

7 - дәріс. MPI-де хабар алмасу. MPI: Point-to-point блоктау. MPI: блоктаусыз point-to-point. MPI: I/O файлдарын басқару. Матрицаны -векторға көбейту, декомпозиция әдістері.

Жоспар:

- MPI-де хабар алмасу
- MPI: I/O файлдарын басқару
- Матрицаны векторға көбейту, декомпозиция әдістері.

MPI-де хабар алмасу

MPI-де хабар алмасу операциялары:

1. Нүктеден нүктеге алмасу (point-to-point)
Екі процесс қатысады – жіберуші процесс және қабылдаушы процесс.
2. Ұжымдық алмасу
Коммуникатордың барлық процестері қатысады.

Point-to-point хабар алмасу

MPI-дегі процестер арасындағы байланыстың негізгі механизмі өзара әрекеттесетін процестердің жұбы арасында деректерді беру болып табылады: біреуі жіберуші, екіншісі қабылдаушы. Бұл байланыс point-to-point деп аталады. MPI тасымалдауға мүмкіндік беретін деректерді беру және қабылдау функцияларының жинағын қамтамасыз етеді. Ол үшін тег ұғымы енгізіледі, оның көмегімен деректер типі деректермен және олардың ұсынымдарымен тікелей байланысады. Деректер типі туралы ақпарат компьютердің әртүрлі архитектураларында деректерді ұсынуды дұрыс түрлендіру үшін қажет.

Point-to-point операцияларының екі негізгі түрі бар:

- Блоктау
Блоктау операциялары олар орындайтын операция аяқталғанша процестің орындалуын тоқтатады (блоктайды).
- Блоктаусыз
Блокталмайтын операциялар басқаруды дереу қайтарады және операция фондық режимде жалғасады.
Сонымен қатар, осы типтердің әрбір жіберу операциясының қосымша түрлері бар:
- Буферленген
Әрекеттесудің буферлі әдісі қолданбалы бағдарлама арқылы буферді алдын ала айқын бөлуді қамтиды, оны жүйе жіберілген деректерді буферлеу үшін пайдаланады.
- Синхрондалған
Синхронды жіберуді тиісті қабылдағыш жіберілген деректерді қабылдауға дайын болғаннан кейін ғана бастауға болады, яғни сәйкес қабылдау функциясы іске қосылғанда.
- Дайын күйдегі
Тасымалдаудың бұл түрі сәйкес қабылдау функциясы іске қосылған жағдайда ұйымдастырылады. Әйтпесе, әрекет қате: оның нәтижесі белгісіз.

MPI: I/O файлдарын басқару

Параллель I/O (енгізу/шығару) нені білдіреді?

- Бағдарлама деңгейінде: Бірнеше процестерден ортақ файлға деректі қатар оқу немесе жазу

- Жүйе деңгейінде: Параллель файлдық жүйе және бір мезгілде қол жеткізуді қолдайтын аппараттық құрал

MPI-IO параллель енгізу/шығару үшін төмен деңгейлі интерфейсті қамтамасыз етеді. MPI-IO деректерді сақтау үшін файлдық жүйеге кіру жолын анықтайды. MPI-IO мақсаты - дискіге немесе одан MPI параллель бағдарламасына немесе одан деректерді алу.

MPI-I/O операциясының қадамдары:

1. Файлды ашу :MPI_File_open
2. Файлға жазу: MPI_File_write
3. Файлды жабу: MPI_File_close

MPI_File_open режимдері:

- MPI_MODE_RDONLY - Read only (Тек қана оқу)
- MPI_MODE_RDWR - Read write (Оқу және жазу)
- MPI_MODE_WRONLY - Write only (Тек қана жазу)
- MPI_MODE_CREATE – Жаңа файл құру
- MPI_MODE_EXCL – Құралып жатқан файл әлдеқайда бар болған жағдайда қателік белгісін беру
- MPI_MODE_DELETE_ON_CLOSE – Жабудан кейін файлды өшіру (уақытша файл)
- MPI_MODE_UNIQUE_OPEN - Файл басқа жерде ашылмайды (код арқылы немесе басқа жүйелерде (сақтық көшірмелер т.б.))
- MPI_MODE_SEQUENTIAL - Файлда файл көрсеткіші қолмен жылжытылмайды
- MPI_MODE_APPEND - Ашу кезінде файл көрсеткішін файлдың соңына жылжыту

Матрицаны векторға көбейту, декомпозиция әдістері

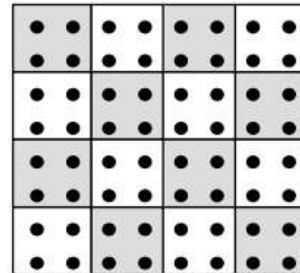
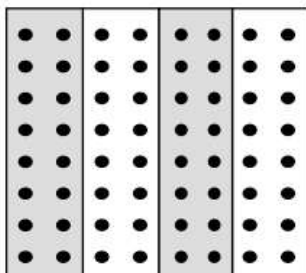
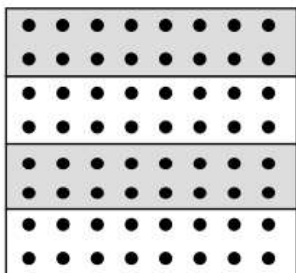
Көптеген матрицалық есептеу әдістері әртүрлі матрица элементтері үшін бірдей есептеу әрекеттерінің қайталануымен сипатталады. Бұл матрицалық есептеулерді орындау кезінде деректер параллелизмінің болуын және матрицаларды пайдаланылатын компьютерлік жүйенің процессорлары арасында бөлу(декомпозициялау) қажеттігін көрсетеді. Матрицаларды бөлу әдісін таңдау нақты параллельді есептеу әдісін анықтауға әкеледі; мәліметтерді бөлудің әртүрлі схемаларының болуы матрицалық есептеулер үшін бірқатар параллельді алгоритмдерді тудырады. Матрицаларды бөлудің ең кең тараған әдістері деректерді жолақтарға (тік немесе көлденең) немесе тікбұрышты фрагменттерге (блоктарға) бөлуден тұрады.

