

УДК 004.715

В.И. Грищенко, Ю.В. Ладыженский, Моатаз Юнис
Донецкий национальный технический университет
victor.grischenko@gmail.com, ly@cs.dgtu.donetsk.ua,
moatazalyounes@gmail.com

Влияние выделенного кэша команд на производительность сетевого процессора

Рассматриваются подходы к моделированию сетевых процессоров пакетной обработки данных. Предложена модель многоядерного сетевого процессора, в котором объем кэша команд вычислительных ядер достаточен для размещения всего кода выполняемых приложений. Сравнивается производительность сетевых процессоров с различной организацией блоков памяти.

Ключевые слова: сетевой процессор, маршрутизатор, IP-протокол, конвейерная обработка, кэш команд

Введение

Увеличение количества пользователей компьютерных сетей и рост объема информации, передаваемой по сетевым каналам, значительно увеличили требования к пропускной способности маршрутизаторов. Процессоры общего назначения не способны обеспечить необходимую производительность [2]. Поэтому в современных маршрутизаторах используются специализированные сетевые процессоры (СП), которые объединяют в себе производительность аппаратной реализации наиболее часто используемых задач и гибкость программируемых устройств.

Высокая производительность сетевых процессоров необходима для поддержания увеличивающихся требований к пропускной способности компьютерных сетей. В работе [1] представлена модель многоядерного СП с общей памятью для кода и данных, показано, что в исследуемой структуре СП память являлась наиболее нагруженным узлом.

В представленной статье будет рассмотрена модель СП, в которой кэш команд у каждого вычислительного ядра достаточно велик для того, чтобы вместить весь код выполняемой задачи. Таким образом, внешний общий блок ОЗУ будет использоваться только для хранения обрабатываемых данных.

Современное состояние проблемы моделирования СП

Анализ эффективности структур сетевых процессоров является важной научной задачей, которая рассматривается несколькими исследовательскими группами во всем мире [3]. Одновременно разрабатывается несколько направлений анализа производительности СП. В работах [4, 5] предложена модель, в которой узлам сетевого процессора соответствуют кривые количества обслуживания, которое испытывает проходящий через них поток данных. Модель

позволяет в значительной мере варьировать набор и взаимосвязи узлов СП и обеспечивает большое пространство исследуемых структур сетевых процессоров. Однако каждое приложение, выполняемое на узлах сетевого процессора, имеет индивидуальную характеристику объема предоставляемого обслуживания и определение этих характеристик весьма трудоемко. Поэтому исследование большого числа сетевых приложений с помощью описанной модели в значительной мере затруднено.

Методика анализа эффективности выполнения сетевых приложений на многоядерных структурах была рассмотрена в работах [6, 7]. Предложенная методика состоит из двух основных этапов: имитационное моделирование работы приложения на эталонном вычислительном ядре и расчет характеристик работы этого приложения на многоядерном сетевом процессоре. Недостатком используемой аналитической модели является невозможность исследования конвейерных структур и возможностей выполнения нескольких приложений на различных вычислительных ядрах.

В работе [8] рассматривается модель СП, основанная на теории массового обслуживания. В ней анализируется структура с одним ведущим ядром и несколькими подчиненными. Ведущее ядро обрабатывает поток данных и, по мере необходимости, подключает подчиненные ядра, переключая на них обработку части входного потока. Модель также не рассматривает конвейерные структуры и ограничена только параллельным выполнением одного приложения.

Конвейерная обработка может значительно увеличить пропускную способность сетевого процессора [2], поэтому модель сетевого процессора должна предоставлять возможность исследовать подобные структуры. В работе [1] авторами была предложена аналитическая модель сетевого процессора, позволяющая исследовать широкий спектр различных структур СП, включая как конвейерные, так и параллельные. В этой статье

представляется развитие предложенной модели СП, в которой весь код, выполняемый на вычислительном ядре, размещается целиком в кэше команд.

