

ОСНОВНЫЕ ПОНЯТИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Параллельные вычисления (параллельная обработка) – это использование нескольких или многих вычислительных устройств для одновременного выполнения разных частей одной программы (одного проекта).

Параллельные вычисления – такой способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно. (Википедия)

Параллельные вычисления – вычисления, которые можно реализовать на многопроцессорных системах с использованием возможности одновременного выполнения многих действий, порождаемых процессом решения одной или многих задач [одного проекта]. (Словарь по кибернетике)

Основная цель параллельных вычислений – уменьшение времени решения задачи. Многие необходимые для нужд практики задачи требуется решать в реальном времени или для их решения требуется очень большой объем вычислений.

Отметим, что увеличение числа процессоров не обязательно приводит к уменьшению времени решения задачи. Например, при выполнении задачи большим числом процессоров (ядер) может оказаться перегруженной коммуникационная сеть.

Использование параллельной обработки данных – не единственный путь уменьшить время решения задачи. Другой подход – увеличивать мощность процессорных устройств. Ограничениями такого подхода являются:

1. Ограниченность скорости переключения. Даже при самых быстрых коммуникациях – оптических – скорость переключения не может превышать скорость света.

2. Ограниченность размеров переключателей. Чем меньше размер компонентов устройства, тем быстрее устройство может работать. Однако существует физический предел размеров компонентов, что связано с их молекулярным и атомным строением.

3. Экономические ограничения. Для увеличения скорости процессора, плотности упаковки, числа слоев в кристалле приходится решать все усложняющиеся научные, инженерные, производственные проблемы. Вот почему каждое новое поколение процессоров дорого стоит. Потребляемая мощность процессора пропорциональна примерно степени 2,5 тактовой частоты. Например, процессор с тактовой частотой 3 ГГц потребляет больше, чем 9 процессоров частотой 1 ГГц. Таким образом, для параллельной программы энергоэффективность массы мелких процессоров будет выше в три раза. Иными словами, многоядерный процессор, с той же потребляемой энергией, будет в три раза производительнее при исполнении параллельного кода.

Всегда найдутся большие задачи, для решения которых потребуются

мощности параллельного компьютера.

Задача параллельных вычислений – создание ресурса параллелизма в процессах решения задач и управление реализацией этого параллелизма с целью достижения наибольшей эффективности использования многопроцессорной вычислительной техники. (Проще говоря, задача параллельных вычислений – получить параллельный алгоритм и эффективно его реализовать на многоядерном устройстве).

Получить параллельный алгоритм решения задачи можно путем распараллеливания имеющегося последовательного алгоритма или путем разработки нового параллельного алгоритма. Возможно, для осуществления распараллеливания алгоритм решения задачи придется заменить или модифицировать (например, устранить некоторые зависимости между операциями).

Терминология параллельных вычислений

Терминологию параллельных вычислений нельзя считать устоявшейся, поэтому некоторые из приведенных ниже понятий могут допускать иную трактовку.

Суперкомпьютер – вычислительная машина, значительно превосходящая по своим техническим параметрам большинство существующих компьютеров.

Кластер – группа компьютеров, объединенных в локальную вычислительную сеть и способных работать в качестве единого вычислительного ресурса.

Кластер предполагает более высокую надежность и эффективность, нежели просто локальная вычислительная сеть. Кластер использует типовые аппаратные и программные решения и поэтому имеет существенно более низкую стоимость в сравнении с другими типами параллельных вычислительных систем.

Распределённые вычисления – способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров, объединённых в параллельную вычислительную систему. Слабосвязанные, гетерогенные вычислительные системы выделяют в отдельный класс распределенных систем – Grid. (Википедия)

Распределённые вычисления – технология обработки данных, в которой большая задача распределяется для выполнения между множеством компьютеров, объединенных вычислительной сетью или интернетом.

Таким образом, распределенные вычисления являются частным случаем параллельных вычислений. Параллельные вычисления могут производиться как одном компьютере (суперкомпьютере или многоядерном компьютере), так и на многих компьютерах. Распределенные вычисления на одном компьютере производиться не могут.

Облачные вычисления – технология обработки данных, в которой компьютерные ресурсы и мощности предоставляются как интернет-сервис.

