МНОГОЯДЕРНОСТЬ И МНОГОПОТОЧНОСТЬ В ИНТЕГРИРОВАННЫХ СЕТЕВЫХ ПРОЦЕССОРАХ

ВЛАДИМИР ЕГОРОВ, к. т. н., в. н. с., Институт проблем информатики РАН

В универсальных высокопроизводительных процессорах максимально совмещаются разнообразные подходы к распараллеливанию вычислений, в т. ч. многоядерность (multicore) и многопоточность (multithreading). Из-за трудности дать количественную оценку эффективности многоядерности и многопоточности, а также вследствие существенной взаимозависимости этих показателей в случае универсальных серверных процессоров компании-разработчики характеризуют свою продукцию на основе внутрифирменных предпочтений или собственных традиций. Анализ особенностей применения интегрированных сетевых процессоров, а также учет специфики работающих на них операционных систем и прикладных программ дают возможность сделать вывод о предпочтительности в сетевых процессорах многоядерности перед многопоточностью. Этот вывод косвенно подтверждается превалированием в сетевых процессорах архитектуры ARM, патентодержатели которой концептуально отвергают многопоточность в пользу многоядерности.

ВВЕДЕНИЕ

Производительность всегда была и остается одной из главных характеристик процессоров. Она постоянно растет по мере освоения микроэлектронной промышленностью новых технологических рубежей. Кроме того, не полагаясь только на достижения технологии, разработчики процессоров вводят в них разнообразные архитектурные и структурные усовершенствования, среди которых важное место занимает параллелизм вычислений во всех его проявлениях: суперскалярности (superscalarity), многопоточности (multithreading), многоядерности (multicore) и мультипроцессорности (multiprocessing). В универсальных высокопроизводительных процессорах все эти подходы максимально совмещаются. Однако уровни параллелизма взаимозависимы, и на практике из-за их взаимного влияния приходится искать разумные компромиссы. Критерии для поиска таких компромиссов во многом зависят от назначения и области применения процессора. Одной из таких областей, предъявляющей специфические требования к параллелизму вычислений, является сетевая инфраструктура, в которой давно укоренились и продолжают с успехом использоваться интегрированные сетевые процессоры (ИСП).

Качественный анализ взаимосоотношения таких широко применяемых форм распараллеливания вычислений в процессорах как многопоточность и многоядерность позволяет выявить специфические с точки зрения этих форм особенности ИСП и оценить предпочтительные для них направления развития параллелизма вычислений.

СООТНОШЕНИЕ МНОГОЯДЕРНОСТИ И МНОГОПОТОЧНОСТИ

На рубеже тысячелетий, когда микроэлектроника преодолела знаковый рубеж в миллиард транзисторов на кристалле кремния, востребованность нового «зарубежья» оказалась под вопросом в рамках традиционной моноядерной парадигмы процессоров. Решить эту проблему обещали, в частности, высокоинтегрированные интеллектуальные системы-на-кристалле (СнК). Такие системы могли объединять в одном микроэлектронном приборе высокопроизводительный процессор с памятью, контроллерами разнообразных интерфейсов ввода–вывода и разного рода специализированными устройствами. Однако более высокая интеграция СнК и включение в них специализированного оборудования неизбежно сопровождалась и более узкой их специализацией, которая ограничивала области применения и, как следствие, потенциальные объемы выпуска. Уменьшение же объемов выпуска удорожало продукты, нивелировало преимущества высокой интеграции и, в конечном счете, ставило под сомнение экономическую целесообразность развития микроэлектроники в этом направлении. Утвердиться на микроэлектронном рынке сумели лишь немногие категории СнК. В их число уверенно вошли ИСП благодаря растущему массовому спросу, который обеспечивался постоянным расширением в мировом масштабе сетевой инфраструктуры и перманентной интеллектуализацией ее компонентов.