Модель сетевого процессора

Модель сетевого процессора с выделенным кэшем инструкций отличается от модели [1] способом расчета вероятности обращения к ОЗУ вычислительного ядра.

На рисунке 1 представлена структура многоядерного СП с выделенным кэшем команд и общим блоком ОЗУ для хранения данных. Каждое вычислительное ядро состоит из арифметико-логического устройства (АЛУ) и имеет отдельные кэши данных (Кд) и команд (Кк). Ядра разделены на несколько этапов конвейера для увеличения пропускной способности СП.

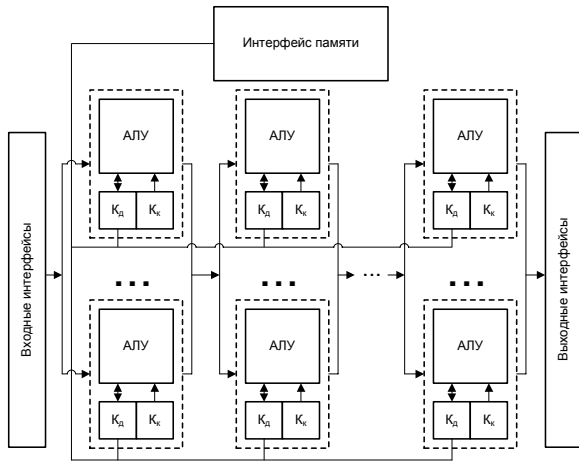


Рисунок 1 – Структура сетевого процессора

После прохождения входных сетевых интерфейсов пакет проходит обработку на первом этапе конвейера. Если этап содержит более одного вычислительного ядра, то пакет передается любому свободному. Если свободных ядер нет, то пакет ожидает завершения обработки предыдущих пакетов. После завершения обработки на первом этапе пакет последовательно обрабатывается на оставшихся этапах и покидает систему через выходные сетевые интерфейсы.

На рисунке 2 показана сеть массового обслуживания, моделирующая работу многоядерного сетевого процессора, представленного на рисунке 1. Узлы $\mu_1 - \mu_n$ моделируют работу этапов конвейера обработки пакетов. Каждый узел может содержать одно и более обслуживающих устройств в зависимости от количества вычислительных ядер выделенных под соответствующий этап конвейера.

Узел μ_{ram} моделирует работу блока ОЗУ. Он имеет одно обслуживающее устройство, и его интенсивность обслуживания зависит от характе-

ристик используемого оборудования.

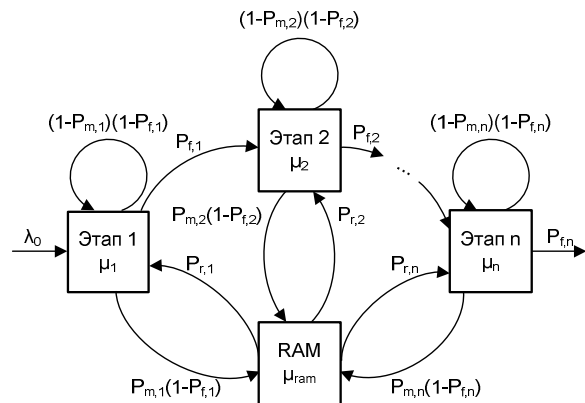


Рисунок 2 – Сеть СМО, моделирующая сетевой процессор

При поступлении пакета на вычислительное ядро выполняется одна инструкция из алгоритма его обработки, после чего, с вероятностью $P_{f,i}$ ($i = \overline{1, n}$) обработка может завершиться. Вероятность $P_{f,i}$ зависит от среднего числа инструкций ($N_{ins,i}$), необходимого для обработки пакета на i -м этапе конвейера:

$$P_{f,i} = \frac{1}{N_{ins,i} + 1} \quad (1)$$

В случае, если обработка не завершилась, то с вероятностью $P_{m,i}$ ($i = \overline{1, n}$) потребуется обращение к оперативной памяти за данными. Вероятность $P_{m,i}$ зависит от интенсивности обращений к ОЗУ и вероятности промаха кэша данных:

$$P_{m,i} = P_{dc,i} P_{d,i} \quad (2)$$

где, $P_{d,i}$ – вероятность обращений за данными, во время обработки пакета на i -м этапе конвейера, $P_{dc,i}$ – вероятность промаха кэша данных. Поскольку кэш команд полностью вмещает весь код обработки пакета, то вероятность промаха кэша команд равна нулю.

