

МОДЕЛИРОВАНИЕ РАБОТЫ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ КЛАСТЕРНЫХ СТРУКТУР

Дежурко А.М., Верхотуров А.Е.

Белорусский государственный университет, г. Минск

Введение

В настоящее время с ростом потребности в больших вычислительных мощностях все большее значение приобретают распределенные вычислительные системы и программное обеспечение, способное эффективно их использовать. Параллельные программы имеют специфические цели и структуру, что налагает определенные требования на процесс их разработки и тестирования [1]. Возникают задачи нахождения алгоритмов, обеспечивающих оптимальную загрузку узлов параллельной системы и наилучшую схему передачи данных. Решение этих задач в настоящее время достигается путём проведения множества экспериментов и подробного анализа их результатов [1,5]. Тестирование параллельных алгоритмов на реальных вычислительных сетях достаточно трудоёмко, и, следовательно, требует значительных материальных затрат. Кроме того, существует проблема проектирования и конфигурирования параметров распределенной вычислительной сети таким образом, чтобы достигалась максимальная вычислительная мощность при решении конкретного класса задач. Таким образом, для увеличения эффективности разработки высокопроизводительных параллельных программ требуется разработать инструменты, позволяющие упростить процесс тестирования параллельных алгоритмов и автоматизировать выбор параметров вычислительной сети.

Постановка задачи

Распределенная вычислительная сеть – это сложный для моделирования объект, имеющий комплексную, многоуровневую структуру и описывающийся множеством различных характеристик. При построении модели выбраны основные, наиболее важные параметры системы, определяющие ее производительность: топология, быстродействие вычислительных узлов, латентность вычислительных узлов, латентность устройств коммутации, скорость передачи данных, загрузка сети. Реализация модели включает описание структуры сети и разработку алгоритма маршрутизации, алгоритма обмена данными и алгоритма учета загрузки сети. Для анализа параллельного алгоритма его необходимо представить в виде набора специальных команд, которые интерпретируются моделирующей программой.

Разработка модели структуры сети

Для описания топологии распределенной вычислительной сети используется объектно-ориентированная модель, так как она дает возможность наиболее наглядно представить структуру сети. Все элементы сети разделены на три типа: вычислительный узел, линия связи, устройства коммутации. Причем вычислительный узел также является устройством коммутации. Каждый элемент сети обладает своим уникальным идентификатором и хранит уникальные идентификаторы своих «соседей». Такая структура обеспечивает двухстороннюю связь элементов сети, что позволяет легко находить маршруты передачи данных. Для сохранения структуры сети выбран формат XML, как наиболее распространенный и удобный для реализации. Данный формат позволяет сохранить сконфигурированную сеть в виде списка элементов сети с указанием связей входящих в него элементов.

Алгоритм обмена данными

Модель реализует парный синхронный обмен данными [3,4]. На рисунке 1 схематично изображен алгоритм обмена данными между двумя узлами.

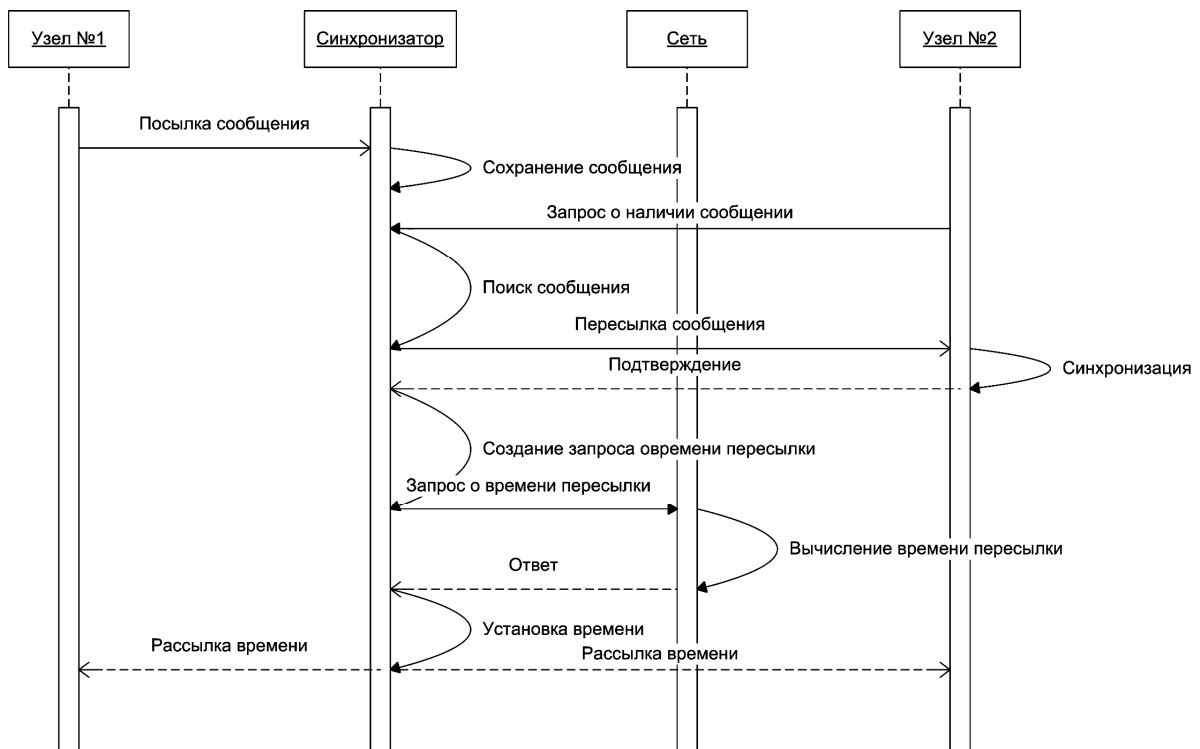


Рис. 1. Алгоритм пересылки данных от вычислительного узла №1 вычислительному узлу №2.

