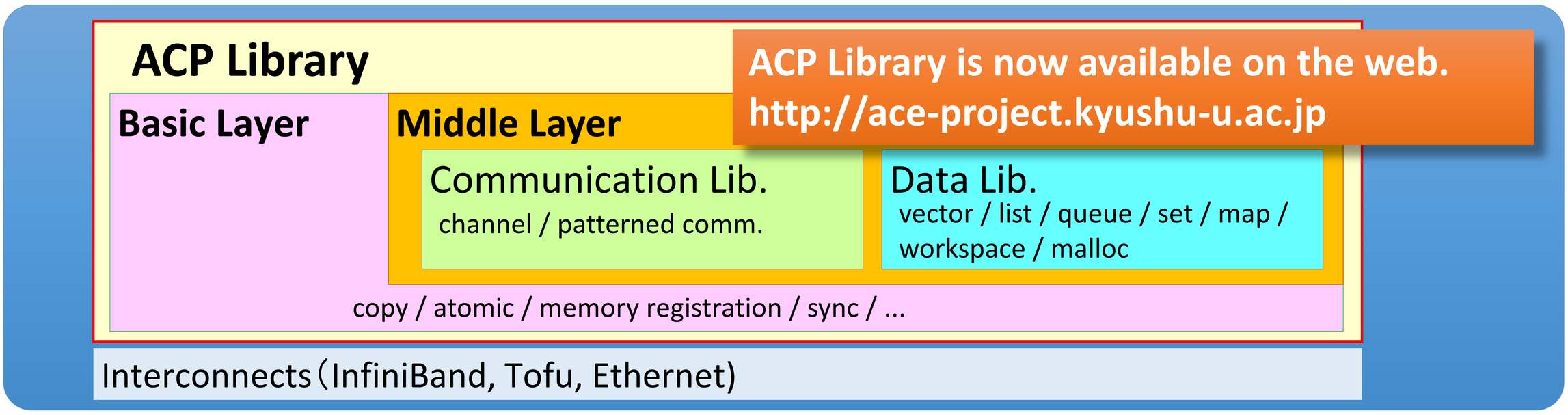
Advanced Communication for Exa (ACE)

- a project on memory-efficient communication library -

Motivation

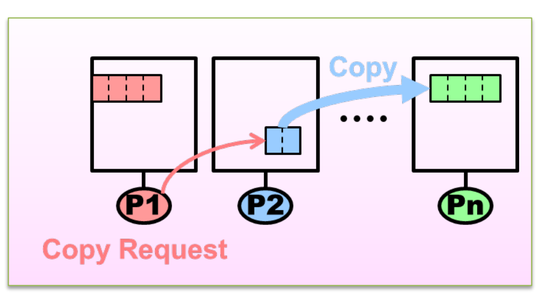
Memory-efficient communication towards exa-scale computing.

Advanced Communication Primitives (ACP) Library



Basic Layer

- PGAS-style global memory management
- Copy and Atomic Op on global address
- Express dependency between accesses

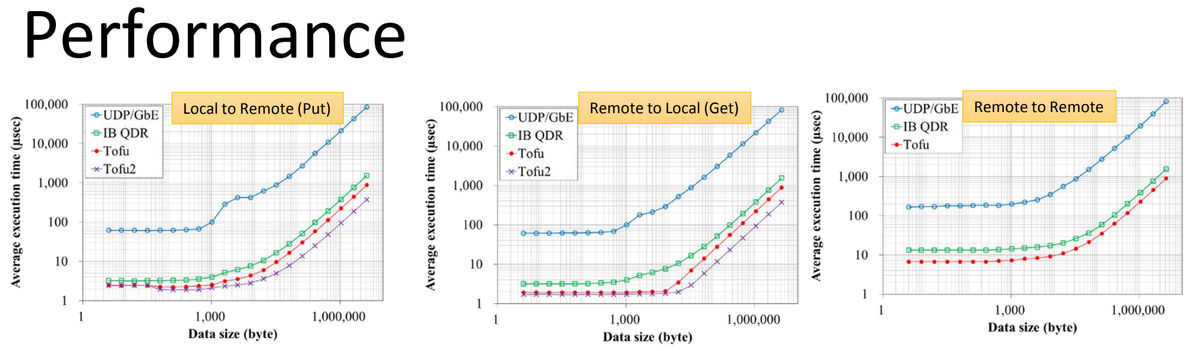



```
acp_handle_t h1, h2;

h1 = acp_copy(dstga1, srcga1, size1,
              ACP_HANDLE_NULL);
h2 = acp_copy(dstga2, srcga2, size2, h1);
acp_complete(h2);
```

