

УДК 621.391; 004.031

## РАСПРЕДЕЛЕНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ МЕЖДУ ОБЪЕКТАМИ ДЕЦЕНТРАЛИЗОВАННОЙ СЕТИ С АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИЕЙ РАСПИСАНИЯ РОТАЦИИ

Ю.А. ДУЙНОВА, К.Р. КОПТЯЕВ, А.В. СМОЛЯК, С.И. ПОЛОВЕНЯ

Учреждение образования «Белорусская государственная академия связи»,  
ул. Ф. Скорины, 8/2, Минск, 220114, Беларусь

Поступила в редакцию 12 марта 2020

Рассмотрены известные системы управления очередями, синтезирован алгоритм работы системы автоматической генерации расписания ротации медиаконтента, описаны способы оптимизации использования ресурсов канала передачи данных.

*Ключевые слова:* генерация расписания, гетерогенная сеть, база данных, ротация медиаконтента.

### Введение

В настоящее время постоянно увеличивающееся количество пользователей систем мобильной связи и их растущие потребности в ресурсах беспроводных сетей обуславливают стремительность развития телекоммуникационных технологий. Ввиду ограниченности частотно-временных радиоресурсов современных технологий мобильного доступа международные комитеты по стандартизации вынуждены были провести напряженную работу для введения в 2020-м году новых спецификаций, определяющих работу сетей связи пятого поколения (5G), для повышения качества обслуживания своих пользователей. Возможности таких гетерогенных беспроводных сетей позволяют качественно улучшить интеграцию уже существующих сетей и развивать более перспективные технологии радиодоступа, что в итоге неизменно приводит к улучшению качества обслуживания пользователей, дает импульс к возникновению принципиально новых научно-технических задач. Не считая услуг мобильного доступа, включающих в себя передачу различного пользовательского трафика, с помощью гетерогенных сетей 5G будут реализованы технологии облачных вычислений и программно-определяемых сетей, а также многие приложения «Интернета вещей», основанные на межмашинном взаимодействии. На пути к реализации «Умного города» в пределах концепции «Интернета вещей» различия между потребительскими и промышленными сценариями становятся все более существенными. Все это, а также аспект подключения носимых устройств, занимающих промежуточное положение между пользовательскими системами и сетями межмашинного взаимодействия, лишь подчеркивает серьезность задачи качественного и эффективного функционирования гетерогенных сетей пятого поколения [1].

В целях улучшения качества обслуживания в системах 5G будет необходимо наладить согласованное взаимодействие сотовых технологий радиодоступа, функционирующих в лицензированном диапазоне частот (к примеру, система LTE комитета 3GPP), с технологиями локального доступа IEEE 802.11 нелицензированного спектра, более известных как Wi-Fi. Современные решения IEEE 802.11ad/ay (WiGig) и такие новейшие технологии 3GPP, как «Новое Радио» (New Radio), реализующие связь, в том числе на крайне высоких частотах, смогут дополнить традиционные системы радиодоступа. Кроме того, непосредственные соединения пользовательских устройств друг с другом также представляют большой интерес, поскольку снижают нагрузку на инфраструктурные каналы связи, позволяя перенаправлять трафик из сотовой сети в сети других технологий радиодоступа, а также сократить общее время загрузки контента на устройства, способные работать в режиме оффлайн.

В большинстве случаев входящие в состав гетерогенных беспроводных сетей технологии радиодоступа излишне фрагментированы и недостаточно интегрированы между собой, что серьезно тормозит их дальнейшее развитие и лишь увеличивает разрыв между эффективностью функционирования современных систем мобильного доступа и требованиями передовых пользовательских приложений. С целью изменения сложившейся ситуации необходимо применение сбалансированного распределения информационных потоков для равномерной загрузки различных технологий радиодоступа в пределах гетерогенной беспроводной сети. Возможно распределять потоки трафика между сотовой сетью и другими (например, локальными) системами радиодоступа, а также активизировать прямые соединения между устройствами и оптимизировать загрузку контента на устройства преимущественно в часы минимальной нагрузки на сеть [2].

