

Построение системы метакомпьютинга Horde на основе Microsoft Silverlight и .NET

А.В. Захаров

В данной статье обозначены основные проблемы, стоящие перед современными системами метакомпьютинга, и предложен подход, позволяющий минимизировать данные проблемы, на основе технологий .NET и MS Silverlight, который лег в основу прототипа системы метакомпьютинга Horde.

1. Введение

Совсем недавно компьютер Roadrunner компании IBM превысил рубеж производительности в 1 Петафлоп и получил право считаться самым быстрым суперкомпьютером в мире [1]. Однако такой же порядок суммарной производительности уже был достигнут ранее с помощью системы распределенных вычислений BOINC [2], построенной на основе обычных персональных компьютеров, взаимодействующих через интернет. Действительно, всемирную глобальную сеть можно рассматривать как гигантский виртуальный суперкомпьютер, обладающий колоссальной мощностью [3]. Однако в силу ряда причин эффективная реализация данной концепции крайне затруднительна.

Системы, объединяющие вычислительные ресурсы интернета или интранета, носят название систем метакомпьютинга [3]. Основные проблемы при разработке систем такого рода связаны, прежде всего, со свойствами используемых ресурсов [4]:

- Неоднородность ресурсов. Компьютеры, объединенные в вычислительную сеть, имеют различные ОС, различную производительность, различные наборы установленного программного обеспечения, различную скорость соединения.
- Динамичность состава среды. В любой момент каждый ресурс может быть потерян, но это не должно приводить к аварийной остановке вычислительного процесса. Более того, к среде могут добавляться новые ресурсы, которые нужно уметь оперативно подключать к текущему расчету.
- Масштабность. Суммарное число активных процессоров, общий объем памяти, число одновременно работающих пользователей и приложений в таких вычислительных средах может быть огромным. Чем больше значения подобных параметров, тем сложнее обеспечить эффективность расчета.
- Распределенность. Вычислительные ресурсы системы могут быть удалены друг от друга на тысячи километров, что вызовет большую латентность в их взаимодействии.
- Различная административная принадлежность. Обладателями ресурсов являются частные лица и организации, которые в большей части случаев не желают отдавать их в полное распоряжение или обременять себя настройкой своих компьютеров для участия в вычислениях. Следовательно, работа программного обеспечения, необходимого для участия в вычислениях, должна быть незаметна, а его установка предельно проста.

На данный момент среди систем метакомпьютинга можно выделить три основных класса [4]:

1. Универсальные. Системы этого класса характеризуются тем, что могут потенциально решать любые возложенные на них задачи. Самым известным представителем данного класса является система Globus. Но за универсальность всегда надо платить, и ценою стала чрезвычайно сложное развертывание этих систем.
2. Проблемно-ориентированные. Системы этого класса создаются лишь для решения набора предопределенных “тяжелых” задач. Наиболее известными проектами, относящимися к этому классу, являются: Great Internet Mersenne Prime Search (поиск простых чисел Мерсенна), SETI@home (поиск внеземного разума на основе анализа полученных из космоса сигналов), LHC@home (анализ информации, полученной от большого адронного коллайдера). Развертывание таких систем на порядок легче, чем систем первого клас-

са. Однако при необходимости добавить новую задачу, придется приостановить работу всей системы и внести массу изменений. С другой стороны, задачи, решаемые этими системами, настолько сложны, что вычисления идут в течение месяцев и даже лет.

3. Инструментарии. Данные системы предоставляют разработчикам инструменты для быстрого решения задач довольно широкого класса в распределенных средах. Они, также как и проблемно-ориентированные системы, относительно просты в развертывании, и зачастую являются одновременно и платформами для их разработки. Так, например, платформа BOINC предоставляет набор библиотек, определяющих организацию распределенных вычислений, способ хранения данных и обеспечения безопасности, на основе которого строятся конкретные проблемно-ориентированные системы [5]. Другие системы, как например, система X-COM, созданная в НИВЦ МГУ, позволяют исполнять распределенные вычислительные задачи, написанные в рамках программного интерфейса (API) системы.

В данной статье будут рассмотрены основные технологии и архитектурные принципы, которые легли в основу разрабатываемой системы метакомпьютинга Horde, которая по своим свойствам ближе всего 3 классу (инструментарии). Как и система X-COM, она предоставляет пользователям распределенную вычислительную среду для выполнения их вычислительных задач, написанных в рамках API системы. При сохранении лучших практик, накопленных в ходе создания аналогов, в системе Horde предлагаются новые решения проблем, связанных с различной административной принадлежностью и неоднородностью вычислительных ресурсов, основанные на особенностях технологии Silverlight.