Memory consumption ratio (1M procs)

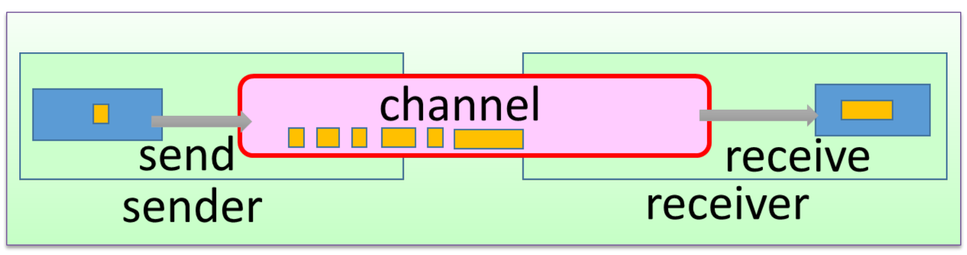
InfiniBand	Tofu	UDP
369 MiB / process	67MiB / process	34MiB / process

Performance



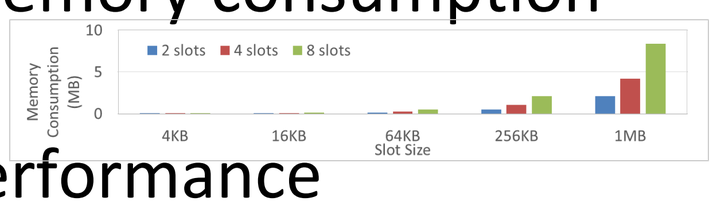
Communication Lib.

- Explicit creation / destruction of channels among processes
- 1 directional send / receive

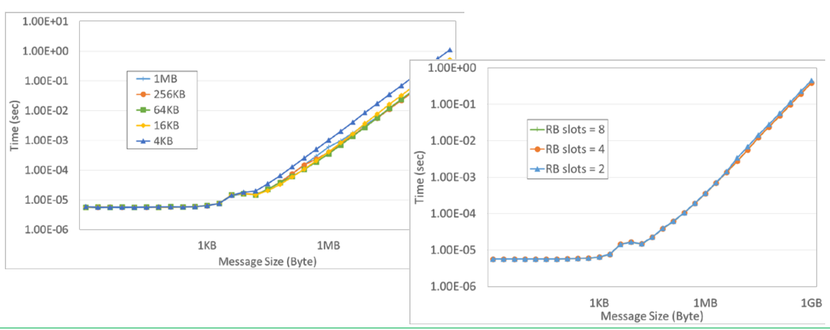



```
ch0 = acp_create_ch(left, myrank);
ch1 = acp_create_ch(myrank, right);

for (...){
  req0 = acp_nbsend(ch0, addr0, size);
  req1 = acp_nbrecv(ch1, addr1, size);
  acp_wait_ch(req0);
  acp_wait_ch(req1);
  calc();
}

req0 = acp_nbfree_ch(ch0);
req1 = acp_nbfree_ch(ch1);
acp_wait_ch(req0);
acp_wait_ch(req1);
```

