Иерархические методы улучшения масштабируемости и эффективности распределенных расчетов в системе метакомпьютинга X-Com *

С.И. Соболев

Система метакомпьютинга X-Com - инструментарий для организации распределенных неоднородных вычислительных сред и проведения расчетов в таких средах. В статье обсуждаются архитектурные решения, примененные в системе X-Com последнего поколения, направленные на улучшение масштабируемости и повышение эффективности распределенных расчетов.

1. Введение

Применение технологий распределенных вычислений для решения больших вычислительно сложных задач стало привычным фактом. Однако уже сейчас на этом пути встают проблемы, связанные с координацией огромного числа взаимодействующих компьютеров и потоков данных между ними. Технологически несложно объединить для работы над единой задачей, скажем, пять суперкомпьютерных комплексов, находящихся в различных регионах России, однако эффективность такого решения будет под вопросом. Предположим, обработка одной независимой порции задачи занимает в среднем 5 минут, а всего над задачей работают 6 тысяч процессоров распределенной среды. Организуя расчеты по обычной клиент-серверной схеме, получаем, что сервер должен корректно обрабатывать до 40 клиентских запросов в секунду. Такая нагрузка на сервер распределенных вычислений имеет тот же порядок, что и суммарная нагрузка на веб-сайты компании Google, обслуживаемые большими фермами серверов. При этом мы не принимаем в расчет затраты ресурсов сервера на генерацию вычислительных порций и обработку результатов. При использовании шифрования данных нагрузка на сервер увеличится еще на порядок. Кроме того, приведенные в качестве примера 6 тысяч процессоров – это реалии сегодняшнего дня, однако уже сейчас в России строятся суперкомпьютеры с десятками тысяч процессорных ядер. Мы естественным образом приходим к необходимости использования распределенных и иерархических технологий при организации самих распределенных вычислений.

В НИВЦ МГУ имени М.В. Ломоносова разрабатывается система метакомпьютинга X-Com [1, 4] – программный инструментарий, предназначенный для организации распределенных неоднородных вычислительных сред и проведения вычислений в таких средах. Базовыми компонентами системы является сервер задачи и клиент X-Com. Сервер задачи отвечает за разбиение конкретной прикладной задачи на независимые вычислительные порции, распределение их на вычислительные узлы и объединение получаемых результатов. Клиенты X-Com, устанавливаемые на узлах, принимают от сервера данные, запускают вычислительные модули прикладной задачи и отправляют результаты обработки порций обратно на сервер.

В последние два года [6, 7] основная работа над системой была сосредоточена на улучшении ее архитектуры с целью повышения масштабируемости и эффективности проводимых с помощью нее расчетов. Наиболее важные реализованные и планируемые архитектурные решения обсуждаются в настоящей статье.

2. Иерархия "сервер задач – сервисы запросов"

Требование обработки сервером задач крайне интенсивного потока запросов от вычислительных узлов приводит к необходимости распределения его работы на отдельные процессы, способные выполняться на разных физических машинах. Естественным представляется разбиение функциональности сервера задач на процесс-координатор, обладающий полной информа-

* Работа выполняется при поддержке гранта Президента РФ для молодых ученых МК-3040.2009.9.

цией о ходе задачи, и сервисы обработки запросов, взаимодействующие непосредственно с вычислительными узлами. Такая архитектура будет особенно эффективна при включенном шифровании данных, требующем дополнительных затрат процессорного времени. В этом случае шифрование и дешифрование может проводиться на выделенных компьютерах, нагрузка на компьютер с процессом-координатором при этом существенно снижается.

В серверной части X-Com выделяются следующие типы запросов и соответствующих им сервисов (Рис. 1). Сервис TSR отвечает за выдачу первоначальной информации о задаче. Файловый сервис – это фактически файловый сервер, у которого клиент запрашивает файлы прикладной задачи и вспомогательные файлы, если они необходимы (отметим, что в качестве альтернативного файлового сервиса может использоваться любой httpd-сервер). Через сервис REQ клиент запрашивает очередную порцию данных у сервера задачи. Результаты выполнения прикладной задачи над данными очередной порции клиент пересылает сервису ASW. Сервис STAT предназначен для предоставления статистической информации о ходе расчета в форматах HTML и XML.

