ПРОБЛЕМЫ ОРГАНИЗАЦИИ МНОГОУРОВНЕВОЙ ПАМЯТИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

М.С. Сущенко, С.П. Сущенко

Томский государственный университет

Введение

Разрыв скорости обработки данных в центральном процессоре и скорости доступа к адресуемым операндам в оперативной памяти современных компьютеров составляет 1-2 порядка. Очевидно, что высокое быстродействие вычислительной системы в значительной мере определяется организацией ее многоуровневой памяти. Главными требованиями, которым должна удовлетворять подсистема памяти, являются достаточно большая емкость, высокое быстродействие и экономическая эффективность с точки зрения технической реализации [1]. Желательно также, чтобы при минимальных физических размерах память обладала как можно большей информационной емкостью. Удовлетворить все эти требования одновременно в одном устройстве невозможно, поэтому обычно комбинируют несколько запоминающих устройств с различными параметрами, добиваясь создания комплексного решения с требуемыми характеристиками. Многоуровневая организация памяти позволяет сгладить разрыв между скоростью обработки данных центральным процессором и скоростью доступа к адресуемым объектам в оперативной памяти вычислителя. Эффективность доступа к иерархической памяти определяется не только быстродействием и емкостью отдельных уровней подсистемы памяти, но и набором архитектурных параметров, важнейшими среди которых являются коэффициент ассоциативности и глубина неблокируемости кэш-памяти. Ассоциативность определяет способ отображения пространства адресов оперативной памяти в кэш-память и, наряду со стратегией вытеснения кэш-строк при конфликте адресов, в значительной мере влияет на степень локализации прикладных задач в памяти верхнего уровня. Степень локализации в свою очередь определяет вероятность «попадания» адресуемого объекта в кэш и время доступа к нему. Как правило, различные уровни иерархической памяти работают асинхронно, поэтому между ними предусмотрены буферные устройства, которые хранят адреса востребованных вычислителем элементов данных и позволяют параллельно выполнять операции доступа к различным уровням. Емкость межуровневого буферного устройства определяет параметр глубины неблокируемости кэша и задает потенциально достижимую степень параллелизма выполнения совокупности транзакций доступа к различным уровням иерархической памяти.

В основе работы кэш-памяти лежит принцип временной и пространственной локальности программ [1,2]. Согласно принципу временной локальности, обращения к памяти носят не случайный характер, а выполняются в соответствии с исполняемой программой. Тогда при считывании данных из памяти с высокой степенью вероятности можно предположить, что в ближайшем будущем программа опять обратится к этим данным, и их целесообразно хранить в течение некоторого времени. В соответствии с принципом пространственной локальности весьма вероятно то, что в ближайшем будущем программа обратится к ячейке, которая следует за той, к которой она обращается в текущий момент времени. Принцип пространственной локальности предполагает считывание в кэш нескольких соседних ячеек памяти (блока информации). Каждый блок хранится в строке буфера, а набор таких строк составляет кэш-память. Таким образом, в пределах небольших временных интервалов обращения к основной памяти за командами и данными сосредоточиваются в ограниченной области адресного пространства.

Проблемы организации подсистемы памяти многопроцессорных систем

При построении вычислительных систем с массовым параллелизмом и вычислительных кластеров в качестве отдельных узлов широко применяются многопроцессорные вычислители с общей памятью (SMP архитектурой). Одной из основных проблем масштабируемых многопроцессорных вычислительных систем с SMP архитектурой является построение распределенной разделяемой памяти [4−6], имеющей единое адресное пространство и доступ к объектам через операции чтения и записи. Важнейшая задача при этом состоит в том, как извещать процессоры об изменениях, вызванных выполнением команд записи. Необходимость увеличения быстродействия вычислителей приводит к применению многоуровневой памяти и механизмов отложенной модификации удаленных копий одного и того же экземпляра данных, находящегося в кэшах различных процессоров и оперативной памяти. При реализации изменений удаленных копий применяется коммуникационный протокол, обеспечивающий доставку изменений, и протокол согласованности или когерентности состояния памяти, предотвращающий использование неизмененных копий модифицированного в одном из процессоров элемента данных. Для сохранения когерентности данных в соответствии с моделью строгой состоятельности памяти операция чтения из разделяемой памяти должна возвращать последнее записанное значение [4,6]. Поэтому при модификации одной из множества копий остальные должны быть либо изменены, либо объявлены несостоятельными (не соответствующими последнему изменению). Выполнение этих действий требует реализации средств быстрого распространения изменений,

однако в любом случае следование модели строгой состоятельности (немедленного изменения значения элемента данных на всех уровнях иерархической памяти) ведет к снижению производительности. Для повышения производительности применяют модели ослабленной состоятельности памяти, допускающие появление несогласованных копий данных в ходе их параллельной обработки с последующим обеспечением когерентности копий, приводящим к тому же результату исполнения приложения, что и при строгой состоятельности.

