ИНТЕГРАЦИЯ СИСТЕМЫ МЕТАКОМПЬЮТИНГА X-COM С СИСТЕМАМИ УПРАВЛЕНИЯ ПРОХОЖДЕНИЕМ ЗАДАНИЙ СУПЕРКОМПЬЮТЕРНЫХ КОМПЛЕКСОВ[1]

С.И. Соболев

НИВЦ МГУ имени М.В. Ломоносова
sergeys@parallel.ru

Система метакомпьютинга X-Com [1], разрабатываемая в НИВЦ МГУ имени М.В. Ломоносова, представляет собой программный инструментарий для организации распределенных неоднородных вычислительных сред и проведения расчетов в таких средах. В качестве вычислительных ресурсов, составляющих основу распределенных сред, могут использоваться компьютеры практически любого типа, от домашних и офисных машин до высокопроизводительных многопроцессорных серверов. Часто распределенные среды строятся на основе узлов и сегментов суперкомпьютерных комплексов. Это позволяет задействовать сразу несколько высокопроизводительных систем для работы над "тяжелыми" задачами в тех случаях, когда ресурсов одного комплекса оказывается недостаточно. Однотипная структура программно-аппаратной платформы таких узлов упрощает их использование в составе распределенной среды как в плане установки программного обеспечения, так и при организации непосредственно вычислительного процесса.

Изначально в X-Com было заложено несколько базовых сценариев использования вычислительных узлов в рамках кластерных систем. В простейшем случае предполагалось, что узлы для распределенного расчета выделяются монопольно. Такой способ оправдан в случае относительно небольшой загрузки вычислительной системы. Однако в настоящее время таких ситуаций практически не бывает – большинство суперкомпьютерных комплексов работают со стопроцентной загрузкой. По той же причине оказывается малоприменим и запуск распределенных расчетов на узлах в моменты простоя, когда они не используются для других приложений. Кроме того, оба этих способа неявно предполагают проведение определенных действий или получение разрешений со стороны администрации вычислительных комплексов. Запуск клиентской части X-Com через штатные системы управления прохождением заданий (СУПЗ) также имеет свои ограничения – возможность запуска только однопроцессорных задач, необходимость отслеживать состояние очереди заданий сторонними средствами, повышенный расход трафика (если несколько клиентов работают на одном узле, то каждый из них независимо от других скачает с сервера исполнимый код задачи и т.д.).

Тем не менее, очевидно, что работать все-таки нужно через штатные СУПЗ – это единственный способ прозрачно и эффективно использовать доступные ресурсы. Кроме того, использование СУПЗ позволило бы запускать в рамках X-Com не только однопроцессорные задачи, но и приложения, использующие MPI и другие технологии параллельного программирования. Поэтому в систему X-Com были добавлены модули взаимодействия и интеграции с наиболее распространенными СУПЗ кластерных систем Torque, Cleo, LoadLeveler, а также с ПО промежуточного уровня Grid-сред Unicore. Последний модуль дает возможность подключить к расчетам

[1] Работа выполняется при поддержке научно-технической программы Союзного государства СКИФ-ГРИД и гранта Президента РФ для молодых ученых МК-3040.2009.9.

сегменты Grid-сред, использующие соответствующее ПО для доступа к удаленным вычислительным ресурсам.

С точки зрения архитектуры X-Com разработанные модули используют интерфейсы, механизмы и функции промежуточных серверов X-Com [2]. В частности, модули осуществляют буферизацию входящих и исходящих порций между центральным (вышестоящим) сервером и целевой системой. Сервер X-Com может быть запущен в режиме взаимодействия с СУПЗ путем указания соответствующих настроек в файле инициализации, при этом запуск осуществляется на головной машине целевого кластера (Рис. 1). Отметим, что клиент X-Com при такой организации расчетов не используется – его функциональность фактически берет на себя промежуточный сервер.

