В. А. Евстигнеев, И. Л. Мирзуитова

РАЗВИТИЕ NUMA-АРХИТЕКТУРЫ: ТЕКУЩЕЕ СОСТОЯНИЕ*

1. ВВЕДЕНИЕ

Данная работа является продолжением исследований, начатых в [1].

Понятие NUMA-архитектуры возникло в конце 80-х годов с появлением интереса к компьютерам с распределенной памятью. Среди них выделялись так называемые DSM-машины, т. е. машины с распределенной общей памятью. В этих машинах общая память была физически разнесена по компьютеру. В силу этого сочетались легкость программирования для машин с общей памятью и серьезные трудности для поддержания режима общей памяти. При этом кусок общей памяти, максимально приближенный к узлу вычислительной системы, имеет время доступа в десятки раз меньше, чем к удаленным кускам. Заметим, что максимально приближенный к узлу кусок общей памяти не есть локальная память этого узла, хотя и имеет много общего с ней.

Как мы уже знаем из [1], такие машины относятся к машинам с NUMA-архитектурой, если они специально нацелены на максимальное сглаживание разницы во времени доступа к разным частям памяти.

Другая проблема, решение которой требует наличия распределенной памяти, есть проблема масштабирования. Разбитая на куски, разнесенная память позволяет организовать параллельные машины в виде сети пар процессор–память. К масштабируемым машинам этого типа относятся машины фирмы BBN (Batterfly, TC 2000, GP 2000), KSR-1 и KSR-2 фирмы Kendall Square Research, NCUBE 2, а также более поздние — IBM SP-2, Convex SPP 1200/XA, Intel Paragon, Thinking Machine CM5, IBM RP3, проект DASH (Стенфорд), проект ALEWIFE (МТИ), Horizon/Tera.

Но многие авторы утверждают, что NUMA-архитектура была разработана в начале 1990-х. На май 1998 г. она поддерживалась только несколькими поставщиками аппаратуры (например, Sequent Computers Inc., Data General Corp., Silicon Graphics Inc. и Hewlett-Packard) и систем баз данных. Это расходится с приведенным выше списком машин,

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 01-01-794) и Министерства образования РФ.

относимых к NUMA-архитектуре. Сторонники NUMA утверждают, что эта архитектура позволяет решить все проблемы, свойственные другим параллельным архитектурам. Однако так было всегда: каждый, кто предлагал новую параллельную архитектуру, заявлял, что она решает все старые проблемы. Вместе с тем, имеются основания ожидать, что архитектура NUMA останется среди реально используемых.

Следующим этапом развития NUMA-архитектуры является появление архитектуры ccNUMA — акроним от "cache-coherent nonuniform memory architecture". ccNUMA — это кэш-когерентный доступ к неоднородной памяти. Архитектура ccNUMA выделяется принципиально. Это архитектура симметричного мультипроцессирования (SMP), обладающая множеством достоинств: простая модель программирования, переносимость приложений и т.д. У истоков создания архитектуры ccNUMA стояла компания Sequent, реализовавшая собственную версию — NUMA-Q.

Следующим этапом развития архитектуры NUMA является появление архитектуры NUMAFlex. 25 июля 2000 г. компания SGI анонсировала новое семейство систем SGI 3000. Эти системы являются первыми системами, построенными на основе технологии NUMAFlex. Технология NUMAFlex, которая является третьим поколением технологии NUMA, дает возможность наращивания и изменения системы вплоть до использования различных процессоров одновременно в рамках единой системы. Не исключено, что технология NUMA станет открытой.

Основным блоком конструкции новых серверов SGI 3000 стал "кирпич" (brick); при этом кирпичи бывают разных типов, в зависимости от их содержимого. Однако основные элементы архитектуры S2MP сохранены, т.е. сохраняется то, как связываются между собой процессоры, оперативная память, концентраторы, маршрутизаторы и подсистема ввода-вывода. То, что ранее было реализовано в виде плат, "превратилось" в кирпичи, а "провода" на системной плате типа midplane заменены кабелями (таких плат в NUMAFlex больше нет).

