УДК 004.382.2

Обзор подходов к реализации программно-аппаратной поддержки общей памяти в многопроцессорных вычислительных системах

М.А. Гильмендинов
науч. сотрудник, ОАО «НИЦЭВТ»
E-mail: gilmendinov@nicevt.ru
А.С. Фролов
начальник отдела, ОАО «НИЦЭВТ»
E-mail: frolov@nicevt.ru

Приведены обзор программно-аппаратных механизмов для поддержания общей памяти в многопроцессорных системах. Рассмотрены системы SGI Altix UV, NumaScale, vSMP и перспективная разработка университетов Валенсии и Гейдельберга под названием MEMSCALE. Рассмотрена возможность реализации полностью прозрачного для пользовательских приложений механизма расширения динамической памяти на базе отечественной коммуникационной сети ЕС8430 подходящего для задач с умеренным тредовым параллелизмом.

***Ключевые слова**: подсистема памяти, коммуникационная сеть, параллельные вычисления.*

In this article we review a hardware-software mechanisms for implementing multiprocessor shared memory systems. SGI Altix UV, NumaScale, vSMP and a perspective project of Valencia and Heidelberg universities under the name MEMSCALE are considered. Possibility of fully transparent mechanism for dynamical extension of main memory implementation based on ES8430 interconnect for moderate thread parallelism applications is considered.

***Keywords:** memory subsystem, interconnect, parallel computations.*

Введение

Объемы данных, с которыми оперируют прикладные задачи, постоянно увеличиваются и ограничиваются только физическими ресурсами вычислительной системы (оперативная память, flash-память, дисковые накопители). В большинстве случаев использование медленной дисковой памяти крайне не эффективно даже с применением технологии подкачки виртуальных страниц. В области бизнес-аналитических систем и баз данных активно развиваются технологии под общим названием in-memory computing, включающие в себя базы данных в памяти (in-memory database), серверы-приложений в памяти (in-memory applications servers), аналитические системы в памяти (in-memory analitics) и др. Для эффективного решения таких задач используются либо специализированные компьютеры с аппаратной поддержкой общей памяти(SGI Altix UV, Bullx, NumaScale) либо программные реализации протоколов работы с общей памятью для кластерных систем без аппаратной поддержки общей памяти(vSMP, Intel Clustered OpenMP). Отдельно стоит отметить идею использования памяти других узлов без соблюдения когерентности памяти, развиваемую в проекте MEMSCALE. В данной работе дается краткий обзор существующих систем с аппаратной и программной поддержкой общей памяти, а также предлагается подход по поддержке динамического расширения памяти в вычислительных системах на базе разрабатываемой отечественной высокоскоростной коммуникационной сети ЕС8430.

Системы с поддержкой общей памяти

Существуют различные подходы к реализации общей памяти в аппаратных вычислительных системах. Примером систем с аппаратной поддержкой общей памяти являются SGI Altix UV и NumaScale. Обе системы реализуют кэш-когерентную общую память(ccNUMA), но используют различные интерфейсы подключения к коммуникационной сети: SGI Altix UV – Intel QPI и NumaScale – HT(рис. 1).

Основным недостатком этих систем является ограниченность масштабируемости, связанная с накладными расходами, вызванными обеспечением когерентности памяти вычислительных узлов.

В отличие от ранее рассмотренных аппаратных систем и программных реализаций общей памяти, MEMSCALE имеет два принципиальных отличия: во-первых, в MEMSCALE не поддерживается когерентность общей памяти, во-вторых, задачи в MEMSCALE могут использовать ядра только в пределах вычислительного узла, в то время как память может выделяться без ограничений на любых вычислительных узлах (рис. 2).

