

лить системе обрабатывать данные на более высоких скоростях. Тем не менее, большие размеры партии также увеличивают интервал задержки между «приходом» данных и получением результатов. Таким образом, необходимо, чтобы система работала с размером пакета, который сводит к минимуму задержки, обеспечивая при этом незамедлительную обработку данных. Кроме того, желаемый размер пакета изменяется в зависимости от скорости передачи данных и других условий эксплуатации. Таким образом, статически заданный размер пакета может либо вызвать излишне высокий интервал задержки при низкой нагрузке или может не справиться со всплеском скорости передачи данных, в результате чего система может дестабилизироваться.

Для решения этих проблем, предлагается адаптивный алгоритм, который позволяет системе автоматически варьировать размер пакета при изменении условий эксплуатации. Пропускная способность потоковой нагрузки может вести себя нелинейно по отношению к размеру пакета. Несмотря на это, при любом уровне нагрузки алгоритм должен быть в состоянии быстро адаптироваться к изменениям и обеспечить низкий интервал задержки. Кроме того, алгоритм должен быть устойчивым по отношению к условиям эксплуатации, которые возникают из непрерывных изменений в скорости передачи данных, имеющихся ресурсах, и т.д.



Рис. 1 – Модель целевой системы с модулем управления

Для того, чтобы динамически адаптировать размер пакета потоковой нагрузки, модуль управления системы обработки должен непрерывно оценивать рабочее состояние и, соответственно, увеличивать или уменьшать размер пакета. Такой алгоритм должен иметь следующие свойства:

*Малый интервал задержки.* Алгоритм управления должны быть в состоянии достичь низкого интервала задержки пакетов и обеспечивать стабильности системы.

*Ловкость.* Алгоритм управления должны быть в состоянии быстро обнаружить любые изменения в операционных условиях и быстро адаптироваться к ним.

*Универсальность.* Алгоритм должен быть применим к любому виду пакетной обработки системы и быть в состоянии справиться с любой нагрузкой. Это важно, т.к. разработчикам нельзя быть озабоченными поведением своих рабочих нагрузок и применимостью алгоритма управления на их рабочие нагрузки.

*Простота в конфигурации.* Алгоритм управления может иметь параметры, которые могут потребоваться настройка в зависимости от предполагаемых условий эксплуатации. Количество параметров должны быть небольшими и должны быть легко настраиваемыми.

Таким образом, как и изображено на рисунке 1, на основе статистики о ходе обработки данных и показателей текущего состояния системы адаптивный алгоритм может определить оптимальный размер пакета пересылаемых данных, который будет поставлен в очередь обработки пакетов.

Список использованных источников:

1. L. Amini, N. Jain, A. Sehgal, J. Silber, and O. Verscheure. Adaptive control of extreme-scale stream processing systems. In Distributed Computing Systems // ICDCS 2006. 26th IEEE International Conference IEEE, 2006 – 71 с.
2. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. // Communications of the ACM, 2008 – 113с.

## БИЛЛИНГ-СИСТЕМА ЛОКАЛЬНОЙ СЕТИ ОБЩЕЖИТИЙ БГУИР

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Асташкевич М.Г.

Лецев А. Е. – ассистент кафедры информатики

Несмотря на постоянное развитие информационных технологий, а также увеличение и удешевление вычислительных мощностей, многие привычные нам сферы деятельности до сих пор остаются неавтоматизированными и требуют для выполнения некоторых простейших действий траты непропорционально большого количества времени. В частности это касается многих сфер жизни, связанных с документооборотом и работой с клиентами. Современный уровень развития технологий может позволить оптимизировать и ускорить выполнение таких операций, как прием оплат за те

или иные услуги, составление отчетов, обработка большого количества данных. Такую задачу предстояло решить автору в ходе работы в ОСП МЦ БГУИР (далее - МЦ), предоставляющего в числе прочего услуги пользования локальной сетью общежитий БГУИР студентам.

Локальная сеть общежитий БГУИР №1 и №2 изначально строилась студентами на добровольных началах. В то время проблемы автоматизации процессов управления не возникало из-за небольшого количества активных пользователей. Однако после введения в эксплуатацию третьего и четвертого общежитий, когда количество активных пользователей резко увеличилось почти в два раза. Тогда необходимость внесения изменений в действующий процесс работы с пользователями стала очевидной.