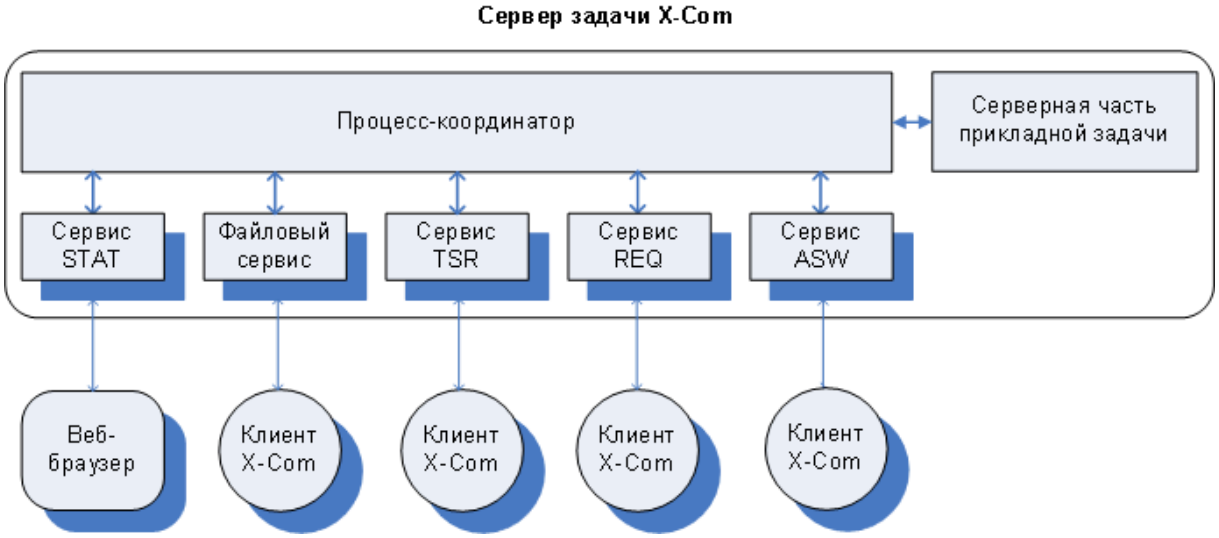

Рис. 1. Структура сервера задачи X-Com

Обмен данными между клиентами X-Com и сервисами запросов производится по специальному протоколу на основе XML поверх стандартного протокола HTTP в сети TCP/IP. Обмен данными между сервисами запросов и процессом-координатором ведется по более простому и экономичному протоколу, причем осуществляться он может как поверх TCP/IP (сервер задач и процессы запросов могут работать на различных компьютерах; поддерживается в любой ОС), так и посредством локальных сокетов UNIX (сервер задач и процессы запросов должны работать на одном компьютере; поддерживается только для ОС семейства UNIX/Linux). Первый способ более универсален, второй обеспечивает более высокую производительность при работе всех серверных процессов на одной физической машине.

Сервер задачи может быть запущен в режиме любого из сервисов вручную (в документации [2] данный режим обозначен как Subserver). При этом в настройках указываются точка доступа (пара хост-порт или имя сокета) процесса-координатора.

3. Иерархия "подсистема управления заданиями – серверы задач"

Работа в масштабной распределенной среде – это, в том числе, огромное число задач и пользователей. Запуск и контроль прохождения заданий в таких средах вручную попросту невозможен – слишком много факторов нужно учесть. Необходимы простые и понятные средства взаимодействия с пользователями, а также механизмы централизованного управления потоками заданий. В системе X-Com эти функции обеспечиваются подсистемой управления заданиям XQSERV. Прототип подсистемы XQSERV был описан в работе [3], настоящая статья описывает актуальное состояние подсистемы.

