ПРОВЕРКА ЦЕЛОСТНОСТИ ДАННЫХ, ПЕРЕСЫЛАЕМЫХ МЕЖДУ MPI-ПРОЦЕССАМИ В MCCT 1.0

А.В. Огородников, М.И. Старов

ООО «Центр компетенций и обучения», Саров

e-mail: mistarov@compcenter.org

Разработка современных параллельных программных комплексов MPI – непростая задача, при которой возрастает вероятность некорректного применения MPI вызовов. Представляемый инструмент MCCT призван значительно ускорить поиск подобных ошибок в реализации параллельного кода MPI приложений. Важной особенностью в версии MCCT 1.0 является новая проверка целостности данных, пересылаемых между MPI-процессами.

Введение

Анализ и отладка современных параллельных программных комплексов MPI [1] – непростая задача, усложняющаяся целым рядом факторов. Среди них и непрерывный рост количества вычислительных ядер, и применение асинхронных обменов, и использование сложных типов данных, и, в целом, наличие большого количества специализированных ресурсов MPI. Неоднозначность стандарта MPI также привносит свой вклад в усложнение процесса разработки – под этим подразумевается как двоякое использование одних и тех же типов данных (например, MPI_Comm) и функций (например, использующих входной аргумент MPI_IN_PLACE), так и разнообразие реализаций библиотеки MPI. Вдобавок к вышесказанному, как правило, ни компиляторы, ни библиотеки MPI не отслеживают ошибки использования MPI.

В результате вероятность некорректного применения MPI-вызовов возрастает. Наличие данных ошибок в свою очередь может привести к искажению передаваемых данных, зависанию программы и к другим сбоям.

Как правило, при поиске проблем в программном коде разработчики используют программы-отладчики, позволяющие пошагово исследовать выполнение программы. Но подобное исследование всего параллельного программного кода, в котором идет постоянный межпроцессный обмен, является весьма длительным и изменяет поведение исследуемой программы.

Соответственно, является логичным переложить все проверки параллельного кода на вычислительную машину, которая будет выполнять их без задержки и по точно закодированным спецификациям. Это поможет программисту точно определить расположение ошибки в коде и сразу заняться её исправлением.

Эту задачу решает инструмент MCCT 1.0 (MPI Correctness Checking Tool) [2], входящий в состав программного комплекса S-MPI [3] и базирующийся на программном обеспечении с открытым кодом MUST 1.1 [4], призванный значительно ускорить поиск подобных ошибок в реализации параллельного кода MPI-приложений.

Одной из важных особенностей MCCT 1.0 является способность обнаружить нарушение целостности передаваемых данных.

Механизм работы и возможности MCCT

Механизм работы MCCT основан на оберточных функциях (wrappers) стандарта MPI-2, в которых производятся все необходимые проверки корректности использования MPI, и из которых вызывается соответствующая PMPI-функция. Инициализация MCCT происходит при вызове функции MPI_Init или MPI_Init_thread.

При каждом входе в функцию MPI MCCT выполняет проверку ряда условий, таких как: проверка правильности аргументов, соответствие получаемых данных их типу, сопоставление коллективных операций и т.д. Для проведения проверок, которые требуют данные от двух и более процессов, так называемых глобальных проверок, MCCT пересылает проверяемые данные служебному MPI процессу; MCCT создает этот процесс автоматически при использовании скрипта запуска. Кроме того, служебный MPI процесс позволяет разгрузить основные счетные процессы MPI задачи. При невыполнении проверяемых условий MCCT сообщает об обнаруженных проблемах в поток вывода STDOUT.

Инструмент MCCT выполняет проверки следующих типов:

- тестирование параметров функций MPI; при этом учитывается контекст вызовов для данной функции MPI, а также, какие ограничения на нее могут быть наложены с учетом языковой принадлежности используемой реализации MPI (C, Fortran);
- проверка переносимости исходного кода на другие реализации MPI;
- проверка целостности используемых данных в вызовах MPI;
- проверка памяти, используемой пользовательскими ресурсами MPI, на своевременное освобождение;
- проверка возможного наложения памяти, в т.ч. внутри сложных типов данных MPI;
- проверка совпадения типов пересылаемых данных между процессами;
- сопоставление коллективных операций, вызванных из вовлеченных в данный коммуникатор процессов MPI, по ряду признаков;
- отслеживание возможных потерь двухточечных сообщений;
- проверка на условия возникновения взаимоблокировок (deadlocks) между MPI-процессами (как реальные, так и потенциальные).

Проверка целостности пересылаемых данных

Во время передачи данных между MPI-процессами в параллельном приложении возможны различные сбои, приводящие к порче или потере пересылаемых данных. Причинами повреждения данных во время пересылки могут являться:

- ошибка разработчика, некорректно использующего асинхронные двухточечные операции в своём приложении;
- возникновение сбоев внутри используемой библиотеки MPI;
- возникновение сбоев внутри коммуникационной среды.

