

УДК 004.94

МОДЕЛЬ РАБОЧЕЙ НАГРУЗКИ ДЛЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ ИЕРАРХИЧЕСКОЙ ПАМЯТИ МУЛЬТИПРОЦЕССОРА И ЕЕ ПРИМЕНЕНИЕ ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОТОКОЛОВ КОГЕРЕНТНОСТИ КЭШ-ПАМЯТИ

Сорока Т.Е., Фельдман Л.П.

Донецкий национальный технический университет

Описывается структура и функциональность имитационной модели мультипроцессора с общей (разделяемой) памятью, предназначенной для моделирования и оценки эффективности протоколов когерентности кэш-памяти на основе наблюдения. Детально рассматривается модель рабочей нагрузки, которая позволяет тестировать алгоритмы поддержки когерентности на различных трассах обращений к памяти. Дается оценка временной сложности модели.

Введение

Различие между быстродействием процессора и памяти всегда было одной из проблем увеличения производительности однопроцессорных вычислительных машин; многопроцессорность вычислительных систем приводит к еще одной проблеме – проблеме одновременного доступа к памяти со стороны нескольких процессоров. Использование общих данных в любой мультипроцессорной системе с разделяемой памятью, где каждый процессор имеет собственную кэш-память, приводит к ситуации, связанной с наличием в нескольких кэшах копий одних и тех же данных, и проблеме когерентности кэш-памяти. Проблема когерентности кэш-памяти является актуальной для нескольких классов вычислительных машин: это современные персональные компьютеры с многоядерными процессорами от Intel и AMD, суперкомпьютеры класса UMA (uniform memory access), например, Cray T90, Intel SHV, Sun E10000, IBM R60, а также более современные суперкомпьютеры класса ccNUMA (cache coherent non-uniform memory architecture), такие как SGI Origin 2000, Origin 3000, Cray T3E, HP Exemplar и др. [1].

Проектирование высокопроизводительных вычислительных систем с разделяемой памятью требует предварительного анализа структуры кэш-памяти и алгоритмов поддержки когерентности, которые затем предполагается реализовать на аппаратном уровне. Система моделирования, позволяющая оценивать эффективность структуры и алгоритмов работы кэш-памяти, позволит разрабатывать эффективные архитектуры для решения различных научных задач, которые смогут максимально использовать как потенциал процессоров, так и имеющуюся в системе память [2].

1 Структура моделирующей системы

Разработанная модель мультипроцессора представляет собой совокупность взаимодействующих моделей нескольких узлов: процессоров, кэш-памяти, кэш-контроллеров и системной шины. Моделируемая система представляет собой мультипроцессор с общей (разделяемой) памятью и отдельным кэшем для каждого ядра на основе общей системной шины (рис. 1). Обобщенная структура имитационной модели в виде параллельно выполняющихся программных потоков представлена на рис. 2, а функциональная структура модели – на рис. 3.

Данные, хранящиеся в основной памяти, не моделируются как таковые (имеет значение только наличие/отсутствие запрашиваемого блока данных в кэш-памяти и его состояние), поэтому и сама основная память, как отдельный объект, в имитационной модели отсутствует (рис. 2).

Систему моделирования также можно представить как функцию, вычисляющую вектор показателей эффективности системы от двух групп входных параметров: вероятностных характеристик модели рабочей нагрузки и некоторых технических параметров мультипроцессора (рис. 3).