Вероятность $P_{d,i}$ может быть рассчитана как отношение обращений к ОЗУ за время обработки одного пакета ($N_{d,i}$) к общему количеству инструкций, выполненных за время его обработки ($N_{prc,i}$):

$$P_{d,i} = \frac{N_{d,i}}{N_{prc,i}} \quad (3)$$

После завершения обращения к ОЗУ, обработка пакета продолжается на том же этапе конвейера, на котором она была прервана для обращения к памяти. Поскольку все этапы конвейера используют один модуль ОЗУ, то необходимо определить, какая часть выходного потока ОЗУ адресуется каждому отдельному

етапу конвейера. Поскольку входной поток от i -го этапа конвейера к ОЗУ равен $P_{m,i}(1-P_{f,i})$, вероятность перехода пакета из ОЗУ на i -й этап конвейера ($P_{r,i}$) определяется по формуле:

$$P_{r,i} = \frac{P_{m,i}(1-P_{f,i})}{\sum_{k=1}^n P_{m,k}(1-P_{f,k})} \quad (4)$$

В случае, если пакет не завершил обработки, и не было осуществлено обращение к ОЗУ, то он, с вероятностью $(1-P_{m,i})(1-P_{f,i})$, вновь поступает в очередь обработки этого же этапа конвейера.

По завершению обработки с вероятностью $P_{f,i}$ ($i=1, n-1$) пакет передается следующему этапу конвейера. Выходной поток $P_{f,n}$ моделирует исходящий поток обработанных пакетов.

Пусть входной поток пакетов для сети СМО, представленной на рисунке 2, распределен по закону Пуассона с интенсивностью λ_0 , тогда можно записать систему уравнений:

$$\begin{cases} \lambda_0 = \lambda_n P_{f,n} \\ \lambda_i = \lambda_{i-1} P_{f,i-1} + \lambda_i (1-P_{m,i})(1-P_{f,i}) + \lambda_{ram} P_{r,i}, i = \overline{1, n} \\ \lambda_{ram} = \sum_{i=1}^n \lambda_i P_{m,i} (1-P_{f,i}), \end{cases} \quad (5)$$

где уравнения для λ_i ($i = \overline{1, n}$) заданы для всех этапов конвейера, а λ_{ram} описывает входной поток блока ОЗУ.

Определим интенсивности обслуживания для всех узлов системы. Будем предполагать, что все вычислительные ядра одинаковы и могут обслуживать только одну инструкцию за такт. Тогда интенсивность обслуживания μ_i равна тактовой частоте процессора, выраженной в герцах (f_{pr}):

$$\mu_i = f_{pr} \quad (6)$$

Допустим, что блок ОЗУ не может обслуживать несколько запросов одновременно, тогда его интенсивность обслуживания будет равна

$$\mu_{ram} = \frac{1}{t_{ram}}, \quad (7)$$

где t_{ram} - среднее время выполнения операции чтения/записи для ОЗУ.

Расчет характеристик сетевого процессора

В таблице 1 представлены параметры, необходимые для расчета характеристик сети СМО.