2. ОСНОВНЫЕ ПАРАЛЛЕЛЬНЫЕ АРХИТЕКТУРЫ

Чтобы понять принципы работы технологии NUMA, нужно знать, как функционирует традиционная симметричная многопроцессорная обработка (SMP). SMP позволяет связать несколько процессоров в одну систему и объединить их вычислительную мощность для выполнения нескольких приложений или одного большого приложения. Эти процес-

соры взаимодействуют друг с другом с помощью так называемой шины межсоединения и используют пул общей памяти. При увеличении в сервере числа процессоров возрастает и трафик в данной шине. В конце концов пропускная способность системы существенно снижается.

NUMA, как и SMP, позволяет получить объединенную вычислительную мощность большого числа процессоров, каждый из которых обращается к общему пулу памяти, однако процессоры организуются в небольшие группы или "узлы", с помощью которых они связываются друг с другом. Архитектура SMP появилась в начале 1970-х гг. и быстро стала фактическим стандартом наиболее распространенных параллельных архитектур. Если компания, производящая компьютерное оборудование, поддерживает хотя бы какой-нибудь вид параллелизма, то с большой вероятностью используется SMP.

Кластеры, вероятно, стали использоваться даже раньше, чем SMP. Причиной возникновения кластерной архитектуры было то, что необходимую для пользователя работу было невозможно выполнить на одном компьютере или эта работа была настолько важна, чтобы приобрести дублирующее оборудование. Много позже этот подход удостоился общепринятого названия (термин "кластер" был введен в обиход компанией Digital Equipment Corp. в середине 1980-х гг.), и его стали поддерживать поставщики систем. Сегодня кластерная архитектура является козырной картой практически каждого поставщика компьютерных систем, ориентированных на применение ОС UNIX и/или Windows NT.

2.1. Архитектура SMP

SMP — это один компьютер с несколькими равноправными процессорами. Все остальное — в одном экземпляре: одна память, одна подсистема ввода/вывода, одна ОС. Слово "равноправный" (как и слово "симметричная" в названии архитектуры) означает, что каждый процессор может делать все, что любой другой. Каждый процессор имеет доступ ко всей памяти, может выполнять любую операцию ввода/вывода, прерывать другие процессоры и т.д. Но это представление справедливо только на уровне программного обеспечения. Умалчивается то, что на самом деле в SMP имеется несколько устройств памяти.

В SMP каждый процессор имеет по крайней мере одну собственную кэш-память (а возможно, и несколько). Наличие кэш-памяти (или просто кэша) необходимо для достижения хорошей производительности, поскольку основная память (DRAM —Direct Random Access Memry)

работает слишком медленно по сравнению со скоростью процессоров, и каждый год это соотношение ухудшается. Уже сейчас скорость основной памяти в 20–40 раз меньше, чем требуется, а в ближайшее время этот показатель будет порядка 100. Кэш работает со скоростью процессора, но эта аппаратура дорогая, и поэтому устройства кэш-памяти обладают относительно небольшой емкостью. В настоящее время размер кэш-памяти составляет 10–100 Кбт, в то время как основная память может иметь объем в 10–100 Мбт. Для процессора кэш исполняет роль "рабочего стола", на котором хранится используемая в текущее время информация: основная память подобна большому шкафу, находящемуся в комнате. При том, что в SMP имеется несколько устройств памяти, программное обеспечение ожидает увидеть только одну общую память. Грубо говоря, из этого следует, что если процессор A сохраняет значение X в ячейке Q, а позже процессор B загружает значение из ячейки Q, то процессор B должен получить X. Но если на самом деле значение X было помещено в кэш-процессор A, то как его сможет получить процессор B?

Имеются три причины, по которым когерентность кэшей является важной. Во-первых, это свойство играет ведущую роль в системах NUMA. Во-вторых, поддержка когерентности серьезно влияет на производительность. И, в третьих, именно из-за поддержки когерентности кэшей архитектура SMP не может обеспечить высокой доступности. Влияние на производительность очевидно. Программа работает гораздо быстрее, в 10–20 раз, если используются данные, уже содержащиеся в кэше. Программы, которые обращаются к другим устройствам памяти, выполняются очень медленно. Даже если для реорганизации кода с целью повышения вероятности использования данных из кэша требуется большее число команд, эти команды гораздо более быстрые.