Параллельные задания – задания, допускающие одновременное (не обязательно независимое) выполнение.

Параллельный алгоритм – алгоритм, операции которого могут выполняться одновременно (не обязательно независимо); подразумевается, что в явном или неявном виде указаны одновременно выполняемые операции или множества операций. Строгое понятие параллельного алгоритма не введено.

Параллельная программа – параллельный алгоритм, записанный в некоторой системе программирования, ориентированной на вычислительные системы параллельной архитектуры.

Параллелизм по данным. Функциональный параллелизм. Если при решении некоторой задачи процессоры выполняют одинаковую последовательность вычислений, но используют разные данные, то говорят о параллелизме по данным. Например, при поиске по базе данных каждый процессор может работать со своей частью базы данных. Если процессоры выполняют разные задания одной задачи, выполняют разные функции, то говорят о функциональном параллелизме. Мы будем в основном рассматривать параллелизм по данным.

Конвейерная обработка данных операции (конвейерный параллелизм на уровне операции). Пусть операция разбита на микрооперации. Расположим микрооперации в порядке выполнения и для каждого выполнения выделим отдельную часть устройства. В первый момент времени входные данные поступают для обработки в первую часть. После выполнения первой микрооперации первая часть передает результаты своей работы второй части, а сама берет новые данные. Когда входные аргументы пройдут все этапы обработки, на выходе устройства появится результат выполнения операции. Таким образом, реализуется функциональный параллелизм. Каждая часть устройства называется ступенью конвейера, а общее число ступеней – длиной конвейера.

Конвейерная обработка уровня операций. Можно рассматривать вычислительный конвейер и при реализации параллелизма по данным: множество операций разбито на макрооперации, входные данные поступают для обработки в первую часть конвейера, после выполнения первой макрооперации первая часть конвейера передает результаты своей работы второй части, а сама берет новые данные (часть данных может быть распределена между частями конвейера заранее).

Реальная производительность вычислительной системы – количество операций, реально выполняемых в среднем за единицу времени.

Пиковая производительность вычислительной системы – максимальное количество операций, которое может быть выполнено системой за единицу времени.

Из определений вытекает, что реальная и пиковая производительности системы есть суммы соответственно реальных и пиковых производительностей, составляющих систему процессоров. Пиковая производительность одного процессора вычисляется как произведение

$n \times f \times k$, где n – количество операций с плавающей запятой, выполняемых за один такт, f – тактовая частота процессора, k – количество ядер в процессоре.

Пиковую производительность еще называют теоретической производительностью. Такое название подчеркивает, что на реальной программе производительность не только не превысит, но никогда и не достигнет этого порога.

Загруженность процессора на данном отрезке времени – отношение времени реальной работы процессора на данном отрезке к длине всего отрезка. Загруженность вычислительной системы, состоящей из одинаковых процессоров – среднее арифметическое загруженностей всех процессоров. Из определения следует, что загруженность p удовлетворяет условию $0 \leq p \leq 1$.

Ускорение (speedup) реализации алгоритма на вычислительной системе, состоящей из одинаковых процессоров – отношение времени выполнения алгоритма на одном процессоре (на одном ядре процессора) ко времени параллельного выполнения. Ускорение зависит от выбора параллельного алгоритма и от того, насколько этот алгоритм адекватен архитектуре вычислительной системы.

Как правило, ускорение выполнения программы не превосходит числа используемых процессоров. Но случается и так называемое суперлинейное ускорение – ускорение, превосходящее число используемых процессоров (чаще всего объясняется эффектом от использования кэш-памяти).

Эффективность (parallel efficiency) реализации алгоритма на вычислительной системе, состоящей из s одинаковых процессоров – отношение ускорения к s . Получение большого ускорения за счёт использования большого числа процессоров часто приводит к снижению эффективности.

Общая (разделенная, совместно используемая) память. Распределенная память. При параллельной обработке используются две основные модели доступа к памяти: общая, когда все процессоры напрямую связаны с одной общей памятью (но каждый процессор имеет и свою кэш-память), и распределенная, когда память физически распределена между процессорами.

Мультипроцессоры – системы с общей разделяемой памятью.