Подсистема управления заданиями XQSERV состоит из серверной части, отвечающей за распределение задач в вычислительной среде, и клиентской, реализующей пользовательские интерфейсы (Рис. 2). Эти интерфейсы позволяют пользователям работать с вычислительной средой с использованием привычных метафор традиционных высокопроизводительных комплексов – поставить задание или набор заданий в очередь, проконтролировать ход их выполнения, удалить задание из очереди. Базовый метод работы с клиентом подсистемы управления заданиями – вызов клиента из командной строки с необходимыми опциями. Общение между клиентом и сервером XQSERV может осуществляться двумя способами – через сокеты UNIX либо поверх TCP/IP. Первый способ работоспособен только в операционной среде UNIX/Linux и только в том случае, когда клиент и сервер XQSERV запущены на одной физической машине. Этот способ средствами операционной системы обеспечивает получение сервером корректной информации об имени пользователя, вызвавшем клиента. Второй способ обмена данными (TCP/IP) такой информации, вообще говоря, не предоставляет, однако он более универсален и может использоваться при работе клиента на удаленной машине с отличной от серверной операционной системой.

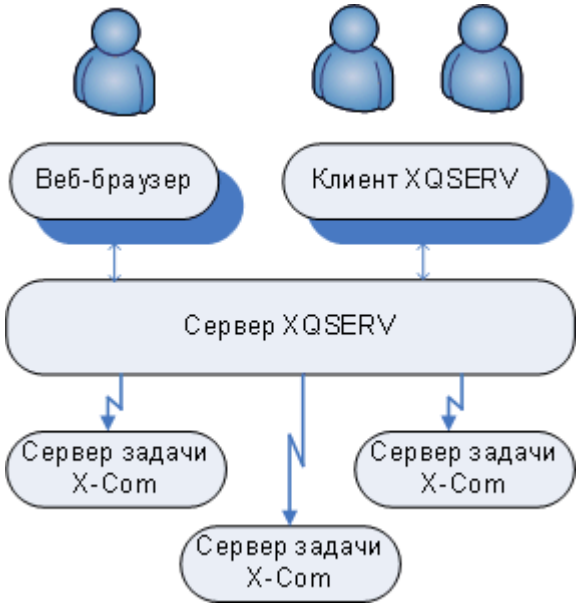

Рис. 2. Подсистема управления заданиями XQSERV

Еще один пользовательский интерфейс, реализуемый подсистемой XQSERV – веб-интерфейс, позволяющий отслеживать общее состояние очереди заданий и ход выполнения каждой задачи из очереди.

Сервер подсистемы управления заданиями XQSERV (управляющий сервер) реализует логику распределения заданий в вычислительной среде. За каждое конкретное задание отвечает свой сервер задачи; с точки зрения управляющего сервера запуск задания означает запуск соответствующего сервера задачи. В простейшем случае управляющий сервер организует линейную очередь, запуская все поступающие от пользователей задания последовательно на всех доступных ресурсах. Такой метод распределения заданий реализует основную идею метакомпьютерных вычислений, а именно использование максимального объема ресурсов для решения задачи, и позволяет достичь максимальной суммарной производительности подключенных ресурсов. Этот метод, однако, не гарантирует максимальной эффективности их использования, в частности, не учитывая требования прикладных задач к тем ресурсам, на которых они будут выполняться. С другой стороны, в ряде случаев может возникнуть необходимость в одновременной работе более одной задачи. Очевидно, необходимы методы динамического разбиения вычислительной среды на сегменты с заданными свойствами, каждый из которых будет выделяться одной задаче, при этом при изменении состава заданий состав сегментов также будет меняться.

4. Иерархическая сегментация среды

Задачи в распределенной среде могут предъявлять различные требования к ресурсам, на которых они должны выполняться. Рассмотрим типичный случай [5]: предположим, что время обработки каждой вычислительной порции достаточно велико, при этом оно 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 самом начале обработки задания, 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 приложений.

При постановке задания в очередь XQSERV можно указать следующие требования к ресурсам: минимальную и максимальную тактовую частоту CPU и/или производительность, минимальный и максимальный объем оперативной памяти, минимальное и максимальное число процессорных ядер, тип операционной системы, процессорную архитектуру. Можно также указать список кластеров, на которых разрешается расчет (принадлежность к кластерам определяется по идентификаторам вычислительных узлов).