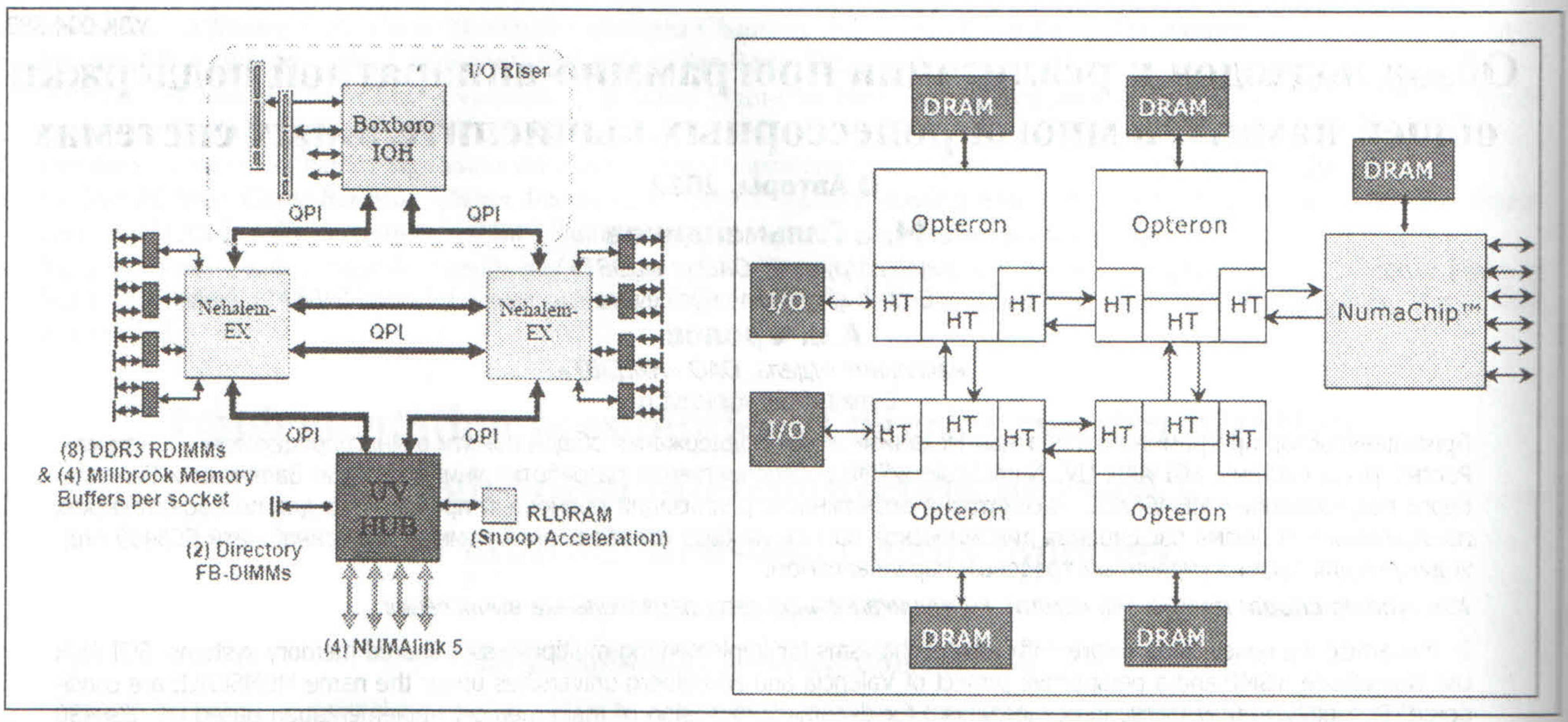

Рис. 1. Схема вычислительных узлов SGI Altix UV и NumaScale

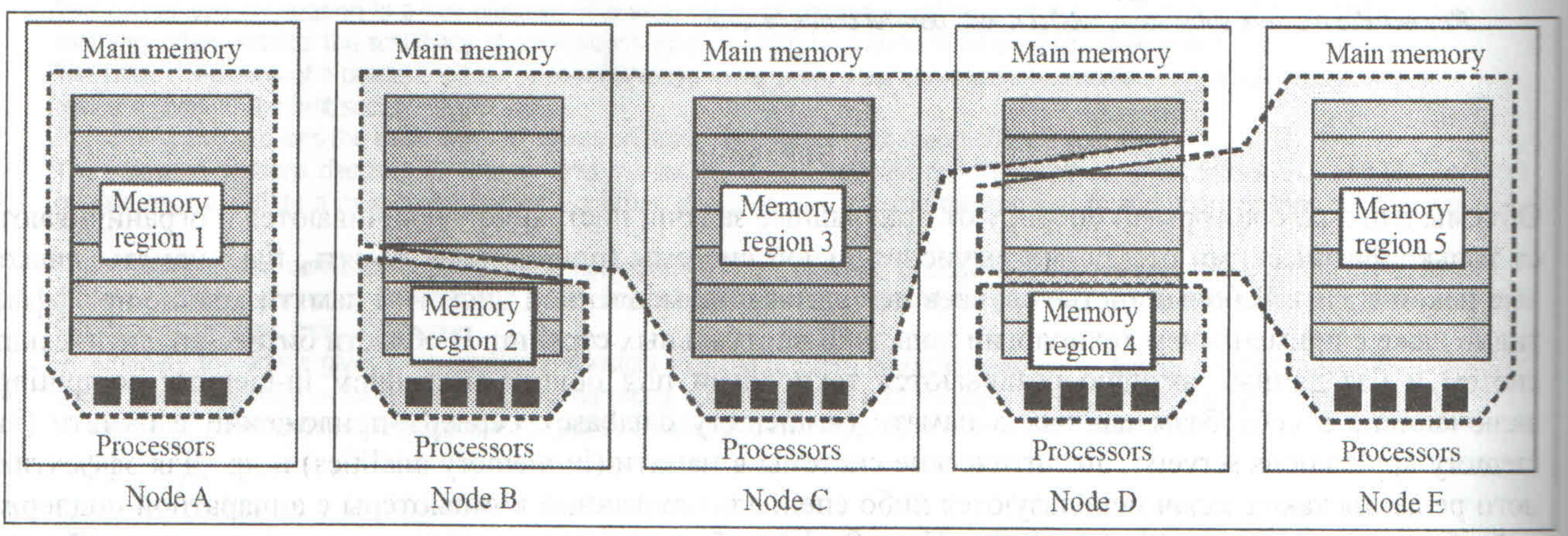

Рис. 2. MEMSCALE: пример распределение памяти между узлами

Такой подход ограничивает применение MEMSCALE задачами с небольшим тредовым параллелизмом, но требующих больших объемов используемой памяти.

При такой организации общей памяти должна сохраняться масштабируемость, а накладные расходы на работу с памятью будут состоять из ограничений накладываемых интерконнектом. Рассмотрим механизм работы более подробно.

На рисунке 3, четыре процессора, находящиеся на одной материнской плате соединены каждый со своим контроллером памяти и друг с другом через HyperTransport(HT). Помимо процессоров к шине HT подключен контроллер удаленной памяти (Remote Memory Controller). При использовании механизма виртуальной адресации, каждому виртуальному адресу соответствует физический адрес памяти доступ по которому осуществляется через контроллер памяти непосредственно соединенный с процессором, через HT к контроллеру памяти подсоединенного к другому процессору или через HT к контроллеру удаленной памяти.