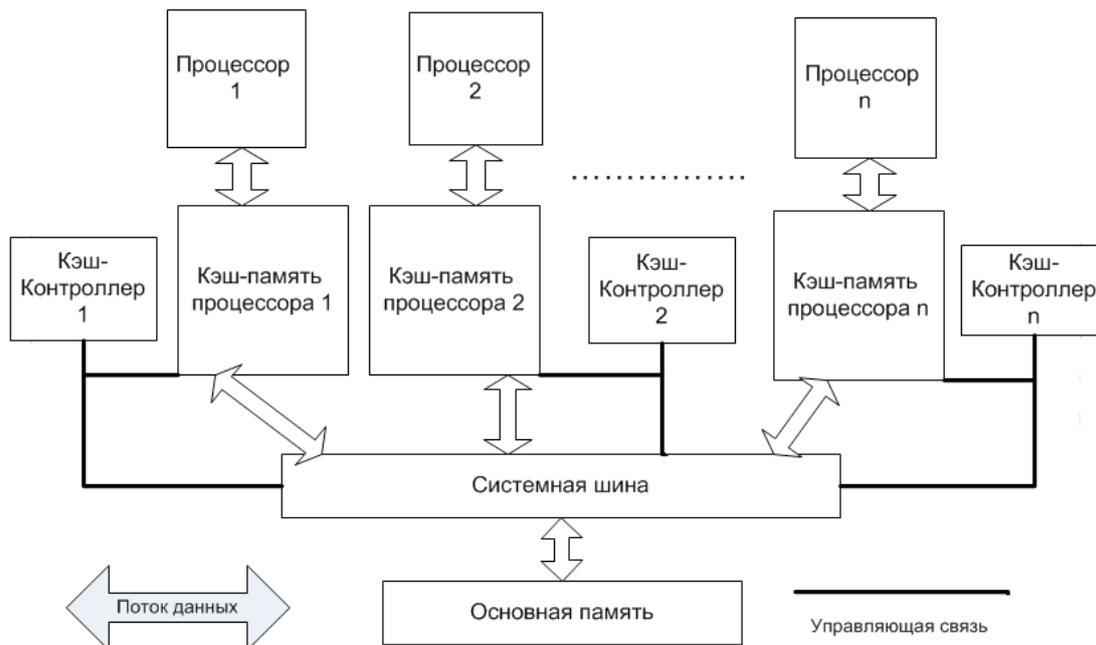


Рисунок 1. Структура моделируемого мультипроцессора



Рисунок 2. Структура имитационной модели мультипроцессора

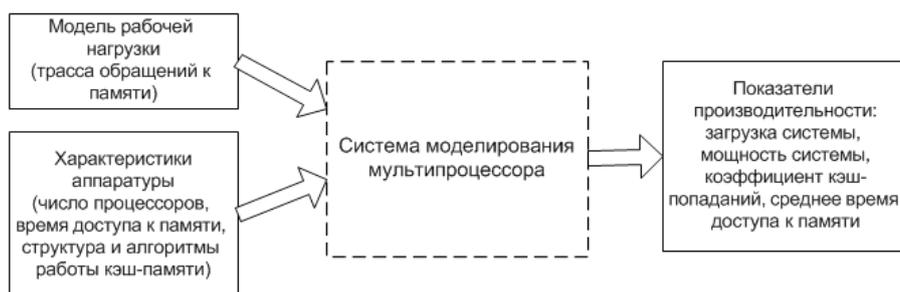


Рисунок 3. Функциональное представление системы моделирования

2 Модель рабочей нагрузки для исследования работы памяти

Рабочая нагрузка для исследования эффективности иерархической памяти представляет собой последовательность запросов к памяти, генерируемых процессором в ходе моделирования. Поскольку процессор в системе моделирования выполняет единственную роль – генератора рабочей нагрузки для памяти, то все его операции целесообразно разделить на 2 группы: запросы к памяти и другие операции, не связанные с памятью (фактически они не моделируются, просто занимая один такт процессорного времени). Запросы к памяти также можно разделить на несколько групп: по цели запроса – это запросы на чтение или на запись, а по адресу – запросы к области частных, либо разделяемых блоков (рис. 4) [3].

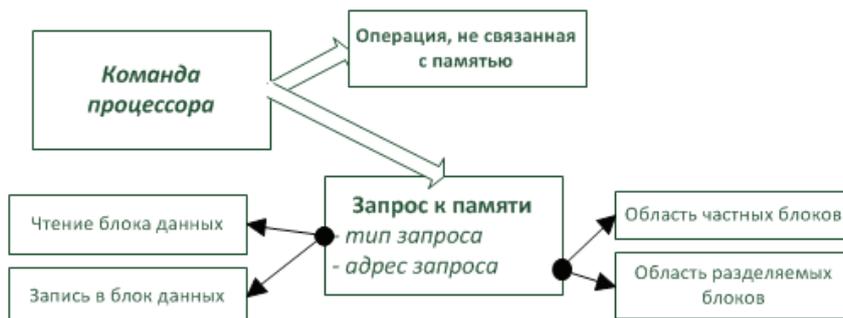


Рисунок 4. Типы команд процессора при моделировании

5

Модель рабочей нагрузки задается вектором из 6 параметров \bar{M} :

$$\bar{M} = \{P_{\text{запрос}}, P_{\text{чтение}}, P_{\text{разд}}, n_{\text{разд}}, \delta, t\}, \tag{1}$$

