УДК 004.715

В.И. Грищенко (аспирант),
Ю.В. Ладыженский (канд.техн.наук, доц.)
Донецкий национальный технический университет
victor.grischenko@gmail.com, ly@cs.dgtu.donetsk.ua

ИССЛЕДОВАНИЕ ВЛИЯНИЯ РАЗДЕЛЬНОЙ ПАМЯТИ НА ПРОИЗВОДИТЕЛЬНОСТЬ МНОГОЯДЕРНОГО СЕТЕВОГО ПРОЦЕССОРА

В статье рассмотрены подходы к моделированию специализированных многоядерных сетевых процессоров пакетной обработки данных. Предложена модель сетевого процессора с раздельной памятью для кода и данных. Представлены результаты моделирования сетевого процессора и проведено сравнение производительности сетевых процессоров с общей и раздельной памятью.

сетевой процессор, маршрутизатор, компьютерная сеть, пропускная способность, модель, сеть массового обслуживания

Введение

Высокая интенсивность обмена данными в современных компьютерных сетях требует высокой производительности маршрутизаторов и процессоры общего назначения не в состоянии обеспечить достаточную скорость обработки данных [2]. Вместо них для обработки потоковых данных в маршрутизаторах используют специализированные сетевые процессоры (СП), объединяющие в себе быстродействие аппаратной реализации наиболее часто востребованных функций и гибкость программируемых систем.

Производительность сетевого процессора — важнейшая его характеристика и ее увеличение является основной задачей при разработке нового устройства. В работе [1] рассматривалась модель многоядерного сетевого процессора с общим блоком памяти для кода и команд. Причем единственный блок памяти разделялся между всеми ядрами СП. Было показано, что в такой структуре память является узким местом и необходимо снижение нагрузок на ОЗУ.

В представленной статье предлагается модель сетевого процессора с разделенной памятью. С целью увеличения производительности каждому этапу конвейера выделены отдельные блоки ОЗУ для хранения кода. Получены характеристики новой структуры СП и проведено сравнение со структурой с общим блоком ОЗУ.

Анализ методик моделирования СП

В настоящее время ведется несколько параллельных исследований в области моделирования структур сетевых процессоров [3]. Основной целью перед собой исследователи ставят построение моделей, способных покрыть максимальное множество возможных структур СП.

В работах [4, 5] рассматривается аналитическая модель сетевого процессора, в которой каждый блок СП описывается двумя кривыми обслуживания. По мере обработки входной поток данных проходит через блоки и уменьшается на величину полученного обслуживания. Предложенная модель позволяет легко менять состав и структуру блоков СП, а также взаимосвязи между ними. С другой стороны каждое сетевое приложение, выполняемое на сетевом процессоре, имеет индивидуальные характеристики нагрузки на каждый узел и изменение исследуемого приложения требует пересчета всех кривых обслуживания для всех узлов СП. Авторами статьи не был предложен простой способа вычисления кривых обслуживания, что затрудняет использование предложенной модели.

Задача определения характеристик сетевых приложений подробно рассмотрена в работах [6, 7]. Авторами предложена комплексная методика моделирования, состоящая из аналитической модели производительности СП и методики использования имитационного моделирования для получения характеристик сетевых приложений. Предложенный подход позволяет исследовать работу практически любого приложения, написанного на языке Си. Главным недостатком комплексной методики является используемая аналитическая модель, которая существенно ограничивает вариативность исследуемых структур сетевых процессоров [3].

В работе [8] предложена модель структуры СП, основанная на теории массового обслуживания. Модель позволяет исследовать работу многоядерного сетевого процессора, в котором все ядра работают параллельно. Процессор имеет одно главное вычислительное ядро, обрабатывающее поток пакетов и, по мере изменения нагрузки, подключающее или отключающее дополнительные вычислительные ядра. Однако, предложенная модель не позволяет исследовать конвейерные структуры СП, которые являются мощным инструментом повышения производительности сетевого процессора [2].

Авторами в работе [1] была предложена методика моделирования сетевых процессоров, объединяющая комплексный подход и аналитическую модель конвейерного СП. Данная статья представляет развитие предложенной модели сетевых процессоров, в которой каждому этапу конвейера выделяется отдельный блок памяти для хранения кода.

Модель сетевого процессора

Одним из способов уменьшить нагрузку на блок ОЗУ является использование нескольких модулей оперативной памяти для выполнения различных функций. Например, можно выделить несколько небольших блоков ОЗУ для хранения кода и основной модуль памяти использовать только для хранения данных (см. рис. 1).