Современные микропроцессоры, применяемые для построения многопроцессорных вычислительных систем, имеют встроенные аппаратные средства обеспечения когерентности данных в иерархической памяти [5]. Когерентность кэшей процессоров вычислителя обеспечивается с помощью межузловых (межмодульных) пересылок. Размер разделяемой процессорами единицы данных обычно составляет кэш-строку. Методы реализации когерентности определяются видом памяти вычислительной системы (сосредоточенная или распределенная) и типом коммуникационной среды, объединяющей память и процессоры. Аппаратная реализация когерентности универсальна и не требует наличия в программах дополнительного кода для организации разделяемой памяти. При этом обеспечивается минимальная задержка согласования копий данных и высокая пропускная способность всего механизма когерентности.

Для вычислительных систем с сосредоточенной памятью, реализуемых, как правило, в симметричных многопроцессорных архитектурах, наиболее широкое распространение получил алгоритм поддержки когерентности кэшей MESI (Modified, Exclusive, Shared, Invalid) [7]. Этот алгоритм обепечивает когерентность кэш-памяти с обратной записью и минимизирует пересылки данных между основной памятью и кэшем [3]. В общем случае в SMP-системах возможны как модификация данных, так и объявление модифицируемых данных несостоятельными.

Алгоритм MESI предполагает, что вычислитель имеет общую разделяемую память, в каждом процессоре содержится локальная кэш-память, все процессоры объединены между собой и с основной памятью посредством шины. Кроме того, к шине подключены внешние устройства. Ключевым в данной конструкции является то, что все действия, выполняемые на шине по инициативе процессоров и внешних устройств с копиями строк в кэшах процессоров и основной памяти, доступны для отслеживания всем процессорам. При этом в каждый момент времени по шине информация передается одним источником, а воспринимают ее все подключенные к шине абоненты. Если в качестве коммуникационной среды вместо шины используется другой тип соединения, то для работоспособности алгоритма MESI необходимо обеспечение указанного «прослушивания» транзакций всеми процессорами вычислительной системы.

Вычислительные системы с памятью, физически распределенной по вычислительным узлам (модулям), но имеющей единое пространство адресов (единую карту памяти), применяют различные алгоритмы реализации когерентности. Простейший из них заключается в том, что при каждом промахе в кэш в любом процессоре инициируется запрос требуемой строки из резидентного блока памяти (того блока памяти, в котором эта строка размещена). Запрос передается через коммутатор или другую коммуникационную систему в узел с резидентным для строки блоком памяти, откуда необходимая строка пересылается в узел, в котором произошел промах. В соответствии с такой процедурой обеспечивается, в том числе, начальное заполнение кэшей. Каждый узел при этом для каждой резидентной строки ведет список узлов, в кэшах которых размещается данная разделяемая узлами вычислителя строка или организуется распределенный по узлам список этих строк. Более совершенным является алгоритм DASH [3]. Явная реализация когерентности предусматривает применение программистом специальных команд для работы с локальной памятью и управления контроллерами каналов коммуникационной среды, связывающей узлы вычислительной системы с массовым параллелизмом. Основная задача разработчика приложения при этом – эффективное программирование параллельных процессов, совмещающих вычисления и передачу данных между узлами и минимизирующих объем передаваемой информации.

Применение явной когерентности в создаваемых вычислительных системах обусловлено либо слишком большим временем, либо недопустимо высокими аппаратными затратами на реализацию неявного механизма когерентности. Причина этого состоит в высокой сложности применения внутреннего механизма когерентности иерархической памяти узла по сравнению с использованием предусмотренного для работы с внешними устройствами механизма когерентности. Как правило, элементами разделяемой памяти при программной реализации когерентности являются страницы, что обусловлено использованием аппаратных средств организации виртуальной памяти для обнаружения операций записи в разделяемые страницы. Однако относительно большой размер разделяемых страниц создает проблему ложного разделения, при котором конфликтными признаются модификации разными процессорами различных ячеек памяти одной страницы, не конфликтующие между собой [3,6].

