СИСТЕМА ТЕСТИРОВАНИЯ КОММУНИКАЦИЙ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ И КЛАСТЕРАХ, ОСНОВАННАЯ НА MPI, И ГРАФИЧЕСКИЙ ИНТЕРФЕЙС К НЕЙ

А.Н. Сальников, Д.Ю. Андреев
Факультет вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова, Москва
Тел.: (495) 939-17-90, e-mail: salnikov@angel.cs.msu.su, andreev@angel.cs.msu.su

Подсистема коммуникаций в современных многопроцессорных и кластерных системах довольно сложна и обычно состоит из большого числа компонентов. Как следствие, оказывается довольно тяжело предсказать время передачи сообщения определённой длины при заданном прогнозе степени загруженности коммуникаций. Трудность предсказания связана в первую очередь именно с большим числом компонентов, составляющих коммуникации. В текущий момент наиболее популярной программной технологией для передачи сообщений является MPI. Следовательно, стоит ожидать появления нового поколения «умных» систем тестирования коммуникаций, которые смогут хотя бы частично решить проблему предсказания длительностей задержек при передаче сообщений через коммуникационные подсистемы кластеров и многопроцессорных систем.

Существует и другая точка зрения на процесс тестирования коммуникаций. Иногда вычислительные кластерные системы имеют несколько тысяч процессоров. Это является причиной того, что тесты в процессе тестирования производят большой объём данных, настолько большой, что человек без использования каких-либо специальных средств не в состоянии их постичь. Следовательно, наряду с интеллектуальными тестами, необходимо создать и достаточно сложные средства визуализации получаемых в процессе тестирования данных.

На текущий момент времени существует некоторое число MPI-тестов, которые определяют задержки при передаче данных через коммуникации, например: NetPIPE [1] и MPI-bench-suit (доступен с сайта http://parallel.ru/testmpi/). К сожалению, большинство средств с открытым исходным кодом лишены средств визуализации результатов тестирования.

Цель нашей работы заключается в разработке нескольких MPI-тестов, которые позволят извлечь статистические данные о задержках при передаче сообщений через коммуникационную среду, а также приложения, осуществляющего визуализацию информации, полученной как результат работы тестов.

Авторами разработано MPI-приложение (*network_test*), в дальнейшем называемое «тест», которое является одним из компонентов системы PARUS [2]. В текущий момент приложение поддерживает шесть режимов тестирования коммуникаций. Для приложения пользователем указывается несколько параметров: интервал длин сообщений, шаг приращения длины сообщения и число повторов на каждом шаге по длине сообщения. Тест начинает свою работу с самой маленькой длины сообщения в интервале, длина сообщения увеличивается на значение шага на каждой итерации до тех пор, пока не будет достигнута верхняя граница интервала. По достижении верхней границы тестирование прекращается. Для каждой длины сообщения тест производит некоторое количество передач через коммуникационную среду. Тип и объём передач зависит от выбранного пользователем режима, который указывается в параметрах командной строки вместе со всеми остальными параметрами. Параметр «число повторов» задаёт число независящих друг от друга итераций для фиксированной длины сообщения, в результате получается некоторая выборка. Выборка в дальнейшем используется для поиска минимального значения, медианы, среднего значения и стандартного отклонения. Найденные значения формируют множество матриц, где каждая матрица отвечает своей длине сообщения. В позиции i,j матрицы содержится задержка при передаче сообщения от MPI-процесса с номером i к MPI-процессу с номером j. Задержка оценивается при помощи функции MPI_Wtime. Данные для минимального значения, медианы и т.д. записываются в отдельные текстовые файлы.

Опишем некоторые наиболее важные режимы тестирования:

- *one_to_one* — в этом режиме одновременно сообщениями обменивается только пара MPI-процессов. Для обмена используются функции MPI_Send и MPI_Recv. В матрицу записывается время выполнения функции MPI_Recv.
- *all_to_all* — в этом режиме все MPI-процессы вовлечены в обмен сообщениями. Сообщения передаются с помощью функций MPI_Isend и MPI_Irecv, а в матрицу записывается промежуток времени между инициализацией MPI_Irecv и окончанием обмена с MPI_Waitany.
- *async_one_to_one* — то же самое, что режим *one_to_one*, но используются неблокированные вызовы функций.
- *test_noise_blocking* — MPI-процессы разделяются на три группы: пара «целевых» процессов, множество «шумящих» процессов и множество «молчащих» процессов. Сперва выбирается пара целевых процессов, остальные помечаются как молчащие, затем из молчащих процессов случайным образом выбирается определённое число шумящих процессов. Число шумящих процессов определяется пользователем в параметрах теста. Целевые процессы производят передачу данных так же, как и для теста *one_to_one*; при этом, время их передачи заносится в матрицу. Шумовые процессы обмениваются сообщениями так же, как и в all_to_all режиме, но задержки для них никуда не сохраняются. Размер и число повторов шумовых сообщений определяются в параметрах теста.

Кроме тестирующего MPI-приложения авторами разработана программа на Sun Java 1.5 для визуализации результатов тестирования. Результаты могут быть отображены в 3-х различных режимах. В первом режиме отображается матрица для одного из размеров сообщений, интерфейс предоставляет возможность

перемещаться от одного размера к другому. В этом режиме предусмотрены два способа нормализации данных: локальная нормализация, внутри одной матрицы, и глобальная, для всего множества длин сообщений. При отображении, длительности задержки соответствует интенсивность серого цвета. Причём нормализованному минимальному значению соответствует белый цвет, а нормализованному максимальному — чёрный. Во втором режиме пользователь может выбрать одну строку или один столбец в матрице и отобразить их в градациях серого для всех длин сообщений. Этот режим используется для отображения характеристик фиксированного MPI-процесса по отношению к остальным процессам. В третьем режиме рисуется график для всех длин сообщений для выбранной пары MPI-процессов.

С помощью описанных приложений были получены данные для машин: mvs100k (кластер с 470x4 процессорами Intel Xeon 5160 связанных Infiniband сетью) и IBM pSeries 690 (SMP система с 16 процессорами в нашей конфигурации). Исходный код доступен со страницы проекта PARUS (**http://parus.sf.net**). Работа проводится при поддержке грантов: РФФИ 08-07-00445 и Президента РФ: МК-1606.2008.9.

Литература

1. Alexey N. Salnikov PARUS: A Parallel Programming Framework for Heterogeneous Multiprocessor Systems //Lecture Notes in Computer Science (LNCS 4192) Recent Advantages in Parallel Virtual Machine and Message Passing Interface, Volume 4192, pp 408-409, 2006, ISBN-10: 3-540-39110-X ISBN-13: 978-3-540-39110-4.

2. Dave Turner, Xuehua Chen, Protocol-Dependent Message-Passing Performance on Linux Clusters, // IEEE International Conference on Cluster Computing (CLUSTER'02), p. 187, 2002, ISBN: 0-7695-1745-5.