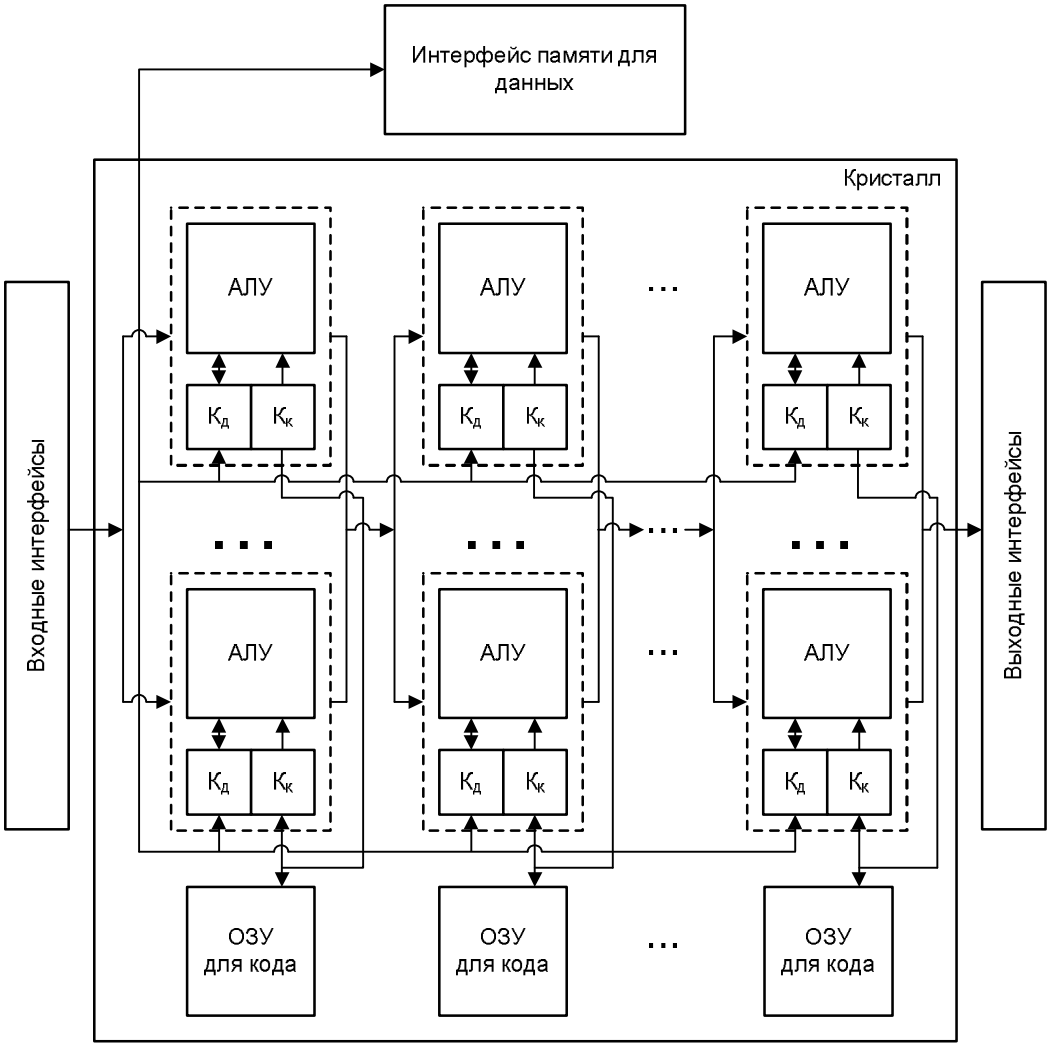

Рисунок 1 – Структура сетевого процессора

В большинстве случаев объем кода сетевых приложений сравнительно невелик, что позволяет разместить блоки ОЗУ кода непосредственно на кристалле сетевого процессора. Таким образом, прирост производительности достигается за счет двух основных факторов: уменьшения количества одновременных обращений к различным модулям ОЗУ и уменьшения задержек при обращении к ОЗУ кода.

На рисунке 1 показана схема сетевого процессора, использующего раздельные блоки ОЗУ для хранения кода и данных. В используемой модели СП вычислительные ядра разделены на этапы конвейера. Все ядра, принадлежащие одному этапу конвейера, выполняют одинаковый набор инструкций, который хранится в отдельном модуле ОЗУ. Все этапы конвейера используют общий модуль оперативной памяти для хранения данных. Это позволяет минимизировать затраты на передачу данных между различными этапами конвейера.

После завершения обработки на одном этапе конвейера пакет переходит к следующему и, благодаря особенностям обработки данных на маршрутизаторах [2], пакет никогда не возвращается на предыдущий этап обработки.