2.2. Кластерная архитектура

Кластер — это связанный набор полноценных компьютеров, используемый в качестве единого ресурса. Под словосочетанием "полноценный компьютер" понимается завершенная компьютерная система, обладающая всем, что требуется для ее функционирования, включая процессоры, память, подсистему ввода/вывода, а также ОС, подсистемы, приложения и т.д. Обычно для этого годятся готовые компьютеры, которые могут обладать архитектурой SMP и даже NUMA.

Другое название кластерной архитектуры — мультикомпьютеры.

Словосочетание "единый ресурс" означает наличие ПО, дающего возможность пользователям, администраторам и даже приложениям считать, что имеется только одна сущность — кластер. Например, система пакетной обработки кластера позволяет послать задание на обработку кластеру, а не какому-нибудь отдельному компьютеру. Более сложным примером являются системы баз данных. У всех ведущих поставщиков систем баз данных имеются версии, работающие в параллельном режиме на нескольких машинах кластера. В результате приложения, использующие базу данных, не должны заботиться о том, где выполняется их работа. СУБД отвечает за синхронизацию параллельно выполняемых действий и поддержания целостности базы данных.

При соединении компьютеров в кластер почти всегда поддерживаются прямые межмашинные коммуникации. Решения могут быть простыми, основывающимися на аппаратуре Ethernet, или сложными с высокоскоростными сетями с пропускной способностью в сотни мегабайт в секунду. К последней категории относятся RS/6000 SP компании IBM, системы компании Digital на основе Memory Channel, ServerNet компании Compaq Computer Corp. Часто, хотя и не всегда, обеспечивается доступ каждого компьютера, входящего в состав кластера, к любому диску. Это не означает, что компьютеры обязательно совместно используют диски каким-либо разумным образом; при разработке СУБД или других подсистем возможен выбор, и в большинстве таких систем применяется подход "sharing–nothing", при котором в любой момент времени диском "владеет" только один компьютер. Исключениями являются Oracle8 и DB2 с IBM OS/390; в них диски активно используются в совместном режиме, образуя большую и медленную память, с помощью которой поддерживается когерентность кэшей базы данных.

2.3. Архитектуры NUMA

Проще всего охарактеризовать NUMA-систему, представив себе большую систему SMP, распиленную пополам, причем половинки связаны кабелем, подключенным к системным шинам, и каждая такая половинка включает собственную основную память и систему ввода/вывода. Это и есть NUMA: большая SMP, разбитая на набор более мелких и простых SMP. Основной проблемой NUMA является обеспечение когерентности кэшей. Аппаратура позволяет работать со всеми отдельными устройствами основной памяти составных частей системы (называемых обычно узлами) как с единой гигантской памятью. Этот подход порож-

дает ряд следствий. Во-первых, в системе имеется одно адресное пространство, распространяемое на все узлы. Реальный (не виртуальный) адрес 0 для каждого процессора в любом узле соответствует адресу 0 в частной памяти узла 0 и т. д., пока не будет использована вся память узла 0. Затем происходит переход к памяти узла 1, затем — узла 2 и т. д. Для реализации этого единого адресного пространства каждый узел NUMA включает специальную аппаратуру (Dir).

NUMA-системы вряд ли смогут заменить SMP при решении задач, требующих частой синхронизации. Ситуация улучшается, если удается разделить приложение на части, каждая из которых располагается в одном узле и взаимодействия между которыми возникают не слишком часто. Например, NUMA-систему с двумя узлами можно эффективно использовать для ведения бухгалтерии корпорации, имеющей два относительно независимых офиса. Эффективно использовать NUMA-системы могут СУБД, оптимизаторы которых в состоянии разбить сложный запрос на независимо выполняемые части (например, сканирующие разные части большой таблицы).

