

УДК 004.94

Л.П.Фельдман, д-р техн. наук, проф.,
Т.В. Михайлова, канд. техн. наук, доц.,
Т.Е. СорокаДонецкий национальный технический университет, г. Донецк, Украина
Feldman@r5.dgtu.donetsk.ua, tanyatv1275@mail.ru, ts.master7@gmail.com

Имитационная модель для оценки эффективности протоколов когерентности кэш-памяти мультипроцессоров с общей памятью

Предложена имитационная модель мультипроцессора с общей памятью для оценки эффективности протоколов когерентности кэш-памяти. Приведена методика оценки быстродействия, модель рабочей нагрузки для системы памяти и сравнительная характеристика эффективности протоколов когерентности.

Ключевые слова: мультипроцессор с общей памятью, протокол когерентности кэш-памяти, оценка производительности вычислительных систем, имитационное моделирование.

Введение

Различие между быстродействием центрального процессора и системы памяти является одной из основных проблем на пути дальнейшего роста производительности вычислительных машин; многопроцессорность вычислительных систем приводит к еще одной проблеме – проблеме одновременного доступа к памяти со стороны нескольких процессоров [1, 2]. 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, то есть к проблеме когерентности кэш-памяти. Данная проблема является актуальной для нескольких классов вычислительных машин: это современные персональные компьютеры с многоядерными процессорами (Intel и AMD), суперкомпьютеры класса UMA, а также более современные суперкомпьютеры класса ccNUMA [1].

Наиболее эффективным (с точки зрения быстродействия) методом разрешения проблемы когерентности кэш-памяти является аппаратная реализация какого-либо алгоритма поддержки когерентности (протокола когерентности). В то же время аппаратная реализация поддержки когерентности является довольно сложной задачей и требует предварительной верификации и анализа на уровне моделей, для чего обычно используется аналитическое и имитационное моделирование [3,4,5].

Данная работа посвящена разработке имитационной модели мультипроцессора с общей памятью, предназначенной для оценки эффективности протоколов когерентности кэш-памяти в

зависимости от структуры кэша и характера нагрузки на систему памяти.

Целью исследования является оценка эффективности протоколов когерентности кэш-памяти мультипроцессора с общей памятью.

Задачей исследования является разработка методики оценки эффективности протоколов когерентности кэш-памяти (на основе наблюдения) с помощью имитационного моделирования, а также разработка модели рабочей нагрузки для системы памяти мультипроцессора.

Постановка задачи

В качестве объекта исследования рассмотрим мультипроцессор с общей (разделяемой) памятью на основе общей системной шины. Обобщенная структура моделируемой системы представлена на рис. 1. Кэш-память каждого отдельного процессора в общем случае является многоуровневой.

Зависимость между организацией кэш-памяти мультипроцессора и производительностью всей системы целесообразно представить в виде некоторой функции $G = F(Y_1, Y_2)$, где Y_1 и Y_2 – векторы входных параметров модели, G – вектор выходных параметров (показателей эффективности системы). Параметры моделирования целесообразно разделить на два вектора, поскольку часть из них представляют собой характеристики аппаратуры (и являются постоянными, независимо от задачи, выполняемой системой), а часть – показателями, зависящими от решаемой задачи (то есть какими-либо характеристиками программы, выполняемой мультипроцессором в конкретный момент времени) [6].

