

4. A Book about Qt 5.[Электронный ресурс] – URL : <https://qmlbook.github.io/> дата обращения (18.10.18)

27.37.17

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

Автор: Карпов Максим Александрович, студент 3 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна»

Научный руководитель: Нурматова Елена Вячеславовна, к.т.н., доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна»

Аннотация

В данной работе рассматривается решение параллельных задач не линейного программирования при линейных ограничениях, как с помощью параллельных вычислений можно ускорить решение поставленной задачи.

In this paper, we consider the solution of parallel problems of non-linear programming with linear constraints, as with the help of parallel calculations, the solution of the problem can be accelerated.

В наше время практически нет такой сферы деятельности, где бы в той или иной форме не использовалось численное моделирование. Современные технологии, относящиеся к областям науки, очень часто обращаются к численным методам. Усложняющиеся технологические задачи, в поисках новых знаний, привели к пониманию, что эмпирический путь решения сложных проблем давно себя исчерпал. Использование накопленного многими поколениями запаса знаний должно быть направлено на построение математических моделей, достаточно адекватно отражающих суть изучаемых явлений или объектов. Как правило, математическая модель представляет собой некоторый комплекс дифференциальных, интегральных, алгебраических или других соотношений, связанных с предметом исследований.

Параллельные вычисления – вычисления, которые можно реализовать на многопроцессорных структурах с выполнением многих действий одновременно, которые, в свою очередь, могут породить процессорные решения одной или многих задач одного проекта.

Основная цель параллельных вычислений – сокращение времени решения задачи. Для решения многих задач в реальном времени требуется большой объём вычислений. Большое количество числа процессоров не всегда приводит к снижению затрат времени для решения задачи.

Задача параллельных вычислений – нахождения параллельного алгоритма в процессах решения задач и управление работой этого параллелизма с целью получения наибольшей эффективности применяя многопроцессорную вычислительную технику. Параллельный алгоритм решения задачи можно найти путём разработки нового параллельного алгоритма или распределением имеющегося последовательного алгоритма. Алгоритм решения задачи необходимо модифицировать для осуществления распараллеливания (например, устранить некоторые зависимости между операциями).

Параллельная обработка данных – это не единственный путь прироста скорости вычислений. Другой подход – повышать мощность процессорных устройств. Ограничениями такого подхода являются:

1. *Лимитированность скорости переключения.* Даже при самых быстрых коммуникациях скорость переключения не может превышать скорость света.

2. *Ограниченность размеров переключателей.* Устройство может работать быстрее с уменьшением его в размерах. Существует физический предел на размер компонентов, что связано с их молекулярным и атомным строением.

3. *Экономические ограничения.* Для увеличения скорости процессора, числа слоев в кристалле, плотности упаковки приходится решать все более сложные научные, производственные проблемы. Вот почему каждое новое поколение процессоров дорого стоит.

При создании параллельного алгоритма очень важно знать тип оперативной памяти, она определяет способ взаимосвязи между частями параллельной программы. В зависимости от организации подсистем оперативной памяти параллельные компьютеры можно разделить на следующие два класса.

Системы с распределенной памятью (мультикомпьютеры). Каждый процессор резервирует свою локальную оперативную память, а другие процессоры в это время не могут подключиться к этой памяти.

Системы с разделяемой памятью (мультипроцессоры). Они имеют одну виртуальную память, а все процессоры имеют равный доступ к данным и командам, держащийся в этой памяти (uniform memory access или UMA). По этому принципу разрабатываются симметричные мультипроцессоры (symmetric multiprocessor или SMP) и векторные параллельные процессоры (parallel vector processor или PVP).

Рассмотрим пример решения параллельных задач не линейного программирования при линейных ограничениях.

Пусть решается задача

$$f(x_1, \dots, x_n) \rightarrow \max$$

При ограничениях

$$g_1(x_1, \dots, x_n) \geq 0$$

$$g_2(x_1, \dots, x_n) \geq 0$$

...

$$g_m(x_1, \dots, x_n) \geq 0$$

и при условиях: g_i – линейны, $x_i \geq 0, i = 1, \dots, m$.

Представим, что гиперплоскости $g_i, i = 1, \dots, m$, возможно, совместно с координатными плоскостями, образуют в n -мерном пространстве выпуклый многогранник R допустимых решений, т.е. область задания функции f .

Обозначим все вершины $\{X_1, \dots, X_N\}, X_l = (x_1^l, \dots, x_n^l), l = 1, \dots, N$, этого многогранника в результате решения C_{m+n}^n систем n линейных уравнений на основе всех заданных и возможных его граней

$$g_1 = 0$$

...

$$g_m = 0$$

$$x_1 = 0$$

...

$$x_m = 0.$$

Тогда множество точек $X = (x_1, x_2, \dots, x_n)$ этого многогранника описывается как $X = \sum_{i=1}^N k_i X_i$, где $0 \leq k_i \leq 1, \sum_{i=1}^N k_i = 1$. Или:

$$\begin{cases} x_1 = k_1 x_1^{(1)} + k_2 x_1^{(2)} + \dots + k_{N-1} x_1^{(N-1)} + \left(1 - \sum_{l=1}^{N-1} k_l\right) x_1^{(N)} \\ \dots \\ x_n = k_1 x_n^{(1)} + k_2 x_n^{(2)} + \dots + k_{N-1} x_n^{(N-1)} + \left(1 - \sum_{l=1}^{N-1} k_l\right) x_n^{(N)} \end{cases}$$

А целевая функция на многограннике R становится функцией параметров $f(x_1, \dots, x_n) = f^*(k_1, \dots, k_{N-1})$.