Основных проверок MCCT, диагностирующих некорректное использование вызовов MPI в приложении и наследуемых из базового продукта MUST, недостаточно для обнаружения повреждения пересылаемых данных.

В связи с этим в версии MCCT 1.0 было решено реализовать новый механизм проверок, основанный на сравнении контрольных сумм переданных данных.

Вычисление контрольных сумм данных происходит в обёрточных MPI-функциях непосредственно перед вызовом PMPI-функции для функций передачи и после вызова PMPI-функции для функций приема данных. Затем контрольные суммы передаются сервисному процессу, который удаленно занимается хранением, сопоставлением и сравнением контрольных сумм, соотвествующих обменным операциям MPI. При обна-

ружении различия значений контрольных сумм для одной пары приема/передачи MCCT выводит соответствующее диагностическое сообщение.

При таком подходе достигается минимальное воздействие на исследуемую программу, т.к. проверяемые данные и алгоритм работы программы не подвергаются изменению.

Для примера диагностики повреждения пересылаемых данных разработан следующий тест:

```
$ cat datacorruption.c
#include <string.h>
#include <mpi.h>
int main (int argc, char **argv)
{
 int rank;
 int buf[10];
 MPI_Request req;
 MPI_Status status;
 MPI_Init( &argc, &argv );
 MPI_Comm_rank( MPI_COMM_WORLD, &rank );
 if( rank == 0 ) {
 memset( buf, 1, 10 );
 MPI_Issend( &buf, 10, MPI_INT, 1, 100, MPI_COMM_WORLD, &req );
 buf[0] = 0;
 MPI_Barrier(MPI_COMM_WORLD);
 }
 if( rank == 1 ) {
 MPI_Barrier(MPI_COMM_WORLD);
 MPI_Recv( &buf, 10, MPI_INT, 0, 100, MPI_COMM_WORLD, &status );
 }
 MPI_Finalize( );
 return 0;
}
```

В данном случае применяется двухточечный обмен Issend/Recv. После инициализации запроса на отправку буфер с данными изменяется, а фактическая пересылка данных задерживается барьером. Таким образом, второй процесс принимает уже испорченные данные.

Диагностическое сообщение для такого примера приведено ниже:

```
$ checkrun -n 2 ./datacorruption

MPI CCT: ERROR: MPI_Recv [rank 1]:
MPI CCT: ERROR: Data corruption during transferring from rank 0 to rank 1
MPI CCT: ERROR: MPI_Recv [rank 1] called from:
MPI CCT: ERROR: #0 main@datacorruption.c:19
MPI CCT: ERROR: #1 __libc_start_main@/lib64/libc-2.12.so:(0x32a2e1ecdd)
MPI CCT: ERROR: #2 _start@/work/examples/datacorruption:(0x400809)

MPI CCT: INFO: Detected 1 errors and 0 warnings. More details is available in
the checker output file.
```

Как видно из примера, диагностическое сообщение содержит информацию о стеке вызовов, что позволяет точно указать на обнаруженное проблемное место MPI-приложения. Данная особенность MCCT использует API библиотеки Stackwalker из LGPL программного продукта Dyninst [5].

В итоге, представленная новая проверка значительно расширяет возможности утилиты MCCT 1.0, предоставляющей полноценный и гибкий комплекс автоматизированного поиска проблемных мест в программном коде сложных параллельных комплексов, что позволяет сократить время на их отладку и оптимизацию.

Можно также отметить, что утилита MCCT оптимизирована для работы в связке с MPI-библиотекой, входящей в состав программного комплекса S-MPI [2, 3].

Литература

1. Message Passing Interface Forum "MPI: A Message Passing Interface". – In Proceedings of Supercomputing ’93. IEEE Computer Society Press, November 1993. P. 878–883.
2. Огородников А.В., Побуринная Н.А., Старов М.И. Компонент проверки корректности использования MPI в программном комплексе S-MPI. – Вопросы атомной науки и техники, сер. «Математическое моделирование физических процессов», 2013, № 2, С. 86-94.
3. Воронов Г.И., Трущин В.Д., Шумилин В.В., Ежов Д.В. Программный комплекс S-MPI для обеспечения разработки, оптимизации и выполнения высокопараллельных приложений на суперкомпьютерных кластерных системах. – Вопросы атомной науки и техники, сер. «Математическое моделирование физических процессов», 2013, № 3, С. 55-60.
4. Hilbrich T., Schulz M., de Supinski B.R., Muller M. MUST: A Scalable Approach to Runtime Error Detection in MPI Programs. – In Proceedings of the 3rd Parallel Tools Workshop, Dresden, Germany, September 2009. P. 53-66.
5. Библиотека для работы со стеком вызовов Stackwalker. – [Электронный ресурс]: http://www.dyninst.org/stackwalker.