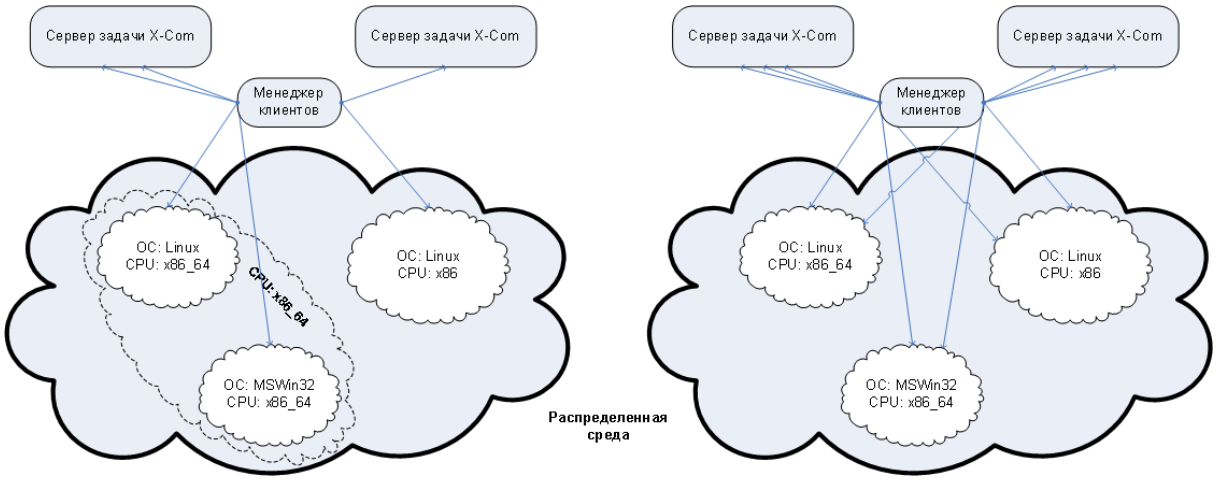

Рис. 3. Примеры сегментации распределенной среды

Чтобы реализовать динамическое перераспределение вычислительных узлов между заданиями, в состав подсистемы управления заданиями XQSERV включен менеджер клиентов. Менеджер клиентов представляет собой процесс-сервер, к которому обращаются при первом запуске все клиенты на вычислительных узлах. Менеджер перенаправляет вычислительные узлы к тем серверам задач, требованиям которых они соответствуют, либо (при отсутствии явно указанных требований) распределяет клиентов поровну на каждую запущенную задачу. Если имеется несколько одновременно работающих задач, и подключающийся узел удовлетворяет требованиям каждой из них, он будет перенаправлен к задаче, запущенной раньше всех. После завершения задачи клиенты на узлах вновь обращаются к менеджеру за новым назначением. Примеры возможной сегментации распределенной среды приведены на Рис. 3.

5. Иерархия промежуточных серверов

Рассмотренные выше механизмы иерархической сегментации среды реализуют логическую группировку вычислительных ресурсов. Однако зачастую ресурсы распределенной среды уже имеют явно выраженную физическую группировку, представляя собой, например, узлы вычислительных кластеров или машины компьютерных классов. Как правило, такие узлы размещены в рамкой закрытой локальной сети, из которой прямой доступ к серверу задач через Интернет может отсутствовать по соображениям безопасности. В этом случае в распределенную среду может быть введен еще один компонент – промежуточный (буферизирующий) сер-

вер. С точки зрения нижележащих узлов промежуточный сервер представляется единственным сервером, доступным данным узлам, с точки же зрения центрального сервера промежуточный сервер сам представляется вычислительным узлом. Помимо организации доступа в закрытую сеть, промежуточный сервер несет еще одну важную функцию – буферизацию обмена данными между "своими" узлами и центральным сервером, снижая тем самым нагрузку на него за счет минимизации числа сетевых соединений (это достигается группировкой нескольких порций в едином запросе) и оптимизации сетевого трафика (клиент на вычислительном узле скачивает необходимые файлы не с центрального, а с ближайшего к нему промежуточного сервера).