Что касается доступности, то NUMA наследует все неприятности, свойственные SMP, по той же причине обеспечения когерентности кэшей и наличия единой ОС. В отличие от кластеров, NUMA-систему любого размера можно считать “одной машиной”. Для достижения высокого уровня доступности нужно использовать кластеры.

3. ОПИСАНИЕ ПЛАТФОРМЫ NUMA-Q ОТ SEQUENT (АРХИТЕКТУРА CCNUMA)

ccNUMA — это кэш-когерентный доступ к неоднородной памяти. В системе ccNUMA физически распределенная память объединяется, как в любой другой SMP-архитектуре, в единый массив. Не происходит никакого копирования страниц или данных между ячейками памяти. Нет никакой программно-реализованной передачи сообщений. Существуют просто одна карта памяти, частями, физически связанными медным кабелем, и очень умные (в большей степени, чем объединительная плата) аппаратные средства. Аппаратно-реализованная кэш-когерентность означает, что не требуется какого-либо программного обеспечения для сохранения множества копий обновленных данных или для передачи их между множеством экземпляров ОС и приложений. Со всем этим справляется аппаратный уровень точно так же, как в любом SMP-узле, с одной копией ОС и несколькими процессами.

3.1. Общая структура

Элементарным блоком платформы NUMA-Q служит квод (NUMA-Q означает ccNUMA с кводами), в котором объединены четыре процессора, блок разделяемой памяти и шины PCI с семью слотами. Несколько кводов могут быть соединены связями с аппаратно-реализованной кэш-когерентностью для формирования более крупного одиночного SMP-узла таким же образом, как процессорные платы добавляются к объединительной плате обычного SMP-узла с большой шиной.

3.2. Задержка — как основная характеристика различия архитектур

Одним из ключевых различий описанных архитектур является диктуемая ими модель программирования, а различия в способах программирования напрямую обусловлены задержками доступа. Достижение организации памяти NUMA-Q заключается в том, что доступ к часто используемым данным происходит за микросекунды, тогда как считывание их с диска требует миллисекунд, а в MPP-системах с разделяемыми дисками доступ к удаленному диску может занимать десятки миллисекунд. В кластерах с отражением памяти между узлами добавляются когерентные соединения с программной поддержкой, в результате чего время доступа к удаленной памяти снижается до сотни микросекунд. Это, конечно, в сотни раз быстрее, чем обращаться к дискам для тех же данных, но сотни микросекунд — это еще в сотни раз медленнее, чем скорость локальной памяти. Поэтому программист должен позаботиться о том, чтобы минимизировать такого рода пересылки, планируя для этого, где только возможно, распределение данных вручную. Следует отметить, что даже в RMC удаленный доступ опирается как на аппаратную, так и на программную поддержку, тогда как доступ к локальной памяти реализуется исключительно аппаратными средствами — очень быстро и с гарантией когерентности. Вот почему программирование в SMP так выгодно для прикладных программистов, которые не должны заботиться о распределении данных в памяти, так как все ее части доступны для любого ЦПУ и доступ к ним одинаково быстр.

3.3. Порядок работы NUMA-Q

В качестве основного строительного блока для SMP-узлов платформы NUMA-Q компания Sequent использует кводы с четырьмя процес-

сорами на блок распределенной памяти. В узле с тремя, например, квадами одна треть физической памяти будет близка (в смысле задержки доступа к памяти) к четырем процессорам квада, а две трети будут "не совсем близкими". Это может привести к выводу о том, что две трети обращений к памяти будут медленными и только одна треть — быстрой. К счастью, без модификации приложений, реализованных для традиционных SMP-архитектур, этого не происходит. В действительности основная часть процессорных доступов к памяти будет очень быстрой, и только маленький процент окажется не столь высокоскоростным. Это происходит из-за большой локальной памяти квада и большого удаленного кэша на плате IQ-Link.