Между тем, в новом тысячелетии обрел второе дыхание застопорившийся было процесс повышения интеграции собственно процессоров. Толчком к этому, возможно, на первый взгляд несколько неожиданно, послужило развитие широкодоступной беспроводной связи, породившее такой феномен как облачные вычисления. Облачная инфраструктура, предоставляемая как сервис многочисленным пользователям разнообразных гаджетов, потребовала громадные объемы процессорных ресурсов, компактно сосредоточенных в серверах облачных датацентров. Так, в число важных показателей экономической эффективности датацентров вошла «плотность процессоров», причем

Таблица. **Показатели многоядерности серверных процессоров ведущих фирм**

Компания	Семейство процессоров	Максимальное число ядер	Число потоков в ядре	Максимальное суммарное число потоков	
AMD	EPYC	64	2	128	
Intel	Xeon Platinum	56	2	112	
Marvell	ThunderX2	54	4	216	
Oracle	SPARC M8	32	8	256	
IBM	POWER9	24	8	96	*)

*) Возможны конфигурации (24×4) или (12×8) потоков

высокопроизводительных, на единицу объема серверного пространства. В конкурентной погоне за этим показателем производители серверов стали быстрыми темпами наращивать в них число процессоров, а их поставщики – число ядер в своих изделиях, ориентированных на серверные применения. Текущие достижения ведущих фирм в области многоядерных серверных процессоров представлены в таблице.

В таблицу не случайно включены характеристики многопоточности. Компании, предлагающие многопоточные процессоры, предпочитают в рекламных целях делать акцент именно на максимальном суммарном числе потоков как наиболее впечатляющей характеристике своей продукции. Более того, нередко многочисленные потоки представляются некими «логическими процессорами» [1] или «виртуальными ядрами» [2]. Однако называть поток логическим процессором или виртуальным ядром допустимо лишь в рекламных брошюрах.

В отличие от процессоров и ядер, потоки разделяют кэши всех уровней и по данным, и по адресам. Вследствие этого многопоточность дает ощутимый эффект только тогда, когда все исполняемые ядром потоки принадлежат одной и той же программе. Распараллеливание потоков одной программы на несколько процессорных ядер может ускорить ее выполнение при условии слабой взаимозависимости потоков по данным. Однако запуск нескольких программ разными потоками на одном ядре лишь ухудшит ситуацию из-за постоянных конфликтов в кэшах, в т. ч. кэшах адресных таблиц, т. к. разные программы работают в разных виртуальных пространствах. Поэтому нет ничего удивительного в том, что нередко при решении на многопоточном процессорном ядре тестовых задач, запрограммированных для разного числа потоков, самая высокая производительность ядра достигалась при работе с одним-единственным потоком [3–4].

Интеграция множества ядер в один кристалл обеспечивала процессору высокую потенциальную производительность и низкую удельную стоимость, приведенную к единице производительности. В свою очередь, доступность высокопроизводительных многоядерных процессоров вкупе с легкостью наращивания их числа в серверах благодаря мультипроцессорным технологиям «сокетов» (sockets) и «лезвий» (blades) во многом стимулировала использование серверов для решения ранее не свойственных им специфических задач, таких, например, как виртуализация сетевых функций. Кроме того, она способствовала активному применению в датацентрах разных форм виртуализации и программного определения их основных функциональных компонентов: сетей (software-defined networking) и хранилищ (software-defined storage). Эти полезные возможности даются чуть ли не даром, поскольку дополнительные накладные расходы на виртуализацию или программное определение зачастую покрываются технологическим избытком серверных ресурсов [5].