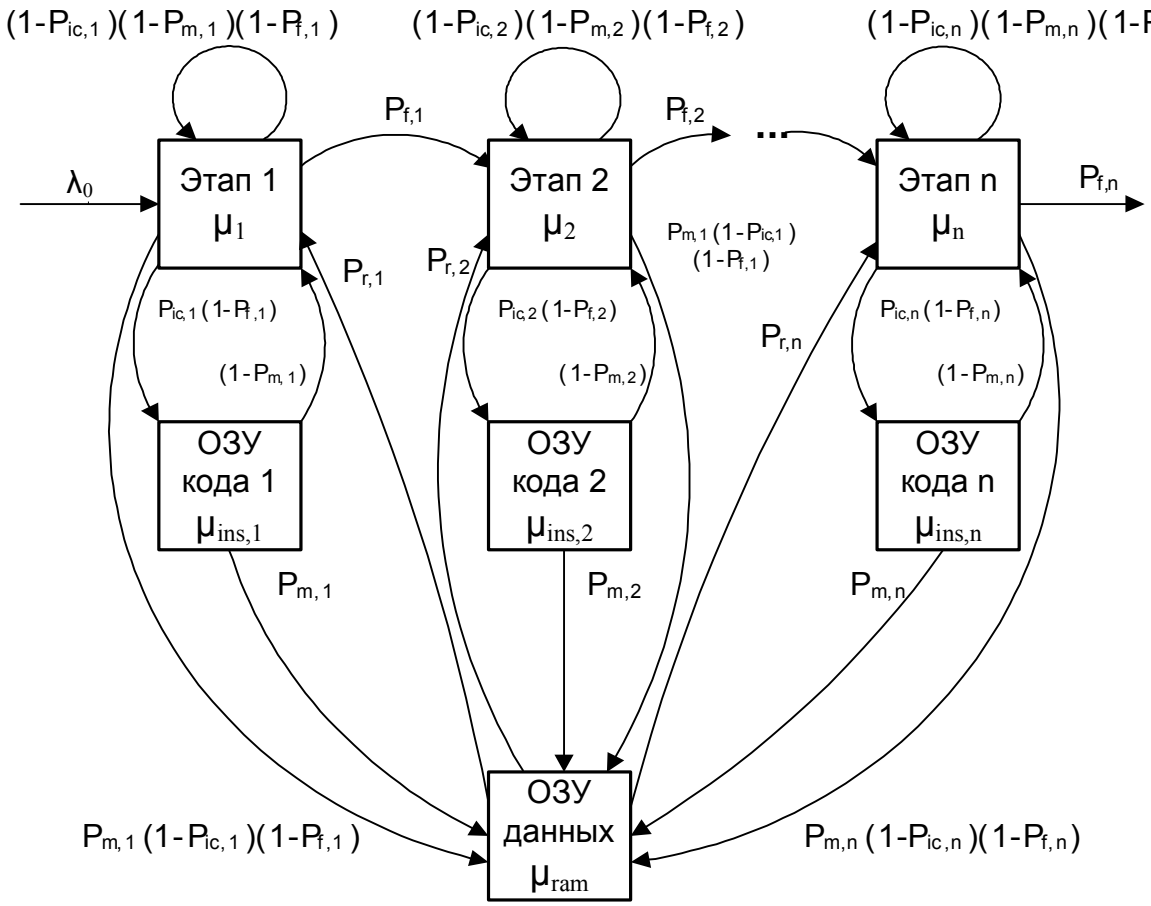

Рисунок 2 – Сеть массового обслуживания, моделирующая сетевой процесор

Сеть массового обслуживания, моделирующая работу многоядерного сетевого процессора, показана на рисунке 2. Здесь узлы с μ_1 по μ_n моделируют работу этапов конвейера обработки пакетов, с $\mu_{ins,1}$ по $\mu_{ins,n}$ — работу модулей ОЗУ для хранения команд и μram — модуль ОЗУ для хранения данных.

Сетевой пакет из входного потока с интенсивностью λ_0 поступает на первый этап конвейера μ_1 и выполняется одна инструкция по его обработке. После этого, с вероятностью $P_{f,1}$ обработка завершается, и пакет переходит на

следующий этап конвейера. Если этого не происходит, то с вероятностью $P_{ic,1}(1\text{-}P_{f,1})$ выполнение следующей инструкции потребует обращения к ОЗУ команд $\mu_{ins,1}$. Получив код команды, с вероятностью $P_{m,1}$ потребуется обращение к ОЗУ за данными, если же данные не нужны, то с вероятностью $(1\text{-}P_{m,1})$ процессор перейдет к выполнению полученной инструкции. Если же следующая инструкция оказалась в кэше, и обращение к ОЗУ команд не требуется, то с вероятностью $P_{m,1}(1\text{-}P_{ic,1})(1\text{-}P_{f,1})$ процессору потребуется обращение к ОЗУ данных.

