

Solution 9

Solution 9

C

```
#include <stdlib.h>
#include <stdio.h>
#include <mpi.h>

int main (int argc, char* argv[]){

    int i,my_id, num_procs,N=50;
    int array[N],array_final_sum[N],array_final_mult[N];
    int r_num,max_value;
    unsigned seed;
    double t0, t1, time;

    MPI_Init(&argc, &argv);      /* starts MPI */
    MPI_Comm_rank (MPI_COMM_WORLD, &my_id);      /* get current process id */
    MPI_Comm_size (MPI_COMM_WORLD, &num_procs);  /* get number of processes */

    /*initialization */
    t0 = MPI_Wtime();
    for(i=0;i<N;i++){
        array[i]=my_id +1 ;
    }

    /* sum and multiplication */
    MPI_Reduce(array,array_final_sum,N,MPI_INT,MPI_SUM,0,MPI_COMM_WORLD);
    MPI_Reduce(array,array_final_mult,N,MPI_INT,MPI_PROD,0,MPI_COMM_WORLD);
    if(my_id == 0) {
        for(i=0;i<N;i++) printf("Final array after sum: %d\n", array_final_sum[i]);
    }
    if(my_id == 0) {
        for(i=0;i<N;i++) printf("Final array after product: %d\n", array_final_mult[i]) ;
    }

    /* random number generation */
```

```

    seed=my_id+1;
    srand(seed);
    r_num=rand();
    printf("my id is %d and r_num is %d\n", my_id,r_num);

/* calculus of maximum */
    MPI_Reduce(&r_num,&max_value,1,MPI_INT,MPI_MAX,0,MPI_COMM_WORLD);

/*sleep(10); */

/* time calculation */
    t1 = MPI_Wtime();
    time = t1 - t0 ;

    if(my_id == 0) {
        printf("Max_value is (AND THE WINNER IS ....): %d\n", max_value) ;
        printf("Total elapsed time [sec] : %f\n", time);
    }

    MPI_Finalize();
    return 0;
}

```

FORTRAN

```

PROGRAM main

    IMPLICIT NONE
    include 'mpif.h'

    INTEGER, parameter :: N=20
    INTEGER :: ierr, i, my_id, num_procs, seed!(2)
    INTEGER, dimension(N)::array
    INTEGER, dimension(N)::array_final_sum
    INTEGER, dimension(N)::array_final_mult
    REAL :: r_num, max_value
    INTEGER :: sizer

    DOUBLE PRECISION :: t0,t1,time

    CALL MPI_INIT( ierr )
    CALL MPI_COMM_RANK( MPI_COMM_WORLD, my_id, ierr )
    CALL MPI_Comm_size ( MPI_COMM_WORLD, num_procs, ierr )

    t0 = MPI_WTIME()

```

```

DO i=1,N
    array(i)= my_id+1
END DO

! Sum
CALL MPI_REDUCE(array, array_final_sum, N, MPI_INTEGER, MPI_SUM, 0 ,MPI_COMM_WORLD, ierr)

IF( my_id .eq. 0) THEN
    WRITE(*,*) " Final array after sum ", array_final_sum(:)
END IF

! Product
CALL MPI_REDUCE(array, array_final_mult, N, MPI_INTEGER, MPI_PROD, 0 ,MPI_COMM_WORLD, ierr)

IF( my_id .eq. 0) THEN
    WRITE(*,*) " Final array after product: ", array_final_mult(:)
END IF

! Random number generation
! CALL RANDOM_SEED(sizer)
seed=my_id+1
! CALL RANDOM_SEED(put=seed)
! CALL RANDOM_NUMBER(r_num)
call srand(seed)
r_num = rand()

WRITE(*,*) "my_id", my_id, " Random number :", r_num, "seed ", seed, "!" "sizer" ,sizer

! Search for the maximum value among generated random numbers...

CALL MPI_REDUCE(r_num, max_value, 1, MPI_REAL, MPI_MAX, 0 ,MPI_COMM_WORLD, ierr)

t1 = MPI_WTIME()
time = t1 - t0

IF( my_id .eq. 0) THEN
    WRITE(*,*) " Maximum generated random number :", max_value
    WRITE(*,*) " Total elapsed time [sec] : ", time
END IF

CALL MPI_FINALIZE(ierr)

END PROGRAM

```