

ИМИТАЦИОННАЯ МОДЕЛЬ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ В СРЕДЕ GPSS WORLD

А.С. Степанова,

магистрант ФИБС МФТИ, кафедра системной архитектуры,
стажер-специалист ООО «Информационные бизнес системы»,
e-mail: astepanova@ibs.ru.

Е.В. Чепин,

кандидат технических наук, доцент, зам. зав. кафедрой КСиТ МИФИ,
зам. зав. кафедрой СА ФИБС МФТИ,
e-mail: evchepin@gmail.com.

Адрес: 127434, г. Москва, Дмитровское шоссе, д. 9Б, ООО «ИБС».

В статье описана функциональная модель многопроцессорной системы и реализация модели в среде моделирования GPSS World. Исследуется производительность многопроцессорной системы в зависимости от числа процессоров, работающих над вычислительной задачей.

Ключевые слова: параллельная обработка данных, параллельное программирование, многопроцессорная система, машина с параллельным случайным доступом, массовая случайная параллель, двуядерность, хаб.

1. Введение

Проектирование мощных вычислительных систем является сложной технической задачей, одним из аспектов которой является проблема создания системы с заданной производительностью при наименьшем количестве затраченных ресурсов. Обычно используются следующие подходы:

- ◆ Эталонные программы, измеряющие производительность (benchmark).
- ◆ Тестирование от экспертных организаций.
- ◆ Математическое моделирование вычислительной системы. Достоинством данного способа является его гибкость и относительно малая ресурсоемкость.

Производительность вычислительной системы значительно зависит от ее архитектуры. Так, производительность SMP-систем ограничена сверху пропускной способностью системной шины, зато программировать их сравнительно просто, т.к. вся память в системе является общей. Кластерная вычислительная система по мере роста начинает проигрывать в производительности из-за сравнительно низкой пропускной способности межузловых коммуникаций и сложности программно-аппаратных средств. ccNUMA-системы объединяют в себе как достоинства, так и недостатки SMP и кластера: все пространство памяти в системе имеет единую адресацию, пропускная способность может масштабироваться по мере добавления новых узлов, однако

существенной проблемой является поддержка когерентности кэшей.

Моделирование многопроцессорных систем имеет большое значение в теории вычислительных систем (computer science). В данной статье не содержится полномасштабного обзора таких исследований, однако можно отметить следующие работы, создавшие основу для современного подхода к моделированию:

❖ Машина с параллельным случайным доступом (PRAM, Parallel Random Access Machine, [1]) обеспечивает абстракцию машины с разделяемой памятью. PRAM является расширением модели последовательной машины с произвольным доступом RAM - Random Access Machine). Основным параметр – число процессоров.

❖ Модель BSP (Bulk Synchronous Parallel, массовая синхронная параллельная, [2]) состоит из процессоров, передающих друг другу сообщения через коммуникационную сеть. Каждый процессор имеет локальную память, время обращения к которой считается нулевым. Основные параметры BSP: p – количество процессоров; g – время передачи единицы данных через сеть, величина, обратная пропускной способности, приходящейся на один процессор; L – латентность сети. Вычислительный процесс считается состоящим из сверхшагов, между которыми происходит синхронизация.

❖ LogP [3] – модель многопроцессорной системы с распределенной памятью, в которой процессоры общаются с помощью сообщений «точка-точка». L – латентность сети (верхняя граница); o – накладные расходы, определяется как время, которое процессор тратит на передачу или прием каждого сообщения; в это время процессор не может выполнять другие операции; g : время передачи единицы данных через сеть, величина, обратная пропускной способности, приходящейся на один процессор; P : количество процессоров и модулей памяти.

В этих моделях топология сети не учитывается.

2. Постановка задачи

В данной работе моделируется ccNUMA SGI Altix - ccNUMA-система с протоколом кэшкогерентности на основе каталогов. Данные приведены согласно технической документации фирм-производителей [4,5]. Моделируется работа процессорных ядер, кэшей и оперативной памяти, выделяются аспекты архитектуры процессора как элемента вычислительной системы. Рассматривается взаимодействие процессоров с памятью в случае МПС, состоящей из 8 узлов. Описана модель кэш-когерентного протокола, гарантирующего корректность доступа к данным. Модель реализована на языке GPSS World [6].