Очевидно, что использование прямых соединений «устройство – устройство» и согласование работы различных технологий радиодоступа подразумевают создание нового подхода к управлению потоками трафика и организации множественного доступа, предоставляющей инструменты для контроля за установлением соединения, назначения мощности передачи и распределения ресурса между пользователями и устройствами. Эти шаги должны обеспечить выполнение таких предъявляемых требований к сетям 5G, как высокая скорость передачи данных, высокая емкость системы (выраженная, например, в числе одновременно обслуживаемых пользователей), приемлемая энергетическая эффективность при доставке трафика, а также низкая задержка доступа в сеть при установлении соединения или передаче данных. Исходя из этого, нужен комплексный подход, учитывающий главные особенности развития гетерогенных беспроводных систем мобильного доступа на современном этапе: высокую плотность расположения пользовательских устройств и узлов сетевой инфраструктуры, взаимную интегрированность разнородных технологий радиодоступа и активное использование прямых соединений между устройствами, — а также использование крайне высоких частот и обеспечение поддержки приложений «Интернета вещей» с учетом особенностей носимых устройств. В итоге появляется важный вопрос создания алгоритмов обеспечения эффективного функционирования устройств в гетерогенных беспроводных сетях и управления ими.

Таким образом, актуальными являются разработка и исследование моделей автоматизации генерации правил ротации контента на узлах сетевой инфраструктуры, а также создание эффективных алгоритмов управления информационными потоками с целью снижения нагрузки на сотовую сеть методом загрузки «тяжелого» контента в периоды снижения трафика в сети.

В данной работе рассмотренные выше задачи предлагается решить путем автоматизации генерации правил ротации контента, что является одним из способов снизить нагрузку на сеть.

### **Анализ управления базами данных**

Работа с базами данных осуществляется с использованием систем управления базами данных (СУБД). Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого и визуализации информации – отбора отображаемых данных в соответствии с заданным критерием, их упорядочения, оформления и последующей выдачи на устройства вывода или передачи по каналам связи [3].

Одними из самых распространенных систем являются реляционные и объектно-реляционные СУБД. Они представляют собой таблицы с упорядоченными столбцами с определенными уникальными названиями [4]. Для управления реляционными базами данных применяется особый язык программирования – Structured Query Language (SQL), позволяющий одним или несколькими запросами работать с данными сразу нескольких таблиц. Схема работы с базой данных представлена на рис. 1 [5].



Рис. 1. Схема работы с базой данных

СУБД MySQL, рекомендованная к применению в небольших или средних проектах, является одной из самых быстрых и удобных в работе [6].

К ее достоинствам относят наличие бесплатной лицензии, обеспечение сопровождающей документацией, многофункциональность, возможность взаимодействия с другими базами данных, включая DB2 и Oracle.

В числе недостатков – отсутствие встроенной поддержки XML или OLAP, сложная реализация некоторых простых задач, к примеру, создание инкрементных резервных копий, исключительно платная поддержка в бесплатной версии.

Для работы с MySQL используется не только текстовый, но и графический режим [7]. Это возможно благодаря приложению phpMyAdmin: для работы в приложении вам даже не нужно будет знать SQL-команды, а администрировать свою базу данных можно прямо через браузер.

Объектно-реляционная СУБД Oracle наравне с SQL использует процедурное расширение под названием PL/SQL, а также язык Java. Oracle используют преимущественно крупные корпорации, поскольку эта система известна высокой стабильностью и безопасностью работы, в ней хорошо отлажена процедура бэкапа, реализована масштабируемость и другие ценные возможности [8].

Среди ее основных достоинств широкий функционал и современные инновации, высочайшая степень надежности.

К недостаткам можно отнести очень высокую стоимость, особенно для небольших организаций, а также серьезные системные требования при установке.

СУБД MongoDB является документно-ориентированной и предназначена для хранения иерархических структур данных. Она представлена в виде документного хранилища без использования таблиц или схем, имеет открытый исходный код. Относится к классу СУБД NoSQL, поэтому во многих случаях приложение, написанное на MongoDB, будет работать быстрее, чем такое же приложение, использующее SQL, поскольку использует значительно более простой объектный язык запросов.

Основные достоинства этой СУБД заключаются в скорости и простоте в использовании, поддержке JSON и других традиционных документов NoSQL, быстром чтении/сохранении данных любой структуры.

Среди недостатков – продолжительное время установки и отсутствие SQL в качестве языка запросов. Несмотря на доступность инструментов для перевода SQL-запросов в MongoDB, они являются лишь дополнением.

MongoDB имеет свои ограничения, поэтому его использование предпочтительно при отсутствии необходимости в сложных и нетривиальных выборках [9]. Сравнение популярных СУБД представлено в таблице 1 [10].