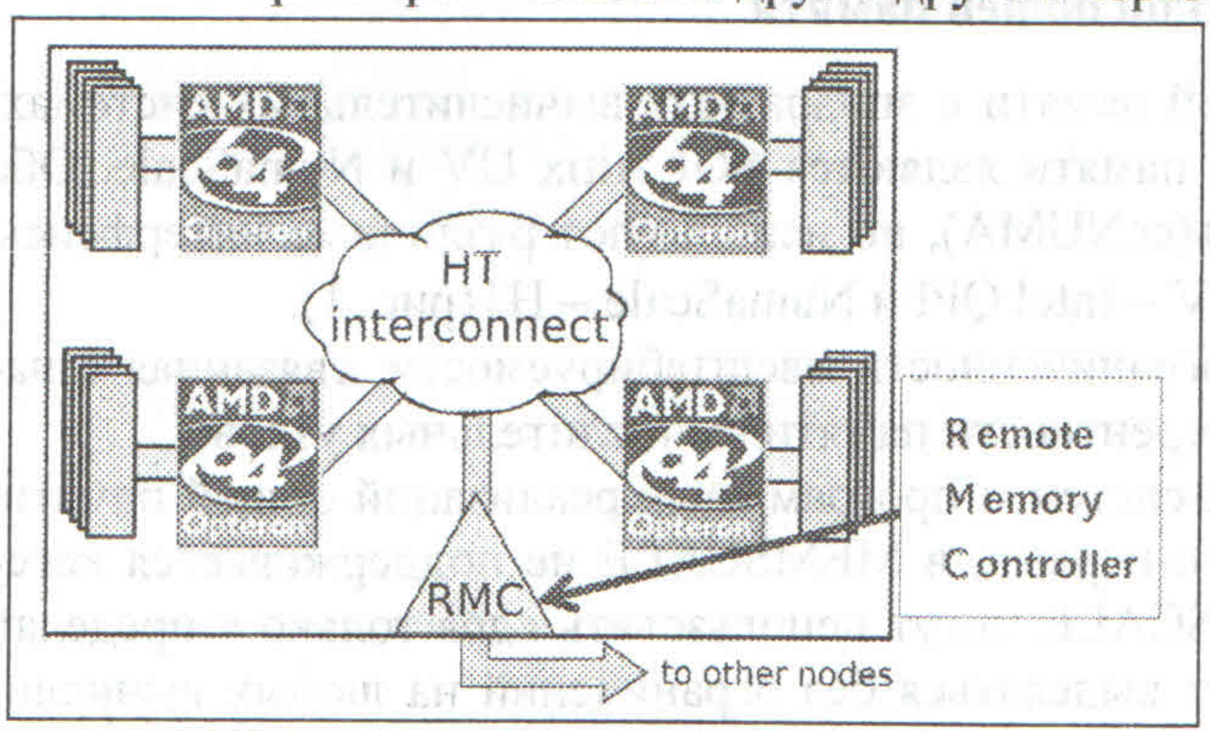

Рис. 3. MEMSCALE: Контроллер удаленной памяти

Выделение памяти на удаленном узле показано на рисунке 4. При запросе на выделение памяти отправляется запрос к RMC, который перенаправляет

его на удаленный узел. На удаленном узле выделяется память, а полученный в результате физический адрес дополняется номером узла и отправляется узлу запросившему память.

На данный момент MEMSCALE поддерживает до 256 узлов. На основе MEMSCALE собран экспериментальный кластер из 64 узлов, соединенных прототипным вариантом сети EXTOLL, реализованном на ПЛИС.