Подставив значения из таблицы 1 в систему (5) получим систему уравнений, описывающую структуру сетевого процессора:

$$\begin{cases} \lambda_0 = \lambda_3 P_{f,3} \\ \lambda_1 = \lambda_0 + \lambda_1 (1-P_{m,1})(1-P_{f,1}) + \lambda_{ram} P_{r,1} \\ \lambda_2 = \lambda_1 P_{f,1} + \lambda_2 (1-P_{m,2})(1-P_{f,2}) + \lambda_{ram} P_{r,2} \\ \lambda_3 = \lambda_3 P_{f,2} + \lambda_3 (1-P_{m,3})(1-P_{f,3}) + \lambda_{OЗУ} P_{r,3} \\ \lambda_{ram} = \sum_{i=1}^3 \lambda_i P_{m,i} (1-P_{f,i}). \end{cases} \quad (8)$$

Таблица 1 – Параметры моделируемого сетевого процессора

Параметр	Значение	Описание
μ_1, μ_2, μ_3	от $5 \cdot 10^7$ до $6 \cdot 10^8$ ком./с	Производительность вычислительных ядер, выраженная в количестве команд, выполняемых в секунду
$\mu_{OЗУ}$	от $5 \cdot 10^7$ до $1 \cdot 10^9$ зап./с	Производительность блока ОЗУ, выраженная в количестве обрабатываемых за секунду запросов
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
$P_{dc,1}$	0.01	Вероятности промахов кэша данных для ядра 1, выполняющего разбор заголовка пакета
$P_{dc,2}$	0.25	Вероятности промахов кэша данных для ядра 2, выполняющего поиск в маршрутной таблице
$P_{dc,3}$	0.01	Вероятности промахов кэша 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

Разделив обе части каждого уравнения из (7) на λ_0 , получим систему для коэффициентов

посещений $\alpha_i, i = \overline{1, n}$ и α_{ram} :

$$\begin{cases} I = \alpha_3 P_{f,3} \\ \alpha_1 = I + \alpha_1(1 - P_{m,1})(1 - P_{f,1}) + \alpha_{ram} P_{r,1} \\ \alpha_2 = \alpha_1 P_{f,1} + \alpha_2(1 - P_{m,2})(1 - P_{f,2}) + \alpha_{ram} P_{r,2} \\ \alpha_3 = \alpha_3 P_{f,2} + \alpha_3(1 - P_{m,3})(1 - P_{f,3}) + \alpha_{ram} P_{r,3} \\ \alpha_{ram} = \sum_{i=1}^3 \alpha_i P_{m,i}(1 - P_{f,i}). \end{cases} \quad (9)$$

Решив систему (8), получаем коэффициенты посещения для всех узлов сети массового обслуживания, моделирующую маршрутизатор (см. рис. 2):

$$\begin{aligned} \alpha_1 &\approx 170 & \alpha_2 &\approx 264 & \alpha_3 &\approx 101 \\ \alpha_{ram} &\approx 3.72 \end{aligned} \quad (10)$$

Результаты моделирования

Главным критерием оценки производительности сетевого процессора является его пропускная способность. Величина максимальной пропускной способности маршрутизатора λ_0^{max} определялась из условия существования установившегося режима для сети массового обслуживания:

$$\lambda_0^{max} = \min \left(\frac{S_{ram} \mu_{ram}}{\alpha_{ram}}, \min_{i=\overline{1, n}} \left(\frac{S_i \mu_i}{\alpha_i} \right) \right). \quad (11)$$

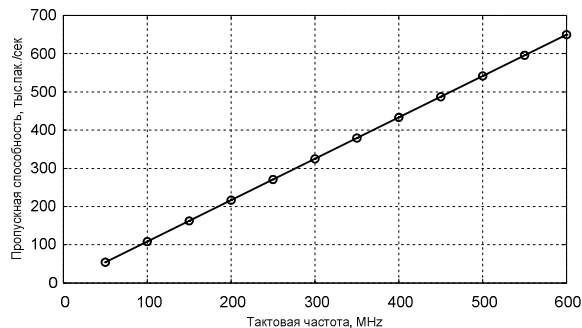


Рисунок 3 – Зависимость максимальной пропускной способности маршрутизатора от тактовой частоты вычислительных ядер

В рассматриваемом случае n равно трем, S_i , μ_i , S_{ram} и μ_{ram} приведены в таблице 1, а α_i и α_{ram} были определены из решения системы (9). На рисунке 3 показана зависимость пропускной способности маршрутизатора λ_0^{max} от производительности вычислительных ядер, выраженной в их тактовой частоте.

