

12 -Дәріс

Таырыбы: MPI негізінде параллель бағдарламалау

Жоспар

1. MPI ұғымы туралы
2. MPI деректерді тасымалдау операциясының типтері
3. MPI хабарламаны тасымалдау деректерінің типі
4. Коммуникатор түсінігі(процессорлар тобы)

Мақсаты: Білім алушыларды MPI кітапханасымен таныстыру

Кілттік сөздер: Параллельді есептеуде MPI моделі, MPI_Init, MPI_Comm_size, MPI_Comm_rank, MPI_Send, MPI_Recv, MPI_Finalize.

Оқытудың әдістемесі мен формасы: Баяндау, дәріс

1.MPI (Message Passing Interface) ұғымы туралы

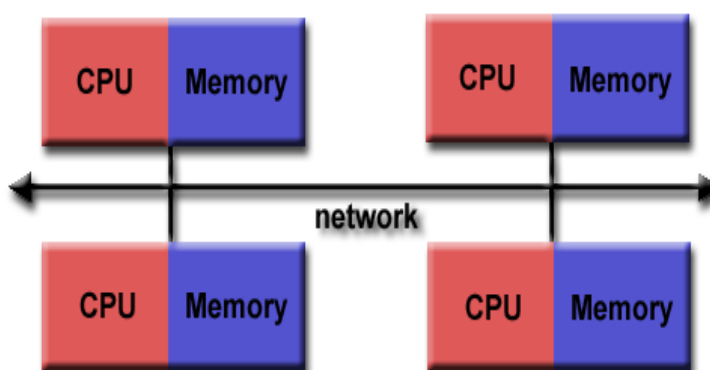
Бөлінген жады жүйелері үшін параллель есептеулерді ұйымдастыру өзара әрекеттесетін процессорлар арасында деректерді берудің әр түрлі әдістерін пайдалану мүмкін. Хабар алмасу моделінің сипатын анықтау және деректерді беру фактісі болып, бір процестің жергілікті жадыдан басқа процестің жергілікті жадыға дейін екіжақты операцияларды (қабылдау және беру) орындалуын талап етуі. Параллель бағдарламаларда пайдаланылатын негізгі ұғымдарды стандарттау және қолданылатын құралдар әзірлеу үшін деректер үлгісін пайдалану қажет. Мұндай әрекеттер жеке аспаптық коммуникациялық кітапханалардың дамуынан кейін бірігуі 1994 жылы деректерді беру интерфейсі стандартының пайда болуына әкелді.

MPI (хабар жіберу интерфейсі)- параллельді компьютерлер үшін революциялық жаңа бағдарламалау әдісі емес, бұл әрекет көптеген хабар алмасу жүйелерінің қажет болса ең жақсы қасиеттерін жинап, оларды стандарттайды.

MPI стандарты UNIX/Linux және Windows жұмыс станцияларына негізделген кластерлер үшін де қол жетімді болып табылады. Қазіргі уақытта MPI-кеңінен қолданылатын және қарқынды дамып келе жатқан интерфейс. MPI технологиясын қолданып кластерде жұмыс жасау үшін Windows операциялық жүйесінде кластерге қатнау үшін putty немесе WinSCP бағдарламалары қажет болады. Linux операциялық жүйесінде кластерде жұмыс жасау командалық

жолдармен де жұмыс жасай беруше болады. MPI деректерді тасымалдау кітапханаларын дамыту үшін негіз болып табылады. Fortran және C-да жазылған бағдарламалар стандартты компиляторлар және MPI кітапханасымен байланысты жасалады. Сонымен қатар MPI хабар алмасу үлгісін білдіреді. MPI пайдалану арқылы жазылған параллельбағдарлама хабар алмасу үрдісінің жиынтығынан тұрады.

Message Passing Programming үлестірілген адрестік кеңістікте параллель есептеледі, 35-сурет (MPI -хабар жіберу интерфейсі. Блез Барни, <https://computing.llnl.gov/tutorials/mpi/>)



Сурет 35-Хабарламаны жіберу интерфейсі (MPI)

MPI-дің негізіне мынадай үш мағлұмат кіреді:

- хабарламаны тасымалдау операцияларының типі;
- хабарламаны тасымалдау деректерінің типі;
- коммуникатор түсінігі(процессорлар тобы).