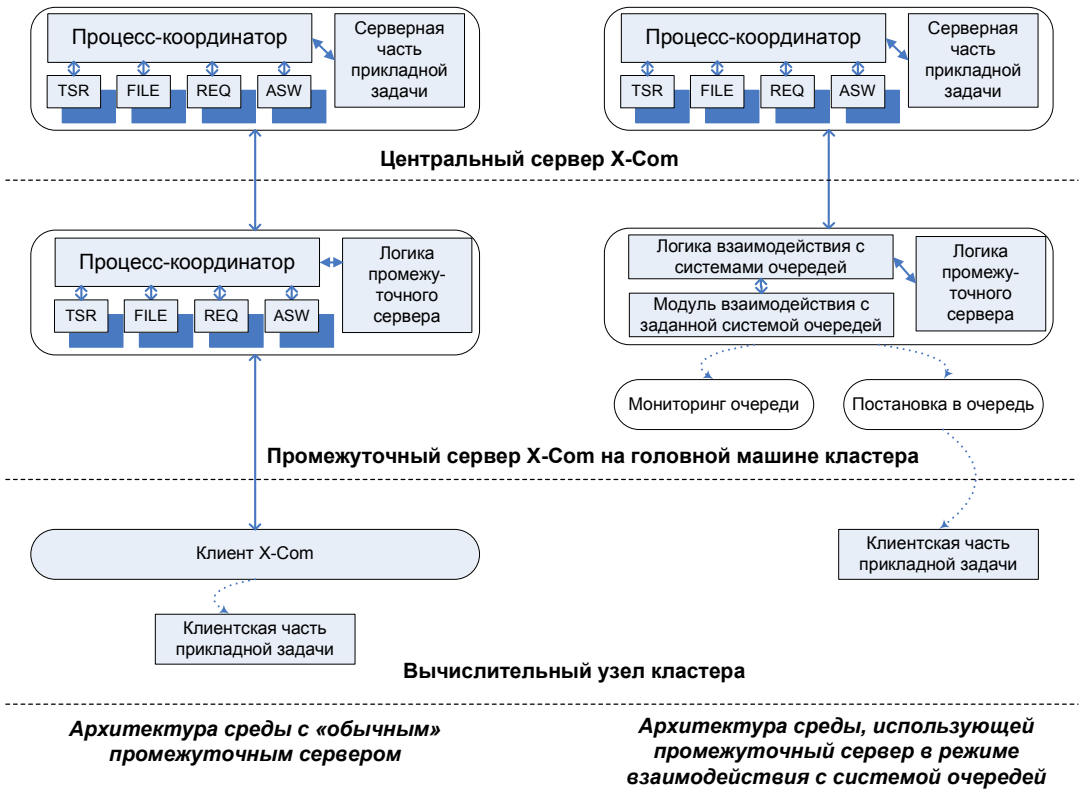

Рис. 1. Архитектура распределенной среды X-Com с использованием "обычного" промежуточного сервера и режима взаимодействия с кластерной СУПЗ

Для непосредственного взаимодействия с СУПЗ модули X-Com вызывают утилиты командной строки соответствующей СУПЗ. Так, для Torque мониторинг и постановка задачи в очередь осуществляется командами qstat и qsub, для LoadLeveler – командами llq и llsubmit, для Unicore во всех случаях используется вызов клиента ucc с нужными опциями. Постановка задач в очередь Cleo производится с помощью команды mpirun, мониторинг с целью экономии ресурсов реализован через чтение служебных файлов, автоматически создаваемых этой СУПЗ. Для каждой СУПЗ генерируется файл описания задания в нужном формате.

Для любой поддерживаемой СУПЗ применимы следующие настройки:

- максимально допустимое время счета задачи;
- число процессоров, на которых будет запущено задание;
- интервал обновления данных о состоянии очереди;
- число попыток поставить задание в очередь, по истечении которого будет осуществлен переход к следующей порции.

Также для каждой поддерживаемой СУПЗ существует набор собственных настроек. Например, для Cleo можно указать название очереди, для Unicore – целевую систему и т.д.

Имеется ряд настроек, отвечающий за буферизацию входящих и исходящих порций. Эти настройки применяются как для "обычного" промежуточного сервера, так и для режима работы совместно с СУПЗ:

- максимальное число входящих порций в одном запросе, буферизуемых промежуточным сервером. Это же значение в настоящий момент используется как максимально допустимое число заданий в очереди;
- минимальное число порций во входном буфере, после достижения которого промежуточный сервер пополнит буфер;
- максимальный размер буфера исходящих порций, после достижения которого начнется отправка порций вышестоящему серверу;
- число исходящих порций, отправляемых вышестоящему серверу в одном запросе.