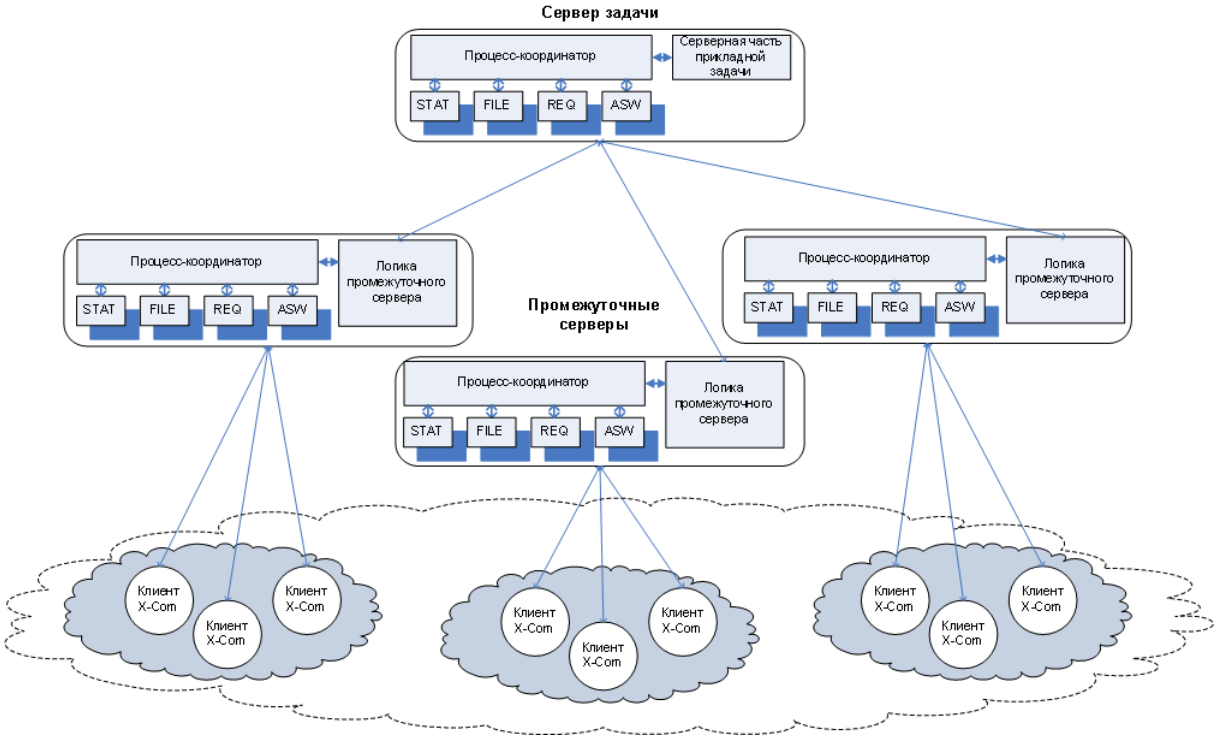

Рис. 4. Организация вычислений с использованием промежуточных серверов

Механизм промежуточных серверов достаточно подробно описан в предыдущих публикациях по системе X-Com, в частности, в [7]. С точки зрения реализации промежуточный сервер – это один из режимов работы сервера задачи (в документации [2] обозначен как Proxy), при котором в качестве "задачи" подключается модуль, реализующий общение с вышестоящим сервером (полностью аналогично клиентской части X-Com). Промежуточные серверы позволяют формировать распределенную среду в виде дерева с произвольным числом ярусов, в корне которого будет центральный сервер X-Com, в узлах – промежуточные серверы, а листьями будут являться клиенты X-Com (Рис. 4).

Хотелось бы отметить, что анализ модели промежуточных серверов и их реализация существенно способствовали внедрению иерархических методов и в другие компоненты системы X-Com.