Из графика видно, что зависимость имеет линейный характер. Это означает, что узким местом маршрутизатора являются вычислительные ядра.

Рисунок 4 демонстрирует зависимость пропускной способности маршрутизатора λ_0^{max} от

производительности блока оперативной памяти μ_{ram} . Из графика видно, что изменение скорости обращения к оперативной памяти не влияет на производительность СП. Это подтверждает вывод о том, что узким местом в исследуемой структуре сетевого процессора являются вычислительные ядра.

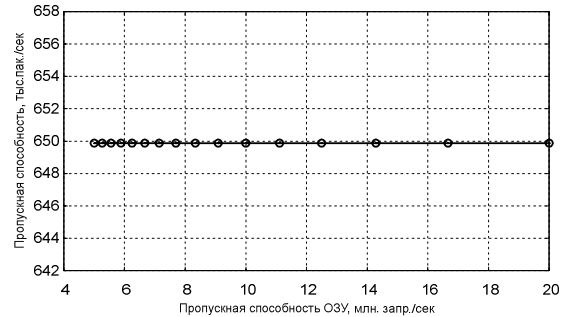


Рисунок 4 – Зависимость пропускной способности маршрутизатора от производительности блока ОЗУ

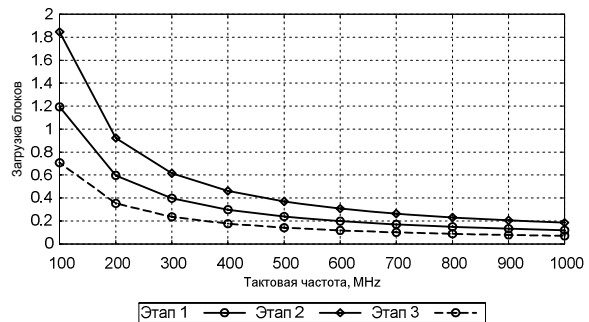


Рисунок 5 – Зависимость загрузки вычислительных ядер от тактовой частоты

Загрузка узлов сетевого процессора определяется как среднее количество занятых приборов:

$$\bar{s} = s - \bar{\rho}, \quad (12)$$

где s – число обслуживающих приборов на узле, а $\bar{\rho}$ – количество свободных приборов, которое вычисляется как

$$\bar{\rho} = \sum_{i=0}^{s-1} (s-i) P_i, \quad (13)$$

где P_i – вероятность пребывания в системе i заявок.

На рисунке 5 показана зависимость загрузки вычислительных ядер от тактовой частоты. Видно, что наиболее нагружен второй этап конвейера, а наименее – третий. С увеличением тактовой частоты различия в загрузке конвейеров процессора становятся незначительными.

Средняя длина очереди к каждому узлу сетевого процессора определяется по формуле:

$$\bar{v} = \frac{(\lambda/\mu)^{s+1}}{(s-1)!(s - (\lambda/\mu))^2} P_0, \quad (14)$$

где P_0 – вероятность отсутствия в системе требований.