2. Достоинства технологии Silverlight с точки зрения разработки систем метакомпьютинга

В октябре 2008 компания Microsoft выпустила новую платформу для разработки RIA (Rich Internet Application) под названием Silverlight [6]. Silverlight представляет собой кросс-браузерный, кросс-платформенный плагин для web-браузеров, содержащий в себе уменьшенный набор библиотек.NET [7]. Он позволяющий создавать сложные графические интерфейсы, работать с различными web-сервисами и использовать для этого .NET языки (C#, VB, а теперь и динамические языки Python и Ruby). В отличие от полного дистрибутива .NET, занимающего около 300 Мбайт, размер плагина – 4.6 Мбайт.

Рассмотрим преимущества платформы Silverlight для создания клиентов системы для распределенных вычислений в среде интернет:

- Silverlight широко распространен. Летняя олимпиада в Пекине транслировалась в интернете с помощью видео плеера, написанного на Silverlight, и по полученной статистике плагин установлен теперь на каждом четвертом компьютере в мире, а в некоторых европейских странах и на каждом втором [5]. Следовательно, мы имеем широко распространенную среду для исполнения программ.
- Silverlight кросс-платформенный (пока только для Windows и Mac OS, поддержка Linux анонсирована на сентябрь 2009) и кросс-браузерный плагин (Internet Explorer, FireFox, Safari, Google Chrome).
- Приложение Silverlight автоматически загружается при посещении сайта, на котором оно расположено, и сразу после загрузки готово к использованию. Следовательно, для установки клиента системы распределенных вычислений, написанного на Silverlight, и подключения к вычислениям достаточно просто зайти на сайт.
- Скорость исполнения Silverlight приложений практически совпадает со скоростью .NET приложений. Производительность компьютера, замеренная на тесте Linpack, реализованного на классическом для вычислительных задач языке C¹, в среднем на 15% больше, чем на портированной на платформу Silverlight версии этого теста.
- Silverlight поддерживает многопоточность, что позволяет использовать многоядерность современных процессоров.

¹ При использовании компилятора Microsoft Visual C++.

Перечисленные преимущества существенно помогут решить проблемы, связанные с неоднородностью и различной административной принадлежностью вычислительных ресурсов, о которых говорилось ранее. Именно эти проблемы тяжелее всего решить разработчикам системы самостоятельно, так как они порождаются, как правило, свойствами ресурсов, а также ограничением технологий, на которых строятся системы. Стоит особо отметить, что Silverlight это технология, ориентированная на персональные компьютеры, и использовать ее на кластере или сервере будет невозможно. Следовательно, и сама система Horde также ориентирована на использование именно на ПК.

Аналогичным потенциалом обладает и технология Adobe Flash. Однако, несмотря на большую распространенность, она уступает в скорости исполнения, имеет менее развитый язык программирования и не поддерживает многопоточность. С другой стороны, архитектура системы в будущем позволит создать клиента и на Flash.

3. Архитектура системы Horde

Любая вычислительная задача, исполняемая системой Horde, состоит из клиентской и серверной части. Клиентская часть задачи пишется на Silverlight, а серверная на .NET.

Основную вычислительную нагрузку несет клиентская часть, именно в ней должны быть сосредоточены все вычисления, которые необходимо выполнить.

Основными функциями серверной части задачи являются:

- Формирования порций входных данных для клиентской части.
- Мониторинг прогресса вычислений.
- Обработка результатов полученных от клиентской части.

Добавление вычислительных задач в системе Horde основано на архитектурном шаблоне проектирования “встраиваемый программный модуль” (плагин) [8]. То есть, как для клиентской, так и серверной части определены интерфейсы, на основе которых они встраиваются в систему, не требуя ее перекompиляции.

В общем виде архитектура системы Horde представлена на рис. 1.

