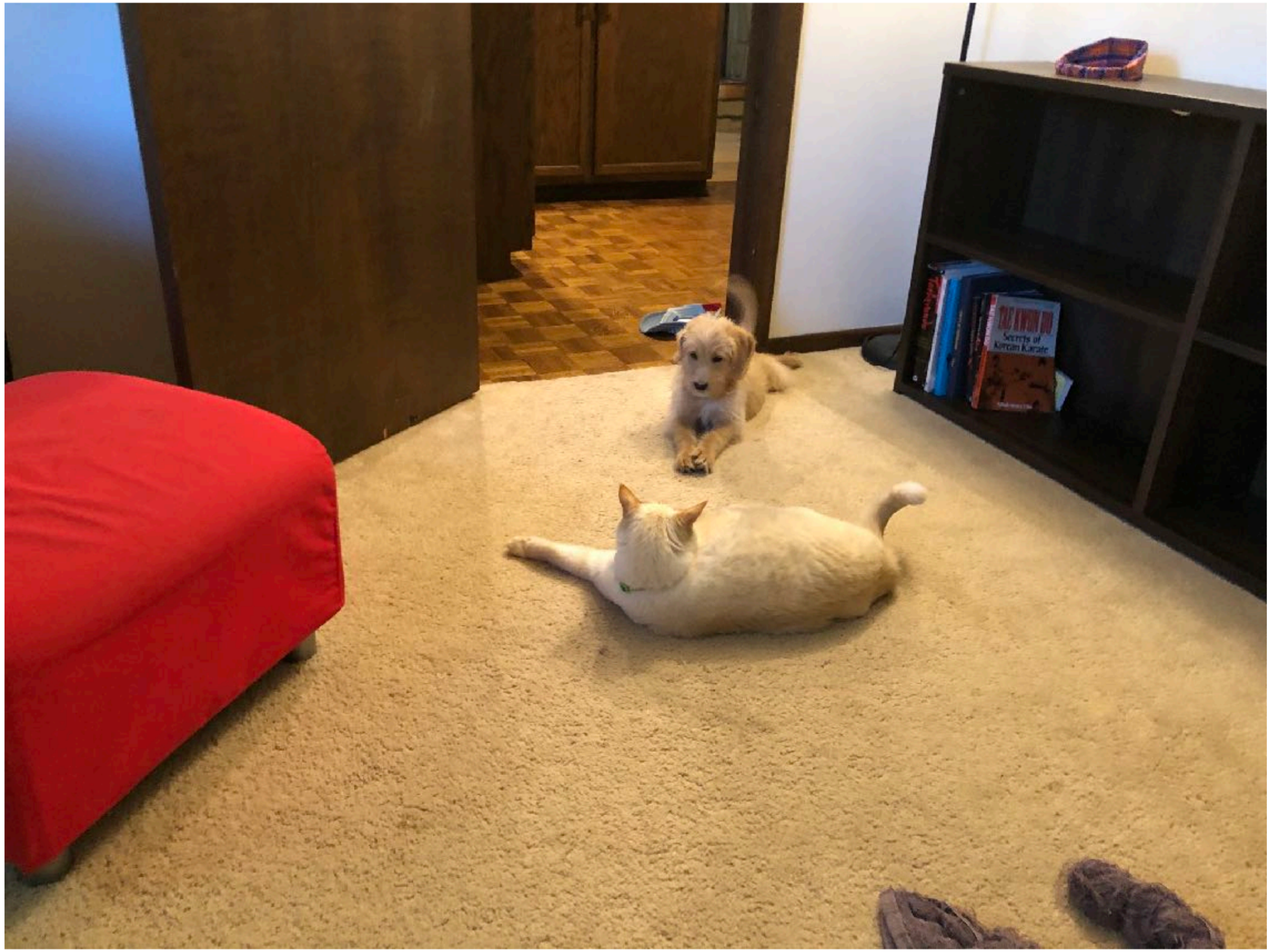
Lecture 9: MPI Collectives 2

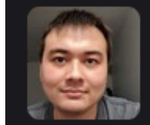
CMSE 822: Parallel Computing

Prof. Sean M. Couch

Puppy time



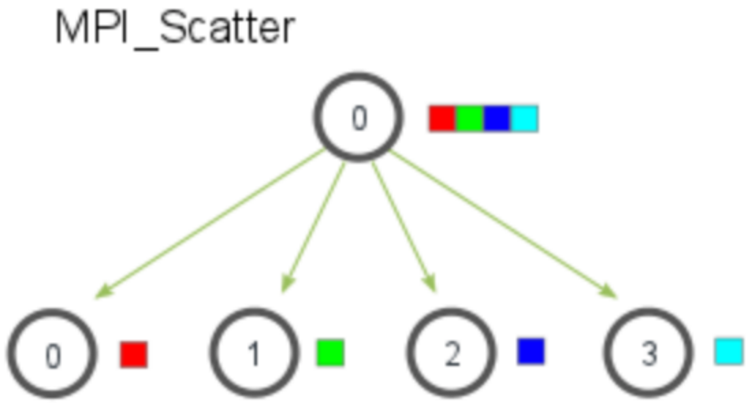
PCA Questions



Matthew Zeilbeck 2:15 AM

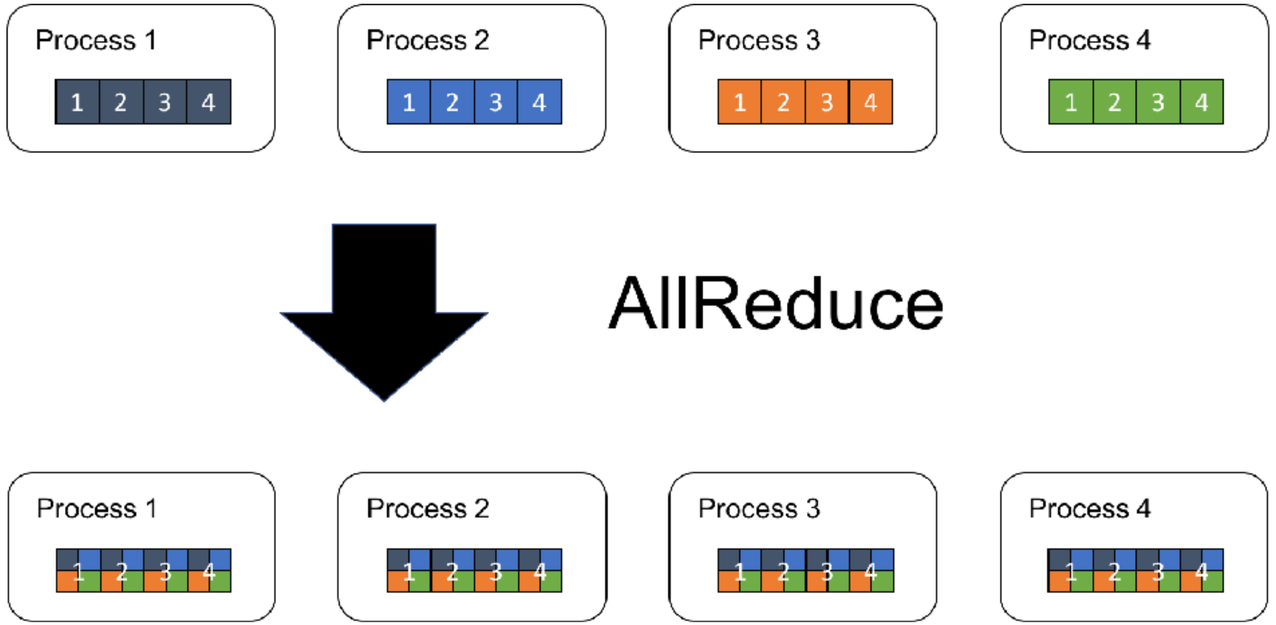
PCA8: When doing operations like gather, scatter, or all-to-all, how do we know the ordering? For a scatter, which element of the send buffer array goes to which process? What is the ordering of elements in the receive buffer array after a gather, or does that not generally matter?