Для моделирования такого обмена реализовано следующее поведение объектов:

- вычислительный узел №1 посылает сообщение объекту «Синхронизатор» о готовности передать данные и ожидает подтверждения от принимающей стороны.
- вычислительный узел №2 посылает сообщение о готовности к приему данных
- «Синхронизатор» пересылает сообщение вычислительному узлу №2
- вычислительный узел №2 синхронизирует время отправки данных
- «Синхронизатор» формирует запрос о времени пересылки данных и отправляет его объекту «Сеть»
- объект «Сеть» на основе сформированного запроса, «зная» маршрут пересылки данных, объем данных, и время отправления, вычисляет время пересылки данных
- время пересылки рассылается вычислительным узлам

Алгоритм маршрутизации

Для корректного расчета времени передачи данных необходимо найти маршрут, по которому эти данные будут передаваться. Обычно существует несколько вариантов различных маршрутов, из которых надо выбирать один на основании определенных критериев. Для построения всех возможных маршрутов передачи данных реализован класс, который с помощью рекурсивных функций осуществлял поиск незамкнутых путей по связанному графу.

Производится нахождение всех путей от каждого элемента вычислительной сети к каждому. Полученные данные образуют N -мерную матрицу, где N – количество узлов вычислительной сети, между которыми происходит передача данных. Элементы матрицы – массивы списков узлов, через которые проходит маршрут.

Алгоритм учета загрузки сети

Для определения времени передачи данных через сеть был использован итерационный подход коррекции скорости передачи данных, основанный на определении загрузки разделяемой среды передачи (РСП) в конкретный момент времени после начала работы алгоритма. Суть подхода состоит в следующем:

- 1) проводится эмуляция работы алгоритма без учета загрузки РСП;
- 2) на основе полученных данных вычисляется загрузка РСП;

- 3) проводится эмуляция работы алгоритма с учетом загрузки РСП;
- 4) если число итераций недостаточно, перейти к пункту 2.

На каждой новой итерации мы получаем данные загрузки РСП, на основе которых корректируем время пересылок в системе, что, в свою очередь, влияет на распределение загрузки РСП. Для получения достоверных результатов число итераций должно быть примерно равно количеству пересылок, использующих РСП.

Экспериментальная оценка работы алгоритма

Модель распределенной вычислительной сети реализована на языке C# в среде разработки Microsoft Visual Studio 2005. Модель использует физические параметры системы: пропускная способность, латентность, количество операций, выполняемых процессором в единицу времени. Эти параметры могут быть получены из технического описания кластера либо экспериментальным путём. Для получения наиболее точных параметров конкретной вычислительной системы проводится ряд специальных тестов.

При сравнении модели с реальным объектом моделирования замерялись такие характеристики параллельных программ, как ускорение и время выполнения программы. В качестве параллельной программы была взята программа вычисления числа π [4]. Ниже в таблице 1 приведены полученные результаты измерений.

Таблица 1. Результаты тестовых измерений.

Количество процессоров	Сеть		Модель	
	Время, с	Ускорение	Время, с	Ускорение
2	8.01	1.87	8.02	1.87
3	6.07	2.47	6.02	2.49
4	4.05	3.70	4.01	3.74
6	3.16	4.75	3.01	4.98
8	2.34	6.4	2.24	6.69

Заключение

Разработана и протестирована модель распределенной вычислительной сети кластерного типа, реализованы алгоритмы маршрутизации, обмена данных и учета загрузки сети. Модель позволяет получить подробную информацию о работе параллельных алгоритмов, наглядно исследовать их поведение. Кроме того, модель может быть полезной при проектировании распределенных вычислительных сетей. Результаты тестирования показали соответствие модели исследуемому объекту.

Литература

1. Воеводин. В.В., Воеводин Вл. В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002.-608 с.
2. Шпаковский Г.И. Организация работы на вычислительном кластере: учеб. пособие для студентов естественнонауч. фак. БГУ / Г.И. Шпаковский, А.Е. Верхотуров, Н.В. Серикова. - Мн.: БГУ, 2004. - 182 с.: ил.
3. Г.И.Шпаковский, А.Е.Верхотуров, Д.А.Стрикелев. Организации эффективных вычислений на локальных сетях в стандарте MPI. Пособие - Мн.: БГУ, 2004.-180 с.
4. Шпаковский Г.И., Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI. Пособие - Мн.: БГУ, 2002.-323 с.
5. Верхотуров А.Е., Шпаковский Г.И., Дежурко А.М. Разработка параллельных численных методов для моделирования полупроводниковых приборов на кластерах // Научно-технический рецензируемый журнал общественного объединения «Белорусская инженерная академия» Инженерный вестник 1(21)/1'2006. с. 14–16.