

# МССТ 1.0 И НОВЫЕ ВОЗМОЖНОСТИ ПРОВЕРКИ КОРРЕКТНОСТИ ИСПОЛЬЗОВАНИЯ MPI

А.В. Огородников, М.И. Старов

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 двойное 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 имеем, что 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 и весьма длительным занятием.

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.

Представляемый инструмент МССТ 1.0 (MPI Correctness Checking Tool), базирующийся на программном обеспечении с открытым кодом MUST 1.1 [2], призван значительно ускорить поиск подобных ошибок в реализации параллельного кода MPI приложений.

Механизм работы МССТ основан на оберточных функциях (wrappers) стандарта MPI-2, в которых производятся все необходимые проверки корректности использования MPI, и из которых вызывается соответствующая RMPi функция. Инициализация МССТ происходит при вызове функции MPI\_Init или MPI\_Init\_thread.

При каждом входе в функцию MPI МССТ 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, т.к. называемых глобальных проверок, МССТ пересылает проверяемые данные служебному MPI процессу; МССТ создает этот процесс автоматически при использовании скрипта запуска. Кроме того, служебный MPI процесс позволяет разгрузить основные счетные процессы MPI задачи. При невыполнении проверяемых условий МССТ сообщает об обнаруженных проблемах в поток вывода STDOUT.

Инструмент МССТ содержит более сотни локальных и глобальных проверок, выполняющих следующее:

- 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;
- проверку целостности используемых данных в вызовах MPI;
- проверку памяти, используемой пользовательскими ресурсами MPI, на своевременное освобождение;
- проверку возможного наложения памяти, в т.ч. внутри сложных типов данных MPI;
- проверку совпадения типов пересылаемых данных между процессами;
- сопоставление коллективных операций, вызванных из вовлеченных в данный коммунитор процессов MPI, по ряду признаков;
- отслеживание возможных потерь двухточечных сообщений.
- проверка на условия возникновения взаимоблокировок (deadlocks) между MPI процессами (как реальные, так и потенциальные).

Благодаря отслеживанию создаваемых ресурсов MPI, МССТ также диагностирует первопричину случившейся взаимоблокировки и предоставляет возможность оценить в целом картину блокирующего обмена. С целью получения графических представлений в случае обнаружения взаимоблокировок на выходе создаются dot-файлы: МССТ\_Deadlock.dot, МССТ\_DeadlockCallstack.dot. Используя инструмент DOT из свободного программного пакета Graphviz [3], на их основе могут быть построены графические файлы в png-формате,

которые могут быть использованы для визуального анализа обнаруженной проблемы. Например, с помощью следующей команды будет получен графический файл MCCT\_Deadlock.png:

```
$ dot -Tpng MCCT_Deadlock.dot -o MCCT_Deadlock.png
```

Реализованное отслеживание стека вызовов помогает более точно определить проблемное место приложения; при этом не оказывается чрезмерного влияния на производительность проверяемого MPI приложения, что позволяет ей выгодно смотреться на фоне конкурентов. При этом используется API библиотеки Stackwalker из LGPL программного продукта Dyninst [4].

Для удобства запуска проверки корректности обеспечен запускающий скрипт, помогающий включить MCCT в работу во время выполнения проверяемого MPI приложения; пример командной строки:

```
$ checkrun -np 4 ./helloworld
```

В ходе выполнения программы MCCT выводит на экран или в файл диагностируемые проблемы, после чего анализ обнаруженных ошибок использования MPI поможет программисту устранить их. Формат выводимой диагностики:

```
MPI CCT: <тип_диагностики>: <сообщение>
```

Типом диагностики могут быть ERROR, WARNING и INFO. Пример выводимой диагностики MCCT:

```
MPI CCT: ERROR: The application issued a set of MPI calls that can cause a deadlock!
```

```
MPI CCT: ERROR: References:
```

```
MPI CCT: ERROR: 1: MPI_Finalize [rank 0] (in main@deadlock_scatter.c:17);
```

```
MPI CCT: ERROR: 2: MPI_Scatter [rank 1] (in main@deadlock_scatter.c:15);
```

В версии 1.0 утилиты MCCT добавлены следующие новые возможности:

- проверка целостности пересылаемых данных между процессами, вынесенная в служебный процесс; это проверка полезна для выявления возможных проблем с порчей памяти передаваемых данных в процессе их пересылки.
- широкие возможности конфигурирования запусков через переменные окружения; включает в себя ограничение/отключение выводимой диагностики для каждой группы проверок.

В итоге, обновленная утилита MCCT 1.0, оптимизированная для работы в связке с MPI библиотекой S-MPI [5,6], предоставляет разработчикам полноценный и гибкий комплекс автоматизированного поиска проблемных мест в программном коде сложных параллельных комплексов, что позволяет сократить время на их отладку и оптимизацию.

#### ЛИТЕРАТУРА:

1. Message Passing Interface Forum "MPI: A Message Passing Interface". // In Proceedings of Supercomputing '93. IEEE Computer Society Press, November 1993, pp. 878–883.
2. T. Hilbrich, M. Schulz, B. R. de Supinski, M. Muller "MUST: A Scalable Approach to Runtime Error Detection in MPI Programs". // In Proceedings of the 3rd Parallel Tools Workshop, Dresden, Germany, September 2009, pp. 53–66.
3. Пакет автоматической визуализации графов Graphviz. // [Электронный ресурс]: <http://www.graphviz.org>
4. Библиотека для работы со стеком вызовов Stackwalker. // [Электронный ресурс]: <http://www.dyninst.org/stackwalker>
5. Г.И. Воронов, В.Д. Трушин, В.В. Шумилин, Д.В. Ежов "Создание программного комплекса S-MPI для обеспечения разработки, оптимизации и выполнения высокопараллельных приложений на суперкомпьютерных кластерных и распределенных вычислительных системах". // Тезисы докладов XIV Международной конференции "Супервычисления и математическое моделирование", Саров, 1-5 октября 2012, с.54-56.
6. А.В. Огородников "Утилита проверки корректности MPI-приложений в программном комплексе S-MPI". // Тезисы докладов XIV Международной конференции "Супервычисления и математическое моделирование", Саров, 1-5 октября 2012, с.131-132.