Вкратце процесс пользования сетью для пользователя можно описать следующим образом: чтобы зарегистрироваться в сети, каждый пользователь должен распечатать договор, заполнить его и лично отнести системному администратору общежития, чтобы тот вручную добавил его данные в систему. Затем нужно оплатить пользование сетью, для чего в свою очередь нужно в один из рабочих дней прийти в один из нескольких пунктов распечатки университета в рабочее время, где заполнить от руки бланк оплаты, произвести оплату и получить чек. Несколько раз в неделю кассир лично относит бланки об оплате в бухгалтерию МЦ, где эта информация также вручную добавляется в систему.

Этот процесс может занять до двух недель в зависимости от суммарного количества оплат и наличия у бухгалтеров других задач (например, отчетность). В виду ручной обработки, информация об оплате иногда теряется, и в таком случае пользователю нужно идти в бухгалтерию и лично решать вопрос о зачислении оплаты. Также не налажен процесс переезда пользователя в другое общежитие, в связи с чем при переезде нужно заново регистрироваться в сети другого общежития, переносить оплаты из одной учетной записи в другую, несмотря на то, что по сути все общежития БГУИР объединены в общую локальную сеть.

Разработанный автором программный комплекс является полностью автоматизированной системой контроля пользования услугами предоставления доступа к сети пользователям всех общежитий БГУИР, которая принимает оплаты в электронном виде и практически полностью исключает участие человека в процессах регистрации пользователей и приема оплат. Архитектура приложения подразумевает наличие головного сервера, который отвечает за сбор и обработку информации о произведенных платежах, регистрацию и авторизацию пользователей и ведение статистики, а также небольшие клиент-серверных приложений-сателлитов, которые предоставляют интерфейс управления пользователями определенного сегмента сети. Данная архитектура позволяет сконцентрировать взаимодействие пользователей и администраторов с базой данных пользователей на работе с одним приложением, вне зависимости от общежития, вести общий учет пользователей по всем общежитиям, но при этом предоставляет возможность относительно легко добавлять новые общежития или другие сегменты сети в систему, в том числе удаленные.

Таким образом разработанное приложение позволяет иметь единую точку авторизации и приема оплат для всех пользователей, что решает проблему смены общежития пользователем. Кроме того благодаря приложениям-сателлитам, разработанным для каждого общежития отдельно, и работающим в режиме реального времени, есть возможность устанавливать достаточно жесткие, но справедливые ограничения на работу пользователей; в частности одна учетная запись может быть активной только в одном сегменте сети, что исключает одновременное использование одной учетной записью людьми из разных общежитий. Также полностью решена проблема личного присутствия пользователя в том или ином месте при регистрации или произведения оплаты: регистрация происходит в режиме онлайн с помощью веб-приложения, а оплаты принимаются через платежную систему ЕРИП. Кроме того разработанное приложение может быть адаптировано для автоматизации других процессов, связанных со студентами, проживающими в общежитиях: обработка заявок пользователей на устранение неисправностей системным администраторам, столяру, сантехнику или электрику; прием оплат за велопарковку; организация выполнения студентами дежурств и общехозяйственных работ, а также организация доступа студентов в общежитие по электронными пропусками при наличии соответствующего оборудования.

Список использованных источников:

1. Дмитрий Котеров, Алексей Костарев. PHP 5 в подлиннике / Алексей Костарев Дмитрий Котеров. — 2-е изд. — BHV (Россия), 2008. — June. — 1104 с.
2. Дюбуа, Поль. MySQL / Поль Дюбуа. — 3-е изд. — «Вильямс», 2006. — 1168 с.

## **ПРАКТИКА ПОСТРОЕНИЯ ВЫСОКОНАГРУЖЕННЫХ СИСТЕМ**

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Божко С. С.*

*Пилецкий И. И. – кандидат физико-математических наук, доцент*

При проектировании и разработке высоконагруженных серверных приложений для обработки и анализа большого объема данных основной задачей является обеспечение надежной и быстрой обработки сообщений от клиентских