13



PCA Questions

MPI_Reduce with MPI_IN_PLACE




```
// allreduceinplace.c
for (int irand=0; irand<nrandoms; irand++)
        myrandoms[irand] = (float) rand()/(float)RAND_MAX;
// add all the random variables together
MPI_Allreduce(MPI_IN_PLACE,myrandoms,
                    nrandoms,MPI_FLOAT,MPI_SUM,comm);
```

PCA Questions

MPI_Reduce with MPI_IN_PLACE

```
if (procno==root)
MPI_Reduce(MPI_IN_PLACE,myrandoms,
                    nrandoms,MPI_FLOAT,MPI_SUM,root,comm);
else
MPI_Reduce(myrandoms,MPI_IN_PLACE,
                    nrandoms,MPI_FLOAT,MPI_SUM,root,comm);
```

works

better

```
float *sendbuf,*recvbuf;
if (procno==root) {
  sendbuf = MPI_IN_PLACE; recvbuf = myrandoms;
} else {
  sendbuf = myrandoms; recvbuf = MPI_IN_PLACE;
}
MPI_Reduce(sendbuf,recvbuf,
          nrandoms,MPI_FLOAT,MPI_SUM,root,comm);
```

PCA Questions

MPI_Reduce with MPI_IN_PLACE

Junjie Han 8:51 PM

PCA8: In the example of 3.2.1, why the result is approximately 0.5 times the number of processes?

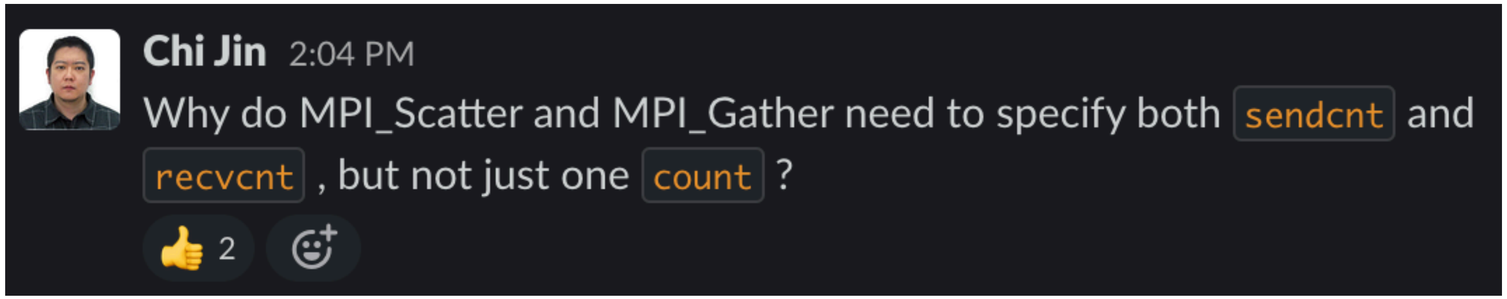
👍 8

rand() returns number between 0 and 1.

Expectation value is then 0.5

```
// allreduce.c
float myrandom,sumrandom;
myrandom = (float) rand()/(float)RAND_MAX;
// add the random variables together
MPI_Allreduce(&myrandom,&sumrandom,
                1,MPI_FLOAT,MPI_SUM,comm);
// the result should be approx nprocs/2:
if (procno==nprocs-1)
  printf("Result %6.9f compared to .5\n",sumrandom/nprocs);
```

PCA Questions



From MPI Standard:

> The type signature of `sendcount`, `sendtype` on each process must be equal to the type signature of `recvcount`, `recvtype` at the root. This implies that the amount of data sent must be equal to the amount of data received, pairwise between each process and the root. Distinct type maps between sender and receiver are still allowed.

