

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ПГУ)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ. ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

Сборник научных статей
VI Всероссийской межвузовской
научно-практической конференции

г. Пенза, 13 марта 2019 г.

Под редакцией Л. Р. Фионовой

Пенза • Издательство ПГУ • 2019

Информационные технологии в науке и образовании.

И74 Проблемы и перспективы : сб. науч. ст. VI Всерос. межвуз. науч.-практ. конф. (г. Пенза, 13 марта 2019 г.) / под ред. Л. Р. Фионовой. – Пенза : Изд-во ПГУ, 2019. – 408 с.

ISBN 978-5-907185-15-9

Материалы конференции посвящены теоретическим и практическим вопросам информационных технологий в науке и образовании. Рассмотрены основные проблемы и перспективы развития информационных компьютерных технологий в системах электронного документооборота; вычислительных системах, сетях и комплексах; современных информационно-коммуникационных технологиях; аналитических и численных методах моделирования; информационной безопасности систем и технологий; приборостроении, электронике и энергетике; машиностроении и транспорте.

Издание подготовлено и опубликовано при содействии Федеральной целевой программы «Исследования и разработки научно-технологического комплекса России на 2014–2020 гг.». Предназначено для широкого круга читателей, интересующихся развитием информационных технологий и вычислительной техники в науке и образовании.

УДК 004 + 001 + 37

Издание включено в Российский индекс научного цитирования
(РИНЦ) – elibrary.ru

ПРИКАЗ

о проведении Всероссийской межвузовской научно-практической конференции
«Информационные технологии в науке и образовании. Проблемы и перспективы»
№ 103/0 от 15.11.2018

ISBN 978-5-907185-15-9

© Пензенский государственный
университет, 2019

Tsung это полнофункциональное решение для тестирования производительности с поддержкой современных протоколов, таких как s HTTP, WebDAV, SOAP, PostgreSQL, MySQL, LDAP, MQTT и Jabber/XMPP сервисы [1, 2, 5]. Tsung работает на Linux/Unix, а архитектура на основе Erlang позволяет моделировать нагрузку в тысячи виртуальных пользователей. Инструмент для записи тестов поддерживает HTTP и Postgres. Tsung собирает не только метрики с тестируемого приложения, но и системные метрики такие как загрузка CPU, количество виртуальной памяти. Гибкие сценарии загрузки позволяют определять и объединять любое количество моделей нагрузки в одном тесте. Тесты могут использовать внешние источники данных, такие как файлы и базы данных. Все операции происходят с использованием bash-скриптов. Tsung, имеет более крутую кривую обучения, по сравнению с другими бесплатными инструментами.

В обзор не попали Tarus и Yandex Tank, так как они являются высокоуровневыми надстройками над другими тестовыми инструментами, хотя Yandex Tank и предоставляет свои собственные модули измерения нагрузки.

Подводя итоги можно сказать, что JMeter остается лидером среди бесплатных инструментов тестирования предоставляя обширный инструментарий и высокую производительность [6]. Существует также довольно много коммерческих и не коммерческих решений таких как например Tarus, которые позволяют запускать тесты JMeter.

Библиографический список

1. The Ultimate Guide to Load Testing Tools. – URL: <https://www.loadview-testing.com/load-testing-tools/> (дата обращения: 20.01.2019).
2. Top Performance Testing Tools. – URL: <https://www.indiumsoft.com/Blog/topperformance-testing-tools/> (дата обращения: 20.01.2019).
3. Top 5 Open Source Performance Testing Tools. – URL: <https://www.dotcom-tools.com/blog/open-source-performance-testing-tools/> (дата обращения: 01.01.2019).
4. Обзор инструментария для нагрузочного и перформанс-тестирования. – URL: <https://habr.com/ru/company/jugru/blog/337928/> (дата обращения: 20.01.2019).
5. Open Source Load Testing Tools: Which One Should You Use? – URL: <https://www.blazemeter.com/blog/open-source-load-testing-tools-which-one-should-you-use> (дата обращения: 20.01.2019).
6. What are the best Performance Testing Tools? – URL: <https://blogs.msdn.microsoft.com/testingspot/2018/08/01/what-are-the-best-performance-testing-tools/> (дата обращения: 20.01.2019).