Вероятность обращения в ОЗУ данных $(P_{m,i})$ определяется по формуле:

$$P_{m,i} = P_{d,i}P_{dc,i}, \qquad (1)$$

где $P_{d,i}$ — вероятность того, что текущей инструкции потребуются данные, а $P_{dc,i}$ — вероятность промаха кэша данных.

После обработки обращения к ОЗУ данных выполняется следующая инструкция по обработке пакета. Однако, поскольку ОЗУ данных разделяется между всем этапами конвейера, возникает необходимость определить вероятность возврата на каждый из этапов конвейера. Эта вероятность определяется по формуле:

$$P_{r,j} = \frac{P_{m,j}(1-P_{ic,j})(1-P_{f,j}) + P_{m,j}P_{ic,j}(1-P_{f,j})}{\sum_{i=1}^{n}\left[P_{m,i}(1-P_{ic,i})(1-P_{f,i}) + P_{m,i}P_{ic,i}(1-P_{f,i})\right]}. \qquad (2)$$

Выражение (2) можно упростить:

$$P_{r,j} = \frac{P_{m,j}(1-P_{f,j})}{\sum_{i=1}^{n}P_{m,i}(1-P_{f,i})}. \qquad (3)$$

Помимо описанных сценариев выполнения команд на процессоре, существует вероятность того, что инструкции не потребуется ни обращение к ОЗУ команд, ни обращение к ОЗУ данных. Эта вероятность равна $(1\text{-}P_{m,1})(1\text{-}P_{ic,1})(1\text{-}P_{f,1})$.

Объединив перечисленные сценарии выполнения команд, можно составить систему уравнений, описывающую потоки в рассматриваемом сетевом процессоре:

$$\begin{cases} \lambda_0 = \lambda_n P_{f,n}, \\ \lambda_1 = \lambda_0 + \lambda_1(1-P_{ic,1})(1-P_{m,1})(1-P_{f,1}) + \lambda_{ins,1}(1-P_{m,1}) + \lambda_{ram}P_{r,1}, \\ \lambda_i = \lambda_{i-1}P_{f,i-1} + \lambda_i(1-P_{ic,i})(1-P_{m,i})(1-P_{f,i}) + \lambda_{ins,i}(1-P_{m,i}) + \lambda_{ram}P_{r,i}, i = \overline{2,n}, \\ \lambda_{ins,i} = \lambda_i P_{ic,i}(1-P_{f,i}), i = \overline{1,n}, \\ \lambda_{ram} = \sum_{i=1}^{n}\left[\lambda_i P_{m,i}(1-P_{ic,i})(1-P_{f,i}) + \lambda_{ins,i}P_{m,i}\right] \end{cases} \qquad (4)$$

Для маршрутизатора, содержащего трехэтапный конвейер, система (4) приобретает вид:

$$
\begin{cases}
\lambda_0 = \lambda_3 P_{f,3}, \\
\lambda_1 = \lambda_0 + \lambda_1 (1 - P_{ic,1})(1 - P_{m,1})(1 - P_{f,1}) + \lambda_{ins,1}(1 - P_{m,1}) + \lambda_{ram} P_{r,1}, \\
\lambda_2 = \lambda_1 P_{f,1} + \lambda_2 (1 - P_{ic,2})(1 - P_{m,2})(1 - P_{f,2}) + \lambda_{ins,2}(1 - P_{m,2}) + \lambda_{ram} P_{r,2}, \\
\lambda_3 = \lambda_2 P_{f,2} + \lambda_3 (1 - P_{ic,3})(1 - P_{m,3})(1 - P_{f,3}) + \lambda_{ins,3}(1 - P_{m,3}) + \lambda_{ram} P_{r,3}, \\
\lambda_{ins,1} = \lambda_1 P_{ic,1}(1 - P_{f,1}), \\
\lambda_{ins,2} = \lambda_2 P_{ic,2}(1 - P_{f,2}), \\
\lambda_{ins,3} = \lambda_3 P_{ic,3}(1 - P_{f,3}), \\
\lambda_{ram} = \sum_{i=1}^{n} \left[\lambda_i P_{m,i} (1 - P_{ic,i})(1 - P_{f,i}) + \lambda_{ins,i} P_{m,i} \right]
\end{cases}
\tag{5}
$$

Разделив обе части каждого уравнения на λ_0, получим систему уравнений для коэффициентов посещений $\alpha_i (i = \overline{0,3})$, $\alpha_{ins,i} (i = \overline{1,3})$ и α_{ram}.