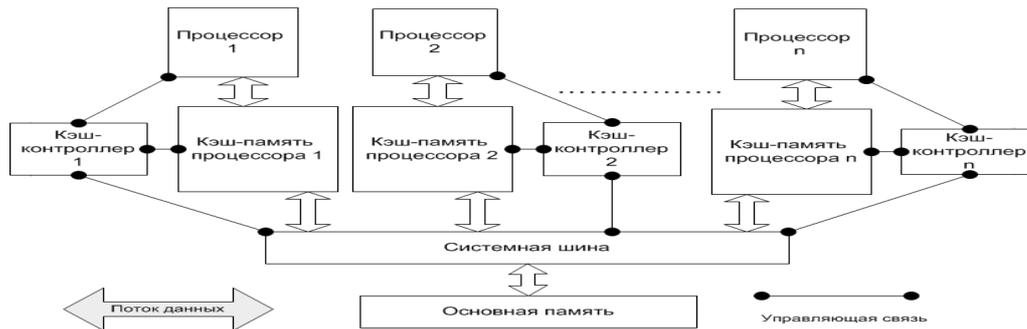


Рисунок 1 – Структура мультипроцессора с общей памятью

В качестве аппаратных характеристик будем рассматривать вектор из 9 параметров:

$$Y_1 = \{N, S_{кэш}, S_{блок}, A, k_{кэш}, k_{шины}, t_{поиска}, t_{кэш}, t_{ОП}\} \quad (1)$$

где N - число процессоров системы, $S_{кэш}$ - размер кэш-памяти, $S_{блок}$ - размер блока кэш-памяти, A - используемый протокол когерентности, $k_{кэш}$ - ассоциативность кэш-памяти, $k_{шины}$ - коэффициент задержки системной шины, $t_{поиска}$ - время поиска указателя в справочнике кэша, $t_{кэш}$ - время чтения/записи в кэш-память, $t_{ОП}$ - время чтения/записи в основную память. С целью упрощения модели длительность выполнения операций чтения и записи предполагаем примерно равной [6].

В качестве параметров моделирования, задающих нагрузку для системы памяти (характеристик решаемой задачи) будем рассматривать вектор из 6 элементов:

$$Y_2 = \{P_{запроса}, P_{чтение}, P_{разд}, n_{разд}, \delta, t_{мод}\}, \quad (2)$$

где $P_{запроса}$ - вероятность генерации запроса к памяти в очередной такт процессорного времени (доля обращений к памяти в общем числе операций процессора), $P_{чтение}$ - вероятность того, что очередной запрос к памяти будет запросом на чтение (доля операций чтения среди всех запросов к памяти со стороны процессора), $P_{разд}$ - вероятность того, что очередной запрос к памяти будет адресован к области разделяемых блоков (уровень разделения), $n_{разд}$ - число разделяемых блоков, δ - среднее отклонение адреса запроса для области частных блоков, $t_{мод}$ - число процессорных тактов, которые должны быть промоделированы (время выполнения программы процессором) [6].

На выходе модели – вектор G показателей эффективности системы памяти:

$$G = \{\rho, h, t_{цикла}, w\}, \quad (3)$$

где ρ - загрузка системы, h - коэффициент кэш-попаданий, $t_{цикла}$ - среднее время цикла системы иерархической памяти, w - мощность системы в расчете на 1000 тактов процессорного времени.

Проектируемая система должна моделировать поведение мультипроцессора с характеристиками Y_1 , в среде с параметрами Y_2 , и как результат моделирования получать вектор характеристик эффективности G [6].

Система считается наиболее эффективной, если:

$$t_{цикла} \rightarrow (t_{кэш} + t_{ук}), \rho \rightarrow 1, h \rightarrow 1. \quad (4)$$

В результате исследования предполагается выявить зависимость между используемым протоколом когерентности кэш-памяти и эффективностью мультипроцессора для различных конфигураций кэш-памяти при рабочей нагрузке на систему памяти различного характера. В данной работе исследуются лишь протоколы когерентности на основе наблюдения: протоколы сквозной, однократной и обратной записи, а также протоколы MSI и MESI. Описания протоколов когерентности можно найти в [1,3].

Требуется проанализировать:

- зависимость среднего времени доступа к памяти, загрузки и мощности системы от используемого протокола когерентности при фиксированных технических характеристиках кэш-памяти, нагрузке на память и коэффициенте кэш-попаданий, для различного числа процессоров;

- чувствительность вышеперечисленных показателей эффективности (для каждого протокола когерентности) к различным видам загрузки на память для различного числа процессоров при фиксированных технических характеристиках кэш-памяти и коэффициенте кэш-попаданий.