ЭФФЕКТИВНОСТЬ МНОГОПОТОЧНОСТИ

В настоящее время быстродействие процессоров настолько выше быстродействия оперативной памяти, что нередко большая часть процессорного времени тратится впустую на ожидание запрошенных из памяти данных. Такие же простои могут возникать при обращении к другим разделяемым ресурсам. Ключевое преимущество многопоточности заключается в возможности переключения программы на другой поток, пока текущий по той или иной причине вынужден простаивать. Для переключения на альтернативный поток необходимо сменить потоковый контекст, что традиционный процессор без специальных средств поддержки многопоточности вынужден делать программно и может потратить на это больше времени, чем теряется в простоях из-за ожидания данных. Для получения выигрыша от многопоточности альтернативные потоки должны быть готовы к немедленному исполнению, а их переключение желательно осуществлять мгновенно. Эту задачу решает одновременная многопоточность (ОМП). Процессоры с ОМП оснащаются специальной аппаратурой, которая каждому потоку предоставляет отдельный персональный контекст, включающий в типичном случае комплект универсальных регистров и средства предвыборки инструкций данного потока. Считается, что такие добавки несущественно увеличивают оборудование процессора, т. к. общей для всех потоков остается его большая часть: рабочие конвейеры, сопроцессоры и кэши всех уровней.

В целом, количественно оценить эффективность ОМП не представляется возможным. Она зависит от слишком многих факторов: особенностей выполняемого алгоритма, методов его программирования, способов реализации многопоточности на уровне аппаратуры процессора и операционной системы. Чтобы ОМП ускорила выполнение программы, необходимо соблюдение целого ряда условий:

- в исходном алгоритме должен присутствовать параллелизм, допускающий одновременное выполнение нескольких действий;
- в рабочем коде программистом или компилятором должны быть организованы соответствующие этим действиям потоки;
- при исполнении кода потоки не должны тормозиться взаимными блокировками на разделяемых аппаратных ресурсах и при доступе к общим данным.

Сложность объективной оценки эффективности ОМП косвенно, но показательно проявляется в разнице подходов, видимо во многом интуитивных, к ее реализации фирмами-поставщиками серверных процессоров. В то время как Oracle в процессорах семейства SPARC или IBM в процессорах семейства POWER традиционно оснащают свои процессорные ядра восемью потоками, Intel во всех процессорах с фирменной технологией HyperThreading ограничивается двумя потоками на ядро. Этой же нормы придерживается и ее прямой конкурент – компания AMD (см. табл.). Разумеется, увеличение числа потоков в ОМП при благоприятном стечении обстоятельств обязано дать выигрыш в производительности. Например, по некоторым оценкам, в восьмипоточном процессоре POWER5 ОМП позволяет повысить суммарную производительность до 60%, в то время как в двухпоточном процессоре Xeon максимальный выигрыш благодаря ОМП равен 30% [6]. Вопрос, однако,

в том, оправдываются ли экономически аппаратные и энергетические затраты на дополнительные потоки итоговым интегральным повышением производительности. Едва ли имеется однозначный ответ на все случаи, но с учетом массовости использования процессоров Intel в самом широком спектре применений, который гораздо больше, чем в случае процессоров SPARC или POWER, компанию трудно заподозрить в некомпетентности или небрежении к запросам пользователей. Более вероятно предположение, что внутрифирменные оценки эффекта от ОМП не побуждают Intel увеличить число поддерживаемых процессорным ядром потоков так, чтобы оно превысило два потока.

Еще более радикальную позицию в отношении ОМП занимает компания ARM. Она полагает более выгодным тиражировать на кристалле целиком процессорные ядра, а не оснащать их средствами многопоточности. По оценкам апологетов архитектуры ARM [7], хотя четырехъядерный процессор и занимает большую площадь кремния, он потребляет меньше энергии, чем одноядерный четырехпотоковый процессор, причем выигрыш достигает 46%. Кроме того, выполнение программы двумя потоками на одном ядре увеличивает число обращений к кэшу на 42% по сравнению с выполнением одним потоком, в то время как при работе той же программы на двух ядрах, наоборот, число обращений к кэшу уменьшается на 37%. Вероятно, следуя этим оценкам, компания ARM практически полностью отказалась в своих процессорах от ОМП. На сегодняшний день в мире имеется лишь один пример многопоточного воплощения архитектуры ARM: ее оригинальная фирменная реализация в семействе процессоров ThunderX2 от компании Marvell (см. табл.), хотя еще предыдущее поколение этих процессоров ThunderX (разработка Cavium) было традиционно однопоточным.