Осыдан MPI-дің негізгі оң аспектілерін атап айтсақ:

- MPI-тасымалдау мәселенің маңыздылығын айтарлықтай төмендетуге мүмкіндік береді жәнәтүрлі компьютерлік жүйелер арасында қатарлас бағдарламалардың MPI кітапханасын пайдалану арқылы жасалған C немесе Fortran алгоритмдік тілдерде әзірленген параллель бағдарламаның тасымалдануы, әдетте, кез-келген есептеу платформаларында жұмыс істейді.

- MPI- өнімділікті арттыру параллель есептеулерде тиімділікті арттыруға көмектеседі, себебі қазіргі уақытта есептеудің барлық түрлеріне компьютерлік техниканың максималды мүмкіндіктерін ескере отырып жасалады.

1. MPI деректерді тасымалдау операциясының типтері

MPI деректерді тасымалдау операциясының функциялары мынадай қызметтер бойынша ажыратылады:

- a) Жұптық(point-to point)-екі процесс арасындағы операциялар.
- b) Коллективті(collective)-бір уақытта бірнеше процесстер арасындағы байланысу операциялары.

a. *Жұптық*(point-to point)-екі процесс арасындағы операциялардың мысалы ретінде MPI_Send және MPI_Recv операциялары жатады. Ол біз қарастыратын «процесс-процесс» типті алмасудың негізгі операциялары болып табылады. Дегенмен, оларды қолдану «нақты» бағдарламаларда, белгілі қиындықтарға кездеседі, алынған параллель бағдарламаларды (көп жағдайда) төменгі өнімділікке әкеледі. Бар мәселе, деректерді жеткізудің стандартты өнімдерін қолдану кезінде хабарламаларды алмасу қай бейнеде болатынына байланысты.

b. *Коллективті*(collective)-осы уақытқа дейін екі процесті қамтитын өзара әрекеттесулер қарастырылды - біреу хабарды жібереді, екіншісі қабылдайды. MPI бірнеше (екіден көп) процестердің қатысуын қамтитын өзара әрекеттесудің күрделі түрлерін қолдайды. Мұндай операциялар коллективті деп аталады. Операциялар MPI негізіндегі бағдарламалар әзірлеуде кеңінен қолданылады. Бұл мұндай операциялардың табиғи сипатына ғана емес (көбіне пайдаланылатын алгоритмдер деректердің барлық есептеу процестеріне таратылуын немесе керісінше, түбірлік процесте есептелген деректерді құрастыруды болжайды), сондай-ақ мұндай операциялардың тиімділігі жоғары.

MPI операциялар типтерінің қызметтері:

Стандартты беру.

Ол алынған немесе алынбағанына қарамай, хабарлама жіберілгеннен кейін, ол толық болып саналады. Хабарламаларды қабылдау басталмаса да, оларды жіберу басталады.

Синхронды беру.

Хабар келмейінше аяқталмайды.

Буферлі беру.

Бірден аяқталады. Буферлік кезде ауысуды басқару дереу қоңырау шалу рәсіміне оралады. Жүйе буферіне және жіберуді күтуге хабарлама бір мезгілде көшіріледі.

Дайындық бойынша беру.

Хабарламаны қабылдау басталғанда басталады, қабылдаудың аяқталуын күтпей аяқталады.

Төрт режимде блогталған және блогталмаған жағдайда орындалуы мүмкін. Осылайша, MPI-де трансмиссиялық операциялардың 8 түрі бар.

MPI қолданылатын операция түрін анықтауға мүмкіндік беретін рәсімдердің атаулары туралы белгілі бір келісім бар. Рәсімнің атауы келесі ережеге сәйкес жасалған:

MPI_[I][R|S|B]Send,

Мұндағы,

- I (Immediate) - блогтау операциясын білдіреді;
- R – дайындығы;
- S – синхронды;
- B – буферлі.