Анализ моделей многоуровневой памяти

Основной целью исследования подсистемы памяти является обнаружение функциональных и численных зависимостей между параметрами памяти различных уровней и операционными характеристиками вычислительной системы. При этом необходимо выделить самые существенные факторы, определяющие данные зависимости, так как преувеличенно точ-

ное детальное описание процессов функционирования вычислителя часто мало способствует достижению цели (получению общих закономерностей), а чрезмерно необоснованное упрощение слишком идеализирует реальные процессы. Среди важнейших факторов следует выделить количество уровней иерархии, емкость, ассоциативность и глубину неблокируемости кэш-памяти различных уровней, размер блока и стратегию вытеснения кэш-строк при конфликте адресов, распределение прикладных задач в оперативной памяти, модель состоятельности, реализованную в многопроцессорной вычислительной системе для обеспечения когерентности многоуровневой памяти [2,8−11]. Основными показателями эффективности подсистемы памяти являются вероятность попадания в кэш, среднее время доступа к адресуемым объектам и пропускная способность. Для понимания узких мест и механизма влияния основных факторов на быстродействие вычислительной системы наиболее эффективным инструментом является аналитическое моделирование вычислительных процессов, позволяющее наглядно увидеть достоинства и недостатки различных архитектур [8]. Аналитические модели обычно представляют собой набор уравнений, связывающих показатели производительности с параметрами системы. Однако формализация вычислительных процессов (особенно допускающая аналитическое решение) часто предусматривает их идеализацию, после чего возникают вопрос адекватности аналитической модели реальной системе и проблема интерпретации полученных результатов. Поэтому наряду с аналитическими методами широко применяются имитационное моделирование и методы тестирования вычислительных систем [16], позволяющие исследовать трудноформализуемые при аналитическом моделировании элементы вычислительного процесса. Кроме того, методы тестирования часто используются для выявления уровня локализации тех или иных приложений в вычислительных системах с заданной архитектурой подсистемы памяти.

Известные исследования быстродействия подсистемы памяти основаны на модели двухуровневой памяти (кэш – оперативная память) [6,8−12,16]. В рамках такой модели при заданном типе распределения прикладных задач в оперативной памяти вычислительной системы обычно анализируются вероятность попадания в кэш и среднее время доступа к иерархической памяти, а также скорость обновления содержимого кэша при переключениях вычислителя на выполнение нового приложения (скорость движения к стационарному состоянию). Анализ факторов, определяющих долю обращений за блоками информации, удовлетворяемых на уровне кэша, выполнен в [8,17−21]. На основе предлагаемых математических моделей, как правило, находятся аналитические и численные оценки операционных показателей для одно- и многопроцессорных вычислительных систем. По результатам исследований отмечается, что основными причинами снижения вероятности попадания в кэш многопроцессорных

вычислителей являются переключения задач на процессорах и снижение полезного объема кэша из-за хранения несостоятельных копий блоков данных в кэшах различных процессоров. В работах [8,9,12,22,23] получены асимптотические соотношения и численные зависимости операционных показателей эффективности подсистемы памяти от объема кэш-памяти, коэффициента ассоциативности, размера блока данных. Из проведенного анализа следует, что рост ассоциативности позволяет снизить отрицательный эффект от явления конфликта адресов одинаково востребованных вычислителем блоков памяти, отображаемых на одну группу кэша. Характерная зависимость вероятности попадания в кэш от коэффициента ассоциативности определяется монотонно возрастающей функцией, а от объема кэш-памяти имеет вид кривой с насыщением. Однако общий аналитический вид операционных характеристик от данных параметров не получен.

