Моделирование и анализ информационных систем. Т. 24, № 4 (2017), с. 434–444
Modeling and Analysis of Information Systems. Vol. 24, No 4 (2017), pp. 434–444

DOI: 10.18255/1818-1015-2017-4-434-444

УДК 004.051

О скоростях передачи данных на шинах между кеш-памятью второго и третьего уровней и между процессором и оперативной памятью в современных компьютерах

Комар М. С.

получена 18 июля 2017

Аннотация. В данной работе рассматривается архитектура используемых в настоящее время центральных процессоров и ограничения их производительности в современном виде. Так как чаще всего для повышения производительности центральных процессоров предлагаются решения, связанные с изменением существующей архитектуры, необходимо иметь представление о скоростях передачи данных внутри процессора и на шинах, подходящих к нему. Это позволит оценить применимость предлагаемых решений и даст возможность их оптимизировать. В этой статье решается задача измерения реальных скоростей передачи данных на интерфейсе между кеш-памятью второго и третьего уровней внутри процессора и на интерфейсе между процессором и оперативной памятью, а также изучения зависимости численных результатов от количества активных ядер, тактовой частоты процессора и типа проводимого теста. В статье приводится методология проведения измерений с помощью программного инструмента Intel Performance Counter Monitor от компании Intel, а также приводятся формулы для получения итогового результата из полученных в ходе измерений значений. Приведено подробное описание тестов, имитирующих реальную нагрузку на центральный процессор, и синтетических тестов. Зависимости скоростей передачи данных от количества активных ядер и от тактовой частоты процессора представлены в виде графиков. Зависимости скоростей передачи данных от типа теста представлены в виде столбиковых диаграмм для трех различных значений тактовой частоты процессора.

Ключевые слова: многоядерные процессоры, оценка скоростей передачи данных, системы на кристалле, сети на кристалле, беспроводные системы на кристалле

Для цитирования: Комар М. С., "О скоростях передачи данных на шинах между кеш-памятью второго и третьего уровней и между процессором и оперативной памятью в современных компьютерах", *Моделирование и анализ информационных систем*, **24**:4 (2017), 434–444.

Об авторах:
Комар Мария Сергеевна, orcid.org/0000-0002-0995-7744, аспирант,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия
магистрант, Технологический Университет г. Тампере,
PO Box 527, FI-33101, Korkeakoulunkatu 10, Тампере, Финляндия
e-mail: maria.s.komar@gmail.com

1. Введение

В настоящее время все большее распространение получают персональные компьютеры, серверы, мобильные телефоны, ноутбуки, нетбуки, электронные книги, игровые консоли, “умные” часы и другие устройства, относящиеся к классу носимой электроники, и многое другое. Практически в каждом доме найдется не один представитель перечисленных выше типов устройств. Объединяет их друг с другом и другими представителями техники наличие центрального процессора. В настоящее время центральные процессоры (ЦП) используются практически повсеместно и являются неотъемлемой частью практически любого электронного устройства, соответственно прогресс в развитии этих устройств неразрывно связан с прогрессом существующих ЦП и созданием новых, улучшенных процессоров.

К сожалению, за последнее десятилетие прогресс в развитии производительности ЦП был не таким существенным, как хотелось бы пользователям и представителям индустрии. Если в 80-е и 90-е года прошлого века процессоры эволюционировали столь стремительно, что раз в 24 месяца количество транзисторов на чипе удваивалось (это эмпирическое наблюдение получило название закона Мура [1]), а раз в 18 месяцев в два раза увеличивалась производительность, то сейчас каждое новое поколение дает улучшение едва ли на 15 процентов [2]. Причина этого кроется в том, за счет чего происходил прогресс в прошлом веке и за счет чего он происходит сейчас. В прошлом веке и самом начале этого электроника бурно развивалась, соответственно была возможность относительно легко и быстро увеличивать количество транзисторов на чипе и повышать тактовую частоту процессора. Однако в настоящее время технологии дошли до той границы, когда ни увеличивать тактовую частоту, ни увеличивать количество транзисторов на чипе не представляется возможным [3]. В связи с этим сейчас ведется поиск новых способов увеличения производительности процессоров. Одно из перспективных направлений, над которым сейчас работают ученые, – это изменение существующей архитектуры центрального процессора на более эффективную. Основным кандидатом для замены являются конструкции из класса беспроводных систем на кристалле (Wireless Networks-on-Chip, WNoC), в которой предлагается часть проводных соединений заменить на беспроводные каналы связи [4], [5]. Однако для того, чтобы оценить, насколько емкими должны быть эти каналы, необходимо иметь представление о том, какова скорость передачи данных в существующих процессорах. Особенно интересна для исследователей загрузка шин данных между процессором и оперативной памятью и между кеш-памятью второго и третьего уровней, так как большинство решений предполагают построение сетей, в которых именно эти шины будут заменены на беспроводные каналы связи [5].

