Эффективная работа в распределенных вычислительных средах

С. И. Соволев*

Статья посвящена анализу факторов, влияющих на эффективность проведения расчетов в распределенных метакомпьютерных средах, построенных на основе программного комплекса X-Com/VMC. Приводятся описания типичных ситуаций, ведущих к снижению производительности распределенной среды, и рекомендации по их устранению. Обсуждаются направления развития программного комплекса X-Com/VMC, нацеленные на повышение эффективности его работы.

1. Введение

Эффективность — одно из ключевых понятий, с которым сталкивается пользователь при работе на любой высокопроизводительной вычислительной системе. Суперкомпьютер может обладать высокой пиковой производительностью, однако реальные прикладные программы крайне редко способны полноценно задействовать все его возможности. Как правило, эффективность решения той или иной задачи определяется соответствием ее внутренней структуры архитектуре высокопроизводительной системы. В качестве примера можно отметить исследование [1], проведенное на основе данных списка Top500 [2]. Согласно приведенным оценкам, эффективность представленных в списке вычислительных систем зависит в первую очередь от их архитектуры и слабо коррелирует с пиковой производительностью. При этом рассматривается только выполнение теста

*Научно-исследовательский вычислительный центр МГУ

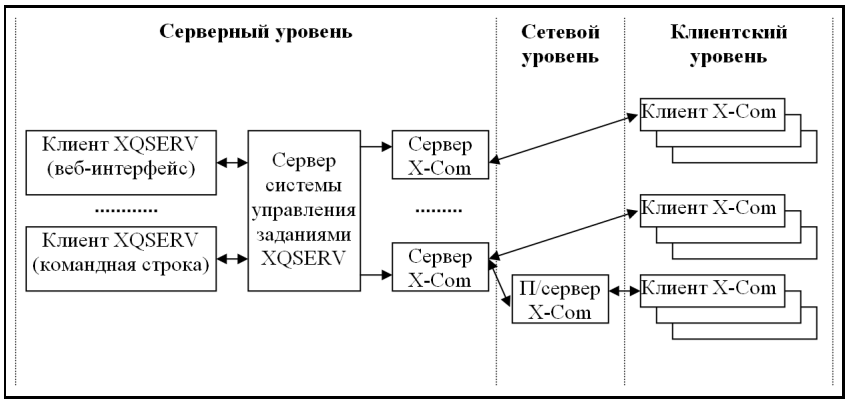

Рис. 1. Архитектура программного комплекса X-Com/VMC

Linpack, на другом классе задач оценки уже могут быть принципиально иными.

Повышение эффективности выполнения конкретного приложения в конкретной вычислительной среде — отдельная и, как правило, весьма нетривиальная задача. При переходе же к распределенным вычислительным средам сложность этой задачи многократно возрастает. Приходится принимать во внимание не только характеристики всего множества вычислительных ресурсов, на которых будет выполняться приложение, но и особенности организации самой среды: ее топологию, надежность и пропускную способность каналов связи, изменчивость состава компонентов среды, накладные расходы, вносимые программными компонентами системы организации распределенных вычислений. Данная статья посвящена обсуждению различных факторов, влияющих на эффективность расчетов в распределенных средах, выявленных в ходе разработки и практического применения программного комплекса X-Com/VMC, а также методам повышения эффективности метакомпьютерных расчетов.

При работе в метакомпьютерной среде, организованной с помощью комплекса X-Com/VMC (рис. 1), любое задание пользователя последовательно проходит три уровня: серверный, сетевой и кли-

ентский. Серверный уровень определяет политику использования доступных ресурсов, сетевой уровень отвечает за топологию среды и транспортировку всех необходимых данных между серверным уровнем и вычислительными узлами, а клиентский уровень обеспечивает взаимодействие с прикладной программой на узлах. Рассмотрим основные факторы, влияющие на эффективность расчета на каждом из уровней.