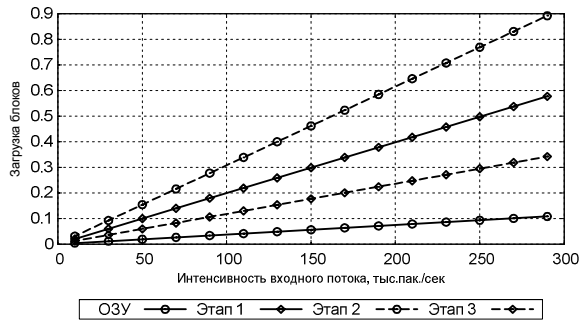


Рисунок 6 – Зависимость загрузки узлов сетевого процессора от интенсивности входного потока

Рисунок 6 демонстрирует зависимость загрузки узлов СП от интенсивности входного потока. В исследуемой структуре модуль ОЗУ оказывается наименее нагруженным. Это означает, что уменьшение производительности этого блока, а, следовательно, и его стоимости, не окажет существенного влияния на производительность всей системы.

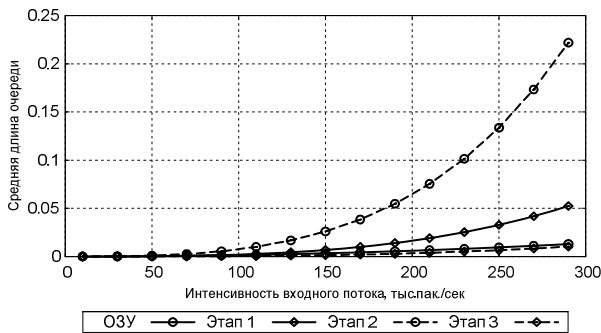


Рисунок 7 – Зависимость средних длин очередей от интенсивности входного потока

Из рисунка 7 видно, что длиннее всего очередь к наиболее нагруженному второму этапу конвейера, а очереди к блоку ОЗУ и третьему этапу конвейера остаются незначительными даже при максимальной интенсивности входного потока. Это еще раз указывает на необходимость повышения производительности именно этого узла.

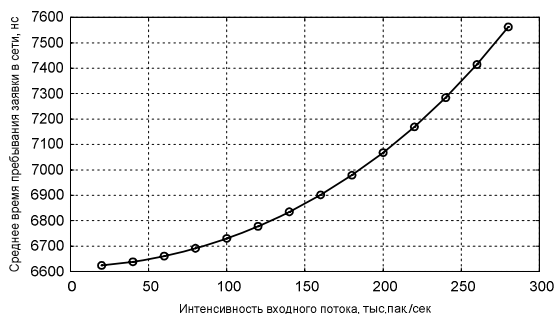


Рисунок 8 – Зависимость среднего времени обработки пакета на СП от интенсивности входного потока

Среднее время обработки пакета определяется как общее время пребывания заявки в сети массового обслуживания (u_t), зависит от времени пребывания в каждом узле (u_i) и коэффициентов посещения узлов (α_i):

$$\bar{u}_t = \alpha_{ram} \bar{u}_{ram} + \sum_{i=1}^n \alpha_i \bar{u}_i \quad (15)$$

Среднее время пребывания заявки в узле вычисляется по формуле:

$$\bar{u} = \frac{\bar{n}}{\lambda} \quad (16)$$

где \bar{n} – среднее число заявок в узле, которое зависит от средней длины очереди, интенсивности входного потока и интенсивности обслуживания:

$$\bar{n} = \bar{v} + \frac{\lambda}{\mu} \quad (17)$$

На рисунке 8 показана зависимость среднего времени обработки пакета от интенсивности входного потока. Видно, что с увеличением интенсивности незначительно растет и время обслуживания пакета. При увеличении входного потока в 14 раз время обслуживания пакета увеличивается всего на 15%. Это говорит о высокой устойчивости исследуемой системы к нагрузке.

Сравнение структур сетевого процессора

Сравним три структуры сетевых процессоров: структуру с общим блоком ОЗУ для кода и данных, рассмотренную в работе [1]; структуру с выделенным блоком кода для каждого этапа конвейера; структуру с большим кэшем команд, достаточным для размещения всего кода сетевого приложения, представленную в данной статье.