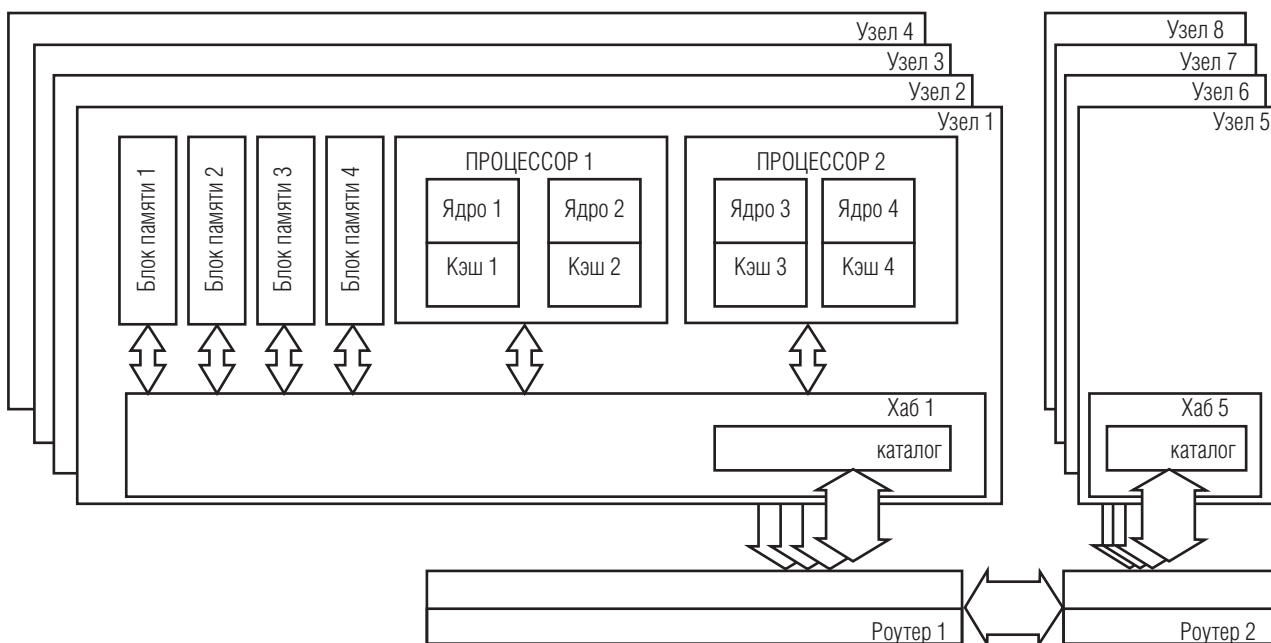


Рис. 1. Схема монтажного блока МПС

3. Функциональная модель

Узел МПС содержит два двухъядерных процессора, разделяющих общую шину, хаб, выполняющий также функции контроллера памяти и др. В модели одновременно протекают два процесса: выполнение программы в ядре процессора и предзагрузка данных в кэш. При этом другие протекающие в ядре процессы оставлены за рамками этого исследования.

Монтажный блок МПС состоит из 8 узлов и 4 роутеров (рис.1), соответственно содержит 32 ядра. Каждый хаб соединен с парой роутеров, которые можно моделировать одним двухканальным устройством. Каждый роутер отвечает за адресное пространство 4 лезвий.

В модели используется предположение о программе как о наборе из 1 000 000 полностью параллелизуемых инструкций (заявок в терминах СеМО). На процессоре последовательно генерируется определенное число заявок соответственно решаемой задаче. В случае кэш-промаха процессор обращается к кэшу уровня L1, самому быстрому, работа процессора тормозится. В слу-

чае, если данные получены, они отправляются в ядро, и процессор возобновляет работу. Иначе отправляется запрос в кэш L2, и т.д. В случае промаха в кэше L3 формируется запрос к основной памяти. Заявка направляется на хаб, также являющийся контроллером памяти, где она получает координаты исходного блока данных, который может располагаться как в локальной, так и в удаленной памяти. Заявка передается на роутер, отвечающий за исходное ядро, затем на роутер, отвечающий за целевое ядро, потом на хаб, к модулю памяти или кэшу, содержащему целевой блок данных, и обратно, до ядра, сформировавшего заявку, обслуживается на ядре и уничтожается. На обратном пути заявка уже содержит строку данных и испытывает дополнительную задержку из-за конечных пропускных способностей шин. Работа процессора возобновляется после получения данных.

События кэш-попаданий, оптимальное расположение данных (локализация внутри узла), обмен данных между узлами моделируются с помощью генератора случайных чисел: задается вероятность нахождения данных на текущем уровне памяти, затем выбирается случайное число от 0 до 1 и реализуется соответствующее событие.