Общая характеристика модели мультипроцессора с общей памятью

В имитационной модели присутствуют 4 типа объектов, которые взаимодействуют между собой: процессоры, кэш-контроллеры, кэш-память и системная шина (рис. 2). Каждый процессор имеет собственную кэш-память и кэш-контроллер. Единственная системная шина служит связующим звеном между кэшами процессоров.

Каждый процессор в модели представляет собой генератор операций. Операции процессора, не связанные с обращением к памяти, выполняются в течение одного такта дискретного времени и фактически не моделируются (для таких операций имеет значение только время загрузки процессора, а не характер операции). Время выполнения такой операции принимается за единицу дискретного времени, длительность любой другой (связанной с обращением к памяти) операции занимает время в несколько тактов [6,7].

Генерация операции в модели фактически происходит мгновенно, не занимая ни одного такта процессорного времени. После генерации операции в зависимости от ее типа процессор либо выполняет операцию в течение одного такта (если она не связана с обращением к системе памяти), либо отправляет запрос в память и переходит в состояние простоя, ожидая ответа. После завершения операции или получения ответа из памяти процессор готов к генерации следующей операции (если не истекло время моделирования).

Все показатели эффективности системы (среднее время цикла памяти, загрузка и мощность системы) определяются именно по временам загрузки и простоя процессора, при чем время простоя процессора в модели регулируется кэшем и системной шиной, обрабатывающими его запросы к памяти. Каждый моделируемый процессор имеет счетчики полезных и холостых тактов, которые наращиваются соответственно в случае выполнения процессором операции и в случае простоя (во время ожидания ответа из памяти) при каждом такте моделирования [6,7].

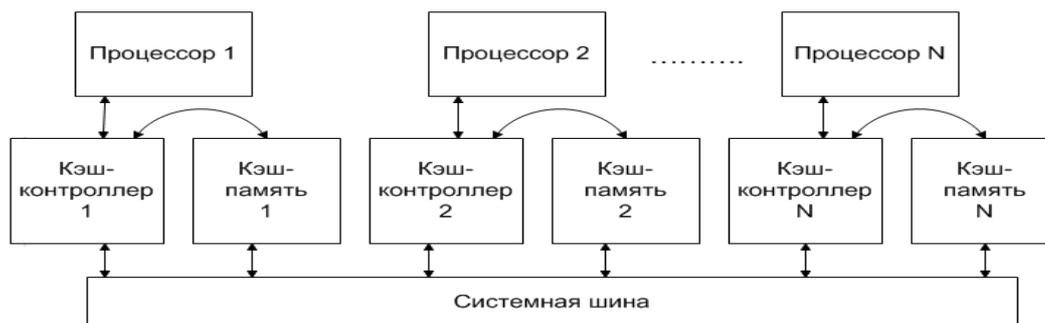


Рисунок 2 – Структура имитационной модели мультипроцессора

Кэш-контроллер обрабатывает запросы процессора к памяти и реализует часть логики моделирования протоколов когерентности. В случае промаха или операции записи в кэш, которая потенциально может привести к несоответствию данных в различных кэшах, контроллер выставляет соответствующий запрос на системную шину. В случае операции, которая не может привести к нарушению когерентности, контроллер прибавляет к счетчику холостых тактов вызывающего процессора время обработки запроса, после чего операция считается выполненной и процессор продолжает работу.

Системная шина последовательно обрабатывает запросы из очереди, реализуя логику моделируемого протокола когерентности.

Программная реализация системы предполагает параллельное моделирование каждого объекта в отдельном потоке команд (рис. 3). Таким образом, для моделирования системы из N процессоров требуется $2N + 2$ потоков. Поток мони-

тора процессоров является служебным и предназначен для опроса состояний процессоров с целью учета времени простоя [6].