Исходя из всего вышеперечисленного, встает задача измерения реальных скоростей передачи данных на интерфейсе между кеш-памятью второго и третьего уровней внутри процессора и на интерфейсе между процессором и оперативной памятью, а также изучение зависимости численных результатов от количества активных ядер, тактовой частоты процессора и типа проводимого теста, решение которой описано в данной статье.

Статья имеет следующую структуру: в разделе 2 описана типичная архитектура современного центрального процессора и упрощенная схема работы кеш-памяти.

Раздел 3 посвящен описанию стенда, использовавшегося для проведения измерений, тестам, проведенным в ходе исследования, методологии их проведения и расчета скоростей передачи данных на интересующих интерфейсах по имеющимся результатам. В разделе 4 приведены численные результаты, полученные в ходе исследования и их анализ. Окончательные выводы сделаны в разделе 5.

2. Архитектура центрального процессора и модель передачи данных внутри него

В современных компьютерах чаще всего используются процессоры, имеющие кеш-память нескольких уровней. Это особый тип памяти, позволяющий получить очень быстрый доступ к хранящейся в ней информации, но имеющий ограниченный размер. Чаще всего используются процессоры с кеш-памятью двух уровней (особенно распространено в процессорах от AMD [6]) и трех уровней (в процессорах от Intel [7] с архитектурой x86 [8], принципиальная схема которого изображена на рис. 1).

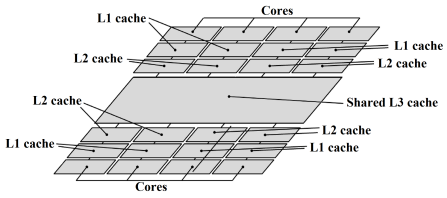

Рис. 1. Архитектура центрального процессора с тремя уровнями кеш-памяти и восемью ядрами

Fig. 1. CPU architecture with 3 cache layers and 8 cores

Первый уровень кеш-памяти является самым быстрым, дорогим и находится ближе всего к ядру процессора. Также он обеспечивает наименьшее время доступа к хранящимся в нем данным. Второй уровень кеш-памяти более медленный, более дешевый, однако тоже достаточно дорог в производстве. Кеш-память первого и второго уровней обычно являются отдельными для каждого ядра в случае многоядерных процессоров. Кеш-память первого уровня чаще всего разделяется на кеш инструкций и кеш данных. Кеш-память третьего уровня чаще всего является общей для всех ядер, имеет самый большой размер и самую большую из трех уровней задержку на доступ к хранимым данным. Сложность работы с кеш-памятью третьего уровня состоит в том, что к ней должны иметь доступ все ядра, например для организации совместных вычислений и своевременного обновления данных, которые находятся в общем доступе нескольких ядер. В такой системе данные, обработанные одним из ядер, могут быть оперативно получены другим. Это приводит к уменьшению времени простоя ядра, то есть повышает производительность системы. Упрощенная модель работы кеш-памяти, позволяющая, однако, реалистично оценить объемы данных, проходящие по шинам внутри процессора и между процессором и оперативной памятью, но лишенная ненужных для этой задачи подробностей, представлена ниже. Она включает в себя основные запросы, связанные с чтением и записью данных. Предположим, что одно из ядер ЦП пытается обратиться на чтение или на запись к определенным данным.