ОСОБЕННОСТИ МНОГОЯДЕРНОСТИ И МНОГОПОТОЧНОСТИ В ИСП

При всем многообразии ИСП их можно поделить на две большие категории по специализации архитектур процессорных ядер. Специализированные и, как правило, предельно упрощенные благодаря специализации ядра позволяют при прочих равных условиях увеличить суммарную производительность ИСП как за счет большей эффективности отдельного ядра, так и благодаря возможности интеграции в СнК большего их числа при одном и том же уровне технологии. Примерами ИСП с оригинальной архитектурой и специализированными ядрами может служить продукция компаний Netronome [8], Tilera (ныне Mellanox) [9], EZchip (ныне тот же Mellanox) [10], Xelerated (ныне Marvell) [11]. К этим изделиям, вероятно, следует отнести ИСП nPower компании Cisco, одного из грандов рынка сетевого оборудования, которая добилась рекордной интеграции в 672 процессорных ядра с фирменной архитектурой [12].

Однако в ИСП широко и в последнее время все чаще используются универсальные процессорные ядра с традиционными архитектурами, в первую очередь, с архитектурой ARM [13]. В качестве примеров можно привести ИСП компаний Motorola/Freescale (ныне NXP) [14], LSI (ныне Intel) [15] или Cavium (ныне Marvell) [16]. Наблюдаемый вполне ощутимый сдвиг предпочтений разработчиков ИСП от специализированных к универсальным процессорным ядрам вполне объясним. Первые сложно программировать из-за специфичности архитектуры и ограниченности средств разработки программного обеспечения, доступных только обладателям ноу-хау архитектуры, тогда как последние имеют гораздо более развитую экосистему с готовыми разнообразными операционными средами и библиотеками, а также удобными апробированными инструментами разработки, в т. ч. от нескольких сторонних поставщиков. В результате стандартные архитектуры процессорных ядер в ИСП существенно упрощают создание потребителями собственных приложений и, как следствие, удешевляют разработку и сокращают время вывода новых продуктов на рынок.

Хотя универсальные процессорные ядра все активнее внедряются в ИСП, это не те же самые ядра, которые применяются в серверных процессорах. Так, на сегодняшний день практически нет примеров применения в ИСП ядер с архитектурой x86. Главное (если не единственное) достоинство последней заключается в огромном объеме наработанного серверного программного обеспечения. Однако ряд его существенных компонентов, таких как виртуализация или программное определение серверов, не являются релевантными для ИСП. В то же время широко применяемые в ИСП архитектуры ARM и MIPS позволяют разместить на кристалле больше процессорных ядер и добиться не только более высо-