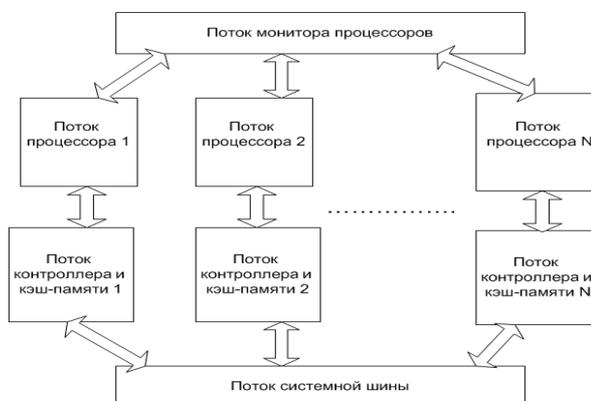


Рисунок 3 – Структура программной реализации имитационной модели мультипроцессора с общей памятью

Модель рабочей нагрузки для системы 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 обращением к памяти (фактически такие операции просто занимают один такт процессорного времени). Запросы к памяти также разделим на несколько групп: по цели запроса – это запросы на чтение или на запись, а по адресу – запросы к области частных, либо разделяемых блоков. Время выполнения операций чтения и записи предполагаем примерно одинаковым, область разделяемых блоков. Частота, адресное распределение и характер запросов к памяти определяются моделью рабочей нагрузки.

Запрос к памяти будем характеризовать двумя параметрами: адресом запроса и типом запроса (чтение или запись). Для того чтобы задействовать при моделировании механизм поддержки когерентности целесообразно разделить поток обращений к памяти на две части: поток обращений к частным блокам (каждый блок данных из этой области может присутствовать в кэш-памяти только одного процессора) и поток обращений к разделяемым блокам (каждый такой блок может присутствовать в кэш-памяти любого числа процессоров системы) [3,7].

Каждый процессор генерирует запросы к одному и тому же линейному пространству адресов (рис. 4). При этом область частных блоков намного больше области разделяемых блоков ($K \gg (N-1)-K$), где N – размер основной памяти в байтах, K – адрес последнего байта области частных блоков. 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. Фактически область частных блоков разделяется на число подобластей (примерно одинакового размера), равное числу процессоров в моделируемой системе, и каждый процессор при обращении к области частных адресов обращается лишь к своей части этой области [7].



Рисунок 4 – Адресное пространство мультипроцессора

Адреса запросов к памяти генерируются по следующей методике [7]:

$$A_{частный_m} = Fnorm(A_{частный_{m-1}}, \delta) \quad (5)$$

$$A_{разд} = Fline(K + 1, N - 1)$$

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

Методика оценки эффективности системы памяти

Поскольку единственными факторами, влияющими на производительность системы в разработанной модели, являются характеристики кэш-памяти и параметры рабочей нагрузки, показатели эффективности всей системы (загрузку и мощность системы) можно рассматривать как показатели эффективности собственно кэш-памяти (в том числе используемого протокола когерентности).

Обобщенный алгоритм взаимодействия процессоров и системы памяти, который реализован в имитационной модели представлен на рис. 5.

От протокола когерентности будет зависеть детализация следующих блоков: изменение состояния блока при записи в него, обработка служебного запроса системной шиной, действия при вытеснении блока. Причем обработка служебного запроса системной шиной предполагает простой всех процессоров, запросы от которых уже помещены в очередь системной шины; обработка шиной запроса аннулирования подразумевает поиск и изменение состояний блоков во всех кэшах, кроме запрашивающего, с последующим возвратом на шину сигнала о завершении обработки [6,7].