3.3.1. Конфигурация памяти NUMA-Q

В традиционном смысле память в каждом кводе не является локальной. Скорее, это одна треть адресного пространства физической памяти, которая имеет собственный адресный диапазон. Адресная карта распределяется по памяти равномерно, при этом каждый квод содержит смежную часть адресного пространства. Как и в любой SMP-системе, работает только одна копия ОС, она размещает в памяти и запускает одновременно на одном или нескольких процессорах любые процессы без какого-либо различия между ними.

Будем называть сегмент памяти, расположенный в кводе, локальной памятью квода, а память из других квадов — удаленной памятью квода. Ясно, что доступ к локальной памяти квода происходит быстрее, чем доступ к удаленной для него памяти. Задержки доступа к единому пространству адресов памяти не одинаковы, вот почему NUMA-Q является истинной NUMA-архитектурой.

В современной реализации архитектуры NUMA-Q компания Sequent использует процессоры Pentium Pro с двумя кэшами L1 и L2 внутри чипа. Как известно, в компьютерных системах кэши устанавливаются в предположении о "пространственной локальности" обращений к памяти. В соответствии с этим предположением основная часть всех кэш-промахов (cache miss) в L2 будет попадать в диапазон локальной памяти квода и, таким образом, будет быстро обслуживаться. Если же адрес находится за пределами диапазона локальной памяти квода, поиск будет распространен на 32-мегабайтный кэш IQ-Link, который называется удаленным кэшем. Доступ к этому кэшу осуществляется с такой же скоростью, как и к локальной памяти квода.

3.3.2. Кэш-когерентность в NUMA-Q

Передача данных по единственной шине в SMP-архитектурах с одной объединительной платой может происходить по разным причинам, и их следует различать. Во-первых, это передача между портом ввода-вывода и памятью, и, во-вторых, ситуации, в которых процессор обращается к памяти при отсутствии данных в L2 кэше. Кроме того, имеются еще и передачи кэш-кэш между разными процессорами, которые называются промахами когерентности.

3.3.3. Нужно ли оптимизировать программы для NUMA-Q?

Эффективность работы приложения на платформе NUMA-Q зависит, главным образом, от того, насколько справедливы следующие предположения.

1. Частота обращений к удаленной памяти существенно ниже, чем частота локальных обращений.
2. Задержка удаленного доступа очень мала.
3. Пропускная способность IQ-Link намного больше, чем та, которая требуется в приложениях OLTP и DSS.

Что должны делать разработчики ПО для систем SMP при переносе его на NUMA-Q? Если некоторые из приведенных пунктов неверны, можно попытаться изменить ПО, чтобы настроить его на NUMA-Q. Однако, если предположения выполняются, тогда тот факт, что малый процент доступов занимает больше времени, чем остальные, может не учитываться программистами так же, как не принимается во внимание работа кэшей.

3.4. Заключение

Определяющим фактором для производительности систем являются задержки доступа к данным. Для уменьшения их влияния в системы вводится дополнительная память. SMP-платформы обеспечили легко программируемую модель, так как время доступа ко всем частям памяти стало одинаково недолгим. Попытки распределять данные между узлами, передавая их между памятью, успешны только в том случае, если альтернативой являются обращения за данными к диску. Однако для оптимизации производительности все еще требуется существенный объем перепрограммирования, так как задержки между узлами значительно больше, чем задержки внутри узла.

Наиболее эффективный способ достичь высокой производительности и сохранить при этом простую модель программирования — построить как можно больший одиночный узел, прежде чем переходить к архитектуре из нескольких узлов. Однако из-за ограничений на размер объединительных плат и системных шин, с использованием кэша слежения этого сделать нельзя, и максимально достигнутое число процессоров не превышает сегодня 32. Для того чтобы выйти за этот предел, можно использовать кэш-протоколы на основе каталогов и архитектуры ccNUMA. Такой подход имеет дополнительные достоинства, так как позволяет подсоединять до 252-х процессоров, получая огромную память и пропускную способность шины ввода-вывода и при этом имея средние задержки доступа к памяти меньше, чем у любой современной системы. Основным же достижением является отсутствие необходимости менять ПО SMP-приложений для того, чтобы воспользоваться всеми этими возможностями.