Таблица 1. Сравнение популярных СУБД

СУБД	Год создания	Поддержка SQL	Популярность
MySQL	1995	да	20 %
Oracle	1979	да	65 %
MongoDB	2009	нет	10 %

В среде web наиболее приспособленной для применения СУБД является MySQL. Ввиду преимущественно ограниченного места для исполнения приложений на большинстве хостинг-площадок необходима эффективная СУБД, обладающая при этом высокой надежностью.

### Анализ работы системы и управления очередями задач

Если выбрать операционную систему (ОС) Ubuntu и язык программирования PHP, можно провести небольшой анализ системы управления очередями задач, поддерживаемых данной ОС и языком программирования.

Очереди сообщений в архитектуре распределенных систем часто используются при дроблении большой системы на компоненты, поскольку они являются простым и в то же время масштабируемым инструментом, позволяющим «подружить» независимые системы и научить их работать совместно.

К их задачам относится предоставление возможности обмена сообщениями различным подсистемам при обеспечении маршрутизации, масштабирования и гарантированной доставки [11].

Очередь сообщений позволяет обеспечить асинхронное выполнение участков программы. Это позволяет:

- увеличить скорость работы приложения;
- обслуживать большее количество посетителей (масштабироваться);
- использовать разные языки разработки в одном приложении.

Принцип системы очередей можно реализовать на MySQL и PHP, однако простота и наличие готовых решений позволяют сделать это быстрее.

Для PHP существует две популярные системы управления очередями задач.

Supervisor – система клиент/сервер, при помощи которой пользователь (администратор) может контролировать подключенные процессы в системах типа UNIX.

Инструмент создает процессы в виде подпроцессов от своего имени, поэтому имеет полный контроль над ними [12]. Схема работы системы управления Supervisor представлена на рис. 2 [13].

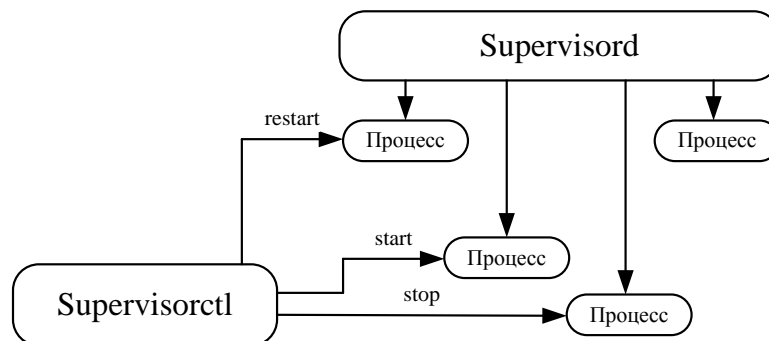


Рис. 2. Схема работы Supervisor

Supervisor состоит из серверной части под названием `supervisord`, которая создает и управляет всеми процессами, и системного/веб-интерфейса `supervisorctl` для управления и мониторинга `supervisord`. Supervisor также включает пользовательский веб-интерфейс `supervisorctl`, который включается при помощи файла конфигурации. Для этого нужно изменить секцию `[inet_http_server]`, вписав туда верные имя пользователя и пароль.

Cron – планировщик задач, автоматически запускающий определенные задания с указанной периодичностью [14]. Удобен при выполнении, например, скрипта, очищающего кэш скриптов или отсылающего уведомления зарегистрированным пользователям сайта.

Редактировать файл `crontab` можно с помощью интерфейса панели управления и используя специальный инструмент с таким же названием – `crontab`. Конфигурационный файл состоит из строк. Каждая строка описывает программу, которая будет запущена по расписанию [15]. В каждой строке шесть полей.

Поля имеют назначение:

- 1) минуты (0–59);
- 2) часы (0–23);
- 3) день месяца (1–31);

- 4) месяц в году (1–12);
- 5) день недели (0–6);
- 6) программа, которая будет запущена.

Сравнение популярных систем управления очередями задач представлено в табл. 2 [16].

Таблица 2. Сравнение популярных систем управления очередями задач

Система	Требует установки	Начало выполнения – после запуска	Сложность настройки
Cron	нет	нет	5
Supervisor	да	да	7

Таким образом, на основе анализа систем управления задач наиболее подходящей является Supervisor. Также присутствует поддержка автоматического перезапуска задачи при возникновении ошибки и запуск одной очереди несколькими воркерами для более быстрого выполнения очереди.

