



Отладка MPI

Алексей А. Романенко

arom@ccfit.nsu.ru

Отладка параллельной программы и MPI в частности является существенно более сложным процессом, чем отладка последовательной программы. Это связано с тем, что параллельная программа — это несколько последовательных программ, которые взаимодействуют между собой, что приводит к появлению новых ошибок, связанных с операциями коммуникации и пр.



Ошибки

- Ошибки как и в последовательных программах
- Ошибки синхронизации
 - Ошибки соревнования
 - мертвые/живые блокировки
- Неравномерность загрузки

Т.е. В параллельной программе возможны ошибки как и в последовательных программах, так и новые ошибки: ошибки соревнования, мертвые блокировки, неравномерность загрузки узлов вычислительной системы.



printf

- Не забывайте печатать ранк процесса/время/строку кода
- Много вывода
- Сложно анализировать

Традиционно один из самых первых способов — это вставка вывода отладочной информации на экран. Стоит порекомендовать не забывать печатать помимо ошибки и служебную информацию по процессу и строке кода.

Некоторые ошибки действительно можно отловить данным способом (но не все), однако при этом придется неоднократно запускать код программы, вставлять дополнительный вывод на экран, анализировать большое количество текстовой информации. При этом есть возможность внесения новых ошибок и маскировки существующих.



TotalView

- Коммерческий продукт
 - <http://www.etnus.com>
- Запуск
 - `mpirun -tv <другие аргументы>`
 - `totalview mpirun <другие аргументы>`
 - `mpirun -dbg=totalview <другие аргументы>`

Более эффективным по сравнению с предыдущим вариантом, может оказаться использование специализированных средств отладки. Одним из наиболее распространенных является коммерческий продукт TotalView.

Варианты запуска представлены на слайде.



mpigdb

- Распространяется с mpich
- Несколько отладчиков gdb и средство перенаправления stdin/stdout
- Менее продвинуто в отличие от TotalView

В комплекте с MPICH — реализацией MPI - поставляется отладчик mpigdb, который по сути управляет отладчиками gdb, запускаемыми на узлах вычислительной системы.

```
arom# mpigdb -np 5 cpi # по умолчанию вывод от всех
(mpigdb) b 29 # установить точку останова для всех
0-4: Breakpoint 1 at 0x8049e93: file cpi.c, line 29.
(mpigdb)r # запустить все 0-4:
Starting program: /home/lusk/mpich/examples/basic/cpi
0: Breakpoint 1, main (argc=1, argv=0xbffffa84) at cpi.c:29
1-4: Breakpoint 1, main (argc=1, argv=0xbffffa74) at cpi.c:29
0-4: 29 n = 0; # все достигли точки останова
(mpigdb)n # пошаговый режим для всех
0: 38 if (n==0) n=100; else n=0;
1-4: 42 MPI_Bcast (&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
(mpigdb)z0 # stdin только для процесса ранга 0
(mpigdb)n # пошаговый режим процесса ранга 0
0: 40 startwtime = MPI_Wtime ();
(mpigdb)n # до останова
0: 42 MPI_Bcast (&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
(mpigdb)z # stdin возвращен для всех процессов
(mpigdb)n # пошаговый режим для всех
...
```

Пример использования `mpigdb` представлен на двух слайдах

```
(mpigdb) n
0-4: 52          x = h * ( (double)i - 0.5)
(mpigdb) px      # вывод от всех процессов
0: $1 = 0.0050000000000000001    # значение x процесса 0
1: $1 = 0.014999999999999999    # значение x процесса 1
2: $1 = 0.0250000000000000001    # значение x процесса 2
3: $1 = 0.0350000000000000003    # значение x процесса 3
4: $1 = 0.044999999999999998    # значение x процесса 4
(mpigdb)c        # продолжить все
0: pi is approximately 3.141600986923, Error is 0.000008333333
0$-4: Program exited normally.
(mpigdb)q        # выход
```

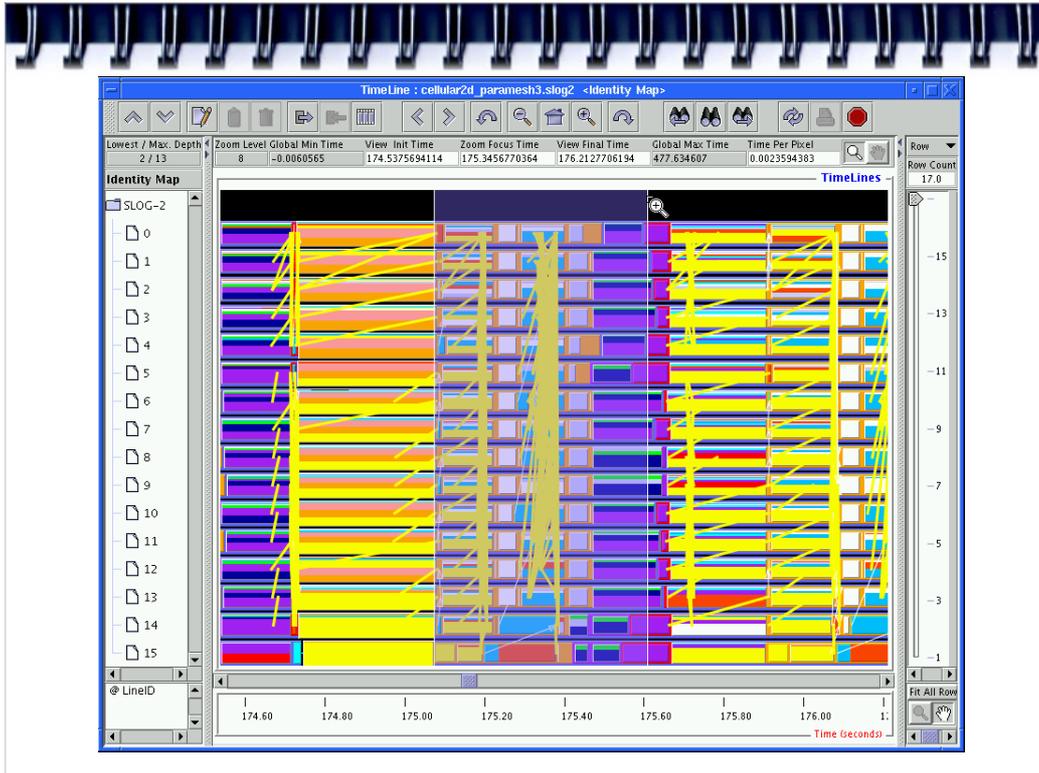


Jumpshot

- Java приложение
- Визуализация трасс в формате clog

Для отладки коммуникаций использовать TotalView или mpigdb не очень разумно. Проще стандартными средствами MPI собрать трассу программы (кто когда и с кем по средством каких функций взаимодействовал) и затем проанализировать полученный файл отчета.

Для этого можно использовать приложение Jumpshot.



Результат визуализации трассы представлен на слайде.

По горизонтали откладывается время, по вертикали процессы с разными ранками. Прямоугольниками разного цвета отображаются MPI - функции



Запуск и трассировка MPI программ

```
export MPE_LOG_FORMAT="ALOG"  
// ALOG, CLOG, SLOG  
mpicc -mpilog -o mpiprog mpiprog.c  
mpirun -np x mpiprog
```

```
mpicc -mpitrace -o mpiprog mpiprog.c  
mpirun -np x mpiprog
```

Для сборки трассы необходимо пересобрать программу с использованием специальной библиотеки, которая идет в комплекте с реализацией MPI.