где $P_{\text{запрос}}$ – вероятность генерации запроса к памяти в очередной такт процессорного времени, $P_{\text{чтение}}$ – вероятность того, что сгенерированный запрос будет запросом на чтение, $P_{\text{разд}}$ – вероятность того, что сгенерированный запрос будет адресован к области разделяемых блоков, $n_{\text{разд}}$ – число разделяемых блоков (определяет размер области разделяемых адресов), δ – среднее отклонение адреса следующего запроса по отношению к предыдущему для области частных блоков, t – общее число операций, которое должен выполнить каждый процессор в ходе моделирования.

Каждый процессор генерирует запросы к одному и тому же линейному пространству адресов (рис. 5). При этом область частных блоков намного больше области разделяемых блоков ($K \gg (N-1)-K$), где N – размер основной памяти в байтах. Каждый процессор при генерации запроса к области частных блоков обращается к адресу, зависящему от случайного начального параметра и среднего отклонения адреса. Среднее отклонение адреса сопоставимо с размером блока, а начальные адреса распределены равномерно по всей области частных блоков, поэтому адрес не отклонится слишком далеко от начального параметра, что практически исключает вероятность образования разделяемых блоков в области частных адресов.

При запросе к области разделяемых блоков адрес запроса будет представлять собой равномерно распределенную случайную величину в области разделяемых адресов $[K+1..N-1]$. Поэтому при достаточно большом числе запросов к области разделяемых блоков большинство этих блоков станут фактически разделяемыми. Алгоритм генерации операции процессора представлен на рис. 6.



