi Andoh,[2] Susumu Okazaki[2]
(1. RIKEN Center for Computational Science, 2. Nagoya University)

## What is MODYLAS ? https://www.modylas.org

- MOlecular DYnamics software for LArge System
- MODYLAS utilizes the fast multipole method (FMM) for the calculation of the electrostatic interactions
- MODYLAS is executed on large-scale supercomputers such as the K computer (right figure)
- Our preliminary evaluation indicates the time required for MPI communication is limited by its latency



SPARC64 VIIIfx 2GHz, DDR3 SDRAM 16GB, Tofu interconnect 5GB/s

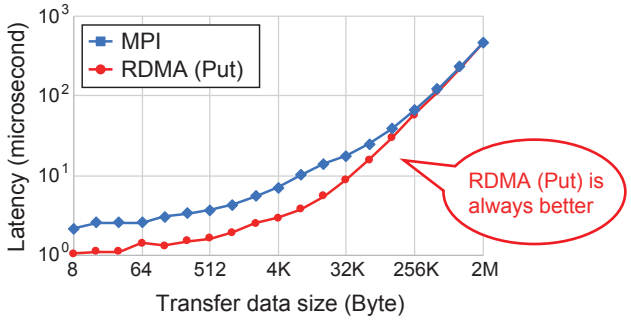
## Summary


- In order to improve the performance of MODYLAS, this research replaces MPI communication with Remote Direct Memory Access (RDMA) on the K computer
- Since the K computer provides the extended RDMA interface for RDMA operations, we implement a library to use the interface from MODYLAS easily
- As a result of measuring the performance of MODYLAS, **the RDMA communication time is 29-42% less than the MPI communication time**


## Approach

### Replacement MPI with RDMA

- The K computer provides users with the extended RDMA interface so that they can issue RDMA operations (Put/Get) with low latency
- This graph shows a comparison of latency between MPI and RDMA (Put) on the K computer using ping-pong benchmark




### Modified code of MODYLAS

We implement a library to use the extended RDMA interface from MODYLAS easily

```
integer(4),allocatable,dimension(:) :: icbufp
allocate(ircbufp(s))
#ifdef RDMA
call rdma_register_addr(ircbufp, s*4)
#endif
  :
#ifdef RDMA
integer(8),pointer :: ircbufp_raddr(:)
type(c_ptr) :: ircbufp_cptr
ircbufp_cptr = rdma_get_raddr(ircbufp)
call c_f_pointer(ircbufp_cptr, fptr=ircbufp_raddr, shape=[nprocs])
call rdma_put_post(ipz_pdest, ircbufp_raddr(ipz_pdest+1), ...)
call rdma_wait(ipz_psrc)
#else
call mpi_irecv(ircbufp, ..., ipz_psrc, ...)
call mpi_isend(icbufp, ..., ipz_pdest, ...)
call mpi_waitall(2, ...)
#endif
```

Register array to use RDMA

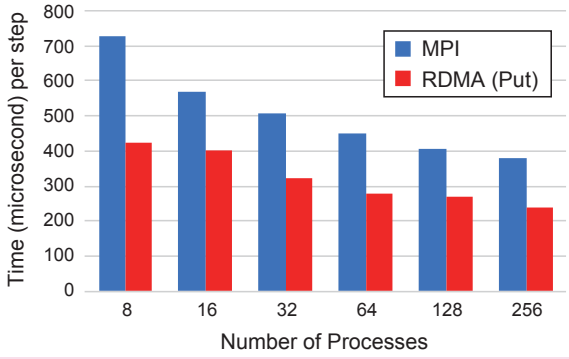
Get information of remote arrays

Perfom RDMA PUT communication

## Evaluation

### Communication time using MPI and RDMA

- RDMA communication time is 29-42% less than MPI communication time on the data set with three FMM levels
- Most transfer data sizes are less than 32K bytes, which is a sufficient size to demonstrate the superiority of RDMA




### Calculation time using MPI and RDMA

- This table shows the total calculation time including communication time per step
- Although the efficiency has increased by a factor of 2.91~4.68% overall, this will further increase for calculations with strong scaling with tuned code for hotspot calculations

| Num. of Proc. | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|
| MPI | 16,129 | 9,973 | 6,941 | 5,636 | 4,624 | 4,151 |
| RDMA (Put) | 15,551 | 9,684 | 6,706 | 5,384 | 4,467 | 4,033 |
| Improvement | 3.72% | 2.99% | 3.50% | 4.68% | 3.51% | 2.91% |

### Future Work

- To make MODYLAS available for reducing communication times in various computing environments, we will utilize coarray features of the Fortran standard
- Since the coarray features provide users with one-sided communication, and its implementation may use RDMA that each machine has