2. Распределение заданий на серверном уровне

Механизмы серверного уровня определяют множество вычислительных ресурсов, на которых будут производиться вычисления, а также порядок выполнения заданий в метакомпьютерной среде. Распределение заданий на доступные ресурсы в X-Com/VMC производится с помощью одного из двух методов: однопоточного либо многопоточного. Однопоточный метод реализует основную идею метакомпьютерных вычислений, а именно использование максимального объема ресурсов для решения задачи. В этом случае каждое задание, поступающее на вход X-Com/VMC, последовательно запускается на всех доступных узлах. Этот метод позволяет достичь максимальной суммарной производительности подключенных ресурсов. Наибольшая эффективность при использовании однопоточного метода достигается в тех случаях, когда прикладная задача разбита на достаточно большое число вычислительных блоков-порций, при этом время обработки каждой порции невелико по сравнению с общим временем проведения расчета. При такой организации вычислений каждый узел среды вносит свой вклад в расчет, обработав какое-то количество порций.

Однако достижение максимальной суммарной производительности ресурсов вовсе не гарантирует оптимальность их использования. Предположим, что во время расчета вычислительная часть прикладной программы достаточно долго обрабатывает каждую порцию входных данных, при этом время обработки порции существенно зависит от тактовой частоты процессора, а среда объединяет узлы как с высокой, так и низкой частотой CPU. В этом случае вполне возможен вариант, при котором слабые узлы, получив свои порции

в начале расчета, не закончат их обработку до момента завершения всего расчета. Подключение таких узлов для данного расчета окажется нецелесообразным; в то же время, их вполне можно было бы использовать, например, для решения относительно небольших задач либо для тестовых запусков других приложений.

Подобную функциональность реализует многопоточный метод распределения заданий. Идея этого метода состоит в динамическом перераспределении доступных ресурсов для решения тех задач, требованиям которых они соответствуют. Требования прикладной задачи должны указываться пользователями при формировании задания. Это могут быть минимальные/максимальные значения тактовой частоты процессора, его тип, объемы оперативной и дисковой памяти, операционная система и другие. При отсутствии требований задание, как и в однопоточном методе, будет отправлено на все узлы.

Многопоточный метод позволяет разбивать всю среду на сегменты по заданным признакам, при этом каждый сегмент будет работать над своей задачей. Данный метод является более универсальным, однако при его использовании могут возникнуть проблемы уже другого характера. Запуск большой серии заданий, затребовавших для себя самые мощные ресурсы вычислительной среды, очевидно, исключит эти ресурсы из работы над другими задачами, а множество незадействованных узлов может уже не представлять интереса для практической работы. Решение подобных проблем — одно из направлений дальнейшего развития программного комплекса X-Com/VMC. Вариантом решения может быть введение в систему механизмов авторизации пользователей и задание административных ограничений на возможность резервирования определенных ресурсов для групп пользователей и их задач. Если распределенная среда используется для решения только одной прикладной задачи с различными наборами входных данных, то в качестве еще одного варианта решения можно предложить такой режим работы серверных компонентов X-Com/VMC, при котором узлам будут последовательно выдаваться порции всех запущенных в данный момент заданий.

3. Обмен данными между компонентами распределенной среды

Сетевой уровень X-Com/VMC формирует топологию распределенной среды и определяет механизмы обмена данными между серверной частью комплекса и вычислительными узлами. В качестве протокола передачи данных комплекс использует модифицированный протокол HTTP, унаследованный от транспортного уровня системы метакомпьютинга X-Com [3]. Основные причины снижения эффективности расчетов, возникающие на сетевом уровне — неоптимальная топология среды и накладные расходы, вызываемые интенсивными обменами между серверной частью X-Com/VMC и вычислительными узлами.

Оптимальным с точки зрения эффективности является такой вариант проведения расчетов, при котором интенсивность сетевых обменов между узлами и сервером невелика, а время передачи входных и выходных данных во много раз меньше времени обработки этих данных на узлах. При передаче больших объемов данных существенное влияние начинают оказывать свойства каналов, которыми связан сервер с узлами. Низкая пропускная способность, большие задержки и невысокая надежность каналов могут свести на нет всю отдачу от подключения удаленных ресурсов. В этом случае необходимо изменить параметры прикладной задачи, а то и саму ее структуру таким образом, чтобы минимизировать объемы пересылаемых данных.