И. Ю. Фролов, Д. В. Патунин, Н. Н. Коннов

РЕАЛИЗАЦИЯ НА ПЛИС АЛГОРИТМА ДИНАМИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В КОММУТАТОРЕ ETHERNET

Пензенский государственный университет, г. Пенза, Россия

В коммутаторах Ethernet с поддержкой качества обслуживания важным механизмом обеспечения требуемых параметров продвижения трафика является дис-

петчеризация очередей кадров, направляемых в каждый выходной порт. Механизм диспетчеризации заключается в правиле выбора обслуживаемой очереди, а так же регулировании выделяемой каждому классу полосы пропускания выходного канала. Эффективность диспетчеризации определяет задержку передачи кадров и ее вариацию (джиттер), значения которых важны при передаче трафика реального времени через сети с пакетной коммутацией [1].

Кадры, продвигаемые коммутатором, помечаются метками класса трафика, базируясь на которые он помещает кадры в очереди, соответствующие указанному в метке классу. Для управления очередями широко используется взвешенные дефицитные циклические алгоритмы диспетчеризации (DRR – Deficit Round Robin), особенностью которых является использование циклического приоритета при выборе обслуживаемой очереди, а длительность непрерывного обслуживания очереди определяется кредитом (допустимым количеством данных, считываемых их каждой очереди за сеанс ее обслуживания), значение которого пропорционально выделяемой для данного класса трафика полосы пропускания выходного канала. Кредит задается в байтах, его значение должно быть не меньше максимальному размеру кадра. При этом величина кредита на новый цикл обслуживания по завершению очередного цикла корректируется (уменьшается) на величину дефицита - превышения кредита в завершенном цикле обслуживания из-за передачи последнего кадра (размер кадра не может быть уменьшен).

Недостатком DRR алгоритма является неконтролируемая величина джиттера коммутируемых кадров из-за разброса времени ожидания кадров в различных очередях и зависимость задержки кадров определенного класса и размера соответствующей ему очереди от выделенной для данного класса полосы пропускания. В результате чего DRR алгоритмы не позволяют эффективно перераспределять недогруженные полосы пропускания, что приводит к увеличению задержки кадров и перегрузки очередей в каналах с малой полосой пропускания при пульсирующем характере трафика.

Для решения этой проблемы был предложен алгоритм динамической диспетчеризации, обеспечивающий одновременно справедливое обеспечение выделяемой для каждого класса трафика полосы пропускания и уменьшение «пачечности»[2,3]. Это достигается сочетанием принципов циклической диспетчеризации и учета времени ожидания обслуживания кадров при наложении ограничений на длительности непрерывного обслуживания каждой очереди при сохранении принципов управления полосой, свойственных DRR алгоритму.

Диспетчер определяет номер обслуживаемой очереди по циклическому приоритету, при этом объем данных, передаваемых в канал из каждой очереди за один сеанс обслуживания, определяется установленным кредитом, пропорциональным выделяемой для соответствующего класса трафика полосе пропускания канала. Сеанс обслуживания каждой очереди разбивается на отдельные подсеансы заданной длительности, причем по окончании каждого подсеанса выполняется передача одного разделительного кадра, выбранного из всех непустых очередей по критерию наибольшей длительности нахождения в очереди. При условии необходимости дальнейшего обслуживания очереди в соответствии с установленным кредитом, после передачи разделительного кадра запускается следующий подсеанс ранее начатого сеанса обслуживания очереди.

В целях оценки временных характеристик и сложности аппаратурной реализации динамического алгоритма было выполнено его моделирование в виде IP ядра ПЛИС, обслуживающего один выходной порт 4 канального коммутатора Ethernet, обслуживающего четыре класса трафика.

Буфер кадров реализован в виде ОЗУ емкостью 32К 32-разрядных слов, причем на очередь каждого класса выделяется по 8 К слов.

На вход коммутатора передаются кадры, которые записываются в четыре очереди. Кадры снабжаются метками, указывающими на размер кадра (11 бит), время его поступления в очередь (16 бит), адрес кадра в ОЗУ (13 бит), а также строб готовности очереди к обработке. Все метки записываются в четыре буфера FIFO для последующей обработки в диспетчере и КПДП (см. рисунок).