Префикстің болмауы стандартты режим қолданылғанын білдіреді. Кез келген типтің қабылдау функциясы (MPI_XXRecv) хабарламаны қабылдау мүмкіндігіне ие екендігін атап өткен жөн.

Нақтылап айтқанда, MPI_Send функциясы стандартты блогтау жіберуін орындайды. Процесті хабарламаны беру әрекеті аяқталғанға дейін бұғаттайды (бұл қабылдаудың аяқталуына кепілдік бермейді!), бұл үлкен деректер жиынтығын жібергенде өте тиімсіз, өйткені процессор есептеулерді орындамайды.

int MPI_Send(void *buf, count, type, dest, tag, Comm)

Кесте 1-MPI_Send аргументтері

Аргумент	Тип	сипаты
Buf		Жіберетін хабарламаның деректері орналасатын жадының алмасу адресі
count	int	Жіберілетін хабарлама элементтер саны
Type		Жіберілетін массив элементінің типі
Dest	int	Процесс қабылдаушының номері(рангі)
Tag	int	Хабарламаны идентификациялауға арналған тэг
Comm	MPI_Comm	Деректерді жіберуге арналған коммуникатор

Бұл құрылым **MPI_Recv** функциясында ұқсас қызметтермен қолданылады:

Кесте 2- **MPI_Recv** аргументтері

Аргумент	Тип	сипаты
Buf		Хабарламаны қабылдауға арналған алмасу буферінің бастапқы адресі
count	int	Қабылданатын хабарлама элементтер саны
Type		Қабылданатын массив элементінің типі
Dest	int	Процесс жіберушінің номері(рангі)
Tag	int	Хабарламаны идентификациялауға арналған тэг
Comm	MPI_Comm	Деректерді алмасуға арналған коммуникатор
status	MPI_Status	Қабылданған хабарлама туралы ақпараттан тұратын құрылым

Мысалы: MPI хабарламасын жіберу және қабылдау

1-есеп жібереді:

```
int buf[10];
MPI_Send(buf, 5, MPI_INT, 1, 0, MPI_COMM_WORLD );
```

2-есеп қабылдайды:

```
int buf[10];
MPI_Status status;
MPI_Recv(buf, 10, MPI_INT, 0, 0, MPI_COMM_WORLD, &status );
```

Сурет 36-MPI хабарламасын жіберу және қабылдау мысалы

Осылайша, пайдаланушыларға хабарды жіберуді (алуды) және басқа операцияларды орындауды жалғастыруға болады[20].

2. MPI хабарламаны тасымалдау деректерінің типі

MPI хабарламасының өту функцияларын қарастырған кезде, осы функциялардың дәлелдерінің бірі жіберілген және алынған хабарлар үшін деректер түрі болып табылады. Бұған дейін талқыланған мысалдарда тек қарапайым деректер типтері қолданылған, ал пайдаланылатын бағдарламалау тіліндегі негізгі деректер типтері-бүтін сандарға, қалқымалы нүктелерге және т.б., сондай-ақ олардың массивтеріне сәйкес келеді. Бұл бөлімде негізгі түрлердің барлығын талқылайтын боламыз, сонымен қатар, MPI-нің туынды түрлері мен құру жолдарын қарастырамыз.

MPI қарапайым деректер типтері кең ауқымды ұсынады. Бұл жиын C тілінің түрлеріне сәйкес келетін барлық негізгі тип түрлерін, сондай-ақ MPI үшін нақты деректердің екі түрі бар: MPI_BYTE және MPI_PACKED. MPI_BYTE байтты сипаттайды, ол 8 бинарлық сан ретінде анықталады.

Бұл түрі MPI_CHAR және MPI_CHARACTER сияқты емес. Біріншіден, MPI_BYTE ауқымындағы сандарды [0..255] көрсетеді, ал char іске қосылғанда 16-бит болуы мүмкін. Екінші себеп - бұл гетерогенді ортада әріптер әртүрлі 18 кодтауға ие болуы мүмкін.

Мысалы, EBCDIC жүйесін пайдалануға қарағанда, ASCII жүйесін пайдалансақ A әрпінің әртүрлі көрінісі бар болады. 1-кестеде MPI негізгі түрлерінің сәйкестігі және Си тілінің деректер типтері көрсетілген.

