

МЕТОДИКИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ В СИСТЕМАХ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ КЛАСТЕРНОЙ АРХИТЕКТУРЫ

METHODS TO IMPROVE THE EFFICIENCY OF DATA TRANSMISSION PROTOCOLS IN SYSTEMS OF HIGH-PERFORMANCE COMPUTING CLUSTER ARCHITECTURE

N. Ahmed

Summary. The paper presents the analysis of techniques in systems of cluster architecture. The model to estimate the acceleration when working with OpenMP uses Amdahl's law. Evaluation of acceleration and characteristics of the computing system, which includes performance and data transfer speed.

Keywords: multiprocessor systems, high performance computing, transactions, database system, decision support system is of high reliability system for high-performance computing, multithreaded systems

Ахмед Набиль Мухаммед Мудхи

*Аспирант, Санкт-Петербургский государственный электротехнический университет СПбГЭТУ «ЛЭТИ»
Aboroan1987@yahoo.com*

Аннотация. В работе представлен анализ методик в системах кластерной архитектуры. Модель для оценки ускорения при работе с OpenMP использует закон Амдала. Оценка ускорения и характеристика вычислительной системы, в которые входит быстродействие и скорость передачи данных.

Ключевые слова: многопроцессорные вычислительные системы, высокопроизводительная вычислительная техника, транзакции, базы данных, системы поддержки принятия решений, системы высокой надежности, системы для высокопроизводительных вычислений, многопоточные системы.

Кластерная архитектура заключается в применении нескольких методик эффективного использования протоколов передачи данных, что в большей части позволяет увеличить производительность процессоров. Отрицательным фактором в увеличении производительности двух узлов является наличие расстояния между персональными компьютерами.

При разработке эффективной производительности в кластерной системе, необходимо решение некоторых аспектов по нахождению оптимального баланса загрузки системы на выполнение вычислительных операций. Нужный баланс нагрузки позволит двум вычислительным узлам работать на равномерной нагрузке, стабильно взаимодействуя подзадачами системы, тем самым это способствует быстрому выполнению задач. В процессе работы учитывается статистика производительности двух узлов. Главным фундаментом в управлении системы кластеров — является подсистема рейтингового баланса, которая приводит к эффективному выполнению задач.

Система разделения нагрузки в оптимальном режиме, заключается в выполнении следующих задач:

Первый период в статике:

1. Осуществляется ряд декомпозиций крупного количества задач, они разбиваются на множество мелких подзадач.
2. Идёт тщательное изучение множеств, определяется наличие взаимосвязей в текущем наборе данных.
3. Завершающие операции над параллельными алгоритмами с целевой направленностью повысить их результативность, методом удаления излишних взаимозависимостей.
4. В результате выполнения второй задачи и завершения третьей задачи в реорганизации подзадач, создаются подмножества подзадач, в каждой из которых идет последовательное исполнение.

Второй период в статике:

1. Идет выстраивание найденных подзадач методом определения трудоёмкости их реализации. Итогом становятся числовые данные, которые обозначаются условными единицами.
2. Идет оценочная деятельность текущего производительного процесса в вычислительных узлах, определяется эффективность рейтингов. Полученный результат, выделяет ресурс с эффективной трудоёмкостью задач.

3. Устанавливается оптимальный и стратегический вариант балансировки процессоров, к примеру, это может величина round-robin.

Третий период в динамике

1. Поступление новой подзадачи в систему дальнейшей обработки. С учетом условий третьей задачи второго этапа диспетчер осуществляет выбор эффективного узла и передает в его систему ряд подзадач. Рейтинг узла уменьшается при обработке операций в каждой подзадаче.
2. Со стороны диспетчера идет корректировочная работа над стратегией выбора с учетом изменяющегося разделения ресурсов. Итоговые результаты после выполнения подзадачи отправляются по запросу.
3. Монитор производительности отмечает поступление и результаты обработки каждой подзадачи, далее определяется время на обработку одной подзадачи, вносятся корректировки в трудоемкий процесс выполнения задачи в каждом узле, учитывается его загруженность.

По данному методу, на первоначальных стадиях кластер готовится к работе, на следующем периоде — работает и выполняет ряд задач. По истечению определенного периода времени данный метод определит эффективное значение трудоемкости в процессе выполнения любой подзадачи на определенном узле. Когда стратегия балансировки видоизменяется в узлах с уменьшающимися ресурсами, требуется перезагрузка кластерной системы. В процессе загрузки узла с подзадачами, идет перерасчет полученных значений производительности узла на мониторе производительности, осуществляется сверка полученных результатов, при необходимости вносят соответствующие корректировки.

Расчеты ускорения процессоров до и после по закону Амдала.

В процессе адаптации алгоритмов записанных в виде вычислений, применяется программная среда OpenMP, работа над параллельными вычислениями операций ведется на компьютере с общим ОЗУ. Это объясняется как системный подход эффективной расстановки большинства директив компилятора для эффективного распараллеливания циклов. В промежуточной стадии распараллеливания программы из-за свойств алгоритма, не все циклы подаются данной процедуре. К примеру, вычисления могут быть однотипными по структуре выполненных процедур, возможен подсчет всех операций, определяется сумма параллельных и последовательных процедур.