Memory consumption

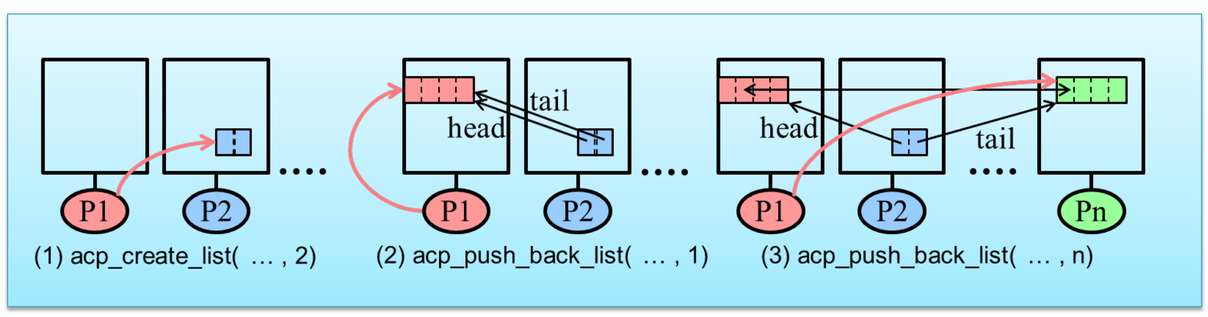


Performance



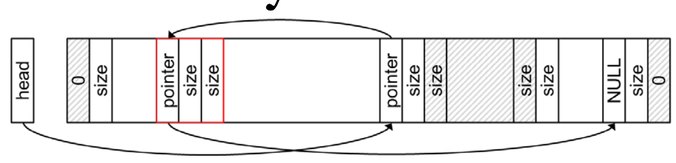
Data Lib.

- Create/modify/destruct data structures on global memory: vector, list, deque, map, set (from STL)

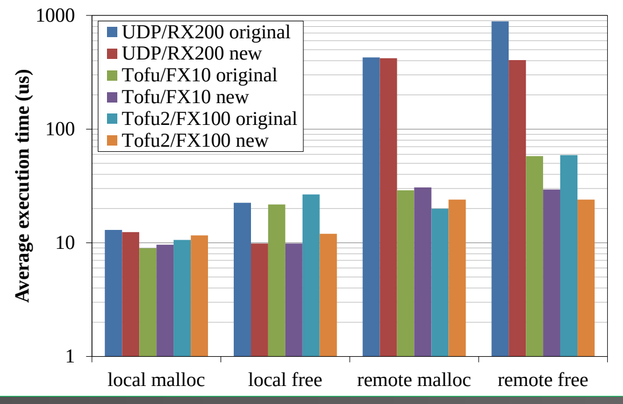


- Based on global memory allocator

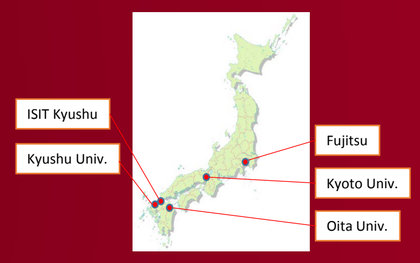
```
acp_ga_t ga;

ga = acp_malloc(size, rank);
  ...
acp_free(ga);
```



Performance



Kyushu Univ.	Takeshi Nanri, Hiroaki Honda, Ryutaro Susukita, Taizo Kobayashi, Yoshiyuki Morie
Fujitsu Ltd.	Shinji Sumimoto, Yuichiro Ajima, Naoyuki Shida, Kazushige Saga, Takafumi Nose
ISIT Kyushu	Hidetomo Shibamura, Takeshi Soga
Kyoto Univ.	Keiichiro Fukazawa
Oita Univ.	Toshiya Takami

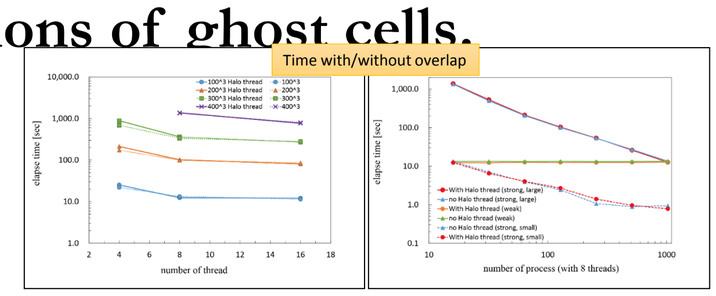


Applications

Halo Communications in MHD Simulation

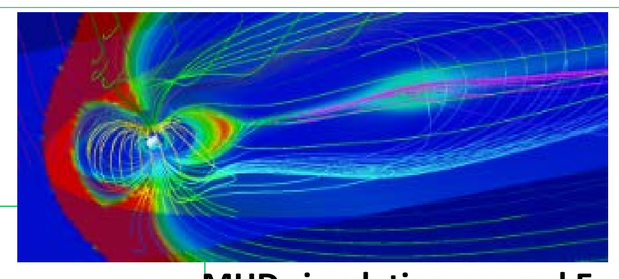
Halo thread

- Dedicated thread for communications and computations of ghost cells.
- Overlap with computations on other cells

