NUMA-системы

Вычислительные системы с неоднородным доступом к памяти

NUMA - системы

В NUMA-системых фигурирует единое адресное пространство, но каждый процессор имеет кэш-память. Доступ процессора к собственной памяти производится напрямую, что намного быстрее, чем доступ к общей памяти через коммутатор или сеть.

При наличии у каждого процессора локальной кэш-памяти существует высокая вероятность того, что нужные команды или данные уже находятся в кэше. Это существенно уменьшает число обращений процессора к общей глобальной памяти

NUMA - системы

В рамках концепции NUMA реализуется несколько различных подходов:

- COMA
- NCC-NUMA
- CC-NUMA
- RM (Reflexive Memory)

Наиболее распространенной из которых является архитектура CC-NUMA.

COMA – системы
(Cache Only Memory Architecture)

В *архитектуре COMA (только с кэш-памятью)* «локальная память» каждого процессора построена как большая кэш-память, которую называют *притягивающей памятью* (attraction memory - AM). Отличие от обычного кэш – нет блока, в который помещаются удаляемые данные. «Удаляемый» из AM одного вычислительного модуля фрагмент должен быть помещён в AM другого модуля.

Кэши всех процессоров в совокупности рассматриваются как общая глобальная память системы.

COMA – системы
(Cache Only Memory Architecture)

АМ содержит для каждого фрагмента данных тэг, включающий адрес и состояние фрагмента. При промахе контроллер просматривает теги АМ для обнаружения данных в «локальной» АМ. При отсутствии данных – вырабатывается запрос на доставку их из другого АМ.

COMA – системы
(Cache Only Memory Architecture)

При запросе данных (фрагмента памяти) из другого ВМ поступившие данные размещаются как в «настоящей» кэш-памяти, так и в АМ.

В системах с *плоской* COMA архитектурой (flat-COMA) в каждом ВМ имеется каталог для указания резидентных данных (фрагментов).

COMA – системы
(Cache Only Memory Architecture)

Для поиска данных (фрагмента) в *иерархических* COMA-системах используются специальные аппаратные средства сети межмодульных связей.

В системе **KSR-1** сеть строится как древовидная иерархия колец, в которой ВМ служат листовыми элементами.

В системе **DDM** - сеть это древовидная иерархия шин.

Каждый уровень имеет каталог, содержащий сведения обо всех данных, размещенных в нижележащих ВМ-листьях.

COMA – системы
(Cache Only Memory Architecture)

Принципиальная особенность COMA выражается в динамике – данные не привязаны к определённому модулю памяти и не имеют уникального адреса. Данные переносятся в кэш-память того процессора, который последним их запросил.

Перенос данных не требует участия операционной системы, но требует сложную и дорогостоящую аппаратуру управления памятью – каталоги кэшей.

\+ Высокая производительность

\- Если переменная (переменные) в строке кэша требуются двум процессорам, строка должна перемещаться между процессорами.

ncc-NUMA – системы
(Non Cache Coherent NUMA)

В системах *с кэш - некогерентным доступом к неоднородной памяти* архитектура памяти предполагает единое адресное пространство, но не обеспечивает согласованности глобальных данных на аппаратном уровне.

Управления использованием таких данных полностью возлагается на программное обеспечение (приложения и компиляторы).

ncc-NUMA – системы
(Non Cache Coherent NUMA)

Программные приемы решения проблемы когерентности позволяют обойтись без дополнительного оборудования или свести его к минимуму.

Задача возлагается на компилятор и операционную систему.

ncc-NUMA – системы
(Non Cache Coherent NUMA)

Привлекательность такого подхода - в возможности устранения некогерентности еще до этапа выполнения программы, однако принятые компилятором решения могут в целом отрицательно сказаться на эффективности кэш-памяти.

Учитывая, что некогерентность возникает только в результате операций записи, происходящих значительно реже, чем чтение, рассмотренный прием следует признать недостаточно удачным.

ncc-NUMA – системы
(Non Cache Coherent NUMA)

Лучшее решение - в ходе анализа программы определять безопасные периоды использования общих переменных и так называемые критические периоды, где может проявиться некогерентность.

Данная архитектура оказывается весьма полезной при повышении производительности вычислительных систем с архитектурой памяти типа DSM (Distribute Shared Memory).

cc-NUMA – системы (Cache Coherent NUMA)

Согласно технологии CC-NUMA (Cache Coherent), каждый узел в системе владеет собственной основной памятью, но с точки зрения процессоров имеет место глобальная адресуемая память, где каждая ячейка любой *«локальной» основной памяти* имеет уникальный системный адрес.

Модули памяти связаны коммутационной сетью (кабелем), а также имеются «умные» аппаратные средства для поддержания когерентности данных.

Не требуется какого-либо программного обеспечения для сохранения множества обновлённых копий или их передачи.

Доступ к «локальным» модулям основной памяти в разных узлах системы может производится одновременно и происходит быстрее, чем к удалённым модулям памяти.