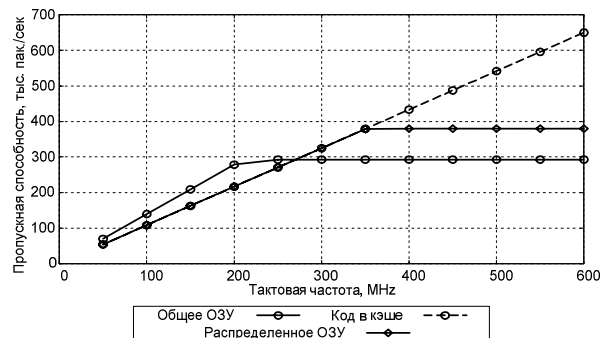


Рисунок 9 – Сравнение максимальной пропускной способности СП в зависимости от тактовой частоты вычислительных ядер

На рисунке 9 сравнивается максимальная производительность СП трех разных архитектур в зависимости от тактовой частоты вычислительных ядер. Из графиков видно, что использование большого кэша команд дает максимальную производительность при больших тактовых

частотах, но до 350 Mhz различие между большим кэшем команд и распределенным ОЗУ незначительно.

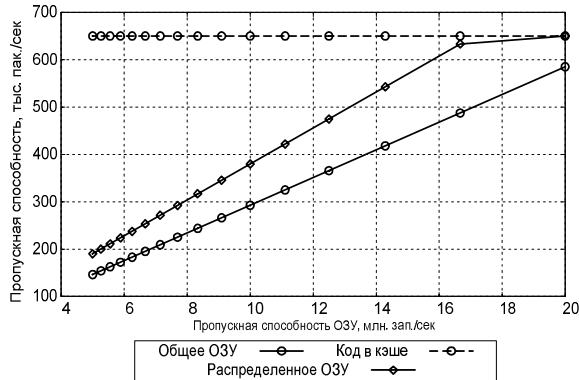


Рисунок 10 – Сравнение максимальной пропускной способности СП в зависимости от производительности блока ОЗУ

На рисунке 10 приведено сравнение пропускной способности в зависимости от производительности блока ОЗУ. Как было показано ранее, в структуре с большим кэшем команд пропускная способность СП не зависит от производительности блока ОЗУ. В то же время, только структура с распределенным ОЗУ способна достигнуть подобной производительности и только в случае использования наиболее производительных блоков ОЗУ. Производительность структуры с общим ОЗУ незначительно отстает от производительности структуры с распределенным ОЗУ и на предельных значениях производительности блоков памяти приближается к структуре с большим кэшем данных.

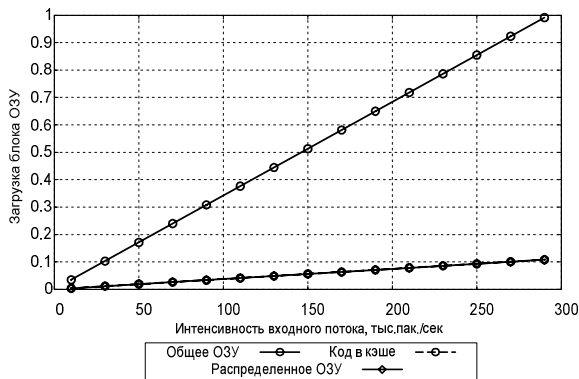


Рисунок 11 – Сравнение загрузки блока ОЗУ в зависимости от интенсивности входного потока

Графики загрузки ОЗУ (см. рисунок 11) подтверждают предположение о значительном снижении нагрузки на этот блок. Интенсивность использования ОЗУ в структуре с распределенной памятью и в структуре с большим кэшем команд одинаковы. Это объясняется тем, что в обеих структурах внешний блок памяти хранит только данные и количество обращений к нему одинаково.

