

# АРХИТЕКТУРА НОВОГО ПОКОЛЕНИЯ СИСТЕМЫ МЕТАКОМПЬЮТИНГА X-COM<sup>1</sup>

С.И. Соболев

НИВЦ МГУ имени М.В. Ломоносова,  
119991, г. Москва, Ленинские горы, д.1, стр.4, НИВЦ МГУ

Система метакомпьютинга X-Com [1], разработанная в НИВЦ МГУ, представляет собой инструментарий низкого уровня, предназначенный для организации распределенных вычислительных сред, а также для адаптации и поддержки выполнения прикладных задач в таких средах. Система X-Com активно используется для решения вычислительно сложных и практически важных задач из области молекулярного моделирования [2], электромагнитной динамики [3], биоинженерии и других. Значительный опыт, полученный при организации распределенных вычислений с помощью системы X-Com, позволил сформировать ряд требований и пожеланий к системе и определить пути ее дальнейшего развития.

На основе анализа требований была спроектирована архитектура системы X-Com следующего поколения (Рис. 1). Прямоугольники на рисунке соответствуют процессам системы, причем каждый такой процесс может выполняться на отдельной физической машине. Общение между процессами осуществляется по сетям TCP/IP. В

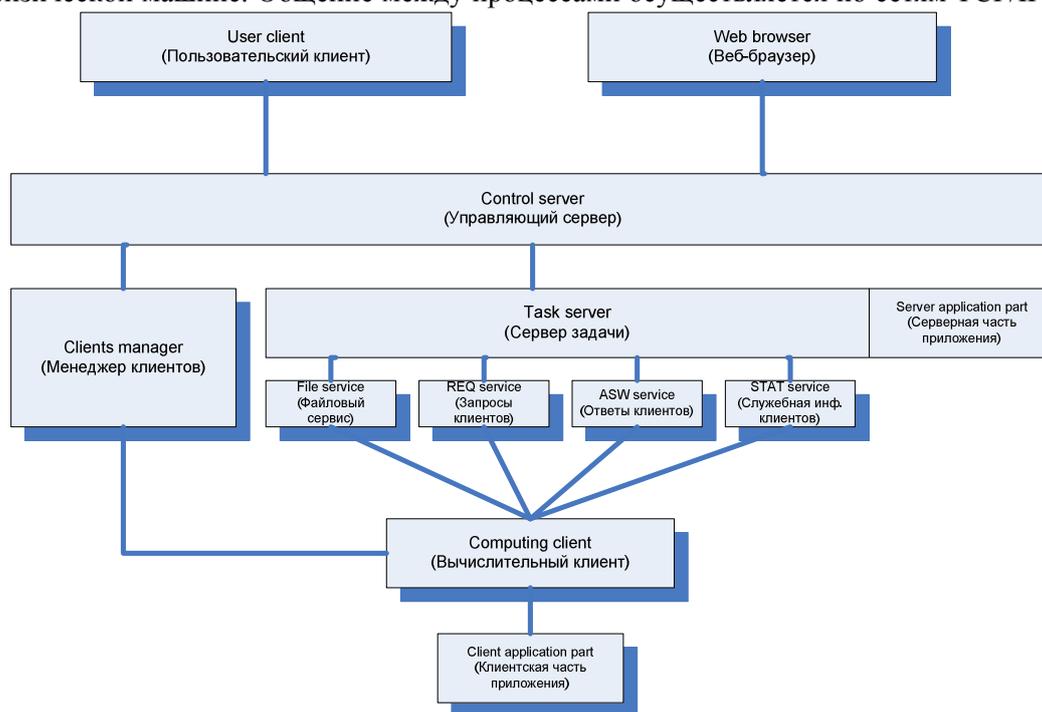


Рис. 1. Архитектура системы X-Com нового поколения

<sup>1</sup> Разработка проекта ведется в рамках государственного контракта № 02.514.11.4035 "Разработка и апробация на реальных приложениях технологий параллельного программирования распределенных вычислительных сред со сверхтерафлопным диапазоном производительности".