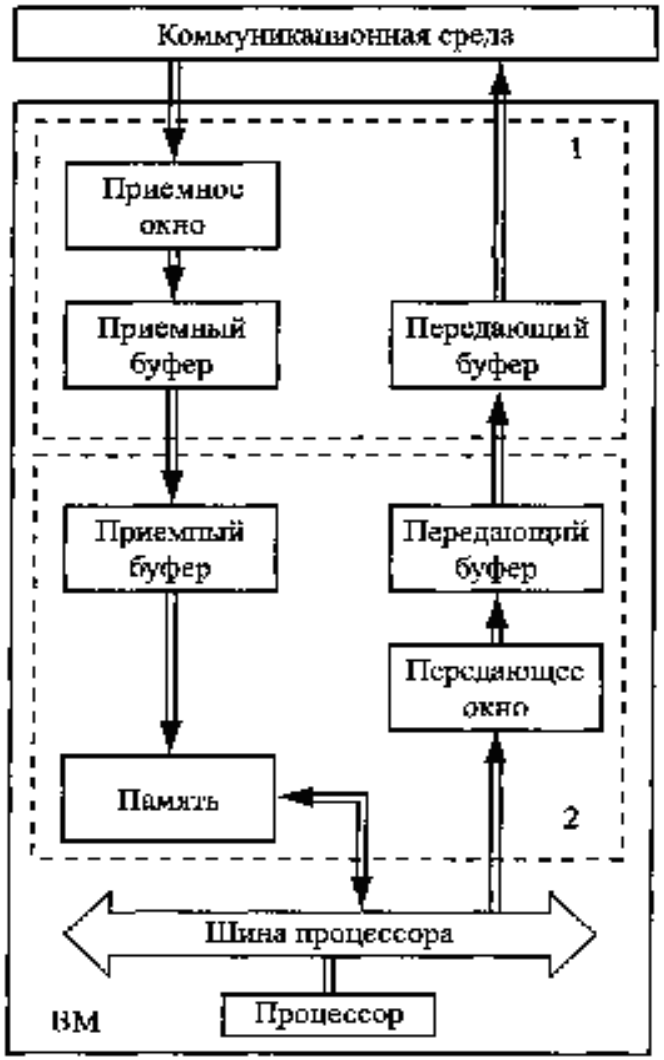
cc-NUMA – системы (Cache Coherent NUMA)

CC-NUMA системы в последнее время рассматривают как самостоятельный вид архитектуры вычислительных систем.

В качестве вычислительных модулей (ВМ) современных NUMA-систем могут выступать небольшие SMP-системы.

Часто NUMA-системы называют архитектурами с виртуальной распределённой общей памятью (virtual shared memory architectures), что не совсем верно. (*DSM!*)

RM – системы
(Reflexive Memory)



Основное свойство рефлексивной памяти – каждая копия разделяемого данного в ЛП любого ВМ имеет одно и то же значение.

Система состоит из совокупности ВМ и коммуникационной среды.

Каждый ВМ имеет процессор, блок памяти, шину процессора, по которой он обращается в память, и адаптер коммуникационной среды (сетевой интерфейс).

Адаптер состоит из приемо-передающей части коммуникационной среды (1) и приемо-передающей части процессора (2).

RM – системы
(Reflexive Memory)

Совокупность страниц ЛП каждого ВМ делится на две группы:

- локальные неразделяемые страницы
- глобальные разделяемые страницы.

Рефлексивная память образуется из всех этих распределенных по различным блокам физической памяти глобально разделяемых страниц памяти, отображенных в глобальное разделяемое адресное пространство.

ВМ, создающий входную страницу (переменную) делает страницу доступной для разделения с другими ВМ.

Отображение глобального адреса в локальный выполняется таблицей преобразования адресов в приёмном окне адаптера. Передающее окно содержит таблицу трансляции адресов разделяемой выходной страницы в адреса глобального адресного пространства.

RM – системы
(Reflexive Memory)

ВМ, создающий выходную страницу, заполняет элемент таблицы преобразования адресов выходной страницы в адреса глобального адресного пространства, задавая в каждом элементе адрес ВМ, в который будет производиться пересылка данных.

После создания входных и выходных страниц доступ к виртуальной памяти выполняется командами ***load, store***. При каждой записи в разделяемую переменную автоматически (прозрачно для исполняемой программы) изменяются все копии этой переменной.

Все чтения данных производятся из локальных памятей ВМ, и только записи в разделяемые переменные используют коммуникационную среду, объединяющую ВМ.

Если ВМ выполняет команду записи в переменную, расположенную в неразделяемой памяти, то эта команда изменяет содержимое только локальной памяти процессора. Иначе, помимо изменения переменной этой страницы, активизирует передающее окно адаптера.

