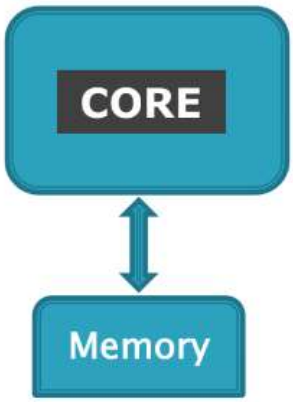
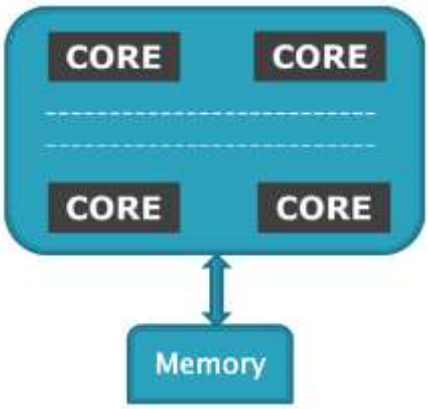
8 - дәріс. Param-Bilim суперкомпьютерінде OpenMP технологияларымен жоғары өнімді есептеулер.

Жоспар:

1. OpenMP ұғымы
2. Орындалу моделі
3. OpenMP директивалары мен фунциялары
4. Бағдарламаны компиляциялау және орындау

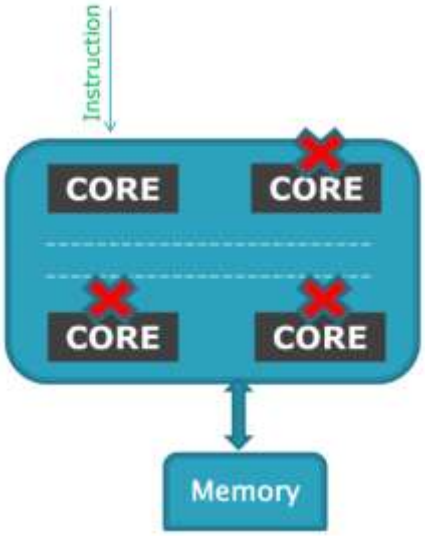
OpenMP ұғымы

Негізгі архитектура:

Процессор	Процессор
	
Ескі процессорда нұсқауларды орындау үшін тек бір процессорлық ядро болды	Процессорларда нұсқауларды орындау үшін 4 немесе одан да көп тәуелсіз CPU ядролары бар

Программаның тізбектей орындалуы

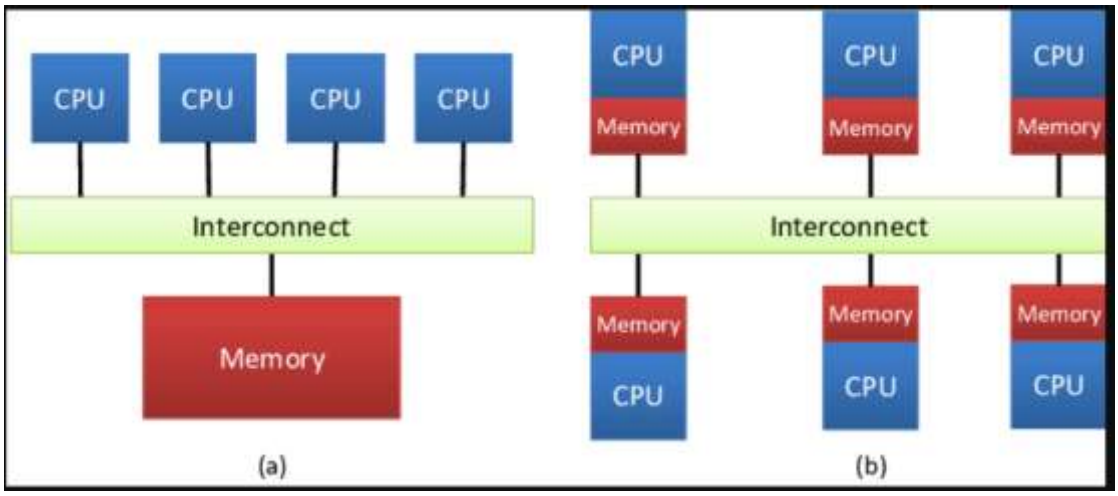
Тізбекті бағдарламаны іске қосқанда бір ғана ядро жұмыс жасап, басқа ядролар жұмыс істемейді. Бұл қолда бар ресурстарды қолданбау болып саналады. Ал параллель бағдарламалау осы ядролардың барлығын жұмысқа қосуға мүмкіндік береді.