архитектуре выделяются три уровня – пользовательский уровень, серверный уровень и уровень вычислительных узлов.

На пользовательском уровне находятся процессы пользовательских клиентов и веб-браузеров. Пользовательский клиент – компонент системы, обеспечивающий интерфейс между пользователями и системой метакомпьютинга. С его помощью пользователь может поставить задачу в очередь на выполнение и отслеживать ее состояние. При постановке задания в очередь пользователь формирует файл описания задания, в котором указываются входные данные прикладной задачи и требования к ее запуску, в том числе требования к ресурсам, на которых задача будет выполняться.

Веб-браузер – обычный интернет-браузер, с помощью которого пользователь может следить за ходом расчетов и состоянием распределенной среды в наглядной форме. При наличии веб-интерфейса к прикладной задаче с помощью браузера также может осуществляться постановка задачи в очередь.

На серверном уровне находятся процессы сервера задачи, управляющего сервера и менеджера клиентов. Сервер задачи – компонент, обеспечивающий выполнение одной конкретной прикладной задачи. Он взаимодействует с серверной частью прикладной задачи, которая генерирует порции данных и осуществляет финальную обработку результатов, приходящих от клиентов с вычислительных узлов. Сервер задачи реализует логику выдачи порций клиентам. С клиентами на вычислительных узлах сервер задачи взаимодействует не напрямую, а через процессы-сервисы, обозначенные на рисунке как File service, REQ, ASW и STAT. Файловый сервис – это фактически файловый сервер, у которого клиент запрашивает файлы прикладной задачи и вспомогательные файлы, если они необходимы. Через сервис REQ (request) клиент запрашивает очередную порцию данных у сервера задачи. Результаты выполнения прикладной задачи над данными очередной порции клиент пересылает сервису ASW (answer). Сервис STAT (statistics) обеспечивает обмен контрольной информацией между клиентами и серверной частью. Выделение сервисов REQ, ASW и других отдельными процессами обусловлено требованием распределения нагрузки между компонентами системы. Это будет особенно важно при интенсивных обменах данными между клиентами и серверной частью при включенном шифровании (см. далее). В этом случае сервисы берут на себя всю предварительную обработку входящих и исходящих порций. В качестве файлового сервера может применяться любой http-сервер.

Управляющий сервер – компонент, обеспечивающий управление прохождением заданий в распределенной среде, в частности, реализующий логику очередей заданий или же логику одновременного выполнения задач. На каждое запускаемое задание управляющий сервер порождает свой сервер задачи. Управляющий сервер транслирует команды – например, на срочное прекращение расчета – серверу задачи, которые затем передаются клиентам, работающим с данным сервером.

Менеджер клиентов – компонент, к которому все клиенты обращаются при первом запуске. Менеджер клиентов перенаправляет вычислительные узлы к тем серверам задач, требованиям которых они соответствуют. По завершению работы с заданием клиенты вновь обращаются к менеджеру.

На уровне вычислительных узлов функционирует клиент X-Com, который получает от сервера задач очередную порцию данных, запускает клиентскую (вычислительную) часть прикладной задачи и передает результаты вычислений обратно на сервер. Очередную задачу клиент получает через менеджера. В ответах на запросы клиентов, формируемые сервером, может быть предусмотрена смена адресов

сервисов, с которыми клиент работал прежде. Таким образом, можно, например, динамически перенаправить клиента для работы с другой задачей.

Подобная архитектура позволяет, в частности, реализовать возможность управления узлами. Каждый узел будет периодически отсылать информацию о своем состоянии, получая в ответ команду на продолжение текущего расчета либо на совершение какого-либо действия, например, на смену задачи. Благодаря такой возможности динамического перераспределения узлов становится возможным реализовать один из вариантов многозадачности в распределенной среде. При появлении очередной задачи среда выделяет часть узлов для ее решения, обеспечивая таким образом одновременное выполнение нескольких задач. Другой режим многозадачности, также заложенный в новой архитектуре, заключается в учете требований прикладной задачи, накладываемых на вычислительные узлы, на которых она может выполняться.