4. NUMAFLEX-АРХИТЕКТУРА

4.1. Введение

Напомним, что система ccNUMA состоит из набора узлов, каждый из которых имеет собственные процессоры, локальную оперативную память и обычно собственные средства ввода-вывода. Это справедливо и для произвольной MPP-системы с распределенной между узлами оперативной памятью, например, для кластерной архитектуры IBM SP2. Следующим шагом являются NUMA-системы, в которых память попрежнему физически распределена между узлами, но адресуема всеми микропроцессорами и логически является общей. Примером такой системы является Cray T3E. Наконец, для автоматического обеспечения согласованности работы всех процессоров с памятью требуется поддержание когерентности их кэшей, что и приводит разработчиков к архитектуре ccNUMA.

Похоже, что именно в направлении ccNUMA архитектура многопроцессорных систем развивается наиболее активно. Следующие ведущие производители предлагают компьютеры с архитектурой ccNUMA: это и HP (еще со времен Convex SPP), и Compaq с новыми компьютерами AlphaServer GS 320, и IBM Sequent NUMA-Q, и Data General AViiON 2x00, и Siemens RM600E, и, наконец, SGI серверы Origin, которые стали несколько лет тому назад основным полигоном практического освоения ccNUMA.

SGI, которая при разработке Origin 2000 опиралась на результаты совместного со Стэнфордским университетом проекта DASH, имеет, как представляется, наибольший опыт в этой области: летом 1999 г. компания представила уже второе поколение ccNUMA — систему Origin 3x00. Архитектура их предшественников Origin 2000 называлась S2MP (Scalable Shared memory MultiProcessing); архитектура же Origin 3x00 носит название NUMAFlex.

Здесь flex, очевидно, есть сокращение от английского flexibility ("гибкость"). NUMAFlex действительно отличается особой гибкостью в построении различных конфигураций системы и ее изменении "на лету" в процессе реального функционирования. В NUMAFlex реализована возможность разбиения всей ccNUMA-системы на разделы (partition), которые являются более "мелкими" ccNUMA или SMP-компьютерами. (Схема разбиения, или парционирования, будет рассмотрена ниже.) Парционирование позволяет преображать ccNUMA-систему в кластерную структуру. Узлами этого кластера могут быть опять-таки ccNUMA-серверы.

4.2. Системы IBM высокой доступности

В конце 90-х гг. фирма IBM приступила к выпуску современных центров обработки данных. Это третье поколение технологии "non-uniform memory access" (NUMA), впервые представленной в 1996 г. компанией Sequent Computer Systems, Inc. К таким центрам относятся две системы: NUMA-Q 2000 и NUMACenter.

4.2.1. Сервер NUMA-Q 2000

NUMA-Q 2000 — это современный центр обработки данных с лидирующей в индустрии производительностью. Серверы предприятия NUMA-Q 2000 базируются на архитектурных решениях, которые обеспечивают высочайшую производительность, доступность и управляемость, что и требуется для обработки огромных информационных массивов и круглосуточной поддержки большого количества запросов. NUMA-Q обладают широкими возможностями.

4.2.2. Сервер NUMACenter

Весной 1999 г. Sequent представила упрощенный вариант серверов — NUMACenter, масштабируемых до 64-х процессоров. Тогда же был про-

демонстрирован режим работы этой системы под одновременным управлением UNIX и Windows NT, интегрированных с помощью Unicenter TNG компании CA. Сервер NUMACenter создан на базе новых процессоров Xeon.

IBM NUMACenter — идеальное решение для эффективной работы с коммерческими приложениями в информационной среде предприятия.

4.2.3. Серверы SGI

Серверы SGI — рыночные лидеры в технических вычислительных приложениях — используются в ключевых отраслях промышленности, в правительстве, индустрии развлечений, связи, энергетики, науки и образовании. Высокомасштабируемые сервера компании также имеют постоянно возрастающее присутствие на коммерческом рынке, с акцентом на стратегический деловой анализ, приложения INTERNET и обслуживание средств мультимедиа.

