

Список литературы

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2003. – 461 с.
2. Knuth D.E. The Art of Computer Programming. Vol. 2: Seminumerical Algorithms. Addison-Wesley Professional; 3 edition. November 14, 1997. – P. 501.
3. Intel® Math Kernel Library. Reference Manual. September 2007.
4. NVidia CUDA. Reference Manual. Version 2.3. July 2009.
5. Craig S., Marc Y., Brian J. Working with Microsoft® Visual Studio® 2005. Microsoft Press.

В.П. Гергель, К.А. Чабан

Нижегородский государственный университет им. Н.И. Лобачевского

СИСТЕМА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ DCS

В настоящее время многие организации имеют в своем распоряжении довольно крупную компьютерную сеть, объединяющую десятки и даже сотни компьютеров. С ростом такой сети все актуальнее становится вопрос об эффективности использования вычислительных ресурсов, которые она объединяет. Зачастую имеющиеся компьютеры загружены не полностью. Поэтому вполне естественно задействовать простаивающие ресурсы для выполнения какой-либо полезной работы – например, использовать их для решения некоторой трудоемкой задачи. Развитие данного направления привело к возникновению концепции *utility computing*.

Основная идея данной концепции состоит в использовании независимых компьютеров, объединенных в сеть, в качестве единой параллельной машины или виртуального суперкомпьютера. Эта идея имеет ряд преимуществ, например низкая стоимость суммарных вычислительных ресурсов.

Быстрый рост сети Интернет позволяет применить эти концепции в намного большем масштабе. К тому же настольные

ПК в корпоративных и домашних условиях нагружены далеко не полностью – обычно используется только одна десятая часть мощности. Поэтому возникает интерес к использованию вычислительных мощностей, которые доступны в виде свободных циклов работы центрального процессора.

В сфере программного обеспечения грид на сегодняшний день доступно множество различных инструментариев для организации распределенных вычислений. В качестве примеров можно привести инструментарии Globus Toolkit [<http://www.globus.org>], Alchemi .NET Framework [<http://www.alchemi.net>], X-Com [<http://x-com.parallel.ru>]. В настоящей работе рассматривается система Distributed Calculation System (DCS), с помощью которой можно организовать распределенную вычислительную среду из имеющихся компьютерных ресурсов. Согласно распространенной классификации данную систему можно отнести к классу *инструментариев*¹, так как она включает в себя базовые средства разработки распределенных приложений и поддерживает необходимые для их работы операции.

Сегодня наблюдается быстро растущий интерес к технологиям грид со стороны коммерческих предприятий и исследовательских организаций. Операционная система Microsoft Windows широко используется большинством фирм. И для массового распространения грид и распределённых вычислений необходимо разработать инструментарии, поддерживающие данную операционную систему. Это позволило бы эффективно использовать свободную вычислительную мощность настольных компьютеров и рабочих станций посредством создания виртуального вычислительного ресурса, стоимость которого значительно ниже стоимости традиционных суперкомпьютеров.

¹ Существующие программные средства грид можно разделить на следующие категории: *программные средства*, *базовые средства грид (инструментарии)* и *платформы*. Такая классификация отражает полноту решения задачи создания и обслуживания грид, поддержки работы пользователей и разработки грид-приложений. Более подробную информацию можно найти на ресурсе <http://www.gridclub.ru>.

Описание системы. Система DCS предназначена для создания распределённой вычислительной сети в соответствии с принципами клиент-серверной архитектуры. Distributed Calculation System состоит из 3 основных компонентов:

1. **DCS Server** – центральная часть системы, которая представляет собой ASP.NET Web-сервис, связанный с экземпляром Microsoft SQL Server 2008. Основные функции сервера можно разделить на 2 группы:

Методы для работы клиента администратора:

- добавление вычислительных заданий;
- получение результатов расчёта;
- оперативное получение информации о прогрессе расчётов;
- сбор сведений о предоставленных сервисом вычислительных ресурсах, их активности;
- on-line настройка параметров сервиса;
- управление ресурсами сервиса.

Методы для работы вычислительного клиента:

- регистрация клиента на сервере,;
- получение части (порции) задания для расчёта;
- передача частичного результата расчёта на сервис для хранения.

2. **DCS Administration Client** – клиент администратора системы. Данное приложение является основным инструментом пользователя при работе с системой.

Можно выделить 3 основных логических модуля данного компонента вычислительной сети. **Менеджер задач**, предназначенный для формирования пакетов вычислительных заданий, отправки их на сервер и последующего получения результатов вычислений. С помощью менеджера производится также удаление заданий и приостановка расчёта. **Монитор активности вычислительных клиентов**. Данный модуль предоставляет сводную информацию об основных параметрах клиентов, подключённых к сервису. С его помощью можно оценить мощность вычислительного ресурса, сформированного с помощью инструментария DCS. **Мастер настроек сервиса** – служебная утилита, помогающая удалённо изменять настройки сер-

виса. С её помощью можно осуществлять перезапуск сервиса, удаление информации о клиентах, задачах, а также чтение журнала работы.

