ПРОГРАММНЫЙ КОМПЛЕКС S-MPI. ОБЕСПЕЧЕНИЕ НАДЕЖНОСТИ КОММУНИКАЦИОННОЙ СИСТЕМЫ

В.В. Шумилин, Н.М. Леонова

При разработке программного комплекса S-MPI [1] были реализованы следующие особенности: отказоустойчивость, масштабируемость, адаптация к современным коммуникационным средам, поддержка смешанных моделей программирования. Одним из аспектов отказоустойчивости является обеспечение надежности коммуникационной системы.

В настоящее время значительное число кластерных систем строится по следующей схеме: счетные узлы объединяются при помощи быстрой сети для передачи данных прикладных задач и дополнительно существует более медленная сеть для служебных сообщений. Если рассмотреть список top500 [2], список самых быстрых суперкомпьютеров, то можно заметить, что примерно половина использует в качестве быстрой сети Infiniband и Ethernet в качестве служебной сети. Для повышения пропускной способности коммуникационной системы (что особенно важно при повышении мощности вычислительных узлов) часто используются несколько коммуникационных каналов.

Большинство расчетов, производимых на высокопроизводительных кластерных системах, требуют значительного времени. Остановка работы по причине сбоя оборудования приведет к потере данных, которые уже были получены. То есть при следующем запуске потребуется начинать расчет с начала задания или с его последней сохраненной контрольной точки. Это будет означать нерациональное использование вычислительных мощностей.

Предоставление возможности продолжить работу даст возможность завершить выполнение задания, пусть и с несколько меньшей производительностью.

Для повышения надежности возможно использование существования дополнительных каналов. Так при наличии нескольких каналов быстрой сети, при сбое на одном из них можно переходить на передачу данных по работоспособным. Еще одной возможностью поддержать работоспособность является использование служебной коммуникационной сети в случае проблем с передачей через основную.

Реализация программного комплекса S-MPI 1.0 базируется на исходном коде OpenMPI [3,4], который имеет компонентную структуру. За организацию передачи сообщений отвечают три набора компонент:

- PML (Point-to-point Management Layer): компоненты этого набора управляют всей передачей сообщений. Они реализует семантику протокола двухточечных обменов MPI.
- BTL (Byte-Transfer-Layer): компоненты этого набора обеспечивают передачу сообщений через конкретные коммуникационные среды и не знают о семантике протокола двухточечных обменов MPI.

1. BML (BTL Management Layer): компоненты этого набора обеспечивают поддержку множества BTL-ей во время запуска задания и при динамическом порождении процессов.

То есть для передачи сообщений используются функции из компоненты PML. Для непосредственной передачи они используют функции необходимых компонент BTL. Выбор компонент BTL осуществляется при помощи функций BML.

Сообщения разбиваются на пакеты, для организации различных протоколов обмена каждому пакету присваивается определенный тип. В реализации OpenMPI определены следующие типы пакетов: MATCH, RNDV, ACK, NACK, FIN, FRAG, PUT, RGET.

На уровне PML реализовано два основных протокола передачи сообщений.

2. Eager-протокол (активная передача), при котором передающий процесс посылает сообщение независимо от наличия запроса на принимающей стороне. При наличии запроса на прием сообщение принимается в указанную в нем память, а при его отсутствии, оно принимается в системную память
3. Rendezvous-протокол (рандеву), при котором передающий и принимающий процессы предварительно договариваются, и передача происходит непосредственно в предназначенную для приема память.

Обычно eager-протокол используется для передачи коротких сообщений, так как он имеет меньшие издержки на старт передачи. Rendezvous-протокол используется для передачи длинных сообщений, так как время на синхронизацию при начале передачи, компенсируются исключением необходимости выделять системную память на приемнике и производить дополнительное копирование данных. Также Rendezvous-протокол используется для реализации специального синхронизирующего режима передачи.

Для обеспечения надежности была модифицирована специальная версия PML компоненты - BFO (BTL FailOver), и сделаны изменения в используемых компонентах BTL. При передаче сообщений в стандартном режиме данные распределяются по всем коммуникационным каналам. В случае обнаружения ошибки, использование коммуникационного канала, на котором обнаружена ошибка, прекращается, и обмены продолжаются по работоспособным каналам.