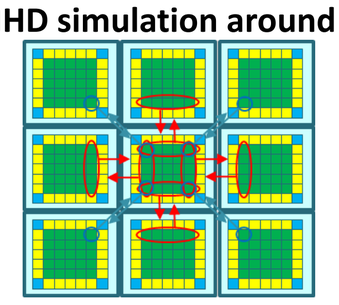


Halo framework

```
i=0;j=0;segnum=26;
hnd=halo_3d_init(nx,ny,nw,matrix,segsize);
for(j=0;j<snum[0];j++)
    halo_3d_isend(hndsxd[0][j],syd[0][j],szd[0][j]
                  ,sxs[0][j],sxs[0][j],sys[0][j],sye[0][j],szs[0][j],sze[0][j]);
for(i=0;i<segnum;i++) {
   for(j=0;j<rnum[i];j++) {
      halo_3d_irecv(hnd,sxd[i][j],syd[i][j],szd[i][j],sxs[i][j],sxs[i][j]
                    ,sys[i][j],sye[i][j],szs[i][j],sze[i][j]);
      halo_3d_wait(hnd);
   }
   compute ghost cells
   for(j=0;j<snum[i+1];j++)
       halo_3d_isend(hnd,sxd[i+1][j],syd[i+1][j],szd[i+1][j],sxs[i+1][j],sxs[i+1][j],
                     sys[i+1][j],sye[i+1][j],szs[i+1][j],sze[i+1][j]);
}
halo_3d_finalize(hnd);
```



MHD simulation around Earth

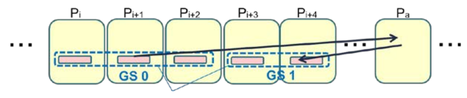


Master-Worker Model in FMO Method

Shared work-space

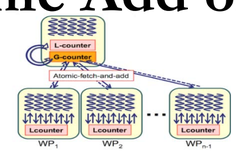
- Common continuous region asynchronously accessible from all of the processes.

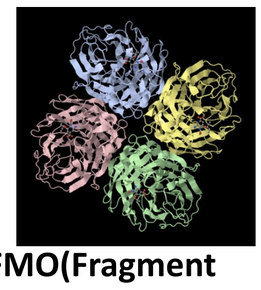
```
sws[0] = acp_create_sws( size0 );
sws[1] = acp_create_sws( size1 );
...
acp_write_sws( gds[0], size_write0, offset_write0, writebuf0 );
...
acp_read_sws(  gds[0], size_read1,  offset_read1,  readbuf1  );
...
acp_free_sws( gds[0] );
acp_free_sws( gds[1] );
```



Global counter

- Dynamic load balance with Atomic Add of ACP



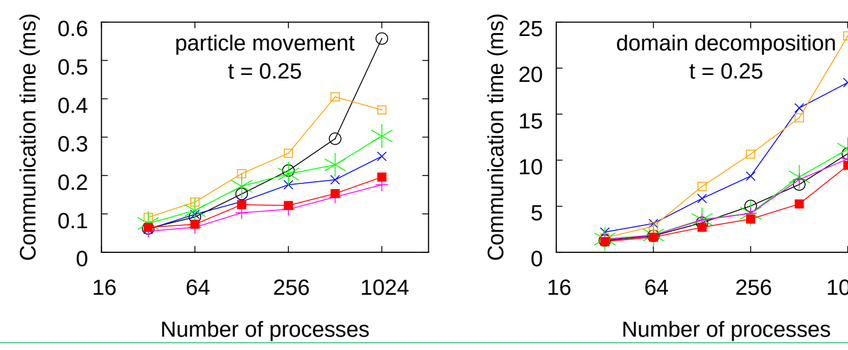


FMO(Fragment Molecular Orbital)