20 – сурет. Программаның тізбектей орындалу схемасы

Параллель бағдарламалаудың көптеген модельдері бар. Олардың ішінде ең көп таралғандары: бөлінген жады(distributed memory) және ортақ жады(shared memory).

a) Ортақ жад жүйелері - жалпы жад кеңістігіне қол жеткізе алатын жалғыз немесе бірнеше процессорларды пайдаланатын HPC жүйелері. Бұл барлық деректер мен нұсқаулар бір жерде сақталады және оларға кез келген процессор қол жеткізе алады дегенді білдіреді.
b) Бөлінген жад жүйелері - бұл өздерінің жергілікті жады бар және желі арқылы бір-бірімен байланысатын бірнеше процессорларды пайдаланатын HPC жүйелері. Бұл әрбір процессордың өз деректері мен нұсқаулары бар екенін және басқа процессорлардың деректеріне тек хабарламаларды жіберу және қабылдау арқылы қол жеткізе алатынын білдіреді.



21 – сурет. а) Ортақ жад жүйелері; б) Бөлінген жад жүйелері

OpenMP (Open Specification for Multi Processing) - дәстүрлі бағдарламалау тілдеріне негізделген ортақ жадты компьютерлерге арналған ең танымал технология болып табылады. Негіз ретінде тізбекті бағдарлама алынады және оның параллель нұсқасын жасау үшін пайдаланушыға директивалар, функциялар және орта айнымалылар жиынтығы беріледі. Жасалған параллель бағдарлама OpenMP API-ін қолдайтын әртүрлі ортақ жад компьютерлері арасында тасымалданатын болады.

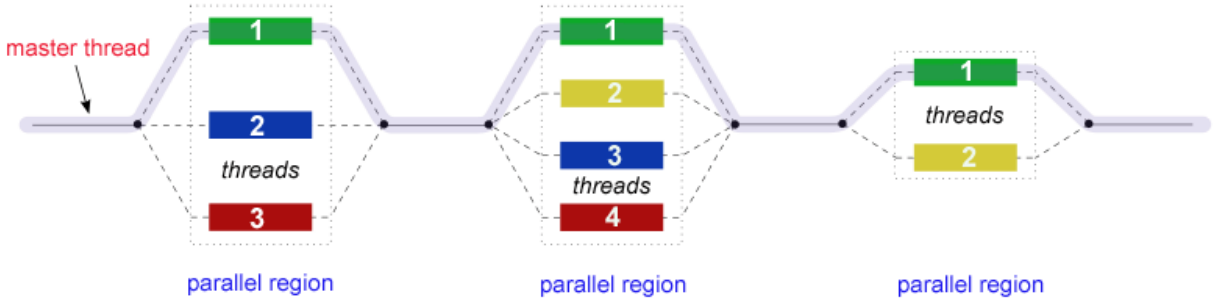
OpenMP артықшылықтары:

- Портативті
  - ортақ жад архитектуралары үшін стандартталған
- Қарапайым және жылдам
  - Бір уақытта қолданбаның кішігірім бөліктері үшін параллелизацияны орындау салыстырмалы түрде оңай жүзеге асырылады
  - Қосымша параллелизация
- Ықшам қолданбалы бағдарлама интерфейсі(API)
  - Директивалардың қарапайым және шектеулі жиынтығы

Орындалу моделі

OpenMP технологиясы пайдаланушыда параллельді және тізбекті орындау үшін бағдарламаның бір ортақ нұсқасы болуын қамтамасыз етуге бағытталған. OpenMP Fork/Join(Айыру және біріктіру) моделі бойынша жұмыс жасайды:

- OpenMP бағдарламасы бір ағынды бастайды
- Қосымша ағындарды жасау үшін пайдаланушы параллель аймақты бастайды
- Қосымша ағындар іске қосылып, команда құрады
- «Басты»(master) ағын «тәуелді»(slave) ағындар жинағын жасайды және тапсырма олардың арасында бөлінеді.
- Параллель аймақтың соңында ағындар біріктіріледі



22 – сурет. Fork/Join(Айыру және біріктіру) моделі

Бірнеше процессоры бар машинада ағындар параллель орындалады деп болжанады және процессорлар саны ағындар санынан үлкен немесе оған тең болуы шарт емес.