Поскольку в большинстве современных вычислительных систем используются архитектуры памяти с тремя и более уровнями иерархии, то, очевидно, следует необходимость разработки моделей многоуровневой подсистемы памяти (с числом уровней больше двух) и уточнения моделей двухуровневой памяти. Кроме того, важной является задача построения подсистемы памяти с оптимальным в смысле некоторого критерия числом уровней иерархии при заданных параметрах каждого уровня (длительностях обращения к памяти, вероятностях промаха, глубинах неблокируемости, ассоциативности, емкости). Оценивание эффективности многоуровневых систем памяти в однопроцессорных вычислителях проводилось в работах [24−26]. В них решается задача минимизации среднего времени обращения к иерархической памяти при известной связи времени доступа к памяти каждого уровня с ее удельной стоимостью и частоты обращения к памяти каждого уровня с ее объемом, а также заданной конфигурации подсистемы памяти (число уровней, вероятности обращения к памяти каждого уровня) и ограничениях на ее общую стоимость. Однако проблема обоснованного выбора рационального числа уровней иерархии не рассматривалась.

Влияние стратегий вытеснения кэш-строк при конфликте адресов обсуждается в [9,11−14,23,27,28]. Отмечается, что наилучшей среди реализуемых стратегией является стратегия LRU (Least Recently Used), которая в то же время характеризуется высокой сложностью и трудоемкостью. На примерах показано, что вероятность попадания в кэш с ростом ассоциативности увеличивается, а от длины блока – имеет унимодальный характер. Наличие оптимального размера блока обусловлено двумя факторами. С одной стороны, размер блока следует увеличивать, чтобы усилить эффект пространственной локализации программного кода и обрабатываемых данных. С другой стороны, рост длины блока приводит к сокращению числа блоков при фиксированной емкости кэша и, как следствие, увеличе-

нию частоты конфликта адресов и, соответственно, интенсивности замещения блоков.

В работах [29−31] отмечается положительное воздействие на операционные характеристики подсистемы памяти увеличения емкости межуровневого интерфейса (параметра глубины неблокируемости). Так, например, в процессоре Pentium 4 отсутствие блокировок кэша обеспечивается наличием 48 и 24 межуровневых буферов для команд чтения и записи соответственно [32]. Однако строгого математического моделирования и обоснования механизма влияния емкости межуровневого буфера на быстродействие иерархической памяти в известных автору исследованиях не проводилось. Из этого следует необходимость построения моделей влияния параметра глубины неблокируемости на вероятностно-временные характеристики подсистемы памяти. Одним из важнейших параметров кэш-памяти, определяющих операционные характеристики подсистемы памяти, является его объем. В целом рост емкости кэша приводит к улучшению вероятностно-временных показателей функционирования иерархической памяти, однако вместе с этим значительно возрастает стоимость вычислительной системы. Тесно связанной с вопросом оптимизации емкости кэш-памяти и общей архитектуры подсистемы памяти является задача определения влияния распределения приложений в оперативной памяти на степень их локализации в кэше и среднее время доступа к адресуемым объектам. Очевидно, что знание этих распределений для различных классов прикладных задач позволит сделать обоснованный выбор архитектуры и параметров подсистемы памяти вычислителя, в том числе емкостей кэш-памяти различных уровней. Однако исследований на эту тему обнаружить не удалось, что обусловливает практическую потребность в анализе взаимосвязи типов решаемых вычислителем задач и соответствующих им рациональных значений параметров подсистемы памяти.

Относительно широко в публикациях представлены исследования эффективности многоуровневой памяти многопроцессорных вычислительных систем [8,10,12,22,27,28]. Основной решаемой здесь задачей является анализ эффективности совместного использования множеством процессоров SMP-вычислителя разделяемой оперативной памяти. Авторы [33,34] предложили математическую модель распределенной иерархической памяти многопроцессорной вычислительной системы, на основе которой получили численные зависимости для среднего времени ожидания запросов на доступ к удаленной памяти на фоне обслуживания более приоритетных запросов к локальной памяти от количества процессоров в системе. Однако при анализе не учитывается параметр глубины неблокируемости кэша. Не рассмотренными также остаются проблемы взаимного влияния на результирующее быстродействие вычислителя существенно неоднородного потока транзакций доступа к иерархической памяти от различных процессоров.

Другой важной проблемой многопроцессорных вычислительных систем является проблема когерентности (согласованности состояния памяти различных уровней), решение которой основано на различных моделях состоятельности памяти [4,6,28]. Работа [4] посвящена классификации моделей состоятельности распределенной памяти и глубокому анализу преимуществ и недостатков каждой из них, а также обсуждению вариантов реализации разделяемой памяти. В [6,28] предложена классификация методов обеспечения когерентности данных в иерархической памяти многопроцессорных вычислительных систем и проведен сравнительный анализ различных методов по критерию эффективного времени обращения процессора к подсистеме памяти. По результатам исследований определены области предпочтения рассмотренных способов реализации когерентности разделяемой различными процессорами информации.