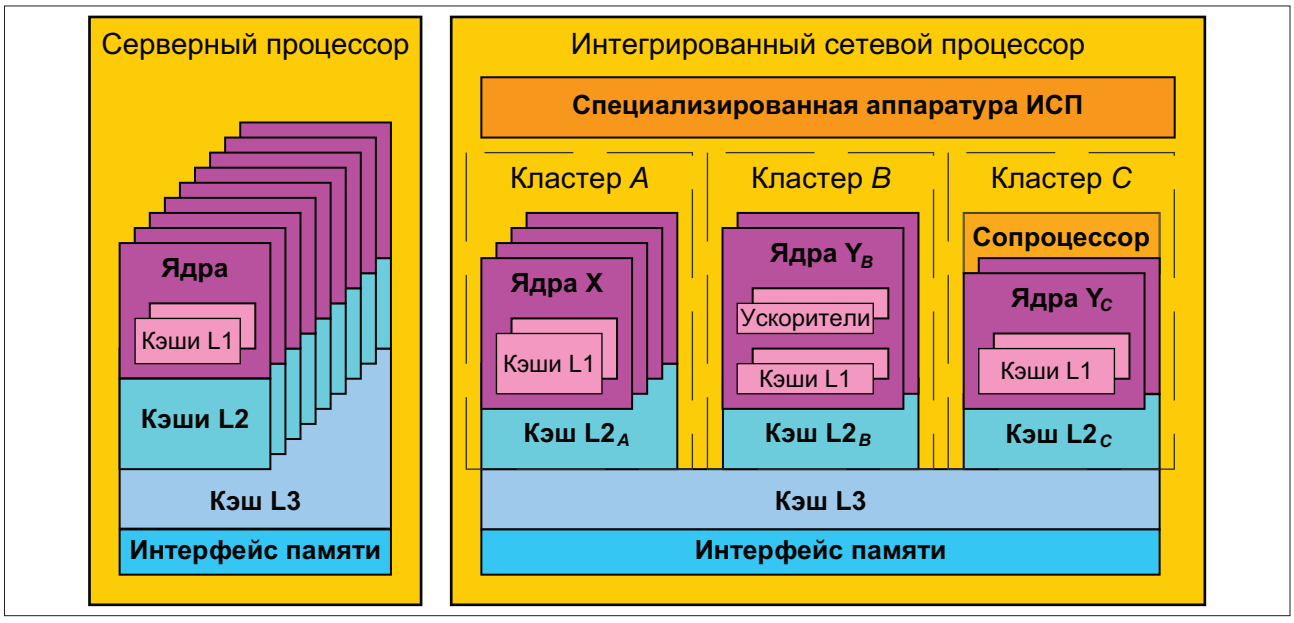

Рис. 1. **Различия в организации кэшей в серверных процессорах и ИСП**

кой производительности, но и лучшей энергоэффективности ИСП.

Кроме того, заметно разнятся в процессорах для серверов и ИСП подходы к организации кэшей. Если в серверных процессорах в целях максимальной независимости ядер от общего для всех интерфейса памяти и приближения функциональности ядра к возможностям отдельного процессора каждое ядро снабжается собственным кэшем второго уровня (L2), для ИСП более предпочтительной может оказаться кластерная организация ядер с общим кэшем L2 для всех ядер одного кластера [17]. Разницу в организации кэшей иллюстрирует рисунок 1.

Отдельно отметим роль гетерогенных кластеров в структуре процессоров. По некоторым оценкам, на уровне рабочего конвейера максимальную отдачу аппаратура дает, являясь гетерогенно кластеризованной в соответствии с требованиями приложения [18]. Правда, остается непонятным, как можно заранее угадать требования приложений при создании процессора для серверов, на которых могут исполняться самые разнообразные задачи в разных операционных средах. Другое дело ИСП, где круг приложений ограничен, а их характеристики действительно заранее известны. Более того, в ИСП гетерогенная функционально специализированная кластеризация процессорных ядер желательна и на архитектурном уровне [17], т. к. она позволяет наиболее эффективно использовать кэши L2 и специализированное оборудование ИСП.

В примере на рисунке 1 три кластера СП реализованы на процессорных ядрах двух типов, имеют кэши разных объемов и включают разное число ядер с разнящейся дополнительной специализированной аппаратурой в ядрах и кластерах. Таким образом, именно в ИСП гетерогенная кластеризация способна наиболее полно проявить свои преимущества.

ИСП не предъявляют каких-либо специфических требований к многопоточности процессорных ядер. Однако на практике ОМП в ИСП может оказаться просто излишней и обернуться неокупаемыми аппаратными затратами по двум причинам. Во-первых, типичные применения ИСП предполагают работу в жестком реальном времени с использованием соответствующих операционных сред. Между тем, операционные системы реального времени традиционно рассчитаны скорее на многозадачность, чем многопоточность. Во-вторых, для ИСП характерен фиксированный комплект рабочих кодов, как правило, относительно небольшого объема, целиком помещающийся в кэшах L2 кластеров либо хранящийся в специальной локальной памяти (tightly-coupled memory) процессорных ядер. Поэтому в типичном ИСП, особенно кластерно организованном, выполнение отдельного потока программы не тормозится обращениями к внешней памяти, и многопоточность как таковая теряет смысл, лишившись своего главного преимущества. Эта особенность ИСП служит дополнительным весомым аргументом в пользу использования в них процессорных ядер архитектуры ARM с ее концептуальным отказом от многопоточности в пользу агрессивной многоядерности.