Повышение интенсивности взаимодействия узлов с сервером, которое может быть вызвано как подключением достаточно большого числа узлов, так и высокой скоростью обработки порций данных, влечет за собой рост нагрузки на серверные компоненты среды и увеличение накладных расходов. Если все узлы общаются с сервером напрямую, до какого-то момента с этой проблемой можно бороться повышением мощности компьютера, на котором работают серверные компоненты. Однако более целесообразным является введение в среду набора промежуточных серверов, позволяющих буферизировать данные между центральным сервером и частью узлов, и тем самым выровнять нагрузку на центральный сервер.

Промежуточные серверы позволяют организовать среду в виде произвольного иерархического дерева. В корне такого дерева будет располагаться центральный сервер X-Com/VMC, листьями всегда будут вычислительные узлы, а во внутренних вершинах будут находиться промежуточные серверы. Помимо буферизации данных, промежуточные серверы также позволяют подключить к расчетам компьютеры, находящиеся внутри закрытой сети, например, узлы вычислительных кластеров. Таким образом, промежуточные серверы увеличивают масштабируемость распределенных расчетов.

4. Подключение вычислительных узлов

Клиентский уровень X-Com/VMC отвечает за взаимодействие с вычислительной часть прикладной задачи непосредственно на узлах метакомпьютерной среды. Инициативу по запросу задания всегда проявляют сами узлы, что позволяет выбрать наилучший в каждом конкретном случае режим участия узлов в расчетах. Очевидно, что с точки зрения прикладной задачи оптимальным режимом для проведения расчетов является монопольный режим, когда задаче полностью предоставляются все ресурсы узла. Однако на практике узлы в таком режиме работы могут быть выделены задаче не всегда, да и время монопольного использования, скорее всего, будет ограничено — узлы вновь могут понадобиться для выполнения обычных задач. Гораздо чаще приходится применять один из методов совместной работы штатных приложений узла и заданий из распределенной среды: фоновый режим работы с пониженным приоритетом либо работа только в те моменты, когда узлы не загружены другими задачами.

Запуск задачи в фоновом режиме основывается на поддержке механизма приоритетов процессов в современных операционных системах. Программа, запущенная в фоновом режиме с пониженным приоритетом, будет получать доступ к ресурсам компьютера только тогда, когда они не требуются приложениям с более высоким приоритетом. В состав программного комплекса X-Com/VMC включен вариант клиента X-Com для ОС семейства MS Windows, реализующий подобную функциональность. Стоит отметить, что примени-

мость данного метода во многом зависит от особенностей той или иной операционной системы и ее настроек на конкретном компьютере, учесть которые крайне сложно.

Более строгим и более «высокоуровневым» методом является запуск распределенного приложения на узле только в те моменты, когда он не занят своей штатной работой. Специальный механизм отслеживает занятость узла; если узел определяется как свободный, он подключается к расчету. Как только штатные процессы узла начинают проявлять свою активность, распределенный расчет на нем останавливается, а все процессы уничтожаются. Различные методы определения незанятости узлов описаны в статье [4].

Понятно, что эффективность использования ресурсов в таком режиме напрямую зависит от их загруженности. Практически то же самое можно сказать и о фоновом запуске. Основная разница между этими двумя методами заключается в том, что при появлении активности на узлах распределенное приложение либо приостанавливается (фоновый режим с пониженным приоритетом), либо полностью прекращается (вычисления в моменты простоя). В первом варианте приложение остается на узле, оно может оказывать влияние на другие процессы узла и мешать их работе. Во втором случае потребуется произвести перерасчет той вычислительной порции, над которой работал узел. Фоновый запуск больше ориентирован на подключение к расчетам рабочих станций, запуск в моменты простоя применяется, как правило, при подключении узлов вычислительных кластеров.

Стоит также отметить еще один способ подключения вычислительных ресурсов к распределенным расчетам — использование возможностей штатных систем управления заданиями. Возможные сложности с работой через такие системы связаны с ограничениями, которые обычно накладываются на максимальное время выполнения приложений, запущенных через них. Поэтому при использовании узлов в подобных режимах необходимо, во-первых, постоянно отслеживать и поддерживать постоянное число процессов распределенного приложения (X-Com/VMC содержит подобный механизм, работающий совместно с системой Cleo [5]), а во-вторых, по возможности самостоятельно задавать ожидаемое время обработ-

ки заданий. Задание максимально возможного времени не всегда оправдано, т.к. в этом случае система управления заданиями может попытаться пропустить сперва более короткие задачи. Наиболее целесообразно указывать время, кратное среднему времени обработки одной вычислительной порции. Тем не менее, по истечению заданного лимита времени процессы распределенного расчета могут быть прерваны системой управления заданиями, что приведет к необходимости их перерасчета. Очевидно, чем больше порций будет обработано за один прием, тем выше будет эффективность.

