

image not found or type unknown



ВВЕДЕНИЕ

С момента появления первых компьютеров одной из основных проблем, стоящих перед разработчиками, была производительность вычислительной системы. За время развития компьютерной индустрии производительность процессора стремительно возрастала, однако появление все более изощренного программного обеспечения, рост числа пользователей и расширение сферы приложения вычислительных систем предъявляют новые требования к мощности используемой техники, что и привело к появлению суперкомпьютеров. Суперкомпьютер это обычная вычислительная система, позволяющая производить сложные расчеты за более короткие промежутки времени.

Суперкомпьютеры

А зачем вообще нужны суперкомпьютеры? На этот вопрос дает ответ представитель компании «Крей ресерч» Вито Бонджорно: Раздвижение границ человеческого знания всегда опиралось на два краеугольных камня, которые не могут, существовать друг без друга, - теорию и опыт. Однако теперь ученые сталкиваются с тем, что многие испытания стали практически невозможными - в некоторых случаях из-за своих масштабов, в других - дороговизны или опасности для здоровья и жизни людей. Тут-то и приходят на помощь мощные компьютеры. Позволяя экспериментировать с электронными моделями реальной действительности, они становятся «третьей опорой» современной науки и производства. Прошло время, когда создатели суперкомпьютеров стремились обеспечить максимальную производительность любой ценой. Специальные процессоры, дорогостоящая сверхбыстрая память, нестандартное периферийное оборудование - все это обходилось заказчикам в круглую сумму. Приобретали суперкомпьютеры либо предприятия ВПК, либо крупные университеты. И те, и другие делали это, как правило, за государственный счет.

Известно, что сегодня существуют две основные архитектуры параллельных компьютеров: симметричные мультипроцессорные системы с общей памятью (SMP) и мультипроцессорные системы с распределенной памятью (MPP).

1.1 Архитектура SMP

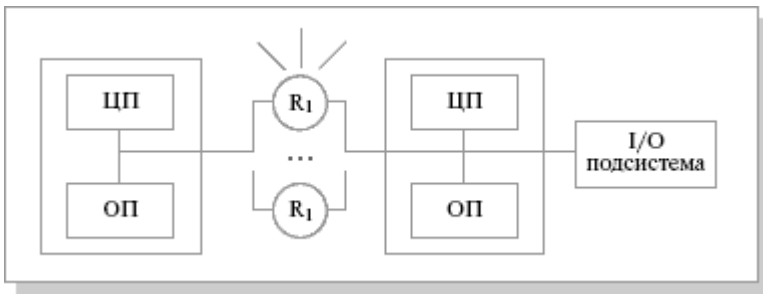
Основное преимущество SMP - относительная простота программирования. В ситуации, когда все процессоры имеют одинаково быстрый доступ к общей памяти, вопрос о том, какой из процессоров какие вычисления будет выполнять, не столь принципиален, и значительная часть вычислительных алгоритмов, разработанных для последовательных компьютеров, может быть ускорена с помощью распараллеливающих и векторизирующих трансляторов. SMP-компьютеры - это наиболее распространенные сейчас параллельные вычислители, а 2-, 4-х процессорные ПК на основе Pentium и Pentium Pro стали уже массовым товаром. Однако общее число процессоров в SMP-системах, как правило, не превышает 16, а их дальнейшее увеличение не дает выигрыша из-за конфликтов при обращении к памяти. Применение технологий типа UPA, основанной на коммутации пакетов вместо общей шины и локальной кэш-памяти большого объема, способно частично решить проблему, подняв число процессоров до 32.

1.2 Архитектура MPP

Альтернатива SMP - архитектура MPP. Каждый процессор имеет доступ лишь к своей локальной памяти, а если программе нужно узнать значение переменной, расположенной в памяти другого процессора, то задействуется механизм передачи сообщений. Процессор, в памяти которого находятся нужные данные, посылает сообщение тому процессору, которому они требуются, а последний принимает его. Этот подход позволяет создавать компьютеры, включающие в себя тысячи процессоров. На нем основаны все машины, имеющие производительность в сотни миллиардов операций в секунду.



1.1



1.2

Основные принципы архитектуры

а) Распределенная память

В суперкомпьютерах nCube используется архитектура распределенной памяти, позволяющая оптимизировать доступ к оперативной памяти, вероятно, наиболее критичному ресурсу вычислительной системы.

