

ЭТАПЫ И ПРИМЕР (МММ) ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ЗЕРНИСТЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ И ОБМЕНА ДАННЫМИ

Этапы получения псевдокода параллельного зернистого алгоритма для реализации на суперкомпьютерах с распределенной памятью (исходный последовательный алгоритм задан гнездом циклов, автоматизация распараллеливания не используется):

- Информационная структура алгоритма: выявление (не обязательно формализованное) информационных зависимостей между операциями.
- Тайлинг (не нарушающий порядок выполнения зависимых операций).
- Запись параллельных зернистых вычислительных процессов (без распределения массивов между процессами и указания обменных операций): псевдокод уровня глобальных циклов, псевдокод уровня операций тайла. Детальное понимание распределения вычислений.
- Распределение входных и выходных данных (следует из распределения вычислений).
- Общее не формализованное представление о работе параллельного алгоритма, об обмене данными и выводе результатов вычислений.
- Выделение массивов. Приватизация (если возможно) массивов.
- Запись (псевдокод) тайла с выделенными массивами.
- Оптимизация вычислений в тайлах (например, введение новых массивов, оптимизация работы с кэшами, вычисление границ цикла вне цикла).
- Детальное понимание коммуникаций. Структурирование коммуникаций (например, бродкаст, трансляция).
- Псевдокод параллельного зернистого алгоритма, включающий пересылку процессам входных данных, коммуникационные операции, вывод результатов вычислений.

Пример: параллельный алгоритм перемножения квадратных матриц (варианты 1 и 2 ЛабМММ)

Дан ijk -алгоритм перемножения двух квадратных матриц порядка N :

```
do  $i = 1, N$ 
  do  $j = 1, N$ 
     $S_1(i,j): c(i,j) = 0$ 
    do  $k = 1, N$ 
       $S_2(i,j,k): c(i,j) = c(i,j) + a(i,k) b(k,j)$ 
    enddo
  enddo
enddo
```

Требуется разработать параллельные алгоритмы согласно варианту 1 и варианту 2 (варианты отличаются только коммуникационными операциями).

Тайлинг: $r_1 = 1$, цикл i глобальный не разбиваемый,

$$Q_2 - \text{параметр, } r_2 = \left\lceil \frac{N}{Q_2} \right\rceil,$$

$r_3 = N$; цикл k локальный не разбиваемый;

s -координата: j ;

коммуникации (вар. 1): трансляция части (согласованной с тайлом) A ,

коммуникации (вар. 2): бродкаст части (согласованной с тайлом) A .

Бродкаст (одновременное распространение) – это передача данного группы процессоров, в которых данное одновременно (на одной итерации) используется как аргумент. Трансляция – это передача данного от процессора к процессору в случае, если элемент массива используется в разных процессорах по очереди.

Рассмотрим этапы получения псевдокода.

Информационная структура алгоритма. Зависимости алгоритма задаются функциями

$$\begin{aligned} \overline{\Phi}_{1,2}(i,j,1) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}, \quad V_{1,2} = \{(i,j,k) \in Z^3 \mid 1 \leq i \leq N, 1 \leq j \leq N, k=1\}, \\ \overline{\Phi}_{2,2}(i,j,k) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad V_{2,2} = \{(i,j,k) \in Z^3 \mid 1 \leq i \leq N, 1 \leq j \leq N, 2 \leq k \leq N\}. \end{aligned}$$

Тайлинг. Цикл с параметром i является глобальным не разбиваемым, цикл с параметром k является локальным не разбиваемым (в задании так сказано). Разобьем цикл с параметром j . Через Q_2 обозначено число итераций в глобальном (т.е. в первом новом) цикле, а через r_2 обозначено (наибольшее)

число итераций в локальном (т.е. во втором новом) цикле; $r_2 = \left\lceil \frac{N}{Q_2} \right\rceil$. Получим

```

do  $i^{gl} = 1, N$ 
   $i = i^{gl}$ 
  do  $j^{gl} = 0, Q_2 - 1$ 
    do  $j = 1 + j^{gl}r_2, \min((j^{gl}+1)r_2, N)$ 
       $S_1:$      $c(i,j) = 0$ 
            do  $k = 1, N$ 
               $S_2:$      $c(i,j) = c(i,j) + a(i,k) b(k,j)$ 
            enddo
          enddo
        enddo
      enddo