Для апробации разработанных модулей и новых режимов работы X-Com проводились серии распределенных экспериментов с использованием приложения John the Ripper [3], осуществляющего перебор паролей, зашифрованных стандартными средствами UNIX. При наличии достаточно большого набора зашифрованных паролей каждый из них может расшифровываться отдельно и независимо от других, поэтому задача элементарно делится на порции достаточно высокой вычислительной сложности, при этом размеры передаваемых данных крайне невелики (десятки байт на каждую порцию). Таким образом, выбранная задача идеально отвечает схеме метакомпьютерных вычислений.

Для проведения расчетов были задействованы 4 суперкомпьютерные системы:

- СКИФ МГУ "Чебышев" (г. Москва);
- СКИФ Урал (г. Челябинск);
- СКИФ Cyberia (г. Томск);
- вычислительный кластер УГАТУ (г. Уфа).

Центральный сервер X-Com был запущен на головной машине кластера СКИФ МГУ "Чебышев". На головных машинах каждого из кластеров запускался промежуточный сервер X-Com в режиме взаимодействия с системой очередей кластера. На кластере СКИФ МГУ "Чебышев" использовался модуль взаимодействия с Cleo, на кластерах СКИФ Cyberia и СКИФ Урал – модули взаимодействия с Torque, на кластере УГАТУ – модуль взаимодействия с LoadLeveler.

На всех кластерах были установлены одинаковые настройки буферизации порций. Размер окна входящих порций составлял 20 порций, размер окна исходящих порций – 5 порций. Согласно этим параметрам, изначально каждый промежуточный сервер запрашивал у центрального сервера по 20 порций и далее всегда поддерживал запас в 20 порций к обработке, при этом число задач в очереди кластера независимо от их состояния также поддерживалось равным 20. При превышении числа готовых порций, равного 5, производилась отправка результатов пакетами по 5 порций. Отметим, что число порций в пакетных запросах на обработку было плавающим (от 1 до 20), число же готовых порций в ответных пакетах всегда было фиксированным (5).

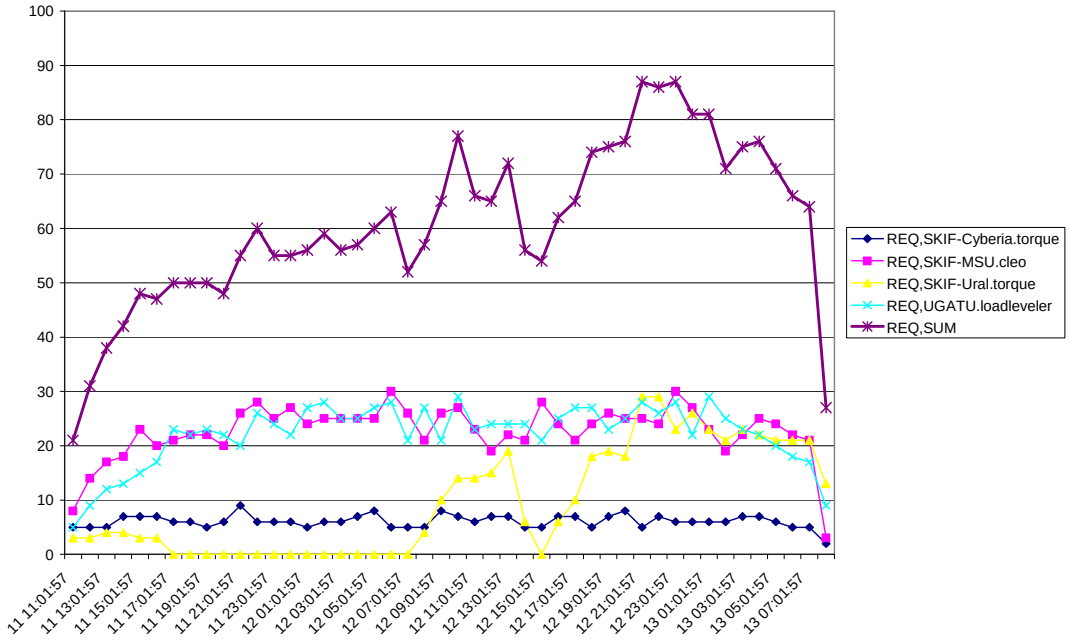

Рис. 2. График запросов исходных данных промежуточными серверами в распределенном запуске John the Ripper