Линия серверов SGI, начинаясь с серверов начального уровня SGI 1100, 1200, 1450, включает сервера среднего уровня SGI Origin 200, Origin 200 GIGAchannel, SGI 2100, 2200, 3200, 3200C и заканчивается серверами высшего уровня SGI Origin 2400, Origin 2800, Origin 3400 и Origin 3800.

Сервера начального уровня имеют архитектуру SMP, от 1 до 2 CPU (1100 и 1200) или от 1 до 4 (1450); в качестве процессоров используются Intel Pentium III или Pentium III Xeon.

Все сервера среднего уровня имеют архитектуру NUMA, от 1 до 4 CPU (Origin 200), от 2 до 8 CPU (Origin 2100, 2200, 3200); в качестве процессоров используются MIPS R12000.

Сервера высшего уровня имеют архитектуру NUMA; Origin 2400 имеет от 2 до 512 CPU, Origin 3400 — от 4 до 32 CPU и Origin 3800 имеет от 16 до 512 CPU; в качестве процессоров используются MIPS R12000.

4.2.4. SGI 3000

25 июля 2000 г. компания SGI анонсировала новое семейство систем SGI 3000. Эти системы являются первыми системами, построенными на основе технологии NUMAFlex.

Данный компьютер построен на базе 64-битового процесора Itanium корпорации Intel. Кроме того, он будет работать под управлением ОС

Linux. Системы SGI 3000 имеют модульную архитектуру, которая позволяет строить системы, полностью отвечающие требованиям любых специализированных задач за счет разделения компонент компьютера на независимые функциональные блоки. По планам компании новый компьютер на базе Intel-Linux будет выпущен сразу же после начала выпуска компанией Intel процессоров Itanium. К сожалению, сроки появления этого процессора постоянно отодвигаются, и когда именно это произойдет, пока до конца не ясно.

4.2.5. Архитектура Origin 3x00

Архитектура NUMAFlex очень близка к той, что была использована в Origin 2000, т.е. S2MP. Большая часть усовершенствований связана с конструктивными изменениями: в Origin 3x00 резко увеличена модульность и надежность системы.

Основным блоком конструкции новых серверов стал "кирпич" (brick); при этом кирпичи бывают разных типов, в зависимости от их содержимого. Однако основные элементы архитектуры S2MP сохранены, т.е. сохраняется то, как связываются между собой процессоры, оперативная память, концентраторы, маршрутизаторы и подсистема ввода-вывода. То, что ранее было реализовано в виде плат, "превратилось" в кирпичи, "провода" на системной плате типа midplane заменены кабелями (таких плат в NUMAFlex больше нет).

Наряду с C-кирпичами используются другие кирпичи (всего 6 типов).

Очень важной особенностью архитектуры NUMAFlex, определяемой применением кирпичей, является исключительно высокая гибкость в построении различных конфигураций и сохранение инвестиций пользователя при модернизации. Заказчик приобретает только те кирпичи, которые ему действительно нужны, и "складывает" из них компьютер нужной конфигурации (конечно, сначала компьютер собирают все-таки на заводе). В случае же приобретения многопроцессорных систем, использующих конструктив общей системной шины на платах типа backplane или midplane, платить приходится за каждую такую плату со всеми расположенными на ней компонентами. Конкретной иллюстрацией такой гибкости может служить проведенное выше сопоставление Origin 3x00 и 2000 в части поддержания слотов XIO. С другой стороны, наращивание конфигурации минимальными порциями по 4 процессора, что, естественно, достаточно дорого, выбивается из общей картины.

5. NEC CENJU-4

В этом разделе мы дадим, следуя работе [2], краткий обзор ЭВМ NEC Cenju-4 — платформы для разнообразных построений.

5.1. Краткий обзор

Cenju-4 есть параллельный компьютер, построенный и выпускаемый компанией NEC.