```

Операторы S_1 и S_2 окружены одним и тем же набором глобальных циклов, поэтому имеется тайл только одного типа:

```

do  $i^{gl} = 1, N$ 
  do  $j^{gl} = 0, Q_2 - 1$ 
    Tile( $i^{gl}, j^{gl}, 0$ )
  enddo
enddo

```

где Tile($i^{gl}, j^{gl}, 0$) имеет вид

```

 $i = i^{gl}$ 
do  $j = 1 + j^{gl} r_2, \min((j^{gl} + 1) r_2, N)$ 
 $S_1:$     $c(i, j) = 0$ 
        do  $k = 1, N$ 
 $S_2:$     $c(i, j) = c(i, j) + a(i, k) b(k, j)$ 
        enddo
      enddo

```

Обоснуем корректность тайлинга (для любого варианта). Имеются зависимости $S_1(i, j) \rightarrow S_2(i, j, 1)$, $S_2(i, j, k-1) \rightarrow S_2(i, j, k)$. Достаточные условия допустимости тайлинга выполняются: для любой зависимости $S_\alpha(I) \rightarrow S_\beta(J)$ имеет место $\beta \geq \alpha$ и, если у I и J есть координата с одинаковым номером, её значение в J не меньше, чем в I . Заметим, что в данном конкретном случае (варианты 1 и 2) порядок выполнения операций при тайлинге не изменяется, поэтому обосновать допустимость тайлинга можно без анализа зависимостей.

Запись параллельных зернистых вычислительных процессов. Из условия следует, что Q_2 – число процессов, предназначенных для реализации алгоритма. Единый для каждого из Q_2 процессов псевдокод параллельного алгоритма (без учета операций обмена данными) можно записать следующим образом ($p = j^{gl}$ – номер процесса):

Для каждого процесса Pr_p , $0 \leq p \leq Q_2 - 1$:

```

do  $i^{gl} = 1, N$ 
  Tile( $i^{gl}, p, 0$ )
enddo

```

Операции тайла Tile($i^{gl}, p, 0$):

```

 $i = i^{gl}$ 
do  $j = 1 + p r_2, \min((p + 1) r_2, N)$ 
 $S_1(i, j):$   $c(i, j) = 0$ 
            do  $k = 1, N$ 
 $S_2(i, j, k):$   $c(i, j) = c(i, j) + a(i, k) b(k, j)$ 
            enddo
          enddo

```

В нулевом процессе Pr_0 осуществляются все вычисления алгоритма, для которых $1 \leq j \leq r_2$; в процессе Pr_1 осуществляются вычисления, для которых $r_2+1 \leq j \leq 2r_2$. В процессе Pr_p , кроме, возможно, (Q_2-1) -го процесса, осуществляются все вычисления алгоритма, для которых $1+p r_2 \leq j \leq (p+1)r_2$; в процессе с номером Q_2-1 осуществляются вычисления алгоритма, для которых $1+(Q_2-1)r_2 \leq j \leq \min(Q_2 r_2, N)$.

Распределение входных и выходных данных. Соответственно распределению вычислений происходит распределение между процессами элементов матриц B и C (согласно заданиям вариантов 1 и 2 элементы матрицы A назначаются процессам динамически). Произвольному процессу Pr_p распределяются столбцы матрицы B с номерами с $1+p r_2$ по $\min((p+1)r_2, N)$. В Pr_p вычисляются столбцы матрицы C с теми же номерами. Распределение входных данных осуществляет нулевой процесс. Процесс Pr_p , $p > 0$, отправляет процессу Pr_0 «свою» часть вычисленной матрицы C . «Свою» (для процесса Pr_p) часть матрицы B обозначим B_p , «свою» часть вычисленной матрицы C обозначим C_p . Нулевой процесс формирует всю вычисленную матрицу C .

Общее представление о работе параллельного алгоритма и об обмене данными. Эти рассуждения здесь не приводим (в лабораторной работе этот пункт тоже можно опустить).

Выделение массивов. Приватизация массивов. B_p и C_p , $0 \leq p \leq Q_2-1$, – матрицы размера $N \times r_2$ – приватизируются процессом Pr_p . AP – вектор размера N – формируется (при каждом фиксированном i^{gl}) в процессе Pr_0 из строк массива $a(i, k)$. Вектор AP нужен (используется, служит) для трансляции или бродкаста.