Доля не распараллеленных операций — $f \in [0;1]$.

T — линейное время выполнения арифметической операции относительно всего количества операций, позволяет определить оптимальное ускорение, достигаемое в текущем выполнении алгоритма.

Каждая многопоточная программа имеет в своем составе последовательную часть, которая заключается системе ввода и вывода данных, синхронизированного отбора информации.

Разработчик Джин Амдал на основе линейного ускорения параллельной части данных, провел ряд исследовательских работ. В 1967 году провел работу в ускорительном процессе выполнения программы на «n» процессорах, используя два значения:

1 — f доля параллельной части;

f доля последовательной части.

В результате работы было получено уравнение — $n / (1 + (n - 1) \times f)$ Ускорение на неограниченное число процессоров составило $1/f$.

Определим ускорение как отношение времени последовательного решения к времени параллельного решения:

$$S = \frac{T_1(n)}{\left(\beta + \frac{1-\beta}{p}\right) T_1(n) + t_{доп}}$$

где $t_{доп}$ — время на накладные расходы (обмены, простои и т.д.);

β — доля операций, выполнение которых невозможно одновременно с другими операциями.

Анализируя полученное соотношение можно вывести следствие, одно из которых называется законом Амдала:

Если $\beta = 0, t_{доп} = 0$, тогда $S = p$ (т.е. алгоритм полностью параллелен, отсутствует последовательная часть и ускорение максимально $S = p$)

Если $\beta <> 0, t_{доп} = 0$, тогда

$$S = \frac{1}{\left(\beta + \frac{1-\beta}{p}\right)} \text{ — закон Амдала.}$$

Если $\beta = \text{любое}$, тогда

$$S = \frac{T_1(n)}{t_{доп}} < 1,$$

$$Speedup_{symmetric}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f \cdot r}{perf(r) \cdot n}}$$

$$Speedup_{asymmetric}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f \cdot r}{perf(r) + n - r}}$$

$$Speedup_{dynamic}(f, n, r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{n}}$$

$Speedup_{symmetric}$ — относительное ускорение в симметричном режиме;

$Speedup_{dynamic}$ — относительное ускорение в динамическом режиме;

$Speedup_{asymmetric}$ — относительное ускорение в асимметричном режиме;

f — часть кода, которая может быть распараллелена; n — общее число ВСЕ; r — число ВСЕ на ядро; $perf(r)$ — производительность из r штук.

Имея большие затраты на обмен данных, простои и синхронизацию $t_{дон} \gg 0$, то все это рано или поздно приведет к неэффективности параллельного алгоритма ($S < I$).

Если 9/10 программы исполняется параллельно, а 1/10 по-прежнему последовательно, то повысить ускорение в 10 раз невозможно, несмотря на качество реализации параллельной части кода и числа используемых процессоров.

Для получения заданного ускорения для процессоров, целесообразно ускорить одну часть кода програм-

мы. Исходя из закона Амдала — чтобы ускорить выполнение программы в q раз необходимо ускорить не менее, чем в $(1 - 1/q)$ -ю часть программы. При необходимости ускорения работы процессоров в 100 раз по сравнению с его последовательным вариантом, то нужно ускорить выполнение на 99,99%.

На суперкомпьютере или на параллельном компьютере — в ряде выполнения работ можно проанализировать заложенный в программе алгоритм, при увеличении доли последовательной операция, то возможно сменить отдельные компоненты в структуре алгоритма.

В процессе работы, обмен данными между узлами осуществляется в комплексной структуре алгоритма с учетом смежных подобластей, которые принадлежат двум узлам. В общем режиме времени доля последовательных частей взаимосвязана от двух условий: конкретный тип расчетной области и вариант разбиения на подобласти. Увеличение участка смежных подобластей нуждается в большем времени на урегулирование всех деталей между вычислительными узлами, и соответственно заинтересовано в ускорительном режиме выполнения операций. Увеличение эффективности вычислений на кластере необходимо на стадии подготовки исходных данных в решении различных задач по минимизации размеров границ сшивки подобластей и зон контакта в кластерной системе.

ЛИТЕРАТУРА

1. Антонов, А. В. Эффективная организация параллельных распределенных вычислений на основе кластерной технологии. Диссертация на соискание ученой степени кандидата технических наук. Пенза, 2005. — 172 с.
2. Богачев К.Ю., Основы параллельного программирования, М.: БИНОМ, Лаборатория знаний, 2003.
3. Воеводин В.В., Воеводин Вл. В., Параллельные вычисления, СПб.: БХВ-Петербург, 2002, С. 608.
4. Гергель В.П., Стронгин Р.Г., Основы параллельных вычислений для многопроцессорных вычислительных систем, Нижний Новгород: Изд-во НГУ.
5. Гергель, В. П. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие / В. П. Гергель, Р. Г. Стронгин — Нижний Новгород: Изд-во ННГУ, 2003. — 184с.
6. Князьков В.С., Потапов А. А. Методика оценки трудоемкости реализации матричных мультипроцессорных систем. — Пенза, Изд-во ПГУ, Сб. тез. АПНО-2003. Т. 2. 2003. — С. 400–402.