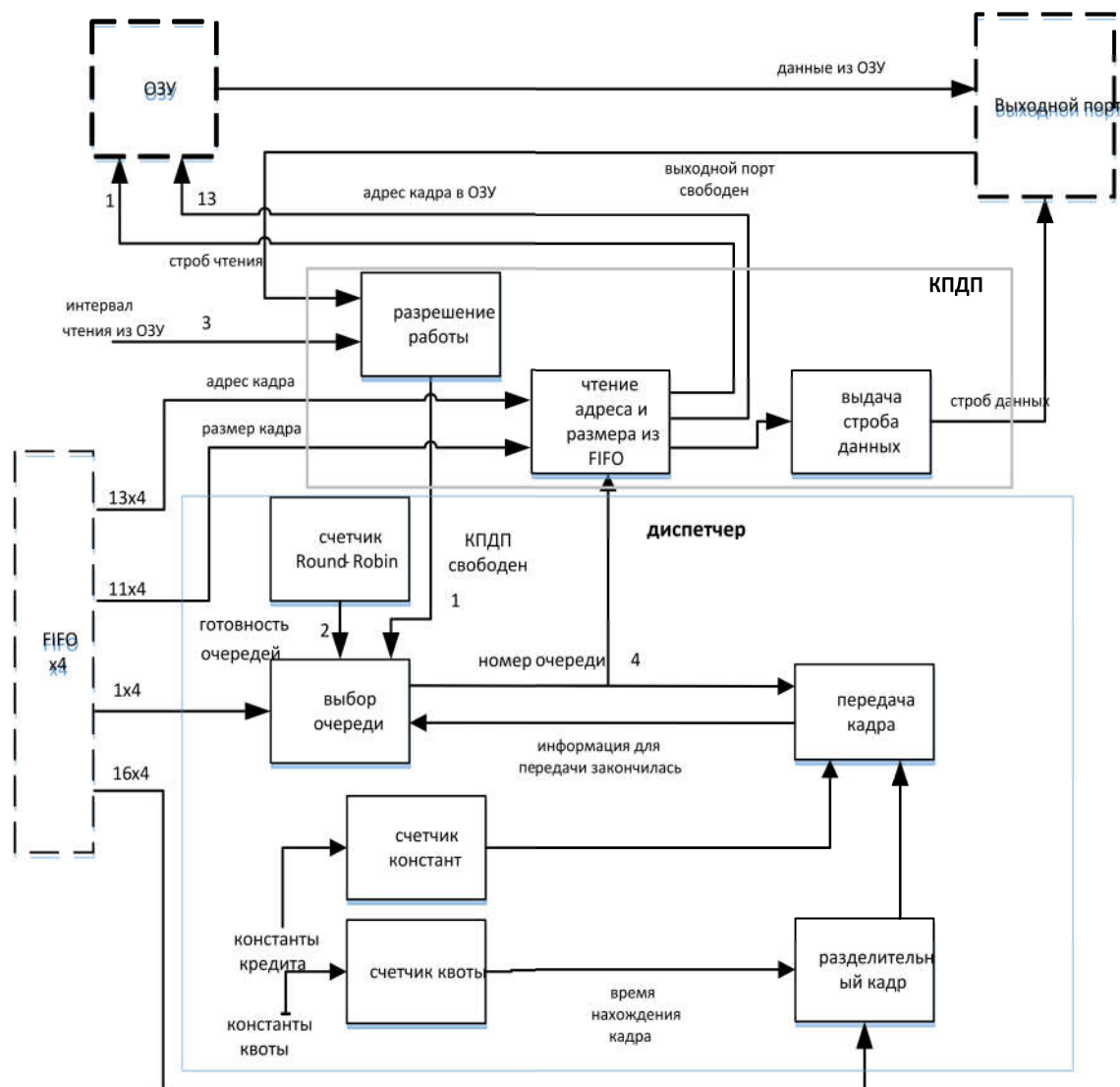


Рисунок. Структурная схема устройства

На вход диспетчера со всех FIFO приходят метки времени нахождения кадра в очереди (16 бит) и строб готовности к обработке. Также на вход поступает сигнал готовности работы КПДП, синхросигнал и строб сброса устройства.

При запуске коммутатора константами задается значение кредита для каждой очереди в байтах; квота в байтах, ограничивающая длительность подсеанса; начальное значение циклического приоритета.

Работу диспетчера регулирует конечный автомат.

1. Автомат изначально находится в состоянии ожидания, а при появлении готовых к обработке очередей и освобождении выходного канала начинается работа диспетчера.