5. Прочие факторы

Рассмотрев причины уменьшения эффективности, возникающие на всех уровнях работы программного комплекса X-Com/VMC, стоит также упомянуть еще несколько моментов. Так, очень важным является вопрос оптимизации самой прикладной программы. Если доступны исходные тексты программы, перед началом расчета имеет смысл провести тестирование ее выполняемого кода, полученного с помощью различных компиляторов, на тех типах программно-аппаратных платформ, которые будут задействованы в расчете. Разница во времени выполнения различных вариантов программного кода может быть весьма значительной. Так, проводя эксперименты с одной из задач электродинамики, мы обнаружили, что версия программы, полученная с помощью компилятора Intel (icc), работает практически в 4 раза быстрее, чем версия, полученная с помощью компилятора GNU (gcc).

При включении в состав распределенной среды многопроцессорных и многоядерных узлов целесообразно запускать по одному вычислительному процессу прикладной программы на каждое процессорное ядро. Однако следует принимать во внимание, что каждый новый расчетный процесс увеличивает расход оперативной памяти компьютера. Кроме того, одновременный запуск одинаковых процессов может привести к замедлению их работы по сравнению с запуском в единственном экземпляре. Причины замедления зависят как от самой прикладной программы, так и от особенностей аппаратной архитектуры вычислительного узла, и могут быть вызваны неэф-

фективным использованием кэш-памяти процессоров, повышением нагрузки на канал процессор-память и системную шину, конфликтами при доступе к устройствам хранения данных. Перед началом реальных расчетов имеет смысл выяснить, будет ли проявляться подобный эффект на доступных узлах, в какой степени он способен повлиять на процесс вычислений в целом и какое количество вычислительных процессов на одном узле обеспечит достаточную эффективность расчета. Для поиска узких мест в прикладной программе может потребоваться ее исследование с помощью специальных отладочных средств.

Принцип централизованного управления распределенными расчетами также вносит свои ограничения. В ходе экспериментов нам удавалось смоделировать ситуацию, когда центральный сервер X-Com/VMC, взаимодействуя с достаточно большим числом вычислительных узлов, с трудом справлялся с обработкой потока входящих и исходящих данных, следствием чего являлось значительное снижение эффективности модельного расчета. Поэтому одно из дальнейших направлений развития программного комплекса X-Com/VMC — децентрализация, и в частности, распределение нагрузки центрального сервера между несколькими компьютерами. Внедрение такого механизма одновременно позволит повысить и общую надежность комплекса, поскольку в случае отказа одной из серверных машин можно выполнить реконфигурацию серверного пула и продолжить расчет.

6. Заключение

Необходимо еще раз отметить важность абсолютно всех аспектов, влияющих на эффективность расчетов в распределенной среде. Количество задач, решаемых с помощью распределенных технологий, неуклонно растет. Не всегда тривиальным оказывается выделение в задаче независимых блоков, формирование вычислительных порций, организация расчета на ста, пятистах, тысяче компьютеров. Решив эти первичные задачи, можно достигнуть значительной суммарной производительности, подключить множество удаленных узлов, провести эффектный эксперимент — но будет ли он эффек-

тивным?

Список литературы

[1] Воеводин Вл.В. Top500: числом или уменьем// Открытые системы, №10, 2005 г. С.12–15. (http://www.osp.ru/os/2005/10/380430/_p1.html)

[2] Top500 Supercomputing Sites, http://www.top500.org/.

[3] Система метакомпьютинга X-Com, http://x-com.parallel.ru/.

[4] Жуматий С.А., Соболев С.И. Оценка загруженности компьютера в различных UNIX-системах (в настоящем сборнике).

[5] Система управления заданиями Cleo, http://parcon.parallel.ru/cleo.html