Particle Data Exchange in *N*-body Simulation

Irregular communication pattern

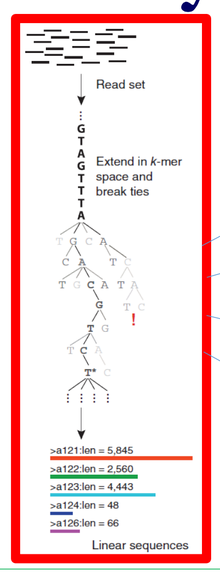
- Inter-process data exchange after particle movement
- Efficient data transfer and synchronization via Global Memory Access (copy and atomic) of ACP
- Better performance than MPI Remote Memory Access



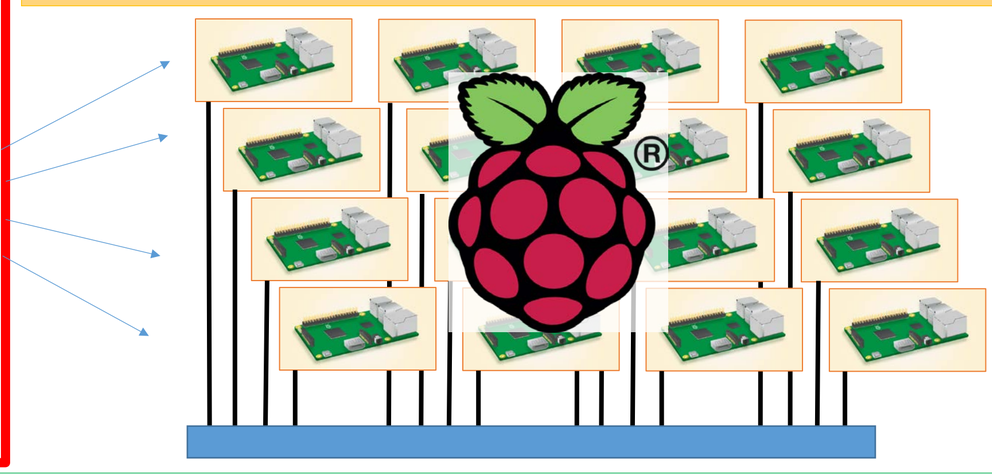
De Novo Transcript Assembly on Python

Concurrent *k*-mer Dictionary on Distributed Memory

- Split NGS short reads into small *k*-mers and store them into distributed dictionary on "map" data structure of ACP.
- Splice *k*-mers to build long sequences.
- Intending enhanced implementation via Python interface.



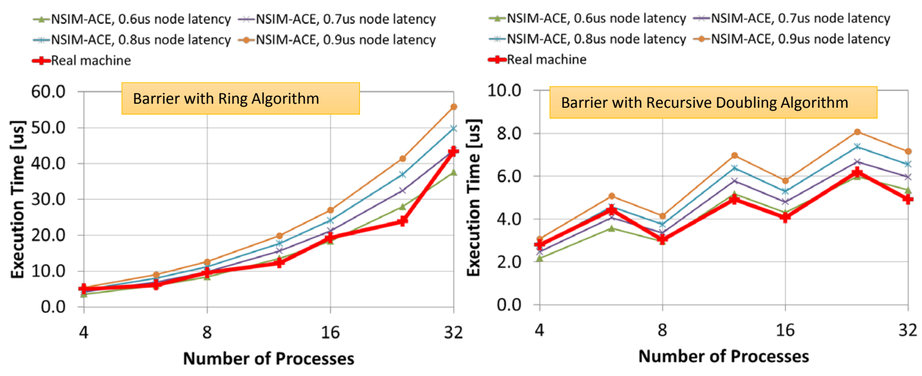
Live demo on a Raspberry Pi cluster



Performance Estimation Tool: NSIM-ACE

- Interconnect simulator for performance evaluation and communication analysis.
- Supports message-passing and/or one-sided communications.
- MGEN program: Skelton of communication pattern.
  = Easily extract from real programs.

```
if(rank == src_rank){
    handle = MGEN_acp_copy(src_rank,
                dest_rank, data_size, tag);
    MGEN_comp(time0);
    MGEN_acp_complete(handle);
}else if(rank == dest_rank){
    MGEN_comp(time1);
    MGEN_poll(tag);
}
```



Find posters in the booth with more detailed information