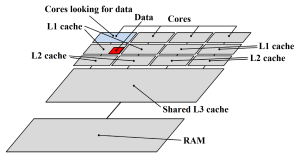

Рис. 2. Данные находятся в кеш-памяти первого уровня того ядра, которое делает запрос

Fig. 2. Piece of data is stored in L1 cache of the core that looks for it

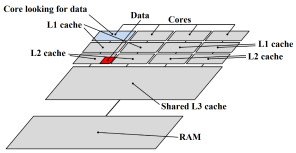

Рис. 3. Данные находятся в кеш-памяти второго уровня того ядра, которое делает запрос

Fig. 3. Piece of data is stored in L2 cache of the core that looks for it

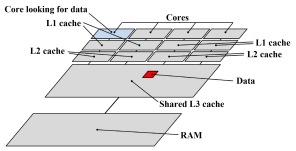

Рис. 4. Данные находятся в общей кеш-памяти третьего уровня

Fig. 4. Piece of data is stored in shared L3 cache

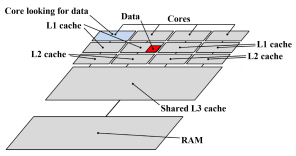

Рис. 5. Актуальная копия данных хранится в кеш-памяти первого уровня другого ядра

Fig. 5. Up-to-date copy of data is stored in the L1 cache of another core

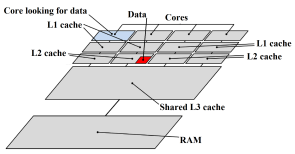

Рис. 6. Актуальная копия данных хранится в кеш-памяти второго уровня другого ядра

Fig. 6. Up-to-date copy of data is stored in the L2 cache of another core

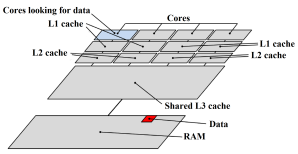

Рис. 7. Данные находятся в оперативной памяти

Fig. 7. Piece of data is stored in RAM

Тогда возможны следующие случаи, проиллюстрированные на рисунках 2 — 7:

1. Данные находятся в кеш-памяти первого уровня того ядра, которое делает запрос.
2. Данные находятся в кеш-памяти второго уровня того ядра, которое делает запрос.
3. Данные находятся в общей кеш-памяти третьего уровня.
4. Данные находятся в оперативной памяти.
5. Актуальная копия данных хранится в кеш-памяти первого или второго уровня другого ядра.

При попытке чтения данных их поиск проводится следующим образом: сначала информация ищется в кеш-памяти первого уровня, если совпадений не обнаружено, происходит так называемый «cache miss» (промах), и поиск продолжается в кеш-памяти второго уровня. Если данные найдены, то происходит «cache hit» (попадание), и результат возвращается ядру. В противном случае проводится поиск в кеш-памяти третьего уровня. Если данные найдены и копия, хранящаяся там,

актуальная, то результат возвращается ядру. Если данные найдены, но копия, хранящаяся в кеш-памяти третьего уровня, не актуальна, то контроллер кеша запрашивает данные у того ядра, которое хранит актуальную копию, и, получив, отдает их запрашивавшему ядру. Если данные не найдены в кеш-памяти третьего уровня, происходит обращение к оперативной памяти, и данные оттуда передаются ядру. Для запроса на запись ситуация практически аналогичная, за исключением случая, когда актуальная копия данных хранится в кеш-памяти другого ядра. В этом случае данные записываются в кеш-память третьего уровня, а данные в кеш-памяти другого ядра признаются неактуальными и обновляются при необходимости.