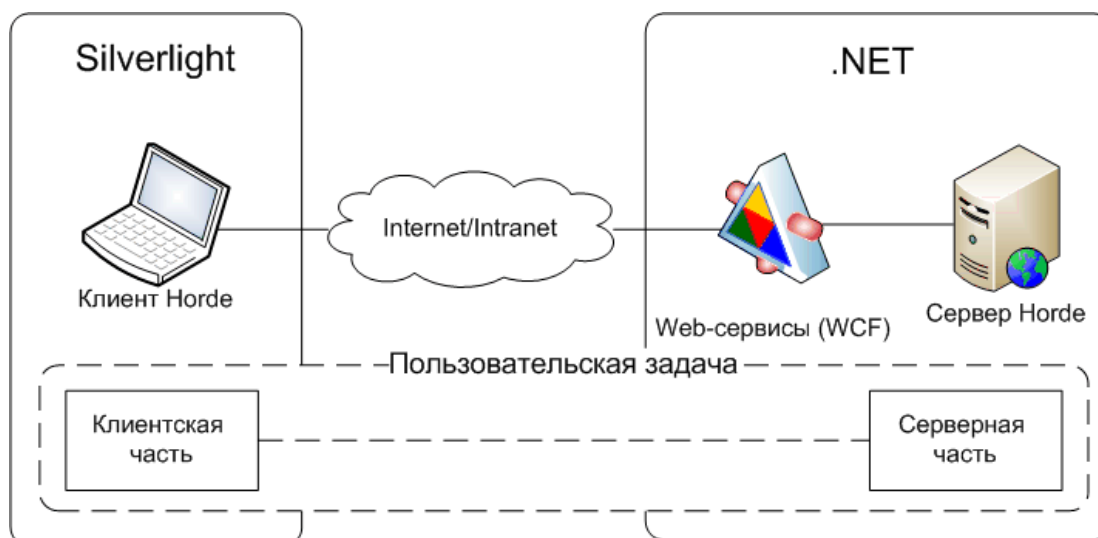


Рис. 1. Архитектура системы Horde.

Система состоит из двух основных частей:

- Клиент системы. Silverlight приложение, которое производит подгрузку клиентских частей задачи, запуск клиентских задач, оценку производительности клиентской машины (на основе некоторой достаточно легковесной вычислительной задачи¹) и скорости соединения (на основе пересылки блока данных определенного размера).

¹ Тестовая задача, используемая для определения производительности, на данный момент не определена.

- Сервер системы. .NET приложение, которое позволяет управлять задачами через web-интерфейс (добавление, удаление, запуск, мониторинг). В нем определен набор web-сервисов (WCF¹), посредством которых производится коммуникация с клиентом.

В системе Норде определено два основных механизма для ввода исходных данных:

1. Разработчик вычислительной задачи создает свой собственный графический интерфейс для ввода исходных данных на базе технологии Silverlight и загружает его вместе с клиентской и серверной частью вычислительной задачи при ее регистрации в системе.
2. Исходные данные считываются через внешний web-сервис, логика работы с которым определена в серверной части задачи. Данный способ эффективен в том случае, когда исходных данных очень много и они хранятся в базе данных, размещенной на сервере пользователя или некотором дата-центре.

Не исключена и возможность их совместного использования.

Для сохранения данных возможны два сценария:

1. Частичные результаты вычислений сериализуются и сохраняются в базе данных, а по окончании вычислений конкатенируются в итоговый результат. Итоговый результат можно получить в формате JSON через REST web-сервис, либо скачать в виде файла.
2. Частичные результаты, полученные от клиентов системы, передаются внешнему web-сервису, логика работы с которым определена в серверной части задачи. Данный способ эффективен в том случае, когда объем результатов очень большой или требуется сохранить их во внешней системе.

На рисунке 2 проиллюстрирован случай, когда для получения исходных данных и записи результатов используются внешние web-сервисы.



Рис. 2. Процесс пересылки исходных данных и результатов вычислений через внешние web-сервисы.

¹ WCF (Windows Communication Foundation) – технология Microsoft для построения сервис-ориентированных приложений.

Основными преимуществами системы Horde являются:

- Простота развертывания системы. Для того чтобы принять участие в вычислениях достаточно лишь зайти на определенную web-страницу проекта, после чего будет автоматически установлена последняя версия клиента системы.
- Устойчивость к динамичному добавлению и отключению вычислительных ресурсов.
- Коммуникации на основе протокола HTTP.
- Разработка пользовательских задач на современных языках программирования (C#, VB.NET, Iron Python).

4. Заключение

На данный момент уже создан первый работающий прототип системы, для которого было реализовано несколько простых задач (поиск простых чисел в заданном интервале, задача 3dep из дистрибутива X-COM [9]).

Для проведения нагрузочного тестирования системы с использованием небольшого числа компьютеров был написан эмулятор клиентов системы, который моментально генерирует фиктивные результаты вычислений определенного размера. Результаты, полученные в ходе эксперимента, приведены на рисунке 3.