RM – системы
(Reflexive Memory)

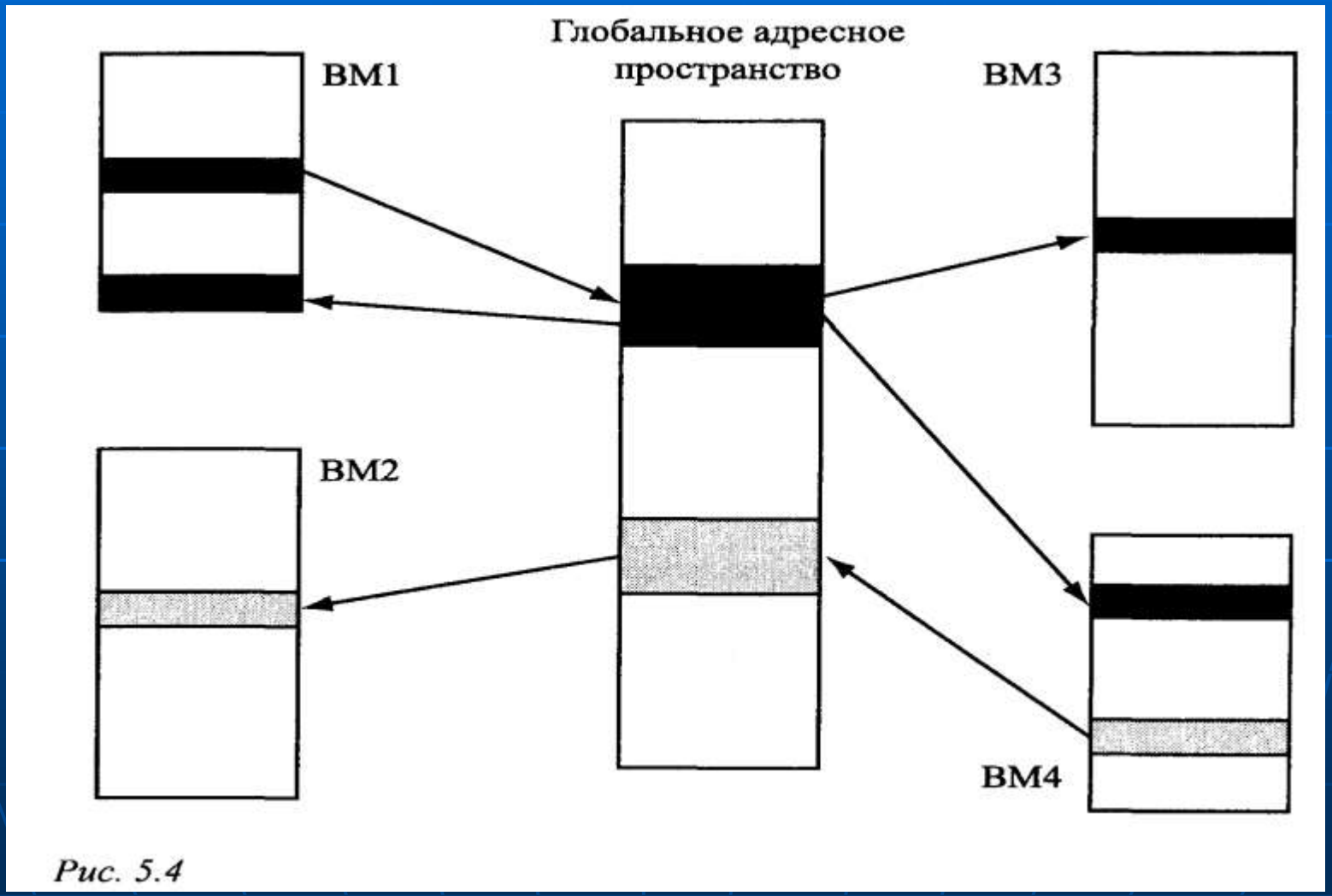


Рис. 5.4

RM – системы
(Reflexive Memory)

Особенность систем с RM - пользователь имеет возможность управлять степенью разделения данных в системе. Число строк таблиц трансляции адресов равно числу разделяемых страниц памяти, и каждой странице однозначно соответствует строка таблицы.

Таким образом, вместо фиксированной глобальной адресации всей памяти применяется выборочное отображение отдельных страниц в глобальное адресное пространство.

Содержимое этих таблиц может задаваться либо статически при загрузке параллельной программы, либо может устанавливаться динамически в ходе исполнения программы.

Вопросы проверки корректности (согласованности и безопасности) заполнения этих таблиц решаются средствами ОС.

RM – системы
(Reflexive Memory)

Недостатки:

- сложность модификации многих копий страниц из-за широковещательных передачах пакетов;
- излишней нагрузкой на коммуникационную среду при многократной модификации одной и той же переменной;
- невозможность использовать в качестве разделяемых страницы, помещаемые во внутрикристальную кэш-память с обратной записью;
- существенно более продолжительная длительность модификации всех кэш-страниц требует введения барьерной синхронизации ВМ при записи в глобальные переменные для поддержания когерентности памяти в рамках модели свободной состоятельности.

Существуют как промышленные, так и экспериментальные системы с рефлекивной памятью. В качестве яркого примера промышленной системы можно упомянуть кластеры фирмы DEC AlphaServer 8xxx, использующие рефлексивную память на базе Memory Channel.

К экспериментальным относятся системы Merlin, Sesame, Shrimp.