Обзор известных систем показал, что на рынке программного обеспечения существует не так много систем удаленного управления медиаконтентом. Известные решения – это «SPINETIX» одноименной компании SpinetiX AG с головным офисом в швейцарской Лозанне, решение от компании Samsung, а также предложение от русской фирмы Galler.

К достоинствам «SPINETIX» можно отнести высокое качество реализации на программном уровне, стабильность всей системы, очень хорошую поддержку, отсутствие необходимости в сервере, встроенную анимацию для любых объектов, а также то, что генератор контента собирает картинку на экране прямо в момент воспроизведения.

Главным недостатком является чрезвычайно высокая цена (премиальный ценовой сегмент).

Решение от корейского гиганта Samsung Business [17] является представителем средне-ценового сегмента рынка. В нем разработан отличный интерфейс и хорошая поддержка.

Среди его достоинств – простой и понятный интерфейс и большое количество клиентов.

В числе недостатков – относительная доступность в зависимости от региона и периодические сбои в работе.

Решение Gallery [18] российской компании принадлежит среднеценовому сегменту рынка. Оно не имеет особых преимуществ, его главное достоинство – это доступная цена.

К недостаткам можно отнести плохую поддержку и частые сбои в работе.

По результатам вышеописанных аналогов можно вывести достоинства и недостатки разрабатываемого программного средства. Среди достоинств – доступность (среднеценовой сегмент рынка), стабильность системы и относительная гибкость в подаче контента.

Единственный недостаток проекта – это слабая поддержка.

Целью создания этого программного обеспечения является привлечение заказчиков на трансляцию своего контента, а для этого важно, чтобы функционал сайта был понятен и приятен.

### Алгоритм работы системы автоматической генерации расписания

Алгоритм работы системы автоматической генерации расписания основан на модульном подходе, его основная идея состоит в объединении различных типов медиафайлов с определенным типом, исключая повторение одного и того же контента более одного раза подряд.

В системе существует три типа медиафайлов:

- общий – основной тип контента;
- коммерческий – тип контента с более высоким приоритетом;
- системный – тип контента, предназначенный для финализации расписания ротации.

Системный файл может быть только изображением.

Схема работы автоматической генерации расписания ротации медиаконтента представлена на рис. 3.

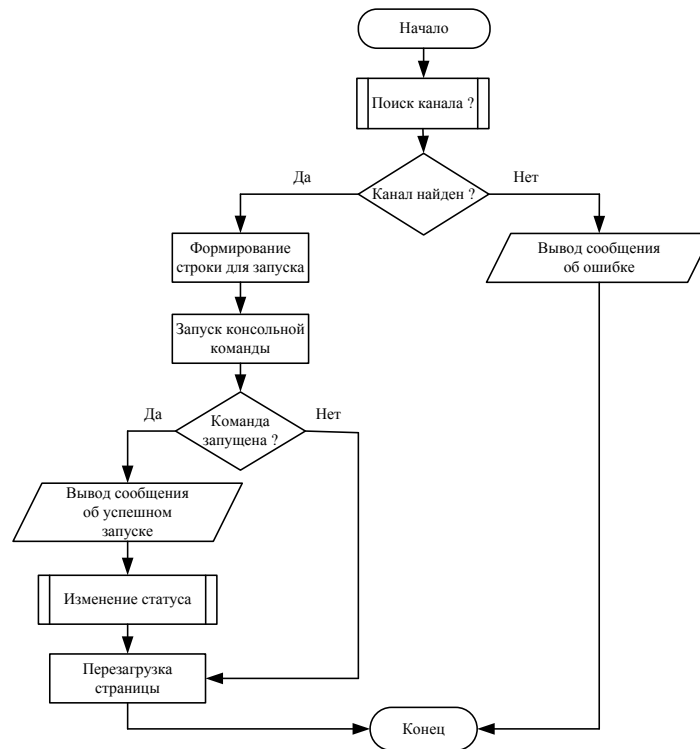


Рис. 3. Схема работы автоматической генерации расписания ротации

Система удаленного управления медиаконтентом сконструирована по архитектуре Representational State Transfer (REST) и удовлетворяет следующим критериям:

- система разделена на клиентов и на сервер;
- сервер не хранит какой-либо информации о клиентах. Вся необходимая информация для обработки запроса и, если необходимо, идентификации клиента, хранится в запросе;
- каждый ответ имеет отметку о том, является ли он кэшируемым или нет;
- имеет универсальный интерфейс между компонентами системы.