Для улучшения масштабируемости и надежности системы при проведении расчетов новая архитектура предусматривает разнесение процессов серверной части системы на физически отдельные машины. Дополнительные возможности включают в себя механизмы передачи файлов между серверной частью X-Com и вычислительными узлами, расширение функциональности клиентской части и другие.

Архитектуру системы X-Com можно рассматривать как множество пар клиент-сервер, обменивающихся данными по сетям, а следовательно, для системы характерны те же аспекты безопасности, что и для любых других сетевых сервисов: возможность подмены одного или нескольких компонентов системы, например, вычислительного клиента или одного из серверов, анализ и модификация трафика между компонентами системы, атаки типа Denial of Service. Для решения описанных проблем вводится регистрация клиентов с получением ими сертификатов. Инфраструктура сертификатов реализуется согласно стандартам X.509. При обнаружении некорректного поведения клиента или какого-либо другого компонента среды его сертификат отзывается. Анализ некорректного поведения клиентов – это отдельная задача, обнаружение такого поведения может производить либо автоматически – в этом случае для каждой задачи необходимо заводить шаблоны «правильного» поведения клиентов – либо вручную по результатам мониторинга.

Опциональное шифрование данных вводится при обменах между серверным уровнем и пользовательским/вычислительным уровнем. При этом мы предполагаем, что все серверные компоненты находятся внутри доверенной сети, и шифрование между ними обеспечивать не нужно.

В новой архитектуре заложена полноценная реализация промежуточных (буферизирующих) серверов. В иерархическую структуру сервер-клиенты вводится еще один компонент – промежуточный сервер, располагающийся между основным сервером и частью вычислительных узлов. С точки зрения основного сервера промежуточный сервер представляет собой обычный узел с достаточно высокой производительностью, с точки зрения ниже лежащих узлов промежуточный сервер является единственным сервером распределенной среды. Функции промежуточного сервера:

- Переадресация запросов между клиентами и сервером. Это необходимо в тех случаях, когда, например, клиенты запущены на узлах вычислительного кластера, находящегося внутри закрытой сети. Промежуточный сервер в этом случае выполняет функцию прокси-сервера.
- Буферизация, объединение нескольких запросов в один. Промежуточный сервер может запросить у центрального сразу несколько вычислительных порций в одном пакете, и аналогичным образом вернуть результаты. При

интенсивных расчетах это позволит снизить нагрузку на серверную часть системы из-за минимизации числа сетевых соединений.

- Автоматическая синхронизация файлов задачи и вспомогательных файлов с центральным сервером.

Дополнительной функциональностью может быть возможность переключения промежуточного сервера между режимами: брать задачи с центрального сервера либо из своей локальной очереди заданий.

При разработке новой архитектуры одним из ключевых требований была необходимость сохранить все сильные стороны текущей версии X-Com, а именно оперативность развертывания распределенных вычислительных сред, поддержку различных программно-аппаратных платформ, различные режимы подключения вычислительных узлов, простоту адаптации прикладных задач. В качестве единой технологической основы для реализации всех компонентов системы была выбрана технология программирования Perl. Perl доступен для практически всех программно-аппаратных платформ, он пригоден для написания переносимого кода. Perl фактически по умолчанию входит в дистрибутивы Linux, имеются хорошие реализации для платформы MS Windows. Perl также добавляет еще один интерфейс для программирования серверной части прикладных задач.

Все протоколы обмена данными между компонентами новой версии X-Com базируются на основе XML. Для облегчения перехода пользователей со старой версии системы X-Com на новую планируется в целом сохранить интерфейсы взаимодействия системы и прикладной задачи.

#### Литература:

1. Система метакомпьютинга X-Com, <http://x-com.parallel.ru/>
2. Соболев С.И. Использование распределенных компьютерных ресурсов для решения вычислительно сложных задач // Системы управления и информационные технологии. 2007, №1.3 (27). С. 391-395./
3. М.Ю. Медведик, Ю.Г. Смирнов, С.И. Соболев. Параллельный алгоритм расчета поверхностных токов в электромагнитной задаче дифракции на экране // Вычислительные методы и программирование. 2005. Том 6, №1. 86-95.