Т.е., задавая разные значения k_1, k_2, \dots, k_{N-1} так, чтобы выполнялось условие $0 \leq k_l \leq 1$, так же как и для $k_N = 1 - \sum_{l=1}^{N-1} k_l \geq 0$, мы можем сделать перебор всех точек R с некоторым шагом h по всем параметрам. Этим мы накладываем N -мерную “сетку” на многогранник R . Каждой точке-узлу этой сетки мы будем находить значение функции f^* и выберем максимальное. Шаг h должен быть выбран так, чтобы обеспечить необходимую точность решения задачи.

Распараллеливание возможно на двух этапах решения:

1. Распределение систем линейных уравнений между процессорами для поиска всех вершин многогранника допустимых решений.
2. Распределение между процессорами узлов сетки – точек многогранника допустимых решений для поиска и изучения в них значений целевой функции.

Например, движение в сторону возрастания функции f может быть направленным, определяемым с помощью конечно-разностных значений частных производных по параметрам k_l в точке идущего анализа функции f . Это вариант так называемого градиентного метода. Эти же значения частных производных могут определять переменный шаг h .

Вывод

В современном мире нет такой сферы человеческой деятельности, где бы не использовалось численное моделирование. Особенно это касается областей науки и современных технологий. Без параллельно вычислительной системы или супер-ЭВМ работать с максимальной эффективностью невозможно. Тщательное согласование структуры программ и алгоритмов с особенностями архитектуры параллельных вычислительных систем даёт большое преимущество в решении сложных задач. В ходе работы был рассмотрен способ решения не линейного программирования при линейных ограничениях, и также было выяснено, что распараллеливание можно провести на двух этапах.

Список использованной литературы

- 1 Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002
- 2 Богачев К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, 2003.
- 3 Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. - М.: МЦНТО, 1999
- 4 Корнеев В.В. Параллельное программирование в MPI. Москва-Ижевск: Институт компьютерных исследований, 2003.
- 5 Цилькер Б. Я., Орлов С.А. Организация ЭВМ и систем. СПб.: Питер, 2006. - 668 с.
- 6 Касперски К. Техника оптимизации программ. Эффективное использование памяти. - СПб.: БХВ-Петербург, 2003. - 464 с.
- 7 Основы параллельного программирования: учебное пособие / К. Ю. Богачёв. — 2-е изд. — М. : БИНОМ. Лаборатория знаний, 2013. — 342 с. : ил. — (Математика)

8 Последовательные и параллельные алгоритмы / Р. Миллер, Л. Боксер ; пер. с англ. — М. : БИНОМ. Лаборатория знаний, 2006. — 406 с. : ил.

РАЗРАБОТКА ИГРОВОГО БОТА ДЛЯ ПРИЛОЖЕНИЯ TELEGRAM

Авторы: Кондратьев Евгений Александрович, Кригер Роман Владимирович, студенты 3 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна»

Научный руководитель: Нурматова Елена Вячеславовна, к.т.н., доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна»

Аннотация

Несмотря на блокировку мессенджера в РФ, разработчики ещё больше проявляют интерес к нему.

Annotation

Despite the blocking of the messenger in the Russian Federation, developers are even more interested in it.

Telegram (телеграм) – это мессенджер, который позволяет мгновенно обмениваться сообщениями и медиафайлами множества форматов. Он поддерживается многими ОС (Android, iOS, Windows Phone, Microsoft Windows, Chrome OS, Классическая Mac OS, Linux и macOS). В нем используется проприетарная серверная часть, исходный код которой закрыт, а также несколько клиентов с открытым исходным кодом.

Бот – это специальный аккаунт, управляемый программой. В основном боты отвечают на специальные команды в различных чатах, но они также могут работать с поиском в интернете или использоваться в целях бизнеса, или даже в развлекательных целях.

Проблема работы исходит из того, что в данный момент ниша игровых ботов пуста, и не так много как хорошего контента, так и сильных конкурентов.

Объектом исследования является разработка Telegram-бота.

Актуальность проблемы обусловлена увеличением количества пользователей мессенджера день ото дня, что рождает спрос на новые сферы применения ботов.

Задачи работы:

1. Подключить бота к аккаунту Telegram
2. Создать обработчик сообщений
3. Написать логику бота
4. Подключить базу данных к проекту
5. Перенести бота на удаленный сервер

Игровая составляющая

Сама игра выполнена в жанре RPG. Начав игру, пользователю предстоит выбрать одну из сторон, за которую он будет сражаться. Ему предстоит сражаться с монстрами, а также с такими же игроками, выполнять всевозможные квесты и участвовать в ивентах, путешествовать по локациям и зарабатывать внутриигровую валюту. Так же для успешного прохождения квестов нужно будет прокачивать навыки своего