Было проведено две серии вычислительных экспериментов на двух различных наборах паролей. В качестве первого набора использовался файл со словарем, размещенный на сайте John the Ripper и составляющий основу его внутренней базы [4]. Ввиду того, что подбор таких паролей, очевидно, не мог

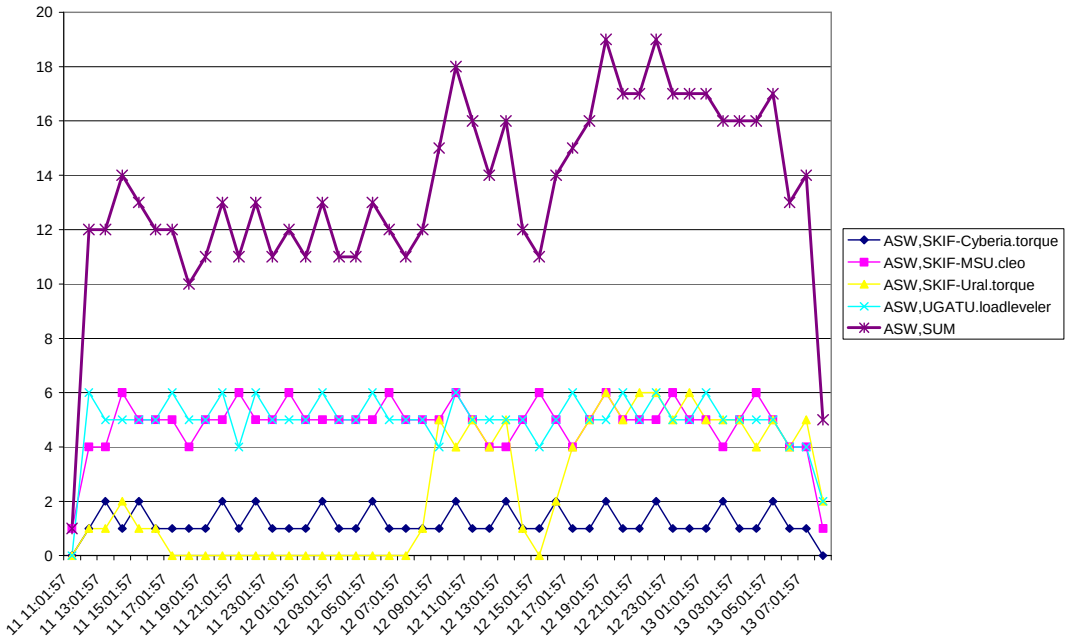

Рис. 3. График возвращения результатов промежуточными серверами в распределенном запуске John the Ripper

представлять большой сложности, каждая порция содержала 50 паролей, а время обработки одной порции было ограничено 15 минутами. Ограничения были заданы как при постановке задачи в очереди кластеров, так и непосредственно в скрипте, вызывающем John the Ripper. В случае неудачи (невозможности за указанное время полностью расшифровать заданный набор) в качестве результата возвращался пустой файл. Также 15-минутное ограничение позволило использовать очередь test на кластере СКИФ МГУ "Чебышев", которая, как правило, более свободна, чем другие очереди на этой машине. Первая серия экспериментов продолжалась 15 часов, из 129000 паролей было расшифровано 95.7%.

Более интересной оказалась вторая серия экспериментов. Набор паролей для нее был сгенерирован с помощью программы на языке Си. Генерировались пароли длиной от 4 до 10 символов, дополнительным условием при генерации была указана возможность легкого произнесения и запоминания паролей, при этом они не должны были совпадать со словарными выражениями. В каждой порции содержался только один пароль, а время подбора была ограничено 1 часом. Эксперимент продолжался более 45 часов, при этом из 3000 паролей было подобрано всего 728, т.е. 24.3%.

Графики интенсивности запросов второй серии экспериментов приведены на рис. 2 и 3. По горизонтальной оси отложено время, по вертикальной – число запросов за данный интервал времени. Видно, что в данном эксперименте сложность обработки каждой порции была примерно одинаковой. Рост суммарных графиков начиная со 2-й половины эксперимента можно объяснить освобождением ресурсов суперкомпьютера СКИФ Урал. Регулярный "пилообразный" характер линий на графике выходных результатов (рис. 3) вызван фиксированным размером выходного окна. На рис. 4 изображен вклад каждого из вычислительных комплексов в решение задачи (по числу обработанных и полученных сервером порций).