Базовым параметром эффективности протокола когерентности (и всей системы памяти) является загрузка мультипроцессора (ρ) [3,7]. Для подсчета загрузки процессора используются два счетчика дискретного времени: счетчик полезных тактов и счетчик холостых тактов. В начале моделирования оба счетчика обнуляются, при

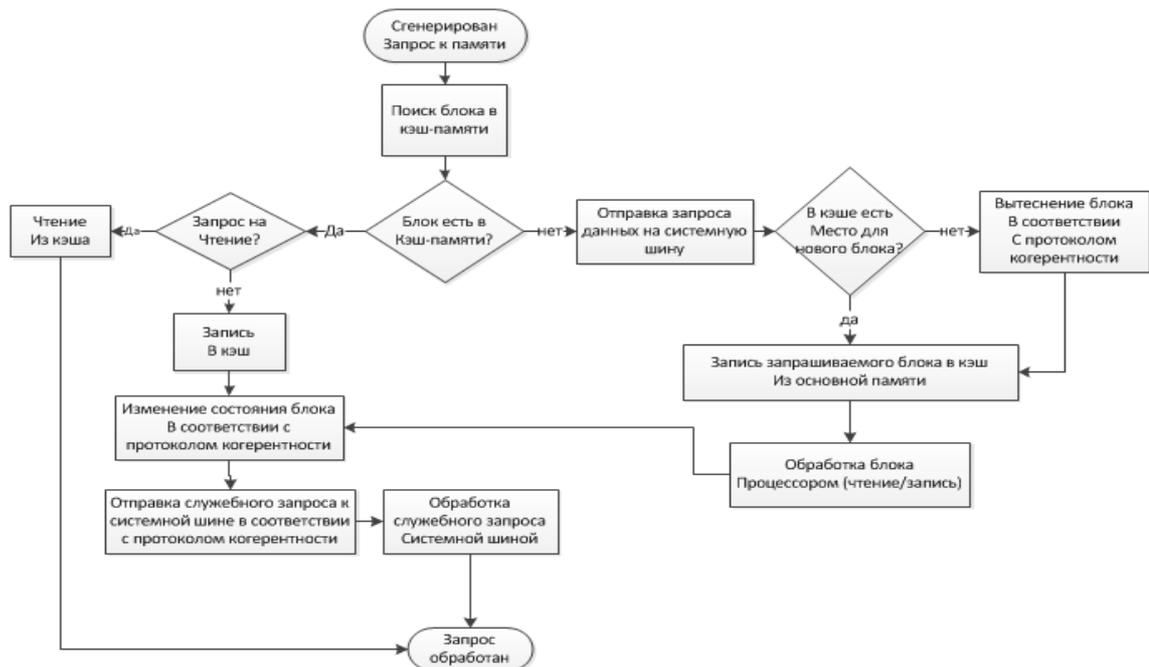


Рисунок 5 – Алгоритм обработки запроса к памяти

генерации процессором любой операции его счетчик полезных тактов увеличивается на единицу, при обработке запроса к памяти счетчик холостых тактов каждый раз увеличивается на соответствующее число тактов с учетом временных параметров модели, моделируемого протокола когерентности и алгоритма взаимодействия основной и кэш-памяти.

Моделирование работы системы памяти мультипроцессора сводится к нескольким действиям: это поиск блоков в справочнике кэш-памяти, изменения тегов и состояний в соответствующих справочниках и наращивании счетчиков полезных и холостых тактов. Выполнение каждой операции системой памяти приводит к соответствующему увеличению счетчика холостых тактов процессора (даже если это операция физически не осуществляется, например, обращение к основной памяти) [6,7].

Эффективность системы будем характеризовать следующими параметрами: среднее время цикла памяти (t_{cp}), загрузка процессора (ρ), коэффициент кэш-попаданий (k), и мощность системы (w). Все параметры кроме мощности системы, рассчитываются по каждому процессору отдельно, а затем по всей системе в среднем. Все параметры на выходе модели, кроме коэффициента кэш-попаданий, определяются исходя из значения счетчиков

полезных ($t_{пол}^{дискр}$) и холостых тактов ($t_{хол}^{дискр}$) и общего числа запросов к памяти ($n_{запр}$).

