Тестирование коммуникационной среды Infiniband для решения задачи балансировки нагрузки на основе сети

М.Р. Халиуллина, А.В. Юлдашев

Узлы современных кластерных вычислительных систем строятся на базе многоядерных процессоров. При выполнении параллельных программ на узлах может находиться множество процессов, конкурирующих за каналы передачи данных. Одним из способов оптимизации использования ресурсов и сокращения времени выполнения программ является балансировка нагрузки – распределение процесса выполнения задач между узлами. Большое количество работ посвящено балансировке нагрузки на основе центрального процессора, оперативной памяти, использования жестких дисков. В то же время балансировка нагрузки в кластерных вычислительных системах на основе сети, по нашему мнению, исследована в недостаточной степени.

В данной работе были поставлены эксперименты для выявления характеристик производительности вычислительной коммуникационной среды кластера УГАТУ Infiniband SDR 4x 10-Gbps при точечном взаимодействии MPI-процессов в режимах конкуренции за канал передачи данных и ее отсутствия. Тесты производились с помощью пакетов mpi-bench-suite и OSU microbenchmarks (OMB). Целью тестирования являлись демонстрация необходимости балансировки нагрузки на основе сети, а также получение экспериментальных данных для построения математической модели коммуникационной среды Infiniband, необходимой при решении задачи динамической балансировки нагрузки в однородных кластерных системах.

Для исследования пропускной способности коммуникационной среды в режиме конкуренции использовался тест osu_mbw_mr, входящий в состав пакета OMB. Указанный тест позволяет измерить суммарную пропускную способность однонаправленных обменов, реализованных через функции MPI_Isend/MPI_Irecv, для нескольких одновременно работающих пар процессов. Тестирование проводилось для размеров сообщений 1, 4, 16…1 048 576 В с использованием библиотеки MVAPICH2 v1.2. В результате эксперимента было получено, что суммарная пропускная способность возрастает при увеличении числа задач, например, для сообщений размером 16 В суммарная пропускная способность увеличивается в 3,01 раза, а для сообщений размером 1 МВ – в 1,07 раза на 8 парах процессов. В то же время средняя пропускная способность на одну пару процессов уменьшается при увеличении числа одновременно выполняющихся задач, причем тем больше, чем больше размер сообщений. Так, для сообщений размером 16 В средняя пропускная способность уменьшается в 2,66 раз, а для сообщений размером 1 МВ – в 7,49 раз на 8 парах процессов. Следовательно, балансировка с учетом нагрузки на каналы передачи данных может принести положительный эффект: снизить время коммуникации в параллельных MPI-программах за счет оптимального использования коммуникационной среды кластерной системы.

Стандарт MPI не предусматривает возможности миграции MPI-процессов, что необходимо для выполнения динамической балансировки. Однако осуществление миграции MPI-процессов в рамках кластерной системы возможно с помощью библиотек поддержки контрольных точек. Наиболее прозрачный подход к поддержке контрольных точек предлагают библиотеки системного уровня, одной из которых является Berkeley Lab Checkpoint/Restart (BLCR), которая позволяет создавать контрольные точки для последовательных и многопоточных программ. В то же время некоторые распространенные реализации MPI поддерживают создание контрольных точек с помощью данной библиотеки: MVAPICH2, OpenMPI и другие. Проведено экспериментальное исследование зависимости времени, затрачиваемого на создание контрольной точки MPI-программы, от используемого объема памяти и количества процессов. В результате представляется возможным использование BLCR для реализации механизма миграции процессов MPI-программ в системе динамической балансировки.

В дальнейшем планируется построить математическую модель коммуникационной среды Infiniband, которая позволит оценить время коммуникации при наличии конкуренции за канал передачи данных. Такая модель может быть использована при разработке алгоритмов динамической балансировки нагрузки на основе сети.