Сервер соответствует всем принципам REST и может считаться RESTfull.

Также система включает полноценный набор средств для упрощенной реализации RESTful API:

- быстрое создание прототипов с поддержкой распространенных API к Active Record;
- настройка формата ответа JSON;
- получение сериализованных объектов с нужной вам выборкой полей;
- надлежащее форматирование данных и ошибок при их валидации;
- коллекция пагинаций, фильтров и сортировок;
- эффективная маршрутизация с надлежащей проверкой HTTP-методов;
- встроенная поддержка методов OPTIONS и HEAD;
- аутентификация и авторизация;
- HTTP-кэширование и кэширование данных;
- настройка ограничения для частоты запросов (Rate limiting);

### Разработка функционала и разработка алгоритма работы с устройствами

На странице устройств представлен список всех подключенных к текущему окружению устройств. Список доступных устройств программного обеспечения представлен на рис. 4.

Активен	Статус	Состояние экрана	Канал	Имя пользователя	Имя	Адрес	Версия клиента	Последнее обращение	Добавлен
<input type="checkbox"/>	Да	Вкл	Новости	device_9b6b732f94b5b04	АЗС 42 (Левый)	улица Шаранговича 63, Минск, Беларусь	71	8 минут назад	21 ноября 2018, 23:20
<input type="checkbox"/>	Да	Вкл	Новости	device_a347b9043d4e03a	АЗС 42 (Центр)	улица Шаранговича 65, Минск, Беларусь	71	9 минут назад	21 ноября 2018, 23:02
<input type="checkbox"/>	Да	Выкл	БКМ1	device_9e41bf70f3d97c2b	Электробус 1	-	69	7 часов назад	27 февраля 2019, 13:56
<input type="checkbox"/>	Да	Выкл	Тест бегущей строки здесь	device_a3ad912d720bc97	device_a3ad912d720bc97	-	71	2 дня назад	16 апреля 2019, 14:44
<input type="checkbox"/>	Да	Выкл	Презентация КИНО	device_a56cc684cceb0331	Презентация для Кино	-	-	Нет данных	26 апреля 2019, 13:30

Рис. 4. Список доступных устройств

Для передачи данных между сервером и устройством (клиентом) используется REST API.

REST – архитектурный стиль, некоторое множество ограничений для построения распределенных приложений. Компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине [19].

Глобальным идентификатором определяется каждая единица информации однозначно, таким как URL. Строго заданный формат, в свою очередь, имеет каждая URL.

Чтобы зарегистрировать устройство в системе, необходимо пройти два этапа.

Вначале следует отправить POST запрос на `url/v1/device` с параметром `username`. После этого устройство появляется в системе, но не отображается в списке, так как не подтверждено. В ответе на запрос приходит логин и пароль.

Для подтверждения регистрации необходимо отправить POST-запрос на `url/v1/device/confirm` с параметрами `username` и `password`. После успешного ответа устройство готово к работе.

После создания DeviceController нам доступны следующие действия:

- GET/device/(Index) – получение списка всех устройств. Это упрощенный список, содержащий только поля идентификатора и названия объекта;
- GET/device/{id}(View) – получение полной информации об устройстве;
- PUT/device/ или POST/user/(Create) – создание нового устройства. Данные передаются в теле запроса без применения кодирования;
- POST/device/{id} или PUT/device/{id} (Edit) – изменение данных с идентификатором {id}, возможна их замена. Разница в том, что при PUT запросе, если нет устройства с указанным {id}, то устройство будет создано;
- DELETE/device/{id} (Delete) – удаление данных с идентификатором {id}.

Рекомендуется применить механизм кэширования для увеличения быстродействия REST-сервисов и сокращения запросов к базе данных [20]. Кэширование можно настраивать как на уровне сервера, так и в самом приложении в зависимости от ситуации.

Нужно создать хорошую и понятную документацию для удобного пользования REST сервисами. REST архитектура очень проста для использования. Сразу можно определить, по виду пришедшего запроса, что он делает, даже не разбираясь в форматах данных, поскольку данные передаются без применения дополнительных слоев. Не надо переводить данные из одного формата в другой, потому что REST считается менее ресурсоемким.

Был разработан алгоритм и функционал работы с пользователями и их правами доступа.