Решив полученную систему, получим значения коэффициентов посещений:

$$
\begin{gathered}
\alpha_1 \approx 170, \qquad \alpha_2 \approx 264, \qquad \alpha_3 \approx 101, \\
\alpha_{ins,1} \approx 0.169, \quad \alpha_{ins,2} \approx 39, \quad \alpha_{ins,3} \approx 0.1, \\
\alpha_{ram} \approx 3.71.
\end{gathered}
\tag{6}
$$

Из решения (6) системы уравнений видно, что среди модулей памяти наиболее нагруженным является модуль ОЗУ команд второго этапа конвейера. Поэтому производительность именно этого блока будет ограничивать пропускную способность всего сетевого процессора.

Расчет характеристик сетевого процессора

В таблице 1 перечислены параметры, используемые при расчете характеристик сетевого процессора. Как было указано ранее, размещение блоков ОЗУ кода на кристалле процессора влечет за собой уменьшение задержек при обращении к ним. В рассматриваемых примерах производительность этих блоков в 1.5 раза больше производительности блока ОЗУ данных. Это облегчает построение зависимостей, но в общем случае, изменять производительности разных блоков ОЗУ можно независимо.

Следует обратить внимание и на расчет интенсивности обслуживания вычислительного ядра. Во всех экспериментах, описанных в данной статье, сделано предположение о том, что вычислительное ядро тратит в среднем 7 тактов на выполнение одной инструкции.

Таблица 1. Параметры моделируемого сетевого процессора

Параметр	Значение	Описание
μ_1, μ_2, μ_3	от $5/7*10^7$ до $6/7*10^8$	Производительность вычислительных ядер, выраженная в количестве команд, выполняемых в секунду. Коэффициент 7 отражает тот факт, что процессор в среднем тратит 7 тактов на инструкцию.
$\mu_{ОЗУ}$	от $5*10^7$ до $1*10^9$	Производительность блока ОЗУ, выраженная в количестве обрабатываемых за секунду запросов
$\mu_{ins,1}$, $\mu_{ins,2}$, $\mu_{ins,3}$	$1.5*\mu_{ОЗУ}$	Производительность блоков ОЗУ кода.
P_{f1}	1/201	Для разбора заголовка пакета требуется в среднем 200 команд
P_{f2}	1/501	Для поиска в маршрутной таблице требуется в среднем 500 команд
P_{f3}	1/101	На сборку пакета 100 команд
$P_{ic,1}$, $P_{dc,1}$	0.001 0.01	Вероятности промахов кэша команд и данных для ядра 1, выполняющего разбор заголовка пакета
$P_{ic,2}$, $P_{dc,2}$	0.15 0.25	Вероятности промахов кэша команд и данных для ядра 2, выполняющего поиск в маршрутной таблице
$P_{ic,3}$, $P_{dc,3}$	0.001 0.01	Вероятности промахов кэша команд и данных для ядра 3, выполняющего сборку пакета
$P_{d,1}$	0.5	Вероятность обращения к данным в ОЗУ для ядра 1, выполняющего разбор заголовка пакета
$P_{d,2}$	0,036	Вероятность обращения за данными в ОЗУ для ядра 2, выполняющего поиск в маршрутной таблице
$P_{d,3}$	0.5	Вероятность обращения за данными в ОЗУ для ядра 3, выполняющего сборку пакета

Величина максимальной пропускной способности маршрутизатора $\lambda_{0,max}$ определяется из условия существования установившегося режима для сети массового обслуживания:

$$\lambda_{0,max} = \min\left(\frac{s_{ram}\mu_{ram}}{\alpha_{ram}}, \min_{i=1,n}\left(\frac{s_i\mu_i}{\alpha_i}\right)\right), \qquad (7)$$

На рисунке 3 показана зависимость максимальной пропускной способности от тактовой частоты вычислительных ядер. Из графиков видно, что до 270 MHz использование общего ОЗУ лучше, чем выделение отдельных блоков ОЗУ под код. Это объясняется тем, что в новой модели увеличились коэффициенты посещения вычислительных ядер. Это при той же производительности влечет за собой уменьшение их пропускной способности. В данном случае узким местом является второй этап конвейера. Но, уже начиная с 350 MHz, пропускная способность второго этапа конвейера становится выше пропускной способности блока ОЗУ кода, относящейся к этому этапу и дальнейшее увеличение тактовой частоты не увеличивает пропускную способность процессора.