6. Иерархия "вычислительный клиент – рабочие процессы"

Клиентская часть X-Com (вычислительный клиент), устанавливающаяся на всех узлах распределенной среды и взаимодействующая с сервером задачи, отвечает за запуск вычислительной части прикладной задачи. Архитектура современных вычислительных узлов позволяет запускать сразу несколько вычислительных процессов на одном узле (обычно по числу процессорных ядер). В настоящее время для реализации этой возможности на каждом узле запускается по несколько клиентов X-Com, которые взаимодействуют с сервером задач и запускают вычислительные процессы независимо друг от друга. Логичным представляется переход к версии клиента X-Com, поддерживающей многопроцессорные и многоядерные конфигурации. Такой

клиент будет однократно скачивать все рабочие файлы прикладной задачи на узел (это позволит уменьшить сетевой трафик и нагрузку на сервер задач) и запускать необходимое число параллельных вычислительных процессов (Рис. 5).

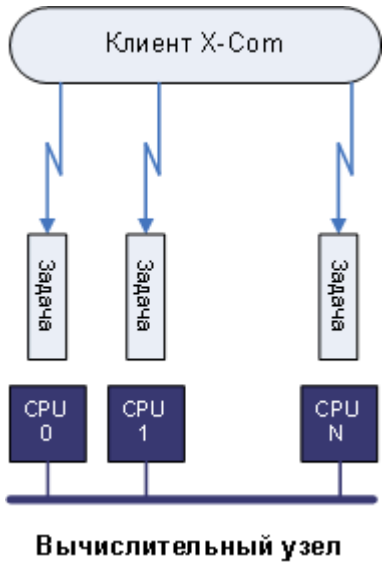

Рис. 5. Перспективная архитектура клиента X-Com

Архитектура такого клиента слегка напоминает архитектуру промежуточного сервера. Автоматически появляется возможность запускать на узлах как чисто последовательные программы, так и программы, написанные с использованием OpenMP. Число запускаемых процессов на узле может указываться при установке клиента, а может быть параметром при запуске каждой задачи.

7. Заключение

Внедрение иерархических и распределенных методов обработки данных на всех уровнях архитектуры системы X-Com способствует снижению накладных расходов системы метакомпьютинга и, как следствие, увеличивает эффективность ее работы. Распределение функциональности сервера задач на независимые процессы, способные работать на отдельных компьютерах, уменьшает загрузку процессора и каналов связи каждой из машин. Подсистема управления заданиями позволяет разбить все множество имеющихся вычислительных ресурсов на подмножества, эффективно решающие имеющиеся прикладные задачи, а кроме того, обеспечивает высокоуровневые пользовательские интерфейсы для работы в распределенной среде. Промежуточные серверы оптимизируют передачу больших объемов данных и снижают загрузку центрального сервера распределенных вычислений. Иерархическая организация клиентской части также позволит экономить сетевой трафик и оптимизировать загрузку вычислительных узлов.

Литература

1. Семейство программ X-Com (официальный сайт) [Электронный ресурс]. -Режим доступа: http://x-com.parallel.ru/ (дата обращения: 5.11.2009).
2. Руководство пользователя системы X-Com^2 [Электронный ресурс]. -Режим доступа: http://x-com.parallel.ru/documentation.html (дата обращения: 5.11.2009).
3. С.И. Соболев. Управление заданиями в Виртуальном метакомпьютерном центре на основе технологий X-Com. Распределенные вычисления и Грид-технологии в науке и образовании. Труды второй международной конференции (Дубна, 26 - 30 июня 2006 г.), Дубна: ОИЯИ, 2007, с. 401-404.

4. Соболев С.И. Использование распределенных компьютерных ресурсов для решения вычислительно сложных задач // Системы управления и информационные технологии. 2007, №1.3 (27). С. 391-395.

5. С.И. Соболев. Эффективная работа в распределенных вычислительных средах // Численные методы, параллельные вычисления и информационные технологии. 2008. 249-258.

6. С.И. Соболев. Архитектура нового поколения системы метакомпьютинга X-Com // Распределенные вычисления и Грид-технологии в науке и образовании. Труды третьей международной конференции. Дубна, 30 июня - 4 июля 2008 г. С. 123-127.

7. Воеводин Вл.В., Жолудев Ю.А., Соболев С.И., Стефанов К.С. Эволюция системы метакомпьютинга X-Com // Вестник Нижегородского государственного университета им.Н.И. Лобачевского. 2009, №4. С. 157-164.