При обнаружении ошибки мы считаем, что корректные данные не были получены на принимающей стороне, поэтому требуется повторная посылка данных. Чтобы была возможность различить оригинальный пакет данных и посланный повторно, требуется использовать порядковые номера. Так как порядковые номера, уже присутствуют в заголовках пакетов MATCH, RNDV, RGET, дополнительные порядковые номера пакетов в BFO не вводятся. Пакеты других типов, таких как PUT или ACK никогда не посылаются повторно, так что не имеет значения то, что они не имеют порядковых номеров. Специальный случай пакет FIN, когда он включает в себя часть заголовка MATCH, а именно включает информацию об источнике, тэге и контексте, которые могут быть использованы для обнаружения дублирования.

Добавлено четыре новых типа пакетов для организации протокола повторной передачи: RNDVRESTARTNOTIFY, RECVERRNOTIFY, RNDVRESTARTACK,RNDVRESTARTNACK.

Когда запрос на передачу находится в состоянии ошибки и завершены все передачи, отправитель посылает пакет типа RNDVRESTARTNOTIFY, чтобы сообщить, что необходимо перепослать сообщение. При получении такого пакета, получатель сначала проверяет, что у него все еще имеется соответствующий запрос на прием, в этом случае он помечает этот запрос как ошибочный. Затем проверяет, что нет незавершенных событий, и посылает пакет типа RNDVRESTARTACK, а если незавершенные события есть, то это сообщение посылается, когда счетчик незавершенных событий опустится до нуля. Если соответствующий запрос на прием не найден, это означает, что сообщение было успешно принято и отсылается сообщение RNDVRESTARTNACK. Такая ситуация происходит, когда последний пакет сообщил на стороне передатчика об ошибке, но был успешно принят на стороне приемника.

Сообщение типа RECVERRNOTIFY добавлено, чтобы получатель мог сообщить передатчику об обнаруженной ошибке. При получении такого сообщения передатчик начинает процедуру повторной посылки с передачи пакета типа RNDVRESTARTNOTIFY.

Также на уровне BTL все пакеты, которые ожидали отправки по неисправному каналу, переназначаются для отправки по оставшимся каналам.

Для того чтобы оценить как повлияет на производительность переход на более медленную коммуникационную сеть были проведены запуски широко используемого теста Linpack [5] на восьми-узловом кластере с использованием сети Infiniband и последующим переходом на использование сети Ethernet(1Gbit).

Очевидно, что производительность (падение производительности) зависит от момента перехода на более медленную сеть. Были произведены имитации сбоя в сети Infiniband в разные моменты времени выполнения.

Можно отметить, что результаты использования этой возможности также зависят от типа задачи. А именно, соотношения объемов передаваемых данных и вычислений, производимых каждым процессом.

Используя полученные результаты, можно разработать различные технологии проведения расчетов на больших кластерных системах. Например, при обнаружении сбоя поведение системы может быть следующим:

- Продолжить вычисления до завершения программы, пусть и с некоторой потерей производительности.
- Передать сообщение о проблеме пользовательской программе, которая будет иметь возможность продолжить работу до ближайшего сохранения контрольной точки.

Данная работа выполнена в рамках контракта с Министерством образования и науки РФ (государственный контракт № 07.524.12.4020).

ЛИТЕРАТУРА:

1. Г.И. Воронов, В.Д. Трущин, 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”. // XIV международная конференция “Супервычисления и математическое моделирование” Саров октябрь 2012 тезисы докладов, [с.54-56]
2. Top500 Supercomputer sites. http://www.top500.org
3. E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. “Open MPI: Goals, concept, and design of a next generation MPI implementation”. // In Proceedings of the 11th European PVM/MPI Users’ Group Meeting, Budapest, Hungary, September 2004, [p. 97–104].
4. Richard L. Graham, Timothy S. Woodall, Jeffrey M. Squyres. “Open MPI: A Flexible High Performance MPI”. // In Proceedings of the 6th International Conference on Parallel Processing and Applied Mathematics in Poznan. Poland. September 2005, [p. 128–139]
5. LINPACK Users Guide, J. Dongarra, J. Bunch, C. Moler and G. W. Stewart, SIAM, Philadelphia, PA, 1979.