Традиционные архитектуры с разделенной памятью удобны для систем с небольшим числом процессоров, однако, они плохо масштабируются по мере добавления процессоров и памяти. Когда в системе с разделением памяти увеличивается число процессоров, возрастает конкуренция в использовании ограниченной пропускной способности системной шины, что снижает производительность соединения процессор-память. Кроме того, добавление процессоров в такую систему требует увеличения количества логики для управления памятью, снижая тем самым производительность системы и увеличивая ее цену.

Эти недостатки отсутствуют в системах с распределенной памятью. В такой системе каждый процессор имеет свою собственную локальную память. Потенциальные узкие места, связанные с шиной процессор-память и необходимостью разрабатывать системы управления кэшем, полностью исключаются. С добавлением процессоров добавляется память, пропускная способность соединения процессор-память масштабируется вместе с вычислительной мощностью.

б) Межпроцессорная сеть

Топология межпроцессорных соединений, обеспечивающая масштабирование до большого числа процессоров без снижения производительности коммуникаций или увеличения времени ожидания, является обязательной для MPP-систем.

Суперкомпьютеры nCube используют сетевую топологию гиперкуба, которая

отвечает этим требованиям. Соединения между процессорами nCube-системы образуют многомерный куб, называемый гиперкубом. По мере добавления процессоров увеличивается размерность гиперкуба. Соединение двух гиперкубов одинаковой размерности образует гиперкуб следующей размерности. N-мерный гиперкуб содержит 2^n процессоров. Двухмерный гиперкуб - это квадрат. Трехмерный гиперкуб образует обычный куб, а четырехмерный гиперкуб представляет собой куб в кубе. Для семейства суперкомпьютеров nCube 2 гиперкуб максимальной размерности 13 содержит 8192 процессора. В системе nCube 3 число процессоров может достигать 65536 (16-мерный гиперкуб).

Эффективность сетевой топологии измеряется, в частности, числом шагов для передачи данных между наиболее удаленными процессорами в системе. Для гиперкуба максимальное расстояние (число шагов) между процессорами совпадает с размерностью куба. Например, в наибольшем 13-мерном семействе nCube 2 сообщения между процессорами никогда не проходят более 13 шагов. Для сравнения, в 2-мерной конфигурации "mesh" (петля) с числом процессоров, вдвое меньшим числа процессоров в максимальной системе nCube 2, наибольшее расстояние между процессорами составляет 64 шага. Задержки коммуникаций в такой системе значительно увеличиваются. Таким образом, никакая другая топология соединения процессоров не может сравниться с гиперкубом по эффективности. Пользователь может удвоить число процессоров в системе, при этом увеличивая длину пути связи между наиболее удаленными процессорами только на один шаг.

Большое число соединений в гиперкубе создает высочайшую пропускную способность межпроцессорных соединений по сравнению с любой другой сетевой схемой. Большое количество путей передачи данных и компактный дизайн гиперкуба позволяют передавать данные с очень высокой скоростью. Кроме того, гиперкубическая схема характеризуется большой гибкостью, так как она позволяет эмулировать другие популярные топологии, включая деревья, кольца. Таким образом, пользователям nCube-систем гарантируется корректное выполнение приложений, зависящих от других топологий.

в) Высокий уровень интеграции

Многие преимущества nCube-систем, и, прежде всего высочайшие показатели - надежности и производительности, являются результатом использования технологии VLSI (Very Large Scale Integration - сверхвысокая степень интеграции). В большей степени, чем какие-либо другие факторы, на надежность компьютера

влияет число используемых компонентов. Большое число компонентов неминуемо увеличивает вероятность сбоя системы. По этой причине nCube интегрирует все функции процессорного узла на одно VLSI-устройство. VLSI-интеграция также сокращает требуемое число соединений, которые могут оказывать решающее влияние на целостность всей системы.

Высокий уровень интеграции сокращает пути передачи данных, повышая тем самым производительность системы. Интеграция процессорного узла на один чип оставляет свободными для контроля только простые соединения с памятью и сетевые интерфейсы вместо сложных сигналов синхронизации, арбитража и управления. Эти простые соединения тестируются и корректируются с помощью методов контроля четности и ECC (Error Correction Code - код коррекции ошибок), упрощая процесс определения и изоляции ошибок.