Общие механизмы OpenMP. Разделение работы

#pragma omp for [*условие* [,*условие*] ...]

```
#pragma omp parallel
{
  #pragma omp for private(i) shared(a,b)
    for(i=0; i<10000; i++)
      a[i] = a[i] + b[i]
} ← sync
```

Общие механизмы OpenMP. Разделение работы

schedule(тип [, размер блока])

Размер блока задает размер каждого пакета на обработку потоком (количество итераций).

Типы расписания:

- **`static`** – итерации равномерно распределяются по потокам. Т.е. если в цикле 1000 итераций и 4 потока, то один поток обрабатывает все итерации с 1 по 250, второй – с 251 по 500, третий - с 501 по 750, четвертый с 751 по 1000. Если при этом задан еще и размер блока, то все итерации блоками заданного размера циклически распределяются между потоками.

Статическое распределение работы эффективно, когда время выполнения итераций на любом потоке равно, или приблизительно равно.

- **`dynamic`** – работа распределяется пакетами заданного размера (по умолчанию размер равен 1) между потоками.

Как только какой-либо из потоков заканчивает обработку своей порции данных, он захватывает следующую.

В отличии от статического планирования, выполняется многократно (во время выполнения программы).

При этом подходе накладные расходы несколько больше, но можно добиться лучшей балансировки загрузки между потоками.

Общие механизмы OpenMP. Разделение работы

schedule(тип [, размер блока])

- **`guided`** – данный тип распределения работы аналогичен предыдущему, однако размер блока изменяется динамически в зависимости от того, сколько необработанных итераций осталось. Размер блока постепенно уменьшается вплоть до указанного значения.

Распределение начинается с некоторого начального размера, зависящего от реализации библиотеки.

При таком подходе можно достичь хорошей балансировки при меньших накладных расходах.

- **`runtime`** – тип распределения определяется в момент выполнения программы. Удобно в экспериментальных целях для выбора оптимального значения **типа** и **размера блока**.

Тип распределения работ зависит от переменной окружения **OMP_SCHEDULE**.

По умолчанию считается, что установлен статический метод распределения работ.

https://stackoverflow.com/questions/42970700/openmp-dynamic-vs-guided-scheduling

Общие механизмы OpenMP. Разделение работы

`#pragma omp sections [условие [,условие...]]`

```
#pragma omp parallel sections (nowait?)
{
  #pragma omp section
  {
    printf("T%d: foo\n", omp_get_thread_num());
  }
  #pragma omp section
  {
    printf("T%d: bar\n", omp_get_thread_num());
  }
} // omp sections ← sync
```

Общие механизмы OpenMP. Разделение работы

`#pragma omp single [условие [, условие ...]]`

```
#pragma omp single
  printf(”Program finished!\n");
```