Загрузку системы определим как отношение времени работы к общему времени работы системы

$$\rho = t_{пол}^{дискр} / (t_{хол}^{дискр} + t_{пол}^{дискр}). \quad (6)$$

Мощность системы определим как суммарное число полезных тактов, по всем процессорам системы в расчете на 1000 тактов дискретного времени:

$$w = \sum_{i=1}^n 1000 \rho_i, \quad (7)$$

где ρ_i – загрузка i -го процессора системы, n – общее число процессоров системы.

Среднее время цикла памяти определим как отношение времени простоя процессора (фактически это время обработки запросов системой памяти) к общему числу запросов к памяти с его стороны

$$c = t_{хол}^{дискр} / n_{запр}. \quad (8)$$

Коэффициент кэш-попаданий (h) определим как отношение числа попаданий в кэш ($n_{попадания}$) к числу запросов к памяти:

$$h = n_{попадания} / n_{запр}, \quad (9)$$

данный параметр зависит лишь от характера рабочей нагрузки и структуры кэш-памяти и не зависит от протокола когерентности. Для правильной оценки протоколов его требуется зафиксировать в определенных пределах.

Сравнительная характеристика протоколов когерентности кэш-памяти по результатам моделирования

Приведем результаты исследования масштабируемости протоколов когерентности. Под масштабируемостью протокола когерентности понимается возможность эффективного применения алгоритма в системах с различным числом процессоров.

Моделирование проводилось для системы с числом процессоров от 2 до 16 с 4-входовой кэш-памятью размером 1 Мб, размер блока – 64 байта, отношение времени цикла основной памяти к циклу кэша – 10 к 1. В ходе моделирования во всех случаях каждый процессор выполнил 30000 операций, коэффициент попаданий кэш-памяти зафиксирован в диапазоне 97-98%. Доля запросов к памяти в потоке операций процессора – 40%, доля запросов на чтение среди всех запросов к памяти – 80%.

Исследование масштабируемости было проведено для 4-х моделей рабочей нагрузки: низкий уровень разделения (5%), средний уровень разделения (10%) и высокий уровень разделения (20%) при объеме разделяемых данных 2 Кб (32 блока по 64 байта).

На рис. 6 представлен график изменения суммарной мощности системы в зависимости от числа процессоров для уровня разделения данных 5% при 32 разделяемых блоках. Наименее эффективным является протокол сквозной записи (write through): максимальная мощность для мультипроцессоров с этим протоколом сопоставима с минимальной мощностью для систем с другими протоколами. Для данной модели рабочей нагрузки шестиядерная система, использующая протокол сквозной записи, будет эквивалентна по мощности двухъядерной с использованием любого другого протокола, что делает данный алгоритм совершенно неэффективным.

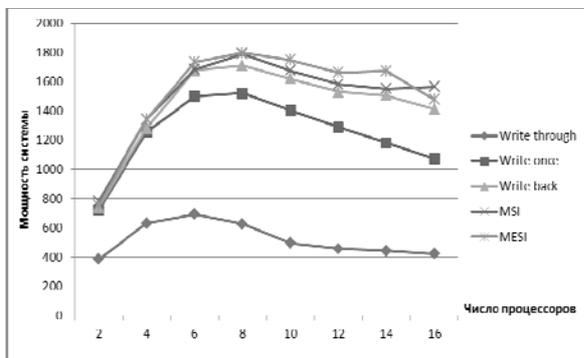


Рисунок 6 - Изменение мощность системы в зависимости от числа процессоров (уровень разделения – 5%)

Наиболее эффективным является протокол MESI, который позволяет достичь максимально

возможной мощности по результатам моделирования, он опережает протокол MSI по результатам 5 экспериментов (6-14 ядер) и лишь для 16-ядерной системы показывает несколько худшие результаты, чем протокол MSI.

На рис. 7 представлен график изменения суммарной мощности системы в зависимости от числа процессоров для уровня разделения данных 10%.