Кесте 3-C тіліне сәйкес MPI деректер типтері мынадай

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

MPI деректер типі - бұл объект осы түрлердің әрқайсысының байттарындағы негізгі деректер типтерінің тізбегін және өрістерді анықтайды. Мұндай ауысулар деректер типін сипаттайтын буферге қатысты саналады. Біз деректер типін негізгі түрлер мен жұптардың тізбегі ретінде ұсынамыз, MPI-де бұл тізбектің түрі карта деп аталады (typemap).

Тип картасы - ерікті деректер типтерін сипаттаудың жалпы тәсілі. Дегенмен, егер нәтиже типінің картасы көп элементтерді қамтыса ыңғайсыз болуы мүмкін. MPI типті картаны нақты жасамай, деректер типтерін жасаудың бірнеше жолдары ұсынылған:

- *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 типті құру 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.

Үздіксіз деректер түрін жасау туралы мысал қарастырайық. Үздіксіз типі MPI_Type_contiguous функциясы арқылы деректердің жаңа типін генерациялау жәнәтүпнұсқалық типтің ескі көшірмелерін ескі түрдегі ұзындықпен көбейте отырып жасауға болады. Мысалы, егер бастапқы деректер түрі (oldtype) {(int, 0), (қос, 8)} типті карта болса, онда MPI_Type_contiguous (2, oldtype, & newtype); (екі, 8), (int, 16), (екі есе, 24) түріндегі картамен жаңа типті деректер түрін жасайды.

Count аргументтің қолдану MPI рәсімдерінде үздіксіз деректер типін пайдаланумен бірдей мөлшерде балама болып табылады ма [21].

MPI_Send (buffer, count, datatype, dest, tag, comm)

Оның балама түрі:

MPI_Type_contiguous(count, datatype, &newtype);

MPI_Type_commit(&newtype);

MPI_Send(buffer, 1, newtype, dest, tag, comm);

MPI_Type_free(&newtype);

Кесте 4- 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	int MPI_Type_hvector	Құрылыстың векторлық әдісіне ұқсас, бірақ элементтер

	(int count, intblocklen, MPI_Aint stride, MPI_Datatypeold_type, MPI_Datatype *newtype);	арасындағы қашықтық байтпен анықталады.
Индекс	intMPI_Type_indexed (int count, intblocklens[], intindices[], MPI_Datatypeold_type, MPI_Datatype *newtype);	Осы типті құрған кезде, түпнұсқалық деректер типінің қалдықтары пайдаланылады; Есепшот бірлігі бастапқы деректер ауқымына сәйкес келеді.
Н-индексі	intMPI_Type_hindexed (int count, intblocklens[], int indices[], MPI_Datatypeold_type, MPI_Datatype *newtype);	Бұл индекс сияқты, бірақ офсет байттармен анықталады.
Құрылым- дық	intMPI_Type_struct (int count, intblocklens[], MPI_Aint offsets[], MPI_Datatype types[], MPI_Datatype *newtype);	MPI типінің картасы пайдаланылады: · Count - типтегі элементтердің саны; · Blocklens [] - типтегі элементтердегі мәндердің саны; · Offsets [] - элементтің басынан бастап жылжуы; · Типтер [] - типтер элементтері үшін мәндердің түрлері; · Newtype - типке сілтеме үшін жаңа айнымалының атауы.

3. Коммуникатор түсінігі (процессорлар тобы)

MPI_Comm_rank функциясы коммуникатор қызметін орындайды.



Сурет 37-Коммуникатор

MPI кемшіліктері:

- тізбектеуден параллель нұсқаға өту үшін бағдарламалау айырмашылықтарының көп болу;
- баптаудың күрделігі;
- жұмыс торабы арасындағы желі байланысының шектеуі.

Тапсырма:

MPI хабарламасын жіберу және қабылдау кодын жазыңыз.

Бақылау сұрақтары:

1. MPI қолданылатын типтерді атаңыз.
2. MPI_Send функциясының қызметін атаңыз?
3. MPI_Recv функциясының қызметін атаңыз?