OpenMP директивалары мен фунциялары

OpenMP функциясының көп бөлігі компилятор директивалары арқылы жүзеге асырылады. Бағдарламаның параллель жұмыс істеуіне мүмкіндік беру үшін оларды пайдаланушы нақты енгізуі керек. Fortran бағдарламаларындағы OpenMP директивалары түсініктемелермен пішімделеді және !OMP, COMP немесе *$OMP таңбаларының тіркесімімен басталады, ал Си тілінде препроцессорға нұсқаулар - #pragma omp-ден басталады. Omp кілт сөзі OpenMP директивалық атауларының басқа атауларға кездейсоқ сәйкес келуін болдырмау үшін пайдаланылады.

C/C++ тіліндегі директива пішімі:
#pragma omp directive-name [опция, опция...]
Fortran тіліндегі директива пішімі:
!$OMP directive-name [опция, опция...]
C$OMP directive-name [опция, опция...]
*$OMP directive-name [опция, опция...]

Барлық OpenMP директиваларын 3 категорияға бөлуге болады: параллель аймақты анықтау, жұмысты бөлу, синхрондау. Әрбір директиваның бірнеше қосымша атрибуттары, яғни опциялары (тармақ) болуы мүмкін.

OpenMP орындау кітапханасының функцияларын пайдалану үшін бағдарламаға omp.h тақырып файлын қосу керек (Fortran бағдарламалары үшін - omp_lib.h файлы немесе omp_lib модулі). OpenMP бағдарламасында қолданылатын барлық функциялар omp_ префиксінен басталады.

Бағдарламаны компиляциялау және орындау

OpenMP технологиясын пайдалану үшін тиісті кілтті көрсете отырып, OpenMP-ді қолдайтын компилятормен бағдарламаны компиляциялау қажет:

gcc –fopenmp <бағдарлама аты> –o <execcutable>

gfortran –fopenmp < бағдарлама аты > –o <execcutable>

ifort < бағдарлама аты > -qopenmp –o <execcutable>

icc < бағдарлама аты > -qopenmp –o <execcutable>

Орындалатын файлды алғаннан кейін оны процессорлардың қажетті санында іске қосу керек. Мұны істеу үшін әдетте OMP_NUM_THREADS ортасының айнымалы мәнін орнату арқылы бағдарламаның параллель аймақтарын орындайтын ағындар санын орнату қажет. Мысалы, команда жолында мұны келесі пәрмен арқылы жасауға болады:

> export OMP_NUM_THREADS=4

> ./name

> time ./name

Іске қосылғаннан кейін бір ағын жұмыс істей бастайды және параллель аймақтарда бір бағдарлама ағындардың барлық жиынтығымен орындалады. Бағдарламаның стандартты шығысы жүйеге байланысты терминалға шығарылады немесе алдын ала анықталған аты бар файлға жазылады.

Бірінші OpenMP бағдарламасы: Hello World!

C тілінде:

```
#include <omp.h>
int main() {
int id;
#pragma omp parallel private(id)
{
        id = omp_get_thread_num();
        if (id%2==1)
        printf("Hello world from thread %d, I am odd\n", id);
        else
        printf("Hello world from thread %d, I am even\n", id);
   }
}
```

Тапсырмалар:

1. Сіздің жүйеңіз OpenMP стандартының қандай нұсқасын қолдайтынын анықтаңыз.
2. Кез келген тізбекті бағдарламаны OpenMP технологиясын қосып, бірнеше ағында орындауға жіберіңіз. Бұл бағдарламаның операторларын нақты қанша ағын орындайды?

Бақылау сұрақтары:

1. Бөлінген жады (distributed memory) және ортақ жады(shared memory) модельдерінің айырмашылығы қандай?
2. OpenMP дегеніміз не? Оның артықшылықтары неде?
3. Fork/Join(Айыру және біріктіру) моделінің жұмыс істеу принципін сипаттаңыз.