Мультикомпьютеры – системы с распределенной памятью.

Синхронизация (работы процессоров). В многопроцессорной системе необходимо координировать обмен данными между процессорами, а также, при использовании общей памяти, между процессорами и общей памятью. Синхронизация – это согласование по времени выполнения параллельных заданий. Она включает в себя ожидание того, что выполнение задачи достигнет особой точки, называемой точкой синхронизации. После того, как все задания достигнут точки синхронизации, выполнение заданий может быть продолжено до следующей точки синхронизации. При использовании модели передачи сообщений (обычно MPI – Message Passing Interface) работу процессоров синхронизируют функции обмена данными.

Синхронизация нужна для того, чтобы согласовать обмен информацией

между заданиями (между параллельно выполняемыми множествами операций). Синхронизация может привести к простоям процессора, так как после достижения точки синхронизации он должен ждать, пока другие задания достигнут точки синхронизации. Задержка с подачей в процессор необходимых данных также ведет к простоям процессора и снижению эффективности параллельной обработки. Эффект вырождения параллелизма из-за интенсивной синхронизации параллельных участков программы называется сериализацией (serialization).

Зернистость. Под зерном вычислений (или тайлом) понимается множество операций алгоритма, выполняемых атомарно: вычисления, принадлежащие одному зерну, не могут прерываться синхронизацией или обменом данными, требуемыми для выполнения этих операций.

Зернистость, при использовании распределенной памяти, – это мера отношения количества вычислений, сделанных в параллельной задаче, к количеству пересылок данных. Мелкозернистый параллелизм – очень мало вычислений на каждую пересылку данных. Крупнозернистый параллелизм – интенсивные вычисления на каждую пересылку данных (данные пересылаются большими порциями). Чем мельче зернистость, тем больше точек синхронизации.

Процесс – выполняемое приложение с собственным виртуальным адресным пространством.

Поток, нить (thread), легковесный процесс – создаваемый операционной системой объект внутри процесса, который выполняет инструкции программы. Процесс может иметь один или несколько потоков, выполняемых в контексте данного процесса. Потоки (нити) позволяют осуществлять параллельное выполнение процессов и одновременное выполнение одним процессом различных частей программы на различных процессорах.

Естественный параллелизм. Алгоритм обладает естественным параллелизмом, если на каком-то этапе выполнения его можно разбить на большие независимо выполняемые части.

Внутренний параллелизм. Алгоритм обладает внутренним параллелизмом, если на каком-то этапе выполнения его можно разбить на большие параллельно (но не обязательно независимо) выполняемые части.

Распараллеливание. Выявление (указание) операций или множеств операций последовательного алгоритма, которые могут выполняться одновременно.

Статическое распараллеливание. Распараллеливание, осуществляемое до начала выполнения алгоритма (программы).

Динамическое распараллеливание. Распараллеливание, осуществляемое во время выполнения алгоритма (программы).

Масштабируемость. Многопроцессорная вычислительная система называется масштабируемой, если она способна увеличивать производительность пропорционально дополнительным процессорам (ядрам) или пропорционально размеру N задачи.

Масштабируемость оценивается относительно конкретного параллельного алгоритма при реализации на конкретной параллельной системе. Различают понятия слабой и сильной масштабируемости.

Сильная масштабируемость (strong scaling) – зависимость означает линейный рост ускорения R с ростом числа P процессоров при фиксированном размере N_{fixed} задачи. Сильная масштабируемость демонстрирует зависимость производительности от количества процессоров (ядер) при фиксированной вычислительной сложности задачи.

Слабая (отнюдь не в смысле «плохая») масштабируемость (weak scaling) означает линейный рост размера N задачи с ростом числа P процессоров при фиксированном времени T_{fixed} выполнения алгоритма. Слабая масштабируемость демонстрирует зависимость производительности от количества процессоров (ядер) при фиксированной вычислительной сложности задачи в пересчёте на один процессор [для случая параллельных компьютеров с распределенной памятью в этой трактовке не учитывается «коммуникационная сложность задачи в пересчёте на один процессор»].

Рассматривают еще масштабируемость вширь (wide scaling) – зависимость ускорения R от размера N задачи при фиксированном числе процессоров P_{fixed} .