Также необходимо примерно оценить размеры пакетов, проходящих по интересующим нас шинам. С точки зрения самого вычислительного ядра его работа с кеш-памятью выглядит следующим образом: оно отсылает адрес, по которому хранятся интересующие его данные, и через какое-то время ему приходит блок информации. Размер адреса в современных ЦП различен, но его верхняя граница в настоящее время - 64 бита, то есть 8 байт [8]. Блок данных, возвращающийся к центральному процессору, всегда составляет 64 байта в случае памяти без корректировки ошибок [8]. Таким образом на один запрос всего по шинам данных проходит не больше 72 байт информации. Если данные не нашлись в кеш-памяти первого уровня, адрес передается в кеш-память второго уровня. Если данные действительно находятся там, то пакет с необходимой информацией уходит обратно к ядру по шине между кеш-памятью первого и второго уровней. Если данные не были найдены в кеш-памяти второго уровня, то запрос с адресом отправляется в кеш-память третьего уровня, а при отсутствии в ней таких данных – в оперативную память. Если информация нашлась в кеш-памяти третьего уровня, то в кеш-память второго уровня идет пакет с найденной информацией. Аналогично этому, если данные нашлись только в оперативной памяти, то найденная информация идет в кеш-память третьего уровня.

Отдельно следует рассмотреть случай, изображенный на рисунках 5 и 6, когда актуальная информация хранится в кеш-памяти первого или второго уровня другого ядра. В этом случае контроллером кеша генерируется сервисное сообщение-запрос на получение актуальной копии данных. Размер этого сообщения так же можно аппроксимировать 8 байтами. В ответ на это в кеш-память третьего уровня отправляется пакет данных стандартного размера – 64 байта. Таким образом, в данном случае по двум разным шинам между кеш-памятью второго и третьего уровней проходит по 72 байта. Для запросов на запись схема следующая: ядро отдает кеш-памяти первого уровня данные, которые необходимо записать, и адрес, по которому их необходимо записать. Кеш-память первого уровня записывает данные и по мере необходимости (или в других случаях, если это определено протоколом) передает данные в кеш-память верхних уровней и через них в оперативную память.

3. Тестовый стенд и методология измерений

Основой тестового стенда был персональный компьютер со следующими характеристиками:

- Процессор Intel Core i7-5960X с архитектурой Haswell семейства Haswell-E с тактовой частотой (без включения режима Turbo Boost) от 1200 МГц до 3000 МГц.
- Материнская плата Asus X99-A LGA2011-3
- Оперативная память 4 модуля по 4 ГБ Kingston HyperX Predator DDR4
- Видеокарта NVIDIA 750GTX
- Накопитель Samsung 850 256ГБ SSD
- Операционная система Linux Kubuntu 14.10 [9]

Оценка загрузки шин данных была получена с помощью натурных измерений. Для этого использовался Intel Performance Counter Monitor (Intel PCM, [10]) – специальный инструмент для оценки производительности процессоров, выпущенный корпорацией Intel. Кроме вывода на экран показаний со счетчиков, он позволяет сохранять их в csv файл, что является большим преимуществом для решения поставленных задач. Стоит отметить, что ни один из известных нам инструментов для проведения натурных измерений не дает количественных значений скорости передачи данных в явном виде, а лишь предоставляет информацию о данных с некоторого набора счетчиков, таких, как количество cache hit (попаданий) и cache miss (промахов) в единицу времени. Однако по этим значениям можно оценить интересующие нас характеристики.

Для тестирования были выбраны как искусственные тесты, так и реальные сценарии использования компьютера. К искусственным тестам можно отнести две программы, написанные на языке C. Они получили условные названия ”хороший стиль программирования” и ”плохой стиль программирования”. В оперативной памяти создавался массив большого объема, который гарантированно не мог поместиться в кеш-памяти третьего уровня. В массиве находились элементы размером 1 байт каждый. Далее в бесконечном цикле производилось чтение данных из этого массива. В случае теста, имитирующего ”хороший стиль программирования” считалось, что при создании программы разработчики позаботились об оптимизации своего продукта и постарались минимизировать количество обращений ядра к оперативной памяти в ходе работы программы, при этом не допуская ее простоя. В ходе этого теста из массива, описанного выше, считывался каждый байт подряд. В таком случае в кеш-память уходили блоки по 64 элемента (байта), такие же блоки перемещались между уровнями кеш-памяти. Тогда ядро запрашивало новую порцию данных только через 63 операции, так как все данные для последующих 63 операций уже хранились в кеш-памяти после первого обращения за данными. Этот же эффект кеш-памяти, но для другой цели использовался в тесте, имитирующем плохой стиль программирования. Целью этого теста было максимально загрузить шины данных, для этого на каждом шаге брался каждый 64-й элемент массива, соответственно информация, переданная в каждом из блоков, не могла быть использована для следующей операции и на каждом шаге процессору приходилось запрашивать новую порцию данных из массива.

