

Программный инструмент для исследования производительности программ

АННОТАЦИЯ. Описывается программа Profile для исследования производительности программ в определяемых пользователем интервалах. Программа пригодна как для использования в традиционных (последовательных) фортранных программах, так и в распараллеленных с использованием технологии MPI.

ANNOTATION. Program Profile for studying of productivity in a program, being investigated by user on the set of intervals of interest is described. Usage of MPI-technology is possible.

Измерение времени работы процессора (процессоров) при исполнении исследуемой программы является чрезвычайно полезным мероприятием. Оно позволяет, во-первых, оценить производительность программы в целом, определяя ее конкурентоспособность в ряду аналогичных программ. Во-вторых, выявить наиболее времяёмкие места изучаемой программы, чтобы в дальнейшем сосредоточить свои усилия именно на них.

Поэтому мы сочли необходимым предложить специальный инструментарий для облегчения подобных деяний при разработке или модификации программ.

Инструмент оформлен как subroutine Profile(N).

Предполагается, что пользователь хочет измерить время работы своей программы на $N < 200$ интервалах ее исполнения. Тогда при $N = -1$ единожды совершается первоначальная засечка времени с помощью стандартной системной программы Seconds. Далее, **в начале и конце** каждого N -го интервала жизни исследуемой программы необходимо вызвать Profile(N). Эти засечки делаются с использованием стандартной программы CPU_Time. Таким образом, общее количество временных засечек должно быть четным!

В конце программы следует вызвать Profile(0) для распечатки всех сделанных засечек. Эта распечатка выглядит примерно так:

```
Time  0 =      5.4 sec.
Time  1 =      6.1 sec.
. . .
Time  <N-1> = 4.8 sec.

WallTime:      6.5  Total CPU_time:      6.1 sec.
```

Здесь, мы надеемся, смысл всех печатаемых времен читателю ясен.

Нумерация счетчиков времен определяется исключительно пользователем, а WallTime – это просто астрономическое время, прошедшее от первой до последней засечки.

Если программа выполняется под управлением пакета MPI, то иницирующий и завершающий этапы измерения ее производительности представляется удобным оформить как вызовы процедур Profile_Start и Profile_Fin. Они также являются частью инструмента Profile.

Приведем небольшой пример измерения времени работы программы:

```
Program Test
. . .
Call Profile_Start  ! Profile: timing initialization
. . .
Repeat
  Call Profile(MyProc+1)      ! start timing
  < do something >           ! code being measured
  Call Profile(MyProc+1)      ! finish timing
  . . . .
  Call Profile(MyProc+10)     ! start timing
  < do something else >
  Call Profile(MyProc+10)     ! finish timing
  . . .
Until done
Call Profile_Fin      ! Profile: timing finalization
End
```

Здесь MyProc - номер процесса, выдаваемый подпрограммой MPI_Comm_Rank пакета MPI, или просто 0, если работаем безо всякого MPI (по умолчанию).

22.10.2013