Работы [35−39] посвящены исследованию влияния структурных факторов векторных, конвейерных и параллельных вычислительных систем на их быстродействие. Получены оценки и границы скорости выполнения вычислений. В целом процесс доступа к объектам, хранимым в многоуровневой памяти, хорошо описывается конвейером с числом фаз, равным количеству уровней иерархии, нарастающим временем обработки в каждой последующей фазе и случайным числом последовательных фаз обработки каждой транзакции доступа к подсистеме памяти. Степень загрузки конвейера при этом целиком определяется параметрами глубины неблокируемости и вероятности промаха на каждом уровне. Наибольшую сложность моделирования конвейера с такими свойствами представляет проблема поиска распределений времени пребывания запроса к подсистеме памяти в различных фазах.

Заключение

При анализе проблем организации эффективной подсистемы памяти вычислительных систем выясняется необходимость совершенствования существующих моделей иерархической памяти и разработки на их основе методов расчета архитектурных параметров многоуровневой памяти. Проведенный аналитический обзор результатов, достигнутых в изучении воздействия архитектурных параметров подсистемы памяти и распределения в оперативной памяти выполняемой вычислителем смеси прикладных задач на скорость доступа к хранимым данным, позволил выявить степень проработки вопроса организации быстродействующей иерархической памяти, сформулировать нерешенные задачи и наметить основные направления исследований.

1. Существенным недостатком известных моделей иерархической памяти является отсутствие явных аналитических зависимостей операционных характеристик подсистемы памяти от коэффициента ассоциативности

и стратегии вытеснения срок кэша. В связи с этим необходимы разработка моделей влияния ассоциативности на быстродействие подсистемы памяти и исследование влияния распределения прикладных задач на вероятность попадания адресуемых объектов в кэш.

2. Существующие подходы к анализу подсистемы памяти не учитывают фактор, определяемый конвейерной природой обработки транзакций доступа к иерархической памяти. Степень параллелизма доступа к адресуемым объектам, хранящимся на различных уровнях памяти, определяется параметром глубины неблокируемости кэша. Поэтому представляется актуальным построение моделей влияния данного параметра на вероятностно-временные показатели функционирования вычислителя.

3. Кроме того, в известных исследованиях отсутствуют модели влияния стратегий сохранения целостности данных, модели подсистем памяти с числом уровней больше двух и решение задачи оптимизации числа уровней иерархической памяти, модели влияния числа процессоров на вероятностно-временные характеристики.

Литература

1. Кохонен Т. Ассоциативные запоминающие устройства. – М.: Мир, 1982. – 384 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
3. Корнеев В.В. Параллельные вычислительные системы. – М.: Нолидж, 1999. – 320 с.
4. Корнеев В.В. Архитектуры с распределенной разделяемой памятью // Открытые системы. − 2001. − № 3. − С. 15−23.
5. Корнеев В.В. Будущее высокопроизводительных вычислительных систем // Открытые системы. − 2003. − № 5. − С. 10−17.
6. Adve S.V., Charachorloo K. Shared memory consistency models: A Tutorial // IEEE Comput. Magn. − 1999. − Vol. 29, № 12. − P. 66−76.
7. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – М.: Нолидж, 1998. – 240 с.
8. Богуславский Л.Б., Ляхов А.И. Методы оценки производительности многопроцессорных систем. – М.: Наука, 1992. – 126 с.
9. Лускинд Ю.И. Буферные запоминающие устройства типа кэш // Зарубежная радиоэлектроника. − 1990. − № 4. − С. 29−42.
10. Ляхов А.И. Многопараметрический анализ эффективности использования кэш-памяти на основе асимптотических методов // Автоматика и телемеханика. − 1989. − № 11. − С. 155−165.
11. Танненбаум Э. Архитектура компьютера. – СПб.: Питер, 2002. – 704 с.