3. ***DCS Calculation Client*** – компонент системы, который организует расчет отдельных порций входных данных с помощью прикладной программы. Клиенты системы DCS отвечают за выполнение следующих действий:

- выполнение вычислений для имеющегося блока (порции) входных данных прикладной задачи;
- запрос новых блоков данных для вычислений от сервера;
- передача результатов вычислений на сервер.

Система DCS подразумевает использование имеющихся вычислительных ресурсов для обработки клиентским приложением, запущенным на компьютере, блоков некоторой задачи, предоставляемых приложению сервером. Следовательно, задача должна допускать декомпозицию входного набора данных на независимые части или, иными словами, должен иметь место параллелизм по данным. После полной обработки данных блоков задача должна считаться полностью решенной. Это требование вызвано необходимостью организации процесса решения задачи в наиболее универсальном виде.

Требование независимости порций данных вызвано идеологией организации любой распределенной вычислительной системы, при которой каждый конкретный блок задачи будет обрабатываться на каком-либо доступном системе компьютере. Причем другие порции данных, возможно, будут обрабатываться на других компьютерах, не связанных с данным. Хотя потенциальный доступ к данным возможен, он существенно замедлит процесс выполнения задачи ввиду низких скоростей передачи данных по сети по сравнению с локальным доступом к данным.

Вычислительное задание в рамках системы DCS является совокупностью следующих элементов. *Вычислительный модуль* – исполняемый файл со всеми зависимостями (динамическими библиотеками, файлами конфигурации и т.д.), реализующий алгоритм расчета. Данное приложение используется клиентами для проведения вычислений. Система DCS не накладывает ограничений на инструменты реализации данного компонента.

Тем не менее пользователь должен учитывать возможные проблемы запуска вычислительного модуля на предоставленных ему вычислительных ресурсах, так как особенностью применения инструментариев данного класса является неоднородность распределённой среды. Другими словами пользователь системы должен обеспечить возможность запуска своего приложения на компьютерах, составляющих вычислительную сеть DCS.

Generator.exe – генератор аргументов задачи. С помощью данной служебной утилиты реализуется алгоритм декомпозиции входного набора данных на независимые друг от друга части. Данное приложение предназначено для генерации входных параметров вычислительного модуля в зависимости от номера текущей порции задания. *Generator.exe* реализуется прикладным программистом системы.

Служебные данные задачи – набор параметров и ограничений, используемых компонентами системы во время вычисления, включающий:

- имя задачи;
- число порций разбиения;
- краткое описание задачи;
- максимальный срок выполнения, после которого задание считается отменённым.

Использование системы. Для демонстрации возможностей системы рассмотрим задачу визуализации четырёхмерных фракталов Julia методом трассировки лучей.

Фракталы Julia генерируются в 4-мерном пространстве кватернионов с помощью итерационной формулы $z_{n+1} = z_n^2 + c$, где c – какая-либо константа.

Для визуализации в привычное для человека трехмерное пространство осуществляется проекция 4-мерного пространства на трехмерное.

Трассировка лучей – это метод машинной графики, позволяющий создавать фотореалистичные изображения любых трехмерных сцен. Трассировка лучей моделирует прохождение лучей света через изображаемую сцену. Фотореализм достигается путем математического моделирования оптических свойств света и его взаимодействия с объектами. Сначала отдельные объекты распо-

лагаются в трехмерном пространстве-сцене, а также задаются физические и оптические свойства их поверхностей и их цвет. Затем определяется, где будут расположены источники света и наблюдатель. И, наконец, с помощью программы трассировки лучей создается математическая модель сцены, света и наблюдателя, с помощью которой вычисляется цвет каждого пикселя графического изображения, получаемого на экране дисплея.

Данная задача была решена с помощью системы DCS. Ниже приведено время эксперимента в зависимости от числа используемых клиентов:

Число клиентов	Время расчёта
1 клиент	23:35
2 клиента	12:45
3 клиента	08:14
4 клиента	06:08

Для экспериментов использовались компьютеры: Intel Core 2 Quad Q9550 2.83 Ghz, 2.5 Gb RAM, Windows XP SP3.

М.В. Гиркин, Д.И. Крыжановский, М.А. Фёдоров

Волгоградский государственный технический университет

МОДЕЛИРОВАНИЕ РАБОТЫ РЕЗИСТИВНОЙ ПАМЯТИ RRAM НА КЛАСТЕРНЫХ СИСТЕМАХ МЕТОДОМ МОНТЕ-КАРЛО

Как известно, суммарное время вычислений, осуществляемых одним последовательным вычислительным устройством, складывается из времени, непосредственно затрачиваемого на выполнение арифметических операций, и времени, затрачиваемого на операции с памятью. Развитие вычислительной техники, наблюдающееся в последние годы и сопровождающееся широким внедрением многоядерных процессоров и других параллельных систем, позволило значительно увеличить скорость вычислений, снизив как среднее время выполнения отдельных