Данный алгоритм ожидает пользовательского ввода имени, после чего создается роль данному пользователю, создаются выбранные права и привязываются к ранее созданной роли посредством `authManager` и RBAC-системы. Благодаря такому подходу исключается повторение ролей, а также преимуществом данного подхода является то, что каждый пользователь имеет свои уникальные права.

### Разработка функционала и разработка алгоритма работы с пользователями и их правами доступа

Вопрос с правами доступа возникает в связи с необходимостью ограничить права пользователя в системе, что подразумевает под собой запрет на совершение каких-либо действий с определенными объектами, например, их просмотр, запись, редактирование и т. д.

Или, наоборот, из-за необходимости дать (расширить) права пользователям в системе, что в реальности чаще всего следует за сообщением системы о нарушении прав доступа (например, недостаточно прав для просмотра) и обращением пользователя к администраторам с просьбой об этом [21].

В системе существует два вида пользователей:

- root – суперпользователь;
- user – пользователь.

Добавление пользователей происходит во вкладке Пользователи – Создать – Создать пользователя. Окно создания нового пользователя представлено на рис. 5.

Рис. 5. Страница создания нового пользователя

Во вкладке Аккаунт указываются основные данные пользователя:

- Имя пользователя – логин пользователя, заполняется латиницей;
- Пароль – пароль пользователя, от шести символов, если не заполнять, то генерируется автоматически;
- Email – email пользователя. Для получения или восстановления пароля;
- Группа – группа пользователя. Основная группа пользователя.

После заполнения основных данных пользователя необходимо перейти во вкладку Права пользователя и выбрать права пользователя:

- Доступ к общим элементам – доступ для CRUD коммерческих и общих элементов;
- Доступ к каналам – доступ для CRUD каналов;
- Доступ к устройствам – доступ для CRUD устройств;
- Доступ к плейлистам – доступ для CRUD плейлистов;
- Работа с оплатой – доступ для CRUD оплаты;
- Email для служебной информации – указываются email через запятую, на которые будет приходить информация об оплате, задолженности и иной служебной информации. Если email не указан, то данному пользователю ничего не придет;
- Доступ к группам – выбираются группы для администрирования.

Окно редактирования прав выбранного пользователя представлено на рис. 6.



Рис.6. Страница редактирования прав выбранного пользователя

При создании пользователя создается роль с названием, как у текущего, и привязывается к нему [22].

```
UserHelper::addingUserRole($this->username, $this->id);
public static function addingUserRoless(string $name, int $user_id) {
    $auth = Yii::$app->authManager;
    if(!$auth->getRole($name)) {
        $role = $auth->createRole($name);
        $auth->add($role);
        $auth->assign($role, $user_id);
    }
}
```

После выбора прав и сохранения происходит добавление прав к роли пользователя.

```
UserHelper::addingUserPermissions($this->accesses, $this->id, $this->username);
public static function addingUserPermissions(array $accesses, int $user_id, string $role_name) {
    $auth = Yii::$app->authManager;
    $role = $auth->getRoless($role_name);
    $auth->removeChildren($role);
    foreach ($accesses as $access) {
        $permission = $auth->getPermission($access);
        $auth->addChild($role, $permission);
    }
}
```

После удаления пользователя роль и связи прав с этой ролью удаляются.

### Заключение

Проведен сравнительный анализ современных стеков технологий, использующихся в создании системы удаленного управления медиаконтентом. Были выявлены преимущества и недостатки СУБД, а также выбрана наиболее подходящая.

Разработан алгоритм автоматической генерации расписания ротации медиаконтента по заданным пользователем правилам и ограничениям. Этот алгоритм, в отличие от ранее опубликованных в литературе, позволяет объединять различные типы медиаконтента для предотвращения коллизии в расписании ротации.

Реализована архитектура основных модулей, серверного и клиентского приложений, на основе анализа недостатков конкурентов, предотвращающих потерю медиаконтента.

## DISTRIBUTION OF INFORMATION RESOURCES BETWEEN OBJECTS OF A DECENTRALIZED NETWORK WITH AUTOMATIC GENERATION SCHEDULES OF ROTATION

Y.A. DUINOVA, K.R. KOPTYAEV, A.V. SMOLYAK, S.I. POLOVENYA

### Abstract

The well-known queue management systems are considered, the algorithm of the automatic generation of the media content rotation schedule generation system is synthesized, methods for optimizing the use of data transmission channel resources are described.