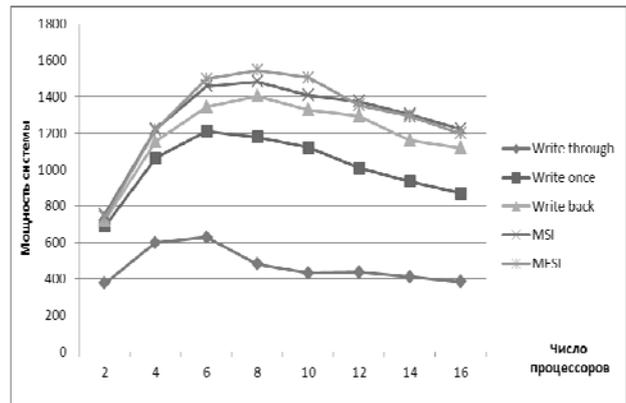


Рисунок 7 - Изменение мощность системы в зависимости от числа процессоров (уровень разделения – 10%)

По результатам эксперимента, использующего модель рабочей нагрузки со средним уровнем разделения, для 6, 8 и 10-ядерных систем наиболее эффективным является использование протокола MESI, который позволяет достичь максимальной мощности, для систем с числом процессоров более 12 протоколы MSI и MESI следует признать одинаково эффективными. Для систем с числом ядер менее 6 также возможно применение протоколов однократной (write once) и обратной (write back) записи, которые позволяют достичь несколько меньшей мощности, чем протоколы MSI и MESI, применение протокола сквозной записи (write back), как и в предыдущем случае является нецелесообразным ввиду слишком низкой достигаемой мощности. Следует отметить, что протокол однократной записи при увеличении уровня разделения с 5% до 10% становится неэффективным уже для систем, с числом ядер больше 6, что свидетельствует о его недостаточной масштабируемости. При увеличении доли обращений к разделяемым блокам среди общего числа обращений к памяти для каждого протокола наблюдается снижение максимальной достигаемой мощности.

Далее рассмотрим результаты эксперимента с повышенной нагрузкой на систему памяти, для чего увеличим уровень разделения до 20% (рис. 8).

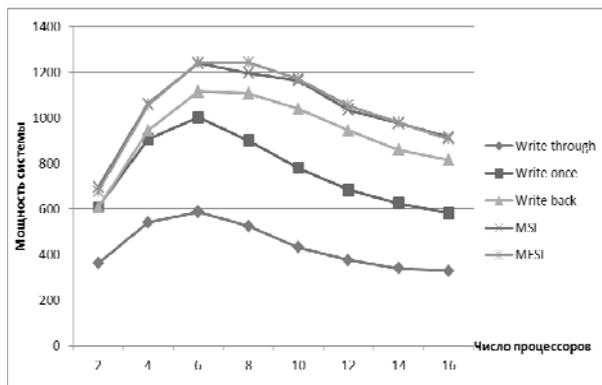


Рисунок 8 - Изменение мощность системы в зависимости от числа процессоров (уровень разделения – 20%).

Для данного эксперимента протоколы MSI и MESI следует считать практически одинаково эффективными (лишь для 8-ядерного процессора протокол MESI показывает несколько лучшие результаты), в то же время все остальные протоколы при столь высоком уровне разделения следует считать неэффективными.

Заключение

Предложенная модель мультипроцессора позволяет оценивать эффективность системы памяти (в том числе протоколов когерентности) для любой комбинации входных параметров.

По результатам исследований определены наиболее эффективные алгоритмы поддержки когерентности (протоколы MSI и MESI). Протокол MESI определен как наиболее масштабируемый и эффективный для всех случаев за исключением слишком высокого уровня разделения ($\geq 20\%$), когда он показывает результаты, почти совпадающие с результатами протокола MSI. Протокол сквозной записи определен как самый неэффективный.

Научная новизна работы заключается в разработке модели рабочей нагрузки для системы памяти мультипроцессора и методики оценки, которые позволяют исследовать зависимость эффективности протоколов когерентности кэш-памяти для различных комбинаций входных параметров.