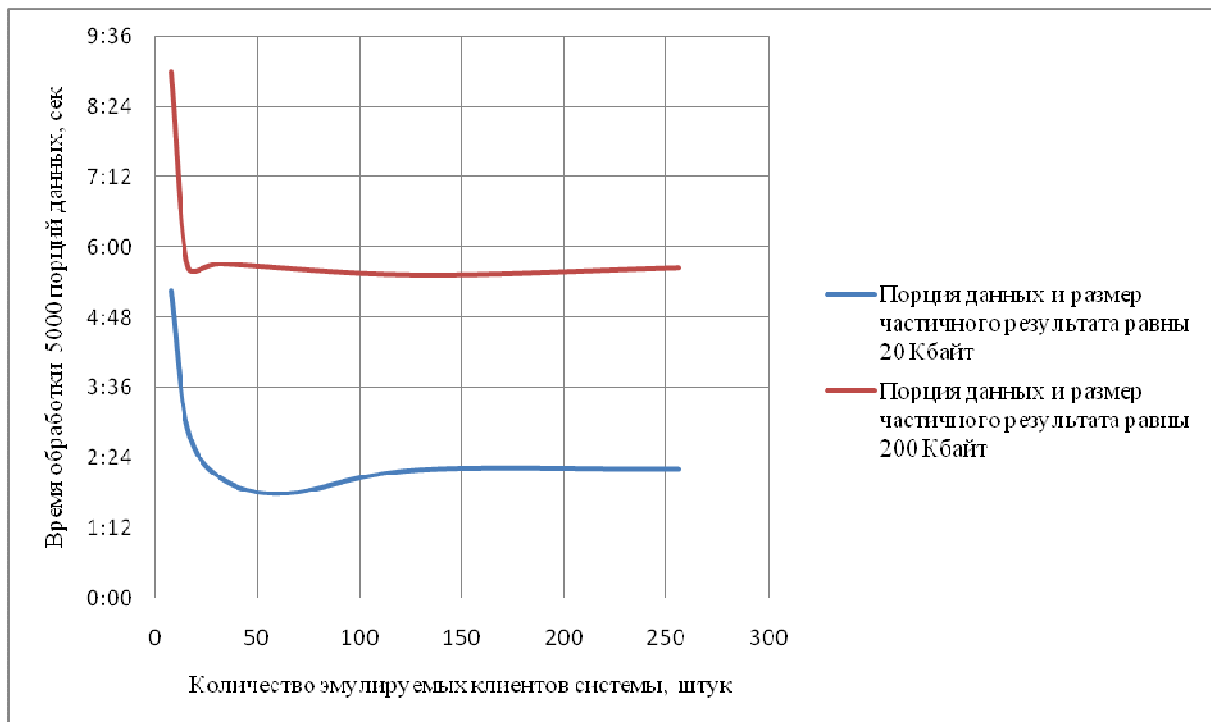


Рис. 3. Результаты работы эмулятора клиентов системы.

Для данного эксперимента использовались 2 компьютера, связанных посредством Fast Ethernet. На одном из них (Intel Pentium D 2800МГц, ОЗУ 2 Гбайта) был развернут сервер прототипа системы Horde, а на другом (AMD Turion X2, ОЗУ 2 Гбайта) запущен эмулятор клиентов системы.

В дальнейшем планируется выбрать одну из последних реальных задач, решенных на базе системы X-COM, и адаптировать ее для работы в рамках системы Horde. После чего произвести ее расчет, объединив ресурсы нескольких вычислительных классов Южно-Уральского государственного университета.

Литература

1. Top 500 RoadRunner Performance/Linpack Data [<http://www.top500.org/system/performance/9707>], 18.11.2008.
2. BOINC stats [http://boincstats.com/stats/project_graph.php?pr=bo], 14.01.2009
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. -Спб.: БХВ-Петербург, 2004. - 608 с.
4. Воеводин Вл.В., Филамофитский М.П. Суперкомпьютер на выходные // Открытые системы, №5, 2003. -с. 43-48.
5. Документация системы BOINC: [<http://boinc.berkeley.edu/trac/wiki>], 27.07.2008.
6. Microsoft объявляет о выходе Silverlight 2: [<http://www.microsoft.com/rus/news/issues/2008/10/silverlight2.mspx>], 20.10.2008
7. Документация по технологии Silverlight: [[http://msdn.microsoft.com/en-us/library/bb404700\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/bb404700(VS.95).aspx)], 16.10.2008
8. Rice D., Foemmel M., Plugin: [<http://martinfowler.com/eaCatalog/plugin.html>], 12.06.2004
9. Система метакомпьютинга X-Com [<http://x-com.parallel.ru>], 14.01.2009