Кроме искусственных тестов, были проведены тесты, представляющие собой сценарии реального использования компьютера. Первым из них была оценка того, насколько сильно загружены шины, когда нет никакой нагрузки на процессор, кроме работы операционной системы. Вторым тестом было шифрование файла с помощью алгоритма AES-256 [11]. Третьим тестом была компьютерная игра. К сожалению, большинство современных игр достаточно сложно запустить в операционной системе Linux без использования дополнительного ПО. По этой причине пришлось ограничиться играми, требующими меньшее количество ресурсов, но запускаемыми в операционной системе Linux без использования дополнительных программ. Для тестирования была выбрана игра Total Annihilation [12], запущенная на максимальных возможных настройках. Четвертым и последним тестом было декодирование видео на центральном процессоре.

Все тесты проводились с частотой измерений 5 раз в секунду в течение 20 минут каждый. При проведении тестов была поставлена задача оценить, какова загрузка интересующих нас шин для каждого из тестов и как она меняется в зависимости от изменения количества активных ядер и тактовой частоты процессора. Если тестирование проводилось для нескольких активных ядер, запускалось количество тестовых программ, равное количеству активных ядер.

Запрос к кеш-памяти третьего уровня от ядра возможен только в случае, когда был выполнен поиск в кеш-памяти второго уровня и необходимые данные были не найдены. Соответственно, чтобы рассчитать количество данных, проходящее по шине между кеш-памятью второго и третьего уровней, необходимо количество промахов в кеш-памяти второго уровня умножить на размер пакета данных, проходящего по шине для одного запроса, которое мы приняли равным 72 байтам. Таким образом, скорость передачи данных в гигабитах в секунду исходя из полученных нами данных можно рассчитать по следующей формуле:

$$S_{\mathrm{L2-L3}} = \frac{L2_{\mathrm{miss,av}} \cdot 10^6 \cdot 8 \cdot 72}{10^9}, \tag{1}$$

где $L2_{\mathrm{miss,av}}$ — это среднее значение количества промахов в кеш-памяти второго уровня за секунду.

Аналогично для шины между кеш-памятью третьего уровня и оперативной памятью запрос к оперативной памяти возможен, только если данные не были найдены в кеш-памяти третьего уровня, поэтому скорость передачи данных можно рассчитать по формуле:

$$S_{\mathrm{L3-RAM}} = \frac{L3_{\mathrm{miss,av}} \cdot 10^6 \cdot 8 \cdot 72}{10^9}, \tag{2}$$

где $L3_{\mathrm{miss,av}}$ — это среднее значение количества промахов в кеш-памяти третьего уровня за секунду.

Хочется отметить, что для шины между кеш-памятью второго и третьего уровней результаты даны для всех ядер сразу, а не в пересчете на шину между кеш-памятью третьего уровня и ядром с кеш-памятью первого и второго уровней. Для шины данных между кеш–памятью третьего уровня и оперативной памятью дана не полная загрузка всей шины, а только ее части, связанной непосредственно с процессором и оперативной памятью, так как в настоящее время по этой шине идут

