

## Построение схемы измерения временных характеристик

Для сравнения различных вариантов реализуемого прототипа необходимо оценить временные характеристики выполнения прототипов на одинаковых наборах исходных данных. В качестве функции для измерения временных характеристик используется функция

`double MPI_Wtime(void)`, которая возвращает время в секундах, прошедшее с некоторого момента в прошлом в виде числа типа `double`. Так же воспользуемся функцией `double MPI_Wtick(void)` для определения точности (разрешения) функции `MPI_Wtime()`.

Для измерения временных затрат используется разница между двумя значениями, полученными в результате вызова функции `MPI_Wtime()` до и после некоторого события.

Пример кода, иллюстрирующего подобную процедуру:

```
// Логический блок
{
    double rTimeBefore; // время до события
    double rTimeDelta; // временные затраты на событие
    // измеряем текущее время
    rTimeBefore = MPI_Wtime();
    // собственно событие
    ...;
    // вычисляем разницу времени - искомые затраты
    rTimeDelta = MPI_Wtime() - rTimeBefore;
}
```

При последовательных измерениях временных затрат возможен другой вариант с целью уменьшения числа дублирующихся вызовов. Сначала в ключевых точках измеряется время, данные сохраняются в массиве, который подлежит последующей обработке.

Пример кода, иллюстрирующего подобную процедуру:

```
// Логический блок
{
    double raTime[nNumPoints]; // временные метки
    // измеряем текущее время
    raTime[0] = MPI_Wtime();
    // собственно событие №1
    ...;
    // измеряем текущее время
    raTime[1] = MPI_Wtime();
    ...
}
```

```
// измеряем текущее время
raTime[nNumPoints-2] = MPI_Wtime();
// собственно событие №«nNumPoints-1»
...;
// измеряем текущее время
raTime[nNumPoints-1] = MPI_Wtime();
// обработка результатов
...;
}
```

Временные затраты на вызов функции `MPI_Wtime()` и вычисление разницы двух времен можно считать пренебрежимо малыми. Обработка данных проводится в основном процессе.

В качестве меток или точек измерения, позволяющих измерить вычислительные и коммуникационные временные затраты для всех элементов, используются все «узловые» точки программы, т.е. выполняются измерения времени во всех промежутках между вычислениями и коммуникациями, а также в начале и в конце программы.

Следует отметить, что при таком выборе гарантируется точное измерение времени, затраченного на вычисления. Но в полученном значении времени, затраченном на коммуникации, также учитывается время «простоя» процессов – время, затраченное на ожидание начала приема или передачу сообщений, которое неизбежно при всех указанных выше топологиях.