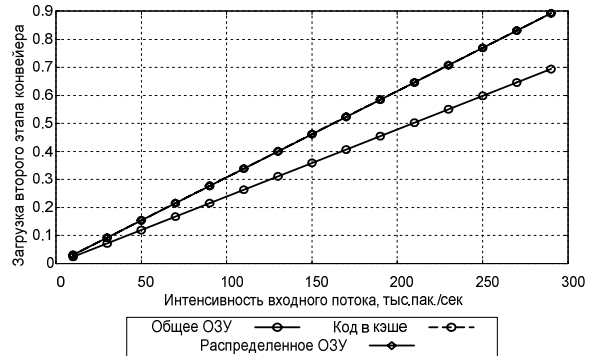


Рисунок 12 – Сравнение загрузки второго этапа конвейера в зависимости от интенсивности входного потока

В противоположность ситуации с блоком оперативной памяти интенсивность загрузки второго этапа конвейера (см. рисунок 12) выросла при использовании распределенной памяти и большого кэша команд. Причем величина нагрузки для обеих структур практически не отличается на всем диапазоне исследуемых значений интенсивности входного потока.

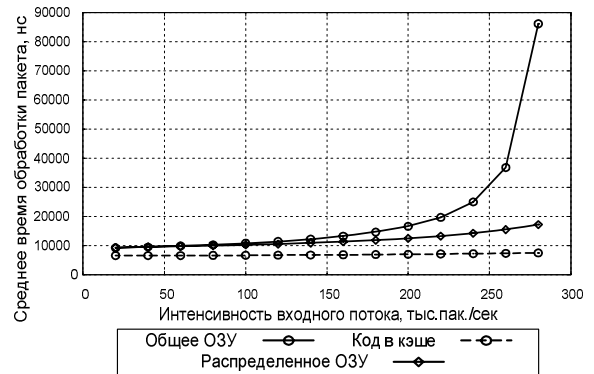


Рисунок 13 – Сравнение среднего времени обработки пакета в зависимости от интенсивности входного потока

На рисунке 13 показано сравнение среднего времени обслуживания пакета в зависимости от интенсивности входного потока. Скорость обработки пакета в структуре с большим кэшем команд практически не уменьшается при увеличении входного потока. Чуть хуже себя ведет структура с распределенной памятью, и больше всего замедляется система с общим ОЗУ для кода и команд.

Выводы

В представленной статье исследована производительность структуры сетевого процессора с кэшем команд достаточно большим для размещения всего программного кода. Моделирование показало, что использование подобной структуры дает существенный прирост производительности и делает многоядерный сетевой процессор более устойчивым к нагрузкам.

Отрицательной стороной такой структуры является рост площади кристалла процессора и, как следствие, увеличение его энергопотребления.

Компромиссом может оказаться структура с распределенной памятью. В ней каждому этапу конвейера обработки пакета выделена собственный блок памяти для хранения кода. Такой подход позволяет исключить избыточное хране-

ние одинакового кода в каждом ядре процессора и уменьшить площадь кристалла. Отрицательной стороной такой структуры по сравнению с большим кэшем команд является падение производительности на высоких тактовых частотах вычислительных ядер и некоторое увеличение времени обслуживания пакетов.

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. Chris Rosewarne. Network Processors: Evaluating Architectures for Leading Edge Applications. White Paper. Calyptech. 19th March 2004.
3. Ладыженський Ю.В. Моделирование сетевых процессоров пакетной обработки данных / Ю.В. Ладыженский, В.И. Грищенко // Матеріали міжнародної науково-практичної конференції «Інтернет – Освіта – Наука – 2006», м. Вінниця, 10 – 14 жовтня р. – 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. M. Gries, C. Kulkarni, C. Sauer, K. Keutzer. Comparing Analytical Modeling with Simulation for Network Processors: A Case Study. Design Automation and Test in Europe (DATE), Munich, Germany, March 2003.
6. Ning Weng and Tilman Wolf, "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. Tilman Wolf, Mark A. Franklin, "Performance Models for Network Processor Design," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 6, pp. 548-561, Jun., 2006.
8. M. Ahmadi, S. Wong, 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 08.02.2011