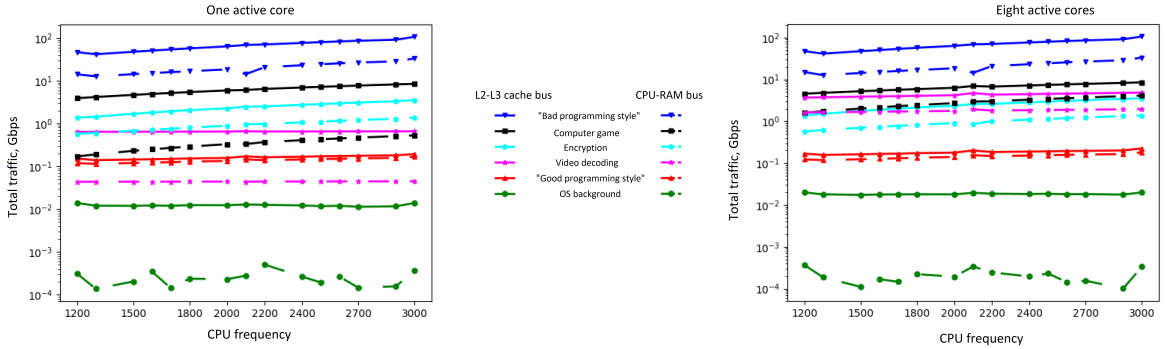

Рис. 8. Зависимость скорости передачи данных от тактовой частоты для одного и восьми активных ядер

Fig. 8. Dependence of total traffic on CPU frequency for 1 and 8 active cores

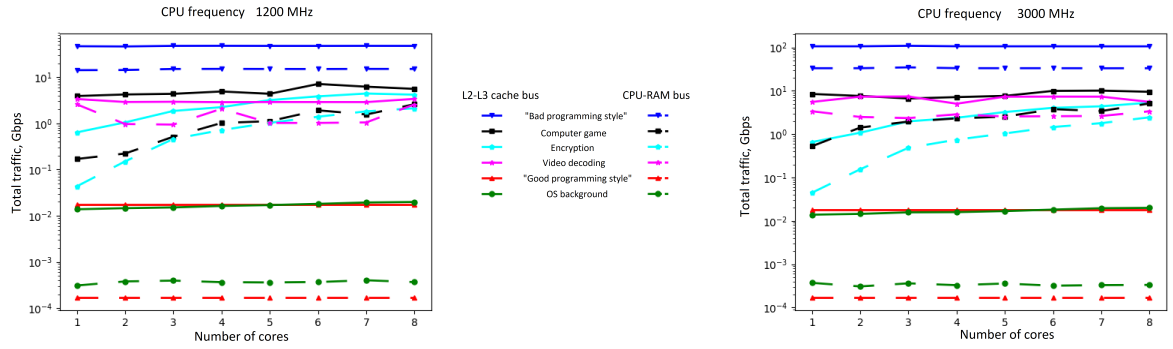

Рис. 9. Зависимость скорости передачи данных от количества активных ядер для минимальной и максимальной тактовой частоты

Fig. 9. Dependence of total traffic on number of active cores for minimal and maximal CPU frequencies

также данные от многих других устройств, но при разработке моделей процессоров на основе беспроводных систем на кристалле они обычно не берутся во внимание, и требуется только оценка скорости передачи данных именно между оперативной памятью и процессором.

В следующей главе рассмотрены полученные результаты и приведены зависимости скорости передачи данных на каждой из интересующих нас шин в зависимости от тактовой частоты процессора и количества активных ядер.

4. Численные результаты

Как видно из рисунков 8, 9 и 10, фоновая нагрузка операционной системы очень слабо влияет как на шину между кеш-памятью второго и третьего уровней, так и на шину между процессором и оперативной памятью. По сравнению с результатами других тестов фоновая нагрузка меньше на порядок. Следовательно, можно считать данные, полученные для всех тестов, достоверными. Также по графикам на рисунке