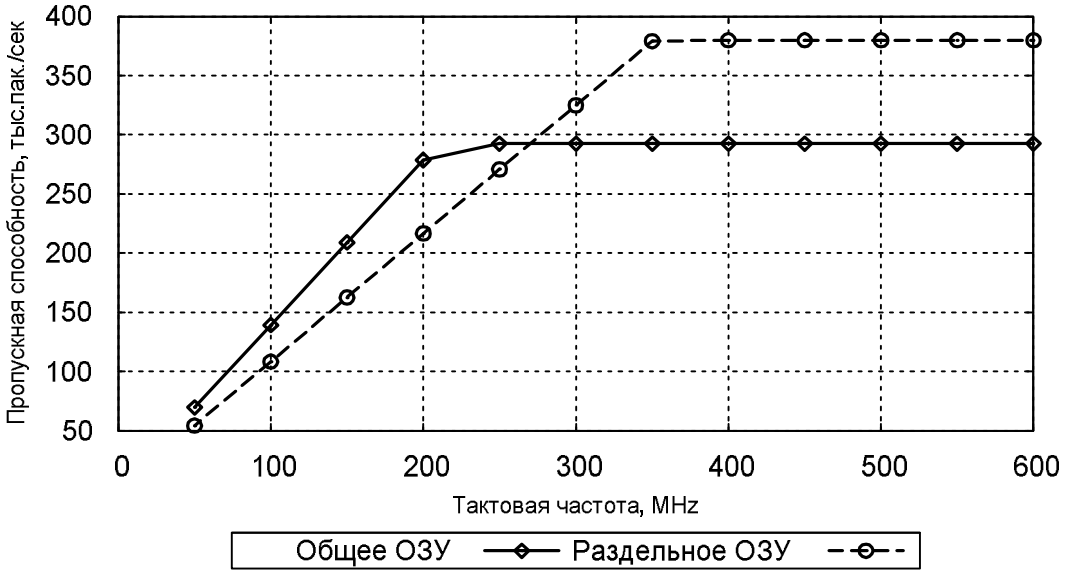

Рисунок 3 – Зависимость максимальной пропускной способности от тактовой частоты вычислительных ядер

На рисунке 4 показана зависимость пропускной способности сетевого процессора от пропускной способности блоков ОЗУ. Здесь по оси абсцисс шкала указана для блока ОЗУ данных. Чтобы получить пропускную способность блоков ОЗУ кода следует значение шкалы умножать на 1.5. Из графиков видно, что пропускная способность СП при использовании отдельных блоков ОЗУ кода незначительно выше производительности базовой структуры. Особый интерес связан с изменением динамики роста производительности при достижении отметки 22 млн. запросов в секунду. Это связано с тем, что пропускная способность при меньших производительностях ОЗУ ограничивалась блоком ОЗУ кода второго этапа конвейере, далее узким местом становится блок ОЗУ данных. В то же время, в случае использования общего блока памяти, коэффициент посещения второго этапа конвейера не столь велик, и даже с такой производительностью блок ОЗУ остается самым узким местом. Следует учитывать, что графики построены для вычислительных ядер, использующих семь тактов на инструкцию, что достаточно много для современных условий. В случае использования более эффективных блоков АЛУ их загруженность оказывается ниже загруженности блока ОЗУ кода второго этапа конвейера. Поэтому с ростом производительности памяти будет расти и пропускная способность всего маршрутизатора.

Общее время обработки пакета на маршрутизаторе u_t, зависит от времени обработки в каждом узле $\overline{u_i}$ и коэффициентов посещения узлов α_i:

$$\overline{u_t} = \alpha_{ram}\,\overline{u_{ram}} + \sum_{i=1}^{n}\left[\alpha_i\,\overline{u_i} + \alpha_{ins,i}\,\overline{u_{ins,i}}\right], \qquad (8)$$

На рисунке 5 показана зависимость времени обработки пакета на маршрутизаторе от интенсивности входного потока при производительности

блока ОЗУ данных равной 10 млн. запросов в секунду и блоков ОЗУ кода — 15 млн. запросов в секунду. Можно сделать вывод, что использование отдельных блоков ОЗУ для кода значительно уменьшает время обработки пакета, особенно при большой интенсивности входного потока.