12. Феррари Д. Оценка производительности вычислительных систем. – М.: Мир, 1981. – 576 с.
13. McKee S.A., Klenke R.H., Wright K.L. et al. Smarter memory: Improving bandwdth for streamed references // IEEE Comput. Magn. − 1998. − Vol. 31, № 7. − P. 54−63.
14. Smith A. Cache memories // ACM Computing Surveys. − 1982. − Vol. 14, № 3. − P. 473−530.
15. Старченко А.В., Есаулов А.О. Параллельные вычисления на многопроцессорных вычислительных системах. – Томск: Изд-во Том. ун-та, 2002. – 56 с.
16. Французов Д. Оценка производительности суперкомпьютеров // Открытые системы. − 1995. − № 6. − С. 48−51.
17. Mattson R.L. Evaluation of multilevel memories // IEEE Trans. Magn. − 1971. − Vol. MAG-7, № 12. − P. 814−819.
18. Patel J.H. Analysis of multiprocessors with private cache memories // IEEE Trans. Comput. − 1982. − Vol. 31, № 4. − P. 296−304.
19. Patel J.H. Performance of processor-memory interconnections for multiprocessors // IEEE Trans. on Comput. − 1981. − № 11. − P. 771−780.
20. Smith J.E., Sohi G.S. The microarchitekture of superscalar processors // Proc. of IEEE. − 1995. − Vol. 83, № 12. − P. 1609−1622.
21. Smith A. Cache evaluation and the impact of work-load choice // SIGARCH Newslett. − 1985. − Vol. 13, №. 3. − P. 64−73.
22. Богуславский Л.Б., Ляхов А.И., Шевчик К.С. Сравнительный анализ стратегий доступа с обратной связью в многопроцессорных системах // Автоматика и телемеханика. − 1996. − № 5. − С. 160−176.
23. Смирнов Р.В. Анализ функционирования двухуровневой иерархической памяти процессора // Вопросы радиоэлектроники. Сер. ЭВТ. − 1980. − Вып. 5. − С. 3−11.
24. Юрченко А.С. К выбору конфигурации иерархической системы памяти // Автоматика и телемеханика. − 1985. − № 4. − С. 137−139.
25. Rege S.L. Cost, performance and size tradeoffs for different levels in a memory hierarchy // Computer. − 1976. − Vol. 19. − P. 43−51.
26. Welch T.A. Memory hierarchy configuration analysis // IEEE Trans. Comput. − 1978. − Vol. 27, № 5. − P. 408−413.
27. Башарин Г.П., Богуславский Л.Б., Штейнберг В.И. Анализ конфликтов в общей памяти мультипроцессорных систем // Автоматика и вычислительная техника. − 1980. − № 6. − С. 27−32.
28. Брехов О.М., Слуцкин А.И. Исследование способов обеспечения идентичности информации в буферной памяти многопроцессорных вычислительных систем // Автоматика и телемеханика. − 1988. − № 8. − С. 156−169.

29. Кузьминский М. Архитектура серверов HP Superdome // Открытые системы. − 1997. − № 3. − С. 9−13.
30. Кузьминский М. Краткий обзор IA-64 // Открытые системы. − 1999. − № 9−10. − С. 8−14.
31. Кузьминский М. Микроархитектура Itanium // Открытые Системы. − 2001. − № 9. − С. 8−14.
32. Carmean D. Inside the Pentium 4 Processor Microarchitecture // Intel Development Forum. − 2000.
33. Корячко В.П., Сускин В.В. Анализ эффективности распределения данных в памяти специализированных микропроцессорных вычислительных систем // Автоматика и вычислительная техника. − 1984. − № 2. − С. 66−68.
34. Корячко В.П., Сускин В.В. Анализ функционирования микропроцессорной вычислительной системы с буферной памятью // Автоматика и вычислительная техника. − 1985. − № 3. − С. 75−77.
35. Головкин Б.А. Анализ факторов, влияющих на скорость вычисления в параллельных и конвейерных системах // Автоматика и телемеханика. − 1988. − № 5. − С. 152−164.
36. Головкин Б.А. Вычислительные системы с большим числом процессоров. – М.: Радио и связь, 1995. – 320 с.
37. Головкин Б.А. Параллельные вычислительные системы. – М.: Наука. Гл. ред. физ.-мат. лит-ры, 1980. – 519 с.
38. Головкин Б.А. Соотношения между показателями производительности супер-ЭВМ и метод ее оценивания // Автоматика и телемеханика. − 1989. − № 12. − С. 141−152.
39. Коган Я.А., Ляхов А.И., Нерсесян С.Г. Асимптотический анализ эффективности использования кэш-памяти в многопроцессорных системах // Автоматика и телемеханика. − 1986. − № 11. − С. 142−151.