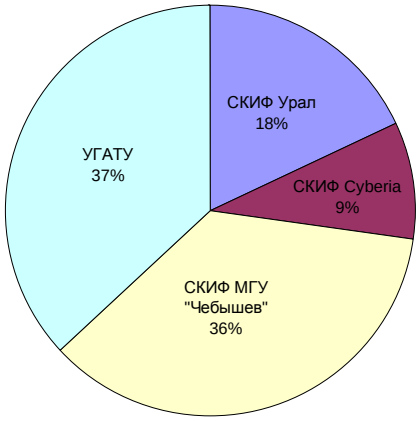

Рис. 4. Вклад каждого из вычислительных комплексов в задаче распределенного запуска John the Ripper

Безусловно, в рамках проведенных экспериментов расшифровка паролей имела лишь академический характере. Тем не менее, решение такой задачи может быть полезно на практике для проверки стойкости паролей пользователей больших систем коллективного доступа в тех случаях, когда пароли фактически являются единственным методом защиты доступа. Стойкость паролей имеет крайне важное значение при обеспечении безопасности таких систем.

Использование модулей взаимодействия X-Com с системами очередей, как уже говорилось выше, в настоящее время видится как основной способ подключения

к распределенным расчетам ресурсов высокопроизводительных вычислительных комплексов. Однако такая модель подключения ресурсов ставит перед системой X-Com ряд новых вопросов и задач. Одна из таких задач – поиск новых оценок эффективности проведения распределенных расчетов. Долгое время в качестве таковых в X-Com применялись две характеристики, условно называемые "серверной" и "клиентской" эффективностью. "Серверная" эффективность позволяла оценить накладные расходы системы метакомпьютинга и вычислялась как отношение времени расчета порции на узле ко времени, затраченному серверной частью X-Com на полную обработку этой порции, включая генерацию, обработку клиентских запросов, передачу данных. При работе через СУПЗ к накладным расходам системы метакомпьютинга добавляются накладные расходы системы очередей, включая время ожидания задания в очереди, которое может быть достаточно велико. Очевидно, что данная оценка при прочих равных будет принимать достаточно высокие значения для относительно свободной вычислительной системы и достаточно низкие значения для загруженной системы, однако об "эффективности" в таком контексте говорить будет уже некорректно.

Вторая используемая оценка – "клиентская эффективность" – вычислялась как отношение числа порций, розданных клиентам, к числу полученных результатов. Эта оценка характеризовала потери порций, вызванные, как правило, сбоями работы клиентской части X-Com на вычислительных узлах. В случае стабильной работы всех клиентов такая оценка имеет высокие значения, при наличии сбоев (перезапуск клиентов, обрывы связи и т.д.) значение ее снижается. Следует отметить, что вследствие особенностей стандартного алгоритма распределения порций в X-Com (после отдачи последней порции сервер X-Com начинает заново раздавать те порции, результаты обработки которых до сих пор не получены) данная оценка почти никогда не достигает 100%, хотя при грамотной организации вычислительного эксперимента ее значения достаточно высоки. При использовании механизмов взаимодействия с СУПЗ возникает новый потенциальный источник "потерь" порций – входные и выходные буферы промежуточных серверов.

Еще одна интересная задача – динамическое определение размеров буферов промежуточных серверов, работающих совместно с СУПЗ, размеров окон приема и передачи порций, а также числа одновременно поддерживаемых в очереди заданий, при которых может быть достигнуть оптимум между накладными расходами на обмен данными, неизбежными потерями порций, оседающих в буферах, и загрузкой конкретной вычислительной системы. Решением этой задачи может быть увеличение числа ручных настроек параметров буферизации, возможность менять их в ходе работы промежуточного сервера, а также использование информации о доступных ресурсах, полученной от СУПЗ.

[1] Система метакомпьютинга X-Com (официальный сайт проекта), http://x-com.parallel.ru

[2] 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. №4. 2009. С 157

[3] John the Ripper, http://www.openwall.com/john/

[4] John the Ripper (библиотека стандартных паролей), http://download.openwall.net/pub/wordlists/all.gz