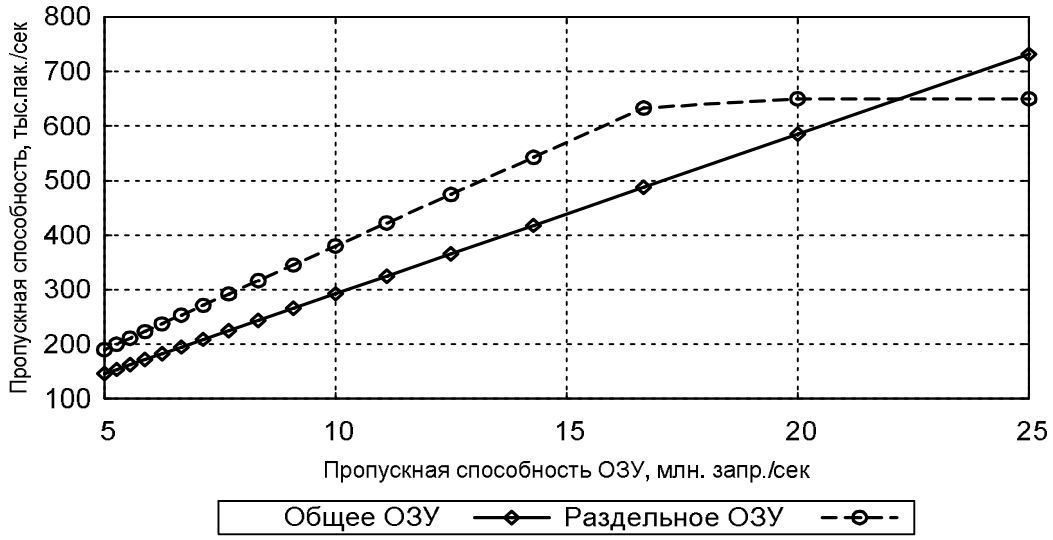

Рисунок 4 – Зависимость максимальной пропускной способности от производительности блока ОЗУ

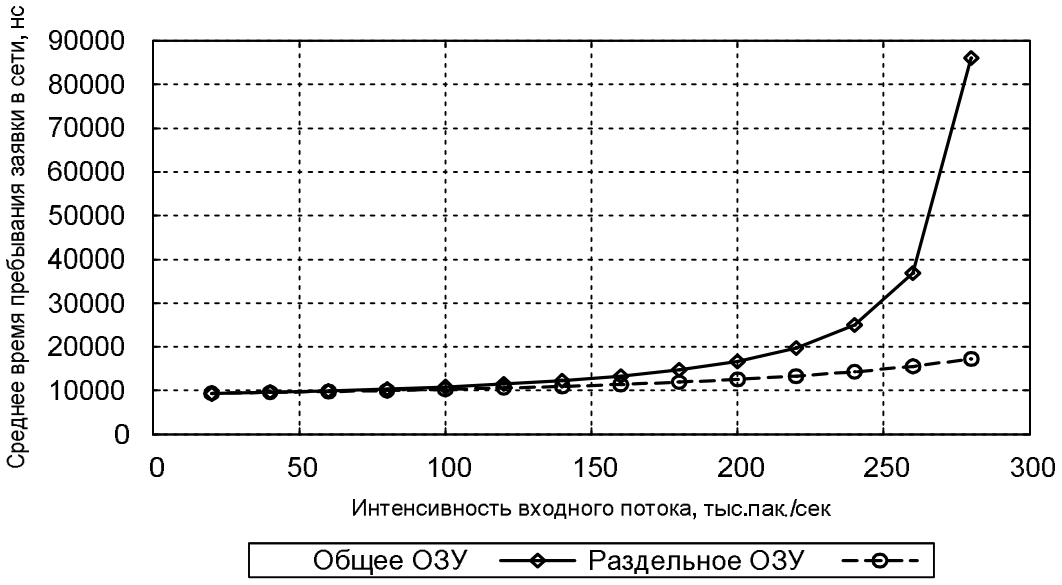

Рисунок 5 – Зависимость среднего времени обработки пакета на маршрутизаторе от интенсивности входного потока

На рисунке 6 показано влияние производительности блоков ОЗУ на время обработки пакета на маршрутизаторе. Как и следовало ожидать, при увеличении производительности блоков памяти разница между двумя моделями уменьшается и становится незначительной уже при 12 млн. запросов в секунду.