- Матрицаны жолақтарға бөлу әдісі

Жолақты бөлу кезінде әрбір процессорға матрицаның жолдар(rowwise) немесе бағандар (columnwise) бойынша белгілі бір ішкі жиыны бөлінеді (18-сурет).

- Матрицаны блоктарға бөлу әдісі

Блокты бөлу кезінде матрица элементтердің тікбұрышты жиындарына бөлінеді - бұл әдетте үздіксіз негізде бөлуді пайдаланады (19-сурет).



сурет. Матрицаны болктарға
бөлу әдісі

бөлу әдісі

Матрицаны векторға көбейту бағдарламасының мысалы (сақина топологиясы бойынша)

```
#include<stdio.h>
#include<stdlib.h>
#include<mpi.h>
#include<time.h>
#include <sys/time.h>
/* Матрицалардың өлшемін анықтау */
#define M 20
#define N 5
/* NUM_DIMS -декарттық топологияның өлшемі.*/
#define DIMS 1
#define EL(x) (sizeof(x) / sizeof(x[0][0]))
/*Бастапқы матрицалардың жолақтарын орнату*/
static double A[N][M], B[N], C[N];
int main(int argc, char **argv)
{
    int rank,size,i,j,k,i1,d,sour,dest;
    int dims[DIMS];
    int periods[DIMS];
    int new_coords[DIMS];
    int reorder = 0;
    MPI_Comm ring;
    MPI_Status st;
    int rt, t1, t2;
    /* MPI кітапханасын инициализациялау*/
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    /* dims массивін босатып, сақина топологиясы үшін periods массивін толтырамыз */
    for(i=0; i < DIMS; i++) { dims[i] = 0; periods[i] = 1; }
    /* dims массивін толтырамыз*/
    MPI_Dims_create(size, DIMS, dims);
    /* comm_cart коммуникаторымен сақина топологиясын құрамыз */
    MPI_Cart_create(MPI_COMM_WORLD, DIMS, dims, periods, reorder, &ring);
    MPI_Comm_rank(ring, &rank);
    MPI_Cart_shift(ring, 0, -1, &sour, &dest);
    for(j = 0; j < N; j++)
    {
        for(i = 0; i < M; i++)
        {
            A[j][i] = 3.0;    B[j] = 2.0;
            C[j] = 0.0;    }
    }
    /* Матрицаны көбейту уатының басталуын анықтаймыз*/
    t1 = MPI_Wtime();
    /* Әрбір тармақ өзінің матрицасы мен векторлық жолақтарын көбейтеді */
    for(k = 0; k < size; k++)
    {
        d = ((rank + k)%size)*N;
        for(j = 0; j < N; j++)
        {
            for(i1=0, i = d; i < d+N; i++, i1++)
            {
                C[j] += A[j][i] * B[i1];
            }
        }
    }
    /* Әрбір тармақ B векторының жолақтарын төменгі дәрежелі көршілес тармақтарына жібереді.
    /* Яғни B векторының жолақтары компьютерлердің сақинасы бойымен жылжиды */
    MPI_Sendrecv_replace(B, EL(V), MPI_DOUBLE, dest, 12, sour, 12, ring, &st);
    }
    /* Көбейту аяқталды. Әрбір тармақ A матрицасының жолдар жолағын B векторының барлық жолақтарына
    көбейтті. Уақытты белгілеп, нәтижені шығарамыз. */
    t2 = MPI_Wtime();
    rt = t2 - t1;
```

```

    printf("rank = %d Time = %d\n", rank, rt);
/* Басқару үшін нәтиженің бірінші N элементін басып шығарамыз*/
    for(i = 0; i < N; i++)
        printf("rank = %d RM = %6.2f\n", rank, C[i]);
/* Барлық тармақтар comm_cart топологиясымен байланысты жүйелік процестерді тоқтатады және
бағдарламаның орындалуы аяқталады*/
    MPI_Comm_free(&ring);
    MPI_Finalize();
    return(0);
}

```

Тапсырма:

Дәрісте көрсетілген C тілінде жазылған матрицаны векторға көбейту бағдарламасын Param–Bіlim суперкомпьютерінде орындаңыз.

Бақылау сұрақтары:

1. MPI-де хабар алмасу операциялары, нүктеден нүктеге алмасу мен ұжымдық алмасудың айырмашылығы неде?
2. Point-to-point операцияларының екі негізгі түрін атаңыз.
3. Параллель I/O (енгізу/шығару) нені білдіреді?
4. Матрицаларды декомпозициалаудың ең кең тараған екі әдісін атаңыз және принциптерін түсіндіріңіз.