Запись тайла с выделенными массивами. Напомним вид тайла $Tile(i^{gl}, p, 0)$:

```

     $i = i^{gl}$ 
    do  $j = 1+p r_2, \min((p+1)r_2, N)$ 
       $S_1(i, j): c(i, j) = 0$ 
      do  $k = 1, N$ 
         $S_2(i, j, k): c(i, j) = c(i, j) + a(i, k) b(k, j)$ 
      enddo
    enddo

```

$Tile(i^{gl}, p, 0)$ с выделенными массивами:

```

 $i = i^{gl}$ 
do  $j = 1 + p r_2, \min((p+1)r_2, N)$ 
   $jp = j - p r_2$ 
 $S_1(i, j): c_p(i, jp) = 0$ 
  do  $k = 1, N$ 
 $S_2(i, j, k): c_p(i, jp) = c_p(i, jp) + a_p(k) b_p(k, jp)$ 
  enddo
enddo

```

Оптимизация вычислений в тайлах. Оптимизацию вычислений рассматривать не будем. В лабораторной работе следует вычисления оптимизировать. Например, вычисление границ цикла следует выполнять вне цикла.

Структурирование коммуникаций. Трансляцию данных (вариант 1), бродкаст данных (вариант 2) опишем непосредственно при записи псевдокода.

Коммуникационную операцию получения массива данных будем представлять в виде

$\text{receive}(\text{Pr}; a; M),$

где первый аргумент обозначает процесс, в котором вычислялся массив, второй аргумент обозначает пересылаемый массив, третий аргумент указывает объем (число элементов) массива. Коммуникационную операцию отправки массива данных будем представлять в виде

$\text{send}(\text{Pr}; a; M),$

где первый аргумент обозначает процесс, которому потребуются вычисленные элементы массива, второй аргумент обозначает пересылаемый массив, третий аргумент указывает объем массива.

При использовании бродкаста будем употреблять breceive и bsend .

Псевдокод параллельного зернистого алгоритма для варианта 1.

Для каждого процесса $\text{Pr}_p, 0 \leq p \leq Q_2 - 1$:

```

{if  $p=0$  сформировать матрицы  $B_q, 0 \leq q \leq Q_2 - 1,$ 
   $\text{send}(\text{Pr}_q; B_q; N \times r_2), 1 \leq q \leq Q_2 - 1 \}$ 
if  $p > 0$   $\text{receive}(\text{Pr}_0; B_p; N \times r_2)$ 
do  $i^{gl} = 1, N$ 
  if  $p=0$  сформировать (из строки  $a(i^{gl}, k)$ )  $AP$ 
  if  $p > 0$   $\text{receive}(\text{Pr}_{p-1}; AP; N)$ 
   $\text{Tile}(i^{gl}, p, 0)$ 
  if  $p < Q_2 - 1$   $\text{send}(\text{Pr}_{p+1}; AP; N)$ 
enddo
if  $p > 0$   $\text{send}(\text{Pr}_0; C_p; N \times r_2)$ 
{if  $p=0$   $\text{receive}(\text{Pr}_q; C_q; N \times r_2), 1 \leq q \leq Q_2 - 1,$ 
  сформировать вычисленную матрицу  $C \}$ 

```

Псевдокод параллельного зернистого алгоритма для варианта 2.

Для каждого процесса Pr_p , $0 \leq p \leq Q_2 - 1$:

```
{if  $p=0$  сформировать матрицы  $B_q$ ,  $0 \leq q \leq Q_2 - 1$ ,  
    send( $Pr_q$ ;  $B_q$ ;  $N \times r_2$ ),  $1 \leq q \leq Q_2 - 1$  }  
if  $p > 0$  receive( $Pr_0$ ;  $B_p$ ;  $N \times r_2$ )  
do  $i^{gl} = 1, N$   
    {if  $p=0$  сформировать (из строки  $a(i^{gl}, k)$ )  $AP$ .  
        bsend( $Pr_q$ ,  $1 \leq q \leq Q_3 - 1$ ;  $AP$ ;  $N$ ) }  
    if  $p > 0$  breceive( $Pr_0$ ;  $AP$ ;  $N$ )  
    Tile( $i^{gl}, p, 0$ )  
enddo  
if  $p > 0$  send( $Pr_0$ ;  $C_p$ ;  $N \times r_2$ )  
{if  $p=0$  receive( $Pr_q$ ;  $C_q$ ;  $N \times r_2$ ),  $1 \leq q \leq Q_2 - 1$ ,  
    сформировать вычисленную матрицу  $C$ }
```