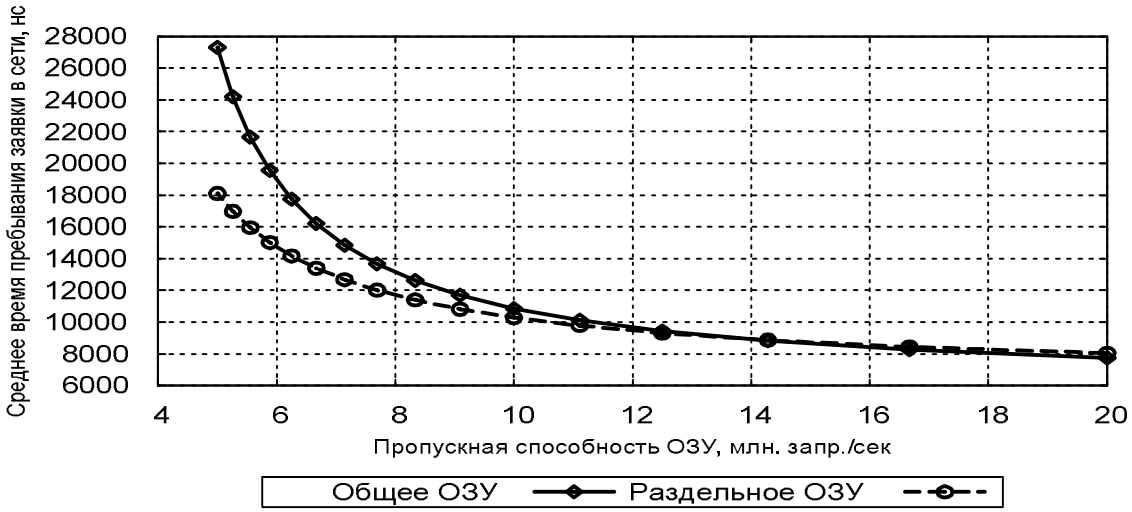

Рисунок 6 – Зависимость среднего времени обработки пакета на маршрутизаторе от производительности блоков ОЗУ

Выводы

Выделение отдельных блоков ОЗУ под код позволяет увеличить производительность сетевого процессора. Увеличение достигает 20% при тактовой частоте процессорных ядер 350 MHz и более. Но если тактовая частота процессора не превышает 300 MHz, то производительность структуры с общим ОЗУ оказывается выше.

Наибольшее влияние разделение ОЗУ оказывает на среднее время обработки пакета. Если в структуре с общим ОЗУ увеличение интенсивности входного потока влекло за собой существенный рост времени обработки пакета, то после выделения отдельных блоков памяти для кода время обработки растет медленнее.

Из полученных результатов следует, что использование структуры с раздельной памятью является эффективным методом повышения производительности сетевых процессоров.

Литература

1. Грищенко В.И. Моделирование маршрутизаторов на многоядерных сетевых процессорах / В.И.Грищенко, Ю.В.Ладыженский // Наукові праці ДонНТУ. Серія: Інформатика, кібернетика і обчислювальна техніка. – 2010. – Вип. 12(165). - С. 169-177.
2. Rosewarne Chris. Network Processors: Evaluating Architectures for Leading Edge Applications. White Paper. Calyptech. 19th March 2004.
3. Ладиженський Ю.В. Моделирование сетевых процессоров пакетной обработки данных / Ю.В. Ладиженський, В.И. Грищенко // Матеріали міжнародної науково-практичної конференції «Інтернет – Освіта – Наука – 2006», м. Вінниця, 10 – 14 жовтня 2006. - Т. 2. - С. 417-422.
4. Samarjit Chakraborty, Simon Kunzli, Lothar Thiele et. al. Performance evaluation of network processor architectures: combining simulation with analytical estimation. Computer Networks: The International Journal of Computer and Telecommunications Networking. Volume 41, Issue 5 (April 2003). 2003. pp. 641 – 665.

5. Gries M., Kulkarni C., Sauer C., Keutzer K. Comparing Analytical Modeling with Simulation for Network Processors: A Case Study. Design Automation and Test in Europe (DATE), Munich, Germany, March 2003.
6. Weng Ning, Wolf Tilman. “Pipelining vs. multiprocessors – choosing the right network processor system topology,” in Proc. of Advanced Networking and Communications Hardware Workshop (ANCHOR 2004) in conjunction with The 31st Annual International Symposium on Computer Architecture (ISCA 2004), Munich, Germany, June 2004.
7. Wolf Tilman, Franklin Mark A. "Performance Models for Network Processor Design," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 6, pp. 548-561, Jun., 2006.
8. Ahmadi M., Wong S. A Performance Model for Network Processor Architectures in Packet Processing Systems, Proceedings of the 19th International Conference on Parallel and Distributed Computing and Systems (PDCS 2007), pp. 176-181, Cambridge, Massachusetts, USA, November 2007.

Надійшла до редакції: 10.12.2010 *Рецензент: канд.техн.наук,проф. Лапко В.В.*

В.І. Грищенко, Ю.В. Ладиженський
Донецький національний технічний університет

Дослідження впливу роздільної пам’яті на продуктівність мережного процесора. У статті розглянуто підходи до моделювання спеціалізованих багатоядерних мережних процесорів пакетної обробки даних. Запропоновано модель мережного процесора з роздільною пам'яттю для коду та даних. Представлені результати моделювання мережного процесора і проведено порівняння ефективності мережних процесорів із загальною і роздільної пам'яттю.
мережний процесор, маршрутизатор, комп'ютерна мережа, пропускна здатність, модель, мережа масового обслуговування

V.I. Grischenko, U.V. Ladyzhensky
Donetsk State Technical University

Research of Separate Memory Influence on Multi-Core Network Processor. In this paper approaches for specialized multi-core network processor modeling are considered. Authors have proposed a model of network processors with separate memory for code and data. The results of network processor modeling are demonstrated and performance of network processors with common and separated memory is compared.
network processor, router, computer network, bandwidth, model, queuing network