PCA Questions

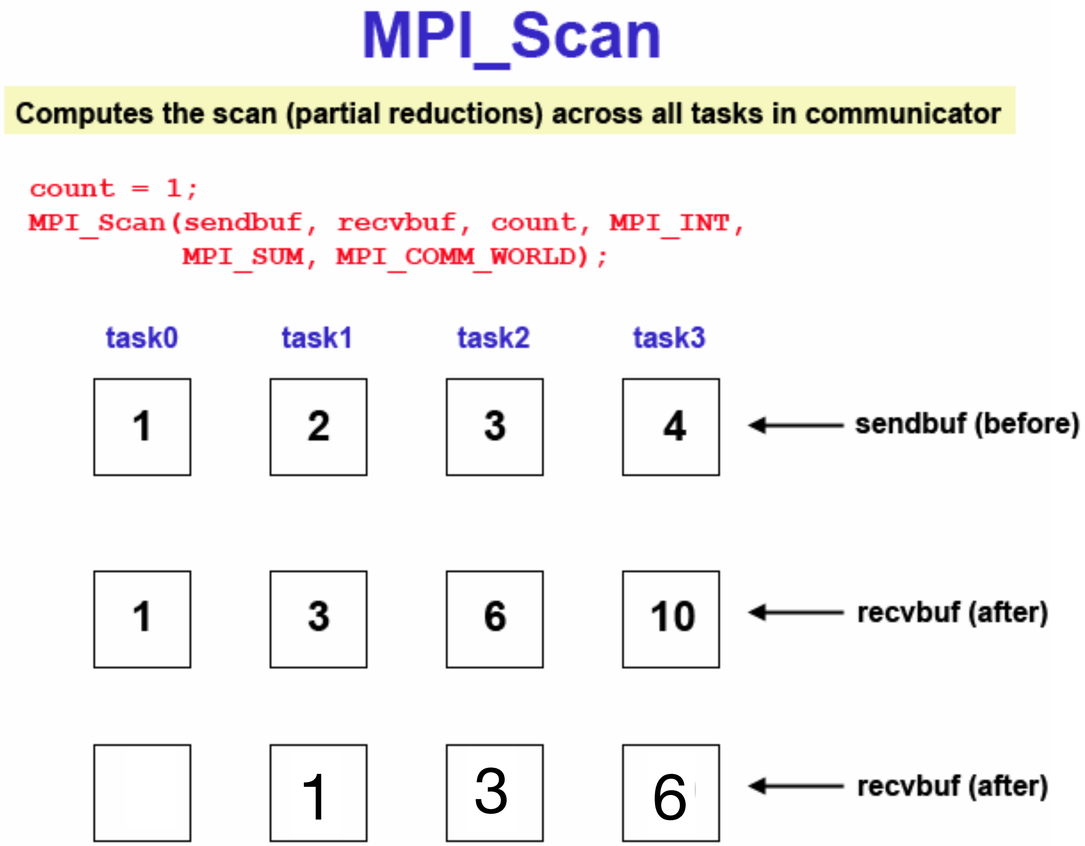
Overlapping and none-blocking

Emily Bolger 4:16 PM

PCA9: When would it make sense to overlap a non-blocking collective with a blocking collective?

PCA Questions

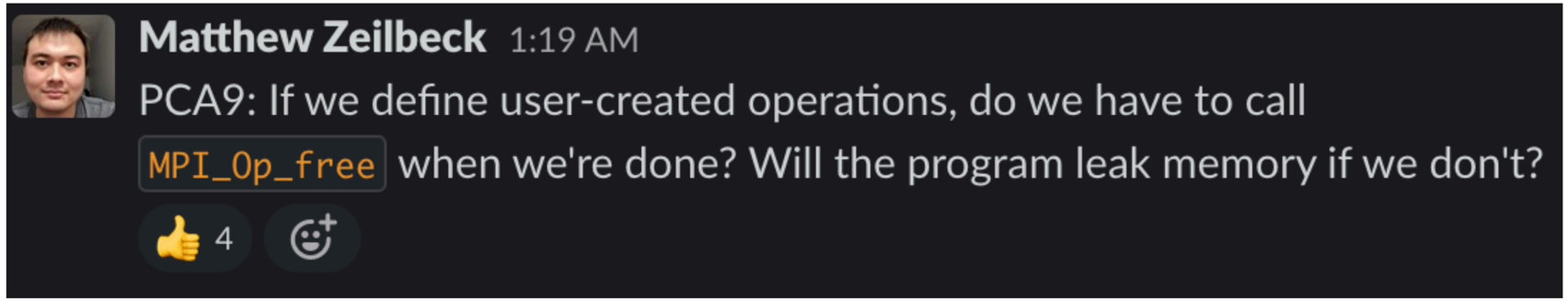
MPI_Exscan



MPI_Exscan:

`MPI_Exscan` is like `MPI_Scan`, except that the contribution from the calling process is not included in the result at the calling process (it is contributed to the subsequent processes, of course).

PCA Questions



- Yes! Program will not result in error, but will leak memory.

Synchronization

```
switch(rank) {
    case 0:
        MPI_Bcast(buf1, count, type, 0, comm);
        MPI_Send(buf2, count, type, 1, tag, comm);
        break;
    case 1:
        MPI_Recv(buf2, count, type, MPI_ANY_SOURCE, tag, comm, &status);
        MPI_Bcast(buf1, count, type, 0, comm);
        MPI_Recv(buf2, count, type, MPI_ANY_SOURCE, tag, comm, &status);

        break;
    case 2:
        MPI_Send(buf2, count, type, 1, tag, comm);
        MPI_Bcast(buf1, count, type, 0, comm);
        break;
}
```