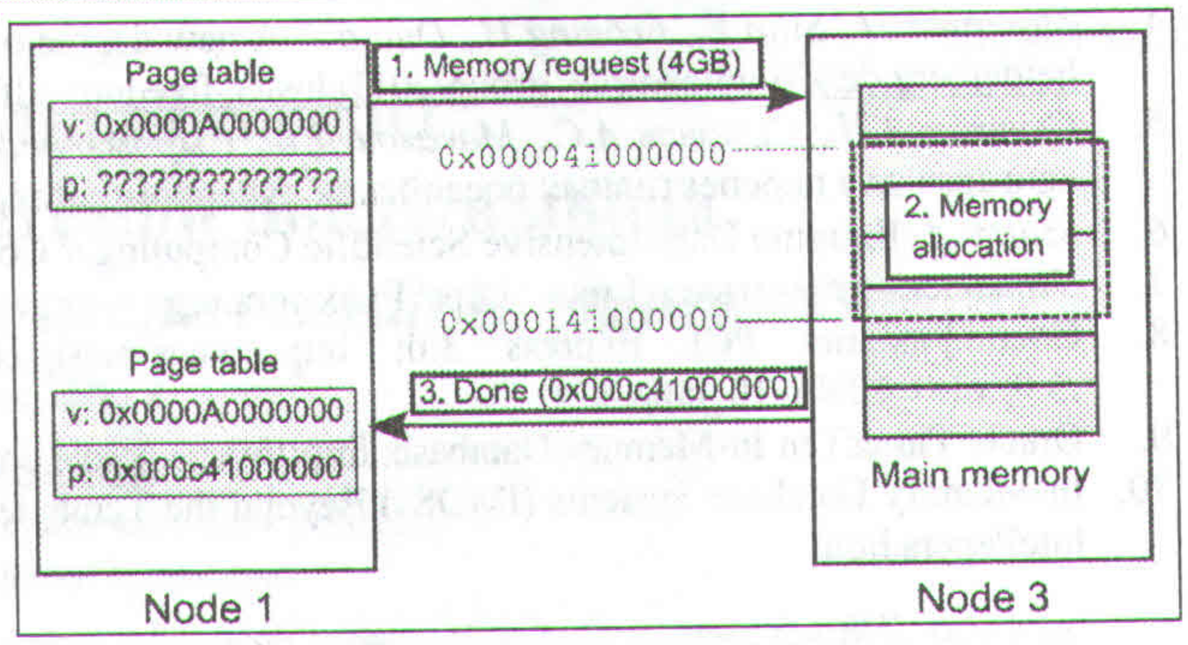

Рис. 4. MEMSCALE: выделение памяти на удаленном узле

Коммуникационная сеть EC8430

В ОАО «НИЦЭВТ» разрабатывается отечественная высокоскоростная коммуникационная сеть EC8430 с топологией 4D-тор. В данной работе рассматривается возможность адаптации и расширения идей MEMSCALE как один из подходов к реализации программно-аппаратного механизма, обеспечивающего динамическое выделение оперативной памяти на удаленных узлах вычислительного кластера, на базе разрабатываемой отечественной коммуникационной сети. Для реализации аппаратной поддержки в маршрутизатор будет добавлен дополнительный конвейер для обработки запросов на чтение/запись удаленной памяти, поступающих от локального узла через интерфейс PCIe. Также необходимо назначение регистров базового адреса (BAR) PCIe на уровне BIOS для перенаправления на них запросов на получение физических страниц. Адресный флит будет формироваться аппаратно на основе данных о номере узла, физическом адресе и типе операции из PCIe пакета. Для приема и посылки страниц данных будут использоваться команды PCIe чтения и записи, в дальнейшем возможно расширение функциональности: атомарные операции(CAS, FADD), а также входящие в спецификацию PCIe 3.0. Поскольку все обращения в память будут производиться из одного узла, то поддержание когерентности между узлами не требуется.

На уровне ОС требуется обеспечить поддержку выделения памяти на удаленном узле. Для этого будет разработан модуль ядра, который будет перенаправлять запросы на выделение памяти удаленной памяти на PCIe устройство и ставить в соответствие виртуальному диапазону адресов непрерывный диапазон физических адресов соответствующих устройству PCIe для узла-потребителя. На узле-доноре модуль ядра будет отвечать за резервирование непрерывного диапазона физических адресов.

Реализация предлагаемого подхода состоит в решении следующих задач:

1. Реализация поддержки обработки удаленных команд обращений к памяти в адаптере коммуникационной сети.
2. Модификация менеджера памяти в ядре ОС для отображения виртуальных страниц на область ввода-вывода для перенаправления команд обращений к памяти в коммуникационную сеть.
3. Разработка библиотеки для выделения памяти с возможностью управления распределением памяти на уровне пользователя.

Заключение

Итак, рассмотрены программные и аппаратные механизмы поддержания общей памяти, включая проект MEMSCALE. Предложен подход к реализации программно-аппаратного механизма расширения динамической памяти, доступной узлу вычислительной системы. Планируется внедрение описанного механизма в 9-узловой макет сети EC8430 и тестирование производительности системы на приложениях, эффективность работы которых зависит от количества доступной памяти. В качестве примера таких приложений можно привести базы данных в памяти.

Литература

1. SGI Altix UV: http://www.sgi.com/products/servers/uv/specs.html
2. ScaleMP: http://www.scalemp.com
3. NumaScale: http://www.numascale.com