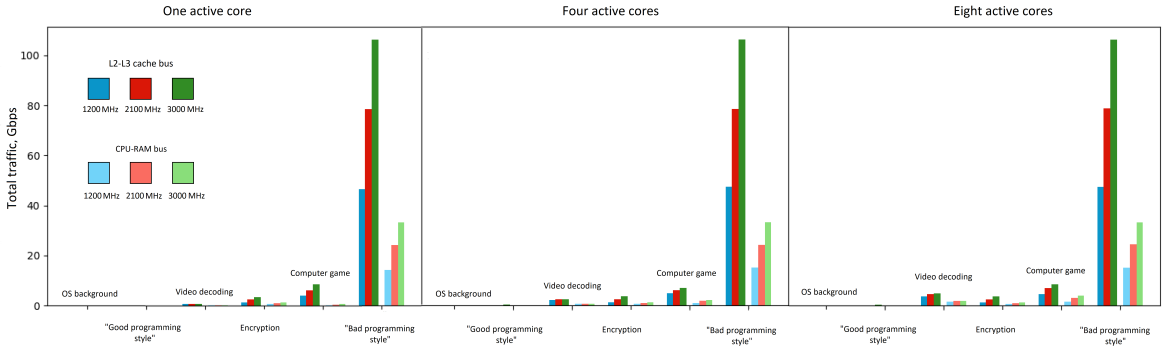

Рис. 10. Зависимость скорости передачи данных от типа теста и тактовой частоты процессора для одного, четырех и восьми активных ядер

Fig. 10. Dependence of total traffic on test type and CPU frequency for 1, 4 and 8 active cores

8 видно, что увеличение тактовой частоты процессора влияет на скорости передачи данных между кеш-памятью второго и третьего уровней, а также на скорости передачи данных между процессором и оперативной памятью, однако значения отличаются не очень значительно. Кроме того, стоит отметить, что характер изменения графиков одинаковый для любого количества активных ядер. Из всего вышеперечисленного можно сделать вывод о том, что тактовая частота процессора оказывает некоторое влияние на количество данных, передаваемых между кеш-памятью второго и третьего уровней и между процессором и оперативной памятью, однако это влияние не очень значительно. Так же мы видим, что скорость передачи данных на интерфейсе между кеш-памятью второго и третьего уровней может превышать 100 Гбит в секунду, а на интерфейсе между процессором и оперативной памятью достигать 40 Гбит в секунду. Графики на рис. 9 показывают зависимость скорости передачи данных между кеш-памятью второго и третьего уровней и между процессором и оперативной памятью от количества активных ядер. На них видно, что количество данных очень сильно зависит от того, какая программа использует процессор. Если для компьютерной игры или декодирования видео значения скорости оказываются в районе 10 Гбит в секунду для шины между вторым и третьим уровнем кеш-памяти и 5 Гбит в секунду для шины между оперативной памятью и процессором, то для теста, имитирующего плохой стиль программирования, эти значения превышают 100 Гбит в секунду для интерфейса между кеш-памятью второго и третьего уровней и приближаются к 40 Гбит в секунду для интерфейса между процессором и оперативной памятью, как было сказано выше. Кроме того, можно отметить, что при увеличении количества ядер количество данных, передаваемых на обоих интерфейсах, увеличивается для всех тестов кроме “плохого стиля программирования”. В случае с “плохим стилем программирования” количество данных, переданных за секунду, не изменяется. Это связано со спецификой теста – он достаточно сильно нагружает шины данных, и это вызывает дополнительные задержки при обращении к оперативной памяти, вследствие чего уменьшается количество операций в секунду, а значит, и промахов при поиске в кеш-памяти второго и третьего уровней. На рисунке 10 наглядно показано сравнение количества передаваемых за секунду данных для всех типов проводимых тестов для одного, четырех и восьми ядер. Дан-

ные представлены для трех различных тактовых частот – 1200 МГц, 2100 МГц и 3000 МГц. На графиках наглядно видно, что количество передаваемых за секунду данных растет по-разному для каждого теста, но зависимость примерно одинаковая для всех частот.

Кроме того, можно заметить, что синтетические тесты “хороший стиль программирования” и “плохой стиль программирования” представляют собой минимальный и максимальный случаи полезной загрузки для шины между кеш-памятью второго и третьего уровней и почти минимальный и максимальные случаи для ОЗУ, и таким образом их можно использовать для тестирования новых дизайнов.

Также стоит отметить, что загрузка шины между процессором и оперативной памятью практически всегда существенно меньше, чем загрузка шины между вторым и третьим уровнем кеш-памяти. Это связано с тем, что существующая архитектура уже крайне эффективна и также подчеркивает ограничения существующих ЦП.