### Список литературы

1. Дуйнова, Ю. А. Модифицированный алгоритм и информационное обеспечение для контроля параметров мультиагентных сетей / Ю. А. Дуйнова, С. И. Половения, Н. М. Бобрик // Веснік сувязі. – 2017. – № 3 (143). – С. 41–45.
2. Половения, С. И. Принципы эффективного управления распределением ресурсов мультисервисной макросети при предоставлении услуг пользователям / Н. В. Федорова, Н. В. Пирогова, С. И. Половения, Ю. В. Мельник // Веснік сувязі. – 2018. – № 2. – С. 34–38.
3. Реляционная база данных и ее особенности. Виды связей между реляционными таблицами [Электронный ресурс]. – Режим доступа : <https://www.yaklass.ru/materiali?mode=cht&chtid=511>. – Дата доступа : 20.02.2020.
4. Архитектура базы данных. Физическая и логическая независимость [Электронный ресурс]. – Режим доступа : <http://bourabai.kz/dbt/dbms/023.htm>. – Дата доступа : 20.02.2020.
5. Что представляет собой MySQL? [Электронный ресурс]. – Режим доступа : <http://www.mysql.ru/docs/man/What-is.html>. – Дата доступа : 20.02.2020.
6. Почему я выбрал Yii2 [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/268269/>. – Дата доступа : 20.02.2020.
7. Шварц, Б. MySQL по максимуму. Полный курс / Б. Шварц, П. Зайцев, В. Ткаченко. – СПб. : Питер, 2018. – 56 с.
8. СУБД Oracle [Электронный ресурс]. – Режим доступа : <http://bourabai.kz/dbt/servers/Oracle.htm>. – Дата доступа : 20.02.2020.
9. MySQL и MongoDB – когда и что лучше использовать [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/322532/>. – Дата доступа : 20.02.2020.
10. Сравнение современных СУБД [Электронный ресурс]. – Режим доступа : <http://drach.pro/blog/hi-tech/item/145-db-comparison>. – Дата доступа : 20.02.2020.
11. Веб-терминал системы управления очередью [Электронный ресурс]. – Режим доступа : <https://mtg-biz.ru/solutions/software/terminal/>. – Дата доступа : 20.02.2020.
12. Установка и настройка supervisor [Электронный ресурс]. – Режим доступа : <https://ruhighload.com/%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA+%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2+%D0%B2+supervisor>. – Дата доступа : 20.02.2020.
13. Схема работы Supervisor [Электронный ресурс]. – Режим доступа : <https://ruhighload.com/fault+tolerance+%D0%B8+%D0%B5%D0%B3%D0%BE+%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F+%D0%B2+php>. – Дата доступа : 20.02.2020.
14. Cron-формат [Электронный ресурс]. – Режим доступа : <http://www.nncron.ru/nncronlt/help/RU/working/cron-format.htm>. – Дата доступа : 20.02.2020.
15. Использование CRON и команды crontab [Электронный ресурс]. – Режим доступа : <http://www.codenet.ru/webmast/php/cron.php>. – Дата доступа : 20.02.2020.
16. Сравнение систем управления очередью [Электронный ресурс]. – Режим доступа : [https://elektroochered.3dn.ru/index/sravnenie\\_sistem\\_ehlektronnoj\\_ocheredi/0-2](https://elektroochered.3dn.ru/index/sravnenie_sistem_ehlektronnoj_ocheredi/0-2). – Дата доступа : 20.02.2020.
17. Решения Samsung для бизнеса [Электронный ресурс]. – Режим доступа : <https://www.samsung.com/ru/business/>. – Дата доступа : 20.02.2020.

18. Размещение наружной рекламы [Электронный ресурс]. – Режим доступа : <http://www.gallerymedia.com/>. – Дата доступа : 20.02.2020.
19. Flanders, J. RESTful. Полный курс / J. Flanders. – 1-е изд. – O'Reilly Media, 2008. – 36 с.
20. Управление информацией сервиса [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/38730/>. – Дата доступа : 20.02.2020.
21. И снова права доступа [Электронный ресурс]. – Режим доступа : <https://community.terrasoft.ru/questions/i-snova-prava-dostupa>. – Дата доступа : 20.02.2020.
22. Настройка Yii2 RBAC [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/post/235485/>. – Дата доступа : 20.02.2020.