Практическая значимость заключается в разработке и программной реализации имитационной модели мультипроцессора, позволяющей исследовать влияние структуры и алгоритмов работы кэш-памяти на эффективность вычислительной системы при нагрузке на память различного характера.

Дальнейшее развитие модели предполагает реализацию более современных протоколов когерентности (MOESI, MESIF, MERSI), дальнейшую детализацию модели рабочей нагрузки и распределенную реализацию системы моделирования.

Список использованной литературы

1. Орлов С.А. Организация ЭВМ и систем: учебник для вузов / С.А. Орлов, Б.Я. Цилькер. – 2-е изд. – СПб.: Питер, 2011. – 688 с.
2. Mayberry W. Cache Boosts Multiprocessor Performance / W. Mayberry, G. Efland // IEEE Micro. – 1988, January. – Vol. 8. – Issue 1.
3. Archibald J. Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model / Archibald J., Baer J.L. // ACM Transactions on Computer Systems. – 1986, November. – Vol. 4. – No. 4. – P. 273-298.
4. Pong F. Verification techniques for cache coherence protocols / F. Pong, M. Dubois // ACM Computing Surveys. – 1997, March. – Vol. 29. – No. 1. – P. 82-126.
5. Pong F. Formal Automatic Verification of Cache Coherence in Multiprocessors with Relaxed Memory Models / Pong F., Dubois M. // IEEE Transactions on Parallel and distributed systems. – 2000, September. – Vol. 11. – No. 9. – P. 989-1006.
6. Сорока Т.Е. Оценка эффективности алгоритмов работы кэш-памяти мультипроцессора с помощью имитационного моделирования / Т.Е. Сорока, Л.П. Фельдман, Т.В. Михайлова // Матеріали IV міжнародної науково-технічної конференції «Моделювання та комп'ютерна графіка» – 5-8 жовтня 2011р, м. Донецьк. – Донецьк, 2011.
7. Сорока Т.Е. Модель рабочей загрузки для системы моделирования иерархической памяти мультипроцессора и ее применение для оценки эффективности протоколов когерентности кэш-памяти / Т.Е. Сорока, Л.П. Фельдман // Матеріали VII Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених «Інформатика і комп'ютерні технології» – 22-23 листопада 2011р, м. Донецьк. – Донецьк: ДонНТУ. – 2011. – С. 244-248.

Надійшла до редколегії 29.03.2012

**Л.П.ФЕЛЬДМАН, Т.В. МІХАЙЛОВА,
Т.Є. СОРОКА**
Донецький національний технічний університет

**L.P.FELDMAN, T.V. MIKHAYLOVA,
T.E. SOROKA**
Donetsk National Technical University

ІМІТАЦІЙНА МОДЕЛЬ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ ПРОТОКОЛІВ КОГЕРЕНТНОСТІ КЕШ-ПАМ'ЯТІ МУЛЬТИПРОЦЕСОРІВ ІЗ ЗАГАЛЬНОЮ ПАМ'ЯТТЮ

SIMULATION MODEL OF SHARED MEMORY MULTIPROCESSOR FOR CACHE COHERENCE PROTOCOLS EFFICIENCY EVALUATION

Запропонована імітаційна модель мультипроцесора із загальною пам'яттю для оцінки ефективності протоколів когерентності кеш-пам'яті. Приведена методика оцінки швидкодії, модель робочого навантаження для системи пам'яті і результати моделювання

Proposed simulation model of shared memory multiprocessor to evaluate cache coherence protocols efficiency. Described method of performance evaluation, workload model for the memory system and simulation results

Ключові слова: мультипроцесор із спільною пам'яттю, протокол когерентності кеш-пам'яті, оцінка продуктивності обчислювальних систем, імітаційне моделювання.

Keywords: shared memory multiprocessor, cache coherence protocol, research of computing systems efficiency, simulation modeling.