Рисунок 5. Пространство адресов мультипроцессора

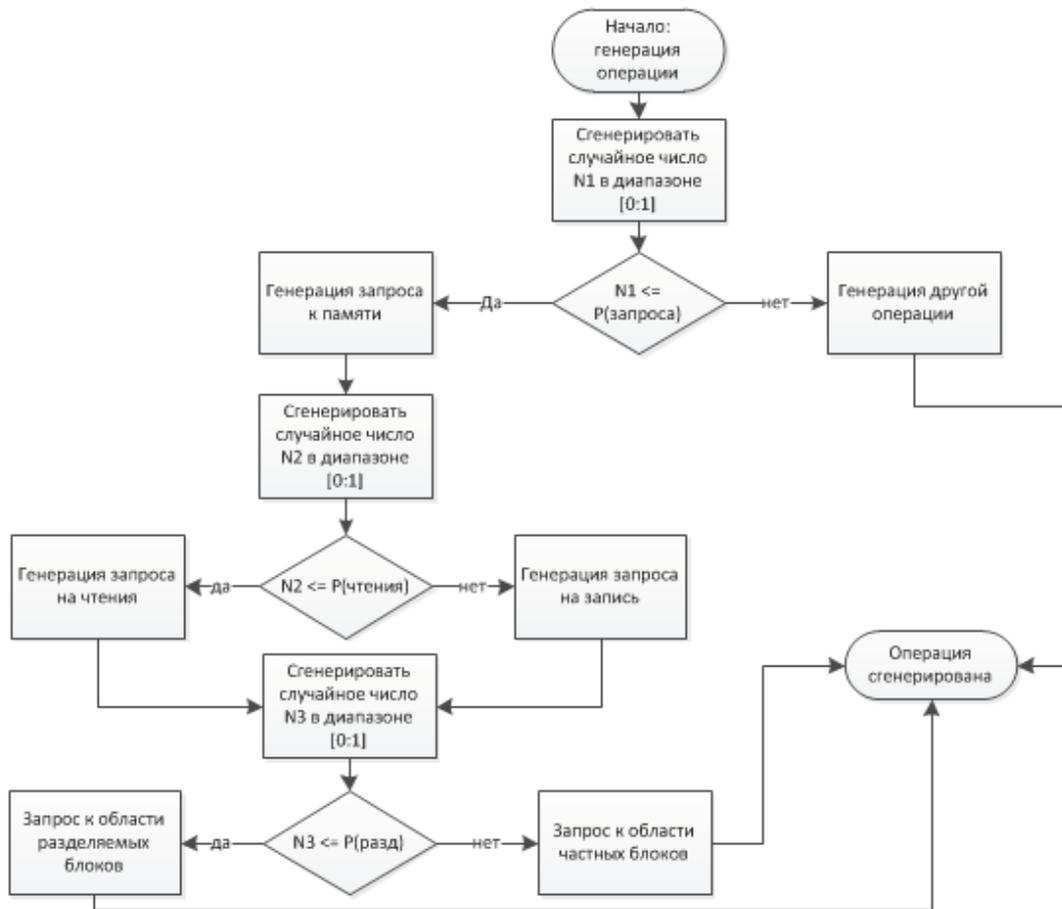


Рисунок 6. Алгоритм генерации операции процессором

Адреса запросов генерируются в соответствии со следующими формулами:

$$\begin{aligned} A_{\text{частный}_m} &= F_{\text{norm}}(A_{\text{частный}_{m-1}}, \delta) \\ A_{\text{разд}} &= F_{\text{line}}(K+1, N-1) \end{aligned} \quad (2)$$

где $F_{\text{norm}}(A_{\text{частный}_{m-1}}, \delta)$ – нормально распределенная случайная величина с математическим ожиданием $A_{\text{частный}_{m-1}}$ (адрес предыдущего запроса к области частных блоков), δ – среднее квадратичное отклонение, $F_{\text{line}}(K+1, N-1)$ – равномерно распределенная в промежутке $[K+1, N-1]$.

3 Оценка временной сложности моделирования

Для того чтобы получить оценку эффективности протоколов когерентности кэш-памяти необходимо, чтобы большинство блоков из области разделяемых адресов стали фактически разделяемыми, то есть были помещены в кэш нескольких процессоров. Для этого в среднем требуется

$$N(\text{операций}) = n_{\text{разд}} / (P_{\text{запроса}} P_{\text{разд}}) \quad (3)$$

операций процессора. Следует отметить, что алгоритмы поддержки когерентности кэша фактически будут задействованы, только если определенная часть этих запросов будут запросами на запись. Аналогично запись во все разделяемые блоки будет произведена в среднем за

$$N(\text{операций}) = n_{\text{разд}} / (P_{\text{запроса}} P_{\text{разд}} (1 - P_{\text{чтения}})) \quad (4)$$

операций процессора. Это число операций, после которого итоговая оценка эффективности уже не будет изменяться. Поэтому целесообразно выбирать продолжительность моделирования в промежутке между значениями (3) и (4).

Выводы

Разработанная модель рабочей нагрузки может применяться для тестирования системы памяти на трассах обращений к памяти самого различного характера. Модель позволяет варьировать долю обращений к памяти среди других операций (позволяет моделировать нагрузку различной интенсивности), долю операций чтения/записи (позволяет анализировать эффективность памяти для «активно» читающих/пишущих программ), долю обращений к области разделяемых адресов и размер этой области (позволяет оценивать эффективность памяти для различных параллельных программ) и разброс адресов при обращении к частным блокам (данный параметр позволяет воздействовать на частоту кэш-попаданий).

Для детального анализа эффективности протоколов когерентности кэш-памяти модель позволяет выявить зависимость между различными группами параметров входной нагрузки и эффективности системы.

Литература

- [1] Орлов С.А., Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.
- [2] Ahmed Louri and Avinash Karanth Kodi, An Optical Interconnection Network and a Modified Snooping Protocol for the Design of Large-Scale Symmetric Multiprocessors, IEEE Transactions on Parallel and Distributed Systems, Vol. 15, NO. 12, December 2004.
- [3] Archibald, J., Baer, J. L., Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model, ACM Transactions on Computer Systems, Vol. 4, No. 4, November 1986.