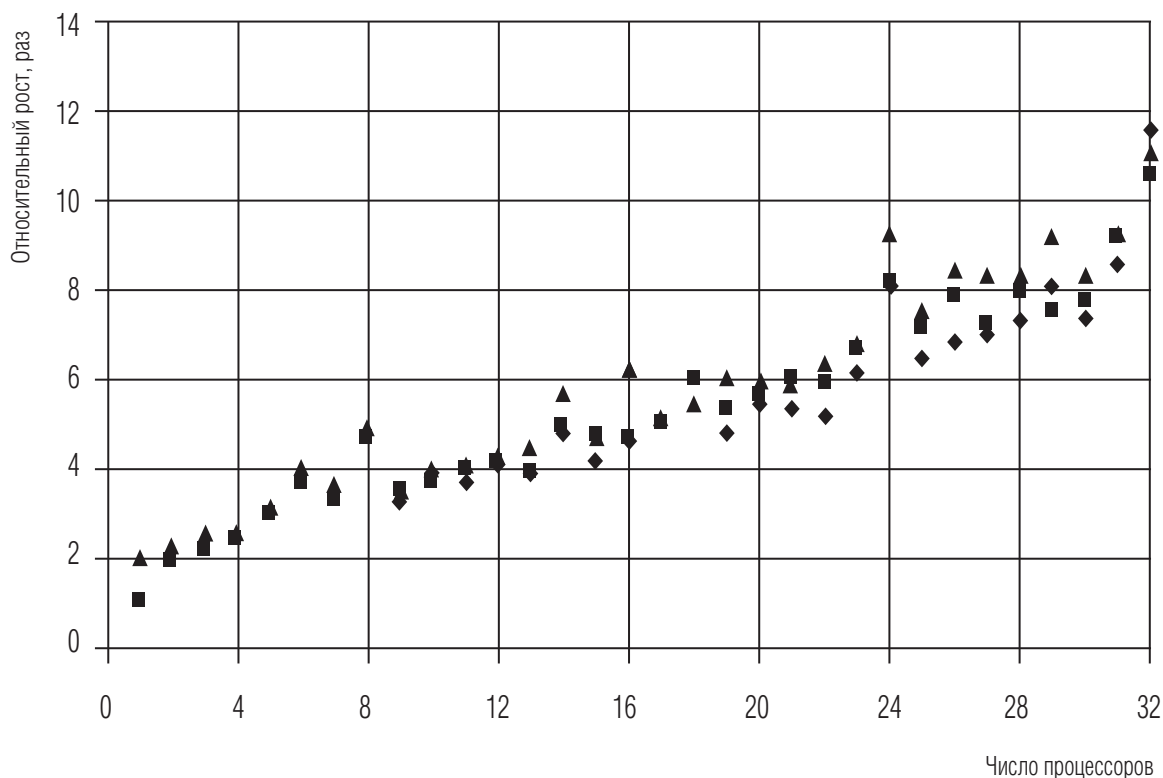


Рис. 2. Относительный рост производительности МПС при увеличении числа процессоров

Значения вероятностей задаются параметрически. Соответствующие статистические данные приведены в [7,8].

Параллельно идет предзагрузка данных. Ядро отправляет заявку на хаб, хаб адресует заявку к памяти, память формирует и передает страницу данных с заявкой по обратному пути. Размер страницы может быть от 4 Кб, что соответствует 32 строкам данных. Период генерации запросов предзагрузки обратно пропорционален размеру страницы.

С введением в модель протокола когерентности кэшей запрос к памяти становится более сложным за счет рассылки служебных сообщений другим процессорам, например, совладельцам запрашиваемого блока данных при запросе на чтение. Используются несколько упрощенные алгоритмы запросов [9].

4. Экспериментальные результаты

В качестве примера применения модели МПС рассматривается рост производительности МПС в зависимости от числа используемых процессоров (рис. 2). Наиболее производительны конфигурации из 8, 16, 24 и 32 процессоров. В комбинациях из 9, 17, 25 производительность системы резко падает, т.к. увеличение вычислительной мощности незначительно, зато накладные расходы на обмен данными между узлами возрастают.

5. Заключение

Модель предназначена для прогнозирования производительности МПС при различных вычислительных нагрузках. Результаты моделирования можно использовать на ранних этапах проектирования для оптимизации вычислений и достижения максимальной производительности МПС.■

6. Литература

1. Fortune S. J, Wyllie J. C. Parallelism in random access machines // Proceedings of the tenth annual ACM symposium on Theory of computing, p.114-118, May 01-03, 1978.
2. Valiant L. G. A bridging model for parallel computation // Communications of the ACM, volume 33, issue 8, p.103-111, August 1990.
3. Culler D. E, Karp R. M., Patterson D. A., Sahay A., Schauser K. E., Santos E. E., Subramonian R., von Eicken T. LogP: towards a realistic model of parallel computation // Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming, p.1-12, 1993.
4. SGI Altix Applications Development and Optimization // SGI, 2003.
5. Dual-Core Update to the Intel Itanium 2 Processor. Reference Manual. For Software Development and Optimization // Intel, 2006.
6. Кудрявцев Е. М. GPSS World. Основы имитационного моделирования. М.: ДМК Пресс, 2004. 320 с.
7. Babka V. Cache Sharing Sensitivity of SPEC CPU2006 Benchmarks. Technical report // Charles University, 2009.
8. Prakash T. K., Peng L. Performance Characterization of SPEC CPU2006 Benchmarks on Intel Core 2 Duo Processor // 2007 SPEC Benchmark workshop: Proc., January 2007.
9. Laudon J. P., Lenoski D. E. The SGI Origin: A ccNUMA Highly Scalable Server // ACM SIGARCH Computer Architecture News, 1997.