

Solution 12

Solution 12

C

```
#include <stdio.h>
#include <mpi.h>
#define DIM_BUF 10

int main(int argc, char **argv){
    int myrank, nproc;
    int buf[DIM_BUF];
    int i, offset, intsize;
    MPI_File fh;
    MPI_Status status;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nproc);
    MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

    for(i=0;i<DIM_BUF;i++) buf[i] = myrank*DIM_BUF+i;

    /* Open the file and write by using individual file pointers */
    MPI_File_open(MPI_COMM_WORLD, "output.dat", MPI_MODE_CREATE|MPI_MODE_WRONLY, MPI_INFO_NULL,&fh);
    MPI_Type_size(MPI_INT, &intsize);
    offset = myrank*DIM_BUF*(intsize);
    MPI_File_seek(fh,offset,MPI_SEEK_SET);
    MPI_File_write(fh,buf,DIM_BUF,MPI_INT,&status);
    MPI_File_close(&fh);

    /* Re-open the file and read by using explicit offset */
    MPI_File_open(MPI_COMM_WORLD,"output.dat",MPI_MODE_RDONLY,MPI_INFO_NULL,&fh);
    MPI_Offset file_size;
    MPI_File_get_size(fh,&file_size);
    offset = file_size/nproc*myrank;
    printf("myid %d, filesize %lld, offset %d\n", myrank,file_size,offset);
    MPI_File_read_at(fh,offset,&buf,DIM_BUF,MPI_INT,&status);
```

```

printf("myid %d, buffer after read:",myrank);
for(i=0;i<DIM_BUF; i++)printf("%d ",buf[i]);
printf("\n\n");
MPI_File_close(&fh);

/* Write the new file using the mpi_type_create_vector. Use the fileview */
MPI_Datatype filetype;
MPI_File_open(MPI_COMM_WORLD, "output_mod.dat", MPI_MODE_CREATE|MPI_MODE_WRONLY, MPI_INFO_NULL,&fh);
MPI_Type_vector(DIM_BUF/2,2,2*nproc,MPI_INT,&filetype);
MPI_Type_commit(&filetype);

MPI_Type_size(MPI_INT, &intsize);
offset = 2*intsize*myrank;
MPI_File_set_view(fh,offset,MPI_INT,filetype,"native",MPI_INFO_NULL);
MPI_File_write_all(fh,buf,DIM_BUF,MPI_INT,&status);
MPI_File_get_size(fh,&file_size);
printf("myid %d, filesize of the second file written %lld, offset %d\n", myrank,file_size,offset);

MPI_Type_free(&filetype);
MPI_File_close(&fh);
MPI_Finalize();
return 0;
}

```

FORTRAN

```

PROGRAM mpi2io

    USE MPI
    IMPLICIT none

    INTEGER :: myrank, nproc,ierr;
    INTEGER, PARAMETER :: dim_buf = 10
    INTEGER :: buf(dim_buf)
    INTEGER :: i, intsize;
    INTEGER(KIND=MPI_OFFSET_KIND) :: offset, file_size
    INTEGER :: fh
    INTEGER :: status(MPI_STATUS_SIZE)
    INTEGER :: filetype

    CALL MPI_Init(ierr);
    CALL MPI_Comm_size(MPI_COMM_WORLD,nproc,ierr)
    CALL MPI_Comm_rank(MPI_COMM_WORLD,myrank,ierr)

    DO i=1,dim_buf
        buf(i) = myrank*dim_buf+i-1;
    END DO

```

```
! Open the file and write by using individual file pointers
CALL MPI_File_open(MPI_COMM_WORLD, "output.dat", MPI_MODE_CREATE+MPI_MODE_WRONLY, MPI_INFO_NULL, fh, ierr)

CALL MPI_Type_size(MPI_INTEGER, intsize, ierr)
offset = myrank*dim_buf*(intsize)
CALL MPI_File_seek(fh, offset, MPI_SEEK_SET, ierr)
CALL MPI_File_write(fh, buf, dim_buf, MPI_INTEGER, status, ierr)

CALL MPI_File_close(fh, ierr)

! Re-open the file and read by using explicit offset
CALL MPI_File_open(MPI_COMM_WORLD, "output.dat", MPI_MODE_RDONLY, MPI_INFO_NULL, fh, ierr)

CALL MPI_File_get_size(fh, file_size, ierr)
offset = file_size/nproc*myrank
write(6,*) "myid ", myrank, "filesize ", file_size, "offset ", offset

CALL MPI_File_read_at(fh, offset, buf, dim_buf, MPI_INTEGER, status, ierr)
CALL MPI_File_close(fh, ierr)

! Write the new file using the mpi_type_create_vector. Use the fileview
CALL MPI_File_open(MPI_COMM_WORLD, "output_mod.dat", MPI_MODE_CREATE+MPI_MODE_WRONLY, MPI_INFO_NULL, fh, ierr)
CALL MPI_Type_vector(dim_buf/2, 2, 2*nproc, MPI_INTEGER, filetype, ierr)
CALL MPI_Type_commit(filetype, ierr)

CALL MPI_Type_size(MPI_INTEGER, intsize, ierr)
offset = 2*intsize*myrank
CALL MPI_File_set_view(fh, offset, MPI_INTEGER, filetype, "native", MPI_INFO_NULL, ierr)
CALL MPI_File_write_all(fh, buf, DIM_BUF, MPI_INTEGER, status, ierr)

CALL MPI_File_get_size(fh, file_size, ierr)
write(6,*) "myid ", myrank, "filesize of the second file written", file_size, "offset", offset

CALL MPI_Type_free(filetype, ierr)
CALL MPI_File_close(fh, ierr)
CALL MPI_Finalize(ierr)

END PROGRAM mpi2io
```