6 - дәріс. Param-Bilim суперкомпьютерінде MPI технологияларымен жоғары өнімді есептеулер. Хабарды жіберу интерфейсі (MPI) дегеніміз не? Қарапайым параллель MPI бағдарламасының мысалы.

Мақсаты: Білім алушыларды MPI кітапханасымен таныстыру

Жоспар:

- MPI ұғымы
- MPI негізгі функциялары
- MPI хабарламаны тасымалдау деректерінің типі

MPI ұғымы

Параллельді бағдарламаларды құрудың қазіргі кездегі ең кең тараған тәсілі хабарларды жіберу кітапханасын пайдалану болып табылады, мұнда процестер басқа процестермен хабарламалар (ақпарат) алмасу үшін MPI (Message Passing Interface) кітапханасына жүгінеді.

MPI – АҚШ пен Еуропаның 40 ұйымнан 60 адамнан тұратын топ әзірлеген хабар алмасу стандарты. MPI стандарты төмендегі міндеттерді орындайтын 129 функцияны қамтиды:

- бағдарламаның параллель бөлігін инициализациялау және жабу;
- жеке процестер арасында хабарламаларды қабылдау және беру;
- процестердің ұжымдық әрекеттесуін жүзеге асыру;
- процестер топтарымен және коммуникаторлармен жұмыс;
- деректердің туынды (құрама) түрлерін анықтау;
- процестердің қарапайым өзара әрекеттесуін қамтамасыз ету үшін виртуалды
- топологияларды құру.

MPI-дің негізіне мынадай үш ұғым кіреді:

- процесс;
- процестер тобы;
- коммуникатор.

Процесс - бұл бағдарламаны MPI жүктелген бір процессор элементімен орындау. Процестер біртұтас байланыс аймағы бар топтарға біріктіріледі, оның ішінде әрбір процестің 0–ден N–1-ге дейінгі диапазонда өзіндік бірегей нөмірі болады, мұндағы N-топтағы процесс саны. Топтағы процестердің өзара әрекеттесуі үшін коммуникатор қолданылады, ол процестер арасында хабарламалар жіберуді (мәліметтер алмасуды) және оларды синхрондауды жүзеге асырады. Коммуникатор алмасуды өзі байланыстыратын топтың аймағында ғана жүзеге асырады.

Әдетте, параллель есептеу жүйелеріне арналған MPI бағдарламасы MPI кітапханасының байланыс функцияларын қолдана отырып, C/C++ немесе FORTRAN тілдерінде жазылады. MPI бағдарламасын іске қос - бұл орындалатын процесстер санын пайдаланушы анықтайтын, параллель процестер жиынтығын бір уақытта іске қосу. Бағдарламаның параллель бөлігі MPI инициализациясынан басталады. Бұл жағдайда барлық белсендірілген процестер mpi_comm_world атауы бар коммуникатормен топқа біріктіріледі. Әрі қарай, MPI көмегімен бағдарламаға іске қосылған процестердің саны беріледі, олардың әрқайсысына өз нөмірі беріледі. Бұл процестердің әрқайсысы параллель MPI бағдарламасының өз тармағын орындауға жауап беретіндігімен ерекшеленеді.

Келесі ережелерді орындау MPI бағдарламаларының көпшілігін құруды жеңілдетеді:

- Барлық MPI бағдарламаларында тақырып файлы болуы керек (C- бағдарламасында mpi.h; FORTRAN-да-mpif.h)

- Барлық MPI бағдарламалары MPI-ді инициализациялау үшін бірінші шақырылған MPI ішкі бағдарламасы ретінде MPI_INIT ішкі бағдарламасын шақыруы керек.
- MPI бағдарламаларының көпшілігі белсендірілген виртуалды машинаның өлшемін, яғни іске қосылған size процестерінің санын анықтау үшін MPI_COMM_SIZE ішкі бағдарламасын шақырады.
- MPI бағдарламаларының көпшілігі 0-ден size-1-ге дейінгі диапазондағы әрбір белсендірілген процестің нөмірін анықтау үшін MPI_COMM_RANK шақырады.
- Әрбір MPI бағдарламасында орындалу сипаты шарт бойынша өзгеруі мүмкін процестер іске қосылады. Процестердің өзара әрекеттесуі MPI_SEND және MPI_RECV функцияларын шақыру арқылы жеке процестер арасында хабарламалар жіберу арқылы жүзеге асырылады.
- Барлық MPI бағдарламалары MPI кітапханасындағы соңғы функция шақыруы ретінде MPI_FINALIZE пайдалануы керек.