5. Заключение

В данной статье были рассмотрены интерфейсы внутри и вне центрального процессора – шины между кеш-памятью второго и третьего уровней и между процессором и оперативной памятью. Была решена задача измерения реальных скоростей передачи данных на интерфейсе между кеш-памятью второго и третьего уровней внутри процессора и на интерфейсе между процессором и оперативной памятью, а также изучена зависимость численных результатов от количества активных ядер, тактовой частоты процессора и типа проводимого теста. На этих интерфейсах с помощью специальных инструментов были проведены натурные измерения на различных частотах и на различном количестве ядер. Эта информация может быть использована для разработки новых архитектур ЦП, в том числе имеющих в своем составе беспроводные каналы связи.

Список литературы / References

[1] Peter J. Denning, Ted G. Lewis, “Exponential Laws of Computing Growth”, *Communications of the ACM*, **60**:1 (2017), 54–65.

[2] Intel Corporation, “7th gen intel core and Intel Xeon processor briefing”, https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/01/7th-gen-intel-core-january-product-brief.pdf.

[3] Thomas Walther, “Scaling through more cores. From single to multi core”, https://wr.informatik.uni-hamburg.de/_media/teaching/wintersemester_2015_2016/nthr-16-walther-scaling_through_more_cores-ausarbeitung.pdf.

[4] Li X., “Survey of Wireless Network-on-Chip Systems”, http://www.eng.auburn.edu/ agrawvd/THESIS/LI/report.pdf.

[5] Ganguly A., Deb S., Belzer B., “Scalable hybrid wireless network-on-chip architectures for multicore systems”, *IEEE Transactions on Computers*, **60**:10 (2011), 1485–1502.

[6] Advanced Micro Devices Inc., “AMD desktop processor solutions”, “AMD desktop processor solutions”, www.amd.com.

[7] Intel Corporation, www.intel.com.

[8] Intel Corporation, “Intel 64 and IA-32 Architectures Software Developer’s Manual”, https://software.intel.com/sites/default/files/managed/39/c5/325462-sdm-vol-1-2abcd-3abcd.pdf.

[9] Kubuntu devs, “Kubuntu 14.10”, www.kubuntu.org/news/kubuntu-14.10.

[10] Intel Corporation, “Intel Performance Counter Monitor”, www.intel.com/software/pcm.

[11] Daemen J., Rijmen V., “AES Proposal: Rijndael”, 1999.

[12] Total Annihilation Universe, “Total Annihilation Universe”, www.tauniverse.com.

Komar M. S., "Data Rates Assessment on L2–L3 CPU Bus and Bus between CPU and RAM in Modern CPUs", *Modeling and Analysis of Information Systems*, **24**:4 (2017), 434–444.

DOI: 10.18255/1818-1015-2017-4-434-444

Abstract. In this paper, a modern CPU architecture with several different cache levels is described, and current CPU performance limitations such as silicone physical limitations or frequency increase bounds are mentioned. As usual, changes of the currently existing architecture are proposed as a way of increasing CPU performance, data rates on the internal and external CPU interfaces must be known. It would help to assess applicability of proposed solutions and allow to optimize them. This paper is aimed at getting real values of traffic on L2-L3 cache interface inside CPU and CPU-RAM bus load as well as show dependencies of total traffic on the interfaces of interest on the number of active cores, CPU frequency and test type. Measurements methodology using Intel Performance Counter Monitor by Intel is provided and equations that allow to get data rates from internal CPU counters are explained. Both real life and synthetic tests are described. Dependency of total traffic on the number of active cores and dependency of total traffic on CPU frequency are provided as plots. Dependency of total traffic on test type provided as bar plot for multiple CPU frequencies.

Keywords: multicore CPUs, data rates assessment, System-on-Chip, Network-on-Chip, Wireless Network-on-Chip, NoC, WNoC

On the authors:
Maria S. Komar, orcid.org/0000-0002-0995-7744, PhD student,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia,
MSc student, Tampere University of Technology,
PO Box 527, FI-33101, Korkeakoulunkatu 10, Tampere, Finland
e-mail: maria.s.komar@gmail.com