Cenju-4 представляет собой мультипроцессор с архитектурой NUMA. Многоступенчатая сеть этой машины связывает до 1024 узлов, используя 4×4-шаговый коммутатор. Сеть имеет следующие особенности: симметричная (in-order) посылка сообщений между двумя любыми узлами, функции группировки и рассыпания, механизм, свободный от дедлоков. Каждый узел Cenju-4 состоит из процессора R10000, 1 Мб вторичного кэша, главной памяти объемом 512 Мб и чипа контроллера. Чип контроллера допускает посылку сообщений на уровне пользователя и DSM-доступ. Не допускается использовать и посылку сообщений, и DSM-доступ на одной и той же странице. Этот атрибут управляется ОС, которая основывается на микроядрах MACH per page bases.

5.2. Распределенная общая память

DSM ЭВМ Cenju-4 реализуется путем использования когерентных кэшей и основанных на директивах протоколах когерентности. DSM имеет следующие четыре характеристики.

- Директория динамически переключает ее представление со структуры указателей на структуру побитового шаблона согласно числу узлов. Эта схема требует постоянной области памяти независимо от числа процессоров, достигая эффективной записи узлов. Это дает масштабирование в аппаратуре и производительности.
- DSM в Cenju-4 использует функции группировки и рассыпания сети для доставки запросов и накапливания ответов. Это уменьшает накладные расходы на сообщения о недопустимости кэша. Cenju-4 также принимает директорию, которая может специфицировать все узлы, кэшируя (cashing) блок с одним доступом к памяти.
- Протокол когерентности кэша, который мешает “зависанию”. Cenju-4 принимает блокирующий протокол для когерентности

кэшей: запросы, которые не могут быть обработаны немедленно, помещаются в очередь в главной памяти для более поздней обработки. Размер буфера этой очереди — 32 Кб для 1024 узлов.

- Свободный от дедлоков механизм с одной сетью. Предлагается механизм, который образует очередь некоторых типов сообщений для когерентности кэша в главной памяти. Размер буфера, требуемого для создания очереди сообщений, равен 128 Кб для 1024 узлов. Этот буфер размещается в области, отличной от предыдущего буфера. Этот буфер и предыдущий для зависания размещаются и используются в различных функциях. Протокол когерентности кэша и свободный от дедлоков механизм гарантируют доступы к общей памяти с окончанием в конечное время.

Пользователи должны добавить вызовы библиотеки, чтобы использовать DSM-функции, так как компилятор, который может генерировать код для утилизации DSM-функции, не доступен. Общее адресное пространство размещается путем использования вызова библиотеки, и общие переменные размещаются или переразмещаются в таком пространстве. В будущем размещенные общие переменные могут быть доступными, так же как и приватные данные.

Существуют некоторые ограничения на использование DSM в Cenju-4: во-первых, наибольший объем общего адресного пространства ограничен объемом физической памяти. Далее, объем общего адресного пространства ограничено 2 Гб. Это объясняется тем, что используется архитектура с MIPS процессором, которая ограничивает адресное пространство пользователя 2 Гб.

6. ЗАКЛЮЧЕНИЕ

При написании данной статьи мы использовали доступные материалы из сети Internet. При этом принимались во внимание как электронные версии статей, так и информационные и рекламные материалы. Были использованы статьи И. Бородина “Архитектура современных суперкомпьютеров” (2001 г.), Е. Коваленко “Система Sequent NUMA-Q” (1997), М. Мосейкина “Параллельные системы и кластеры: проблема выбора” (1998), М. Сонгини, Д. Коннора “Возможные перспективы архитектуры NUMA” (1998), М. Кузьминского «“Кирпичные” компьютеры». Серверы нового поколения архитектуры NUMA компании SGI (2000) и др.

СПИСОК ЛИТЕРАТУРЫ

1. **Евстигнеев В. А.** NUMA-архитектура: некоторые особенности компиляции и генерации кода // Поддержка супервычислений и Интернет-ориентированные технологии — Новосибирск: ИСИ СО РАН. 2001. — С. 44–53.
2. **Kusano K., Sato M., Hosomi T., Seo Y.** The Omni OpenMP Compiler on the distributed shared memory of Cenju-4 // Lect. Notes Comput. Sci. — 2001. — Vol. 2104. — P. 20–30.