ВЫВОДЫ

Растущие требования к производительности процессоров заставляют их разработчиков применять всевозможные методы повышения быстродействия, в т. ч. распараллеливание вычислений в вариантах многопоточности и многоядерности. Хотя изготовители процессоров иногда рекламируют многопоточность своих изделий как альтернативу многоядерности, поток не эквивалентен процессорному ядру и тем более процессору, а многопоточность не заменяет многоядерность. Из-за отсутствия универсальных оценок эффективности многопоточности и уверенности в ее абсолютной эффективности разработчики разных компаний руководствуются интуицией, внутрифирменными оценками и традициями при выборе технических решений для своих процессорных ядер. В результате на рынке представлены процессорные ядра с разным числом потоков. Есть и примеры принципиального игнорирования многопоточности.

Если многопоточность – не замена многоядерности, то многоядерность во многих случаях может успешно заменить многопоточность. Она более универсальна, поскольку применима для распараллеливания и потоков одной задачи, и целиком задач. Более того, даже при распараллеливании потоков она может оказаться более выгодной, в частности, с точки зрения энергопотребления.

Особенно успешно многоядерность, в т. ч. как альтернатива многопоточности, применима в ИСП. Особенности внутренней организации ИСП и специфика работающего в них программного обеспечения объективно ограничивают многопоточность в возможностях продемонстрировать свои лучшие свойства. В то же время они позволяют в полной мере проявиться преимуществам многоядерности. Не случайно все более массовое применение в ИСП находит архитектура ARM, патентообладатели которой отдают безусловное предпочтение многоядерности перед многопоточностью.

ЛИТЕРАТУРА

1. Г. Речистов. Процессоры, ядра и потоки. Топология систем//habr.com/ru/company/intel/blog/243385.

2. QorIQ T4240/T4160/T4080 Multicore Communications Processors//www.nxp.com.

3. Why is my multi threading not efficient?//stackoverflow.com.

4. А. А. Кадомский, В. А. Захаров. Эффективность многопоточных приложений. Научный журнал. 2016//scientificmagazine.ru.

5. В. Б. Егоров. Накладные расходы виртуализации и влияющие на них факторы. Системы и средства информатики. 2017. Т. 27. № 3.

6. Y. Rua, E. Nahum, V. Pai, J. Tracey. On the effectiveness of simultaneous multithreading on network server workloads//pdfs.semanticscholar.org.

7. C. Mulligan. ARM is no fan of HyperThreading//www.theinquirer.net.

8. В. Егоров. Интегрированные сетевые процессоры потоков от Netronome. Электронные компоненты. 2014. № 10.

9. В. Егоров. Многоядерные сетевые процессоры от Tilera. Электронные компоненты. 2014. № 6.

10. В. Егоров. Сетевые процессоры EZchip. Электронные компоненты. 2011. № 7.

11. В. Егоров. Сетевые процессоры Xelerated. Электронные компоненты. 2011. № 10.

12. J. Markevitch, S. Malladi. A 400 Gbps Multi-Core Network Processor//www.hotchips.org.

13. В. Егоров. Тенденция к ARMированию многоядерных интегрированных сетевых процессоров. Электронные компоненты. 2018. № 6.

14. В. Егоров. Интегрированные сетевые процессоры NXP с ARM-ядрами покоряют верхний «ярус». Электронные компоненты. 2018. № 3.

15. В. Егоров. Коммуникационные процессоры Axxia от LSI. Электронные компоненты. 2014. № 4.

16. В. Егоров. Интегрированные сетевые процессоры Cavium. Электронные компоненты. 2011. № 4.

17. В. Б. Егоров. Современные тенденции в развитии архитектур интегрированных сетевых процессоров. Системы и средства информатики. 2014. Т. 24. № 3.

18. C. Acosta, A. Falcón, A. Ramirez, M. Valero. A complexity-effective simultaneous multithreading architecture. Proceedings of the International conference on parallel processing//upcommons.upc.edu.