Төмендегі мысалда C тілінде жазылған параллель MPI "Hello world" бағдаламасы көрсетілген:

```
#include <stdio.h>
#include <mpi.h>
main(int argc, char **argv)
{
 int rank, size, tag, rc, i;
 MPI_Status status;
 char message[20];
 rc = MPI_Init(&argc, &argv);
 rc = MPI_Comm_size(MPI_COMM_WORLD, &size);
 rc = MPI_Comm_rank(MPI_COMM_WORLD, &rank);
 tag = 8;
 if (rank == 0) {
 strcpy(message, "Hello, world!");
 for (i=1; i<size; i++)
 rc = MPI_Send(message, 13, MPI_CHAR, i, tag,
MPI_COMM_WORLD);
 }
 else
 rc = MPI_Recv(message, 13, MPI_CHAR, 0, tag,
MPI_COMM_WORLD, &status);
 printf( "process %d : %s\n", rank, message);
 rc = MPI_Finalize();
}
```

MPI хабарламаны тасымалдау деректерінің типі

Кесте – 1. С тіліне сәйкес MPI деректер типтері

MPI_Datatype	C тіліндегі типтер
MPI_BYTE	
MPI_CHAR	signed char
MPI_DOUBLE	Double
MPI_FLOAT	Float
MPI_INT	Int
MPI_LONG	Long
MPI_LONG_DOUBLE	long double
MPI_PACKED	
MPI_SHORT	Short
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long
MPI_UNSIGNED_SHORT	unsigned short

Деректер типтерін жасаудың жолдары:

- *Үздіксіз:*бұл жаңа типті жобалаудың ең қарапайым тәсілі. Ең қарапайым мысал - белгілі бір типтегі элементтер жиыны;
- *Вектор:* бұл жалпылама үздіксіз типі,ол өрістердегі тұрақты үзілістерге мүмкіндік береді. Элементтер түпнұсқа деректер түрінің ұзындығының аралықтарымен бөлінеді.Мысалы, матрицаны екі өлшемді массив ретінде анықтаған кезде матрицаның бағандары;
- *Н-векторы:*векторға ұқсас, бірақ элементтер арасындағы қашықтық байтпен анықталады;
- *Индекс:*осы типтіқұру кезде, түпнұсқалық деректердің өрістерінің жиынын пайдаланылады, ауыстыру бастапқы деректер түрінің ұзақтығы бойынша өлшенеді;
- *Н-индексі:*индекске ұқсас, бірақ байттармен анықталады;
- *Құрылымдық:*жалпы сипаттама береді. Іс жүзінде, бастапқы дәлелдер негізгі MPI түрлерінен тұрса, көрсетілген құрылым - дәл құрылатын типтің картасы.

Кесте – 2. Деректер типтерін құру үшін MPI функциялары

Деректер типі	**Функция**	**Сипаттама**
Векторлық иллюстрациялар	int MPI_Type_vector(int count, int blocklen, int stride, MPI_Datatype old_type, MPI_Datatype *newtype);	Екі түрдегі элементтері бар вектор x = (x1, ..., x20). Тек біркелкі элементтерді пайдаланыңыз. MPI_Type_vector (10, 1, 2, MPI_DOUBLE және жаңа түрі).
Н-векторы	intMPI_Type_hvector(int count, intblocklen, MPI_Aint stride, MPI_Datatypeold_type, MPI_Datatype *newtype);	Құрылыстың векторлық әдісіне ұқсас, бірақ элементтер арасындағы қашықтық байтпен анықталады.
Индекс	intMPI_Type_indexed(int count, intblocklens[], int indices[], MPI_Datatypeold_type, MPI_Datatype *newtype);	Осы типтіқұрған кезде, түпнұсқалық дерек-тер типінің қалдықтары пайдаланылады; Есепшот бірлігі бастапқы деректер ауқымы-на сәйкес келеді.

H-индексі	intMPI_Type_hindexed(int count, intblocklens[], int indices[], MPI_Datatypeold_type, MPI_Datatype *newtype);	Бұл индекс сияқты, бірақ офсет байттармен анықталады.
Құрылымдық	intMPI_Type_struct(int count, intblocklens[], MPI_Aint offsets[], MPI_Datatype types[], MPI_Datatype *newtype);	MPI типінің картасы пайдаланылады: · Count - типтегі элементтердің саны; · Blocklens [] - типтегіэлементтердегі мәндердің саны; · Offsets [] - элементтің басынан бастап жылжуы; · Типтер [] - типтер элементтері үшін мәндердің түрлері; · Newtype - типке сілтеме үшін жаңа айнымалының атауы.

Тапсырма:

1. Дәрісте көрсетілген C тілінде жазылған параллель MPI “Hello world” бағдаламасын Param–Bilim суперкомпьютерінде орындаңыз.
2. Param–Bilim суперкомпьютерінде массивтің элементтерінің қосындысын есептеу үшін MPI қолданатын параллельді бағдарламасын жазыңыз.

Бақылау сұрақтары:

1. MPI стандартының функциялары орындайтын міндеттерді атаңыз.
2. MPI-дің негізіне кіретін қандай үш ұғым бар?
3. Қандай деректер типтерін жасаудың жолдары бар?