DOI: 10.15593/2499-9873/2021.01.07
УДК 62; 004; 007

О.В. Логиновский, А.А. Шинкарев, М.Е. Коваль

Южно-Уральский государственный университет, Челябинск, Россия

РАЗРАБОТКА АРХИТЕКТУРЫ СИСТЕМ ИНФОРМАЦИОННОГО ПОИСКА НА ОСНОВЕ ОЧЕРЕДЕЙ СООБЩЕНИЙ В КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Сегодня поиск информации стал не только рядовым навыком пользователей информационных систем, но и важнейшей частью бизнеса. Изо дня в день растут объемы данных, которыми владеют организации. Соответственно, затрудняется и поиск нужной информации в этих данных, особенно если они являются частью разных хранилищ. Для обеспечения поиска в корпоративных информационных системах применяются системы информационного поиска, которые предоставляют единый графический интерфейс для взаимодействия с ними конечных пользователей и осуществляют поиск в разных источниках данных. Компании могут приобрести уже готовые программные комплексы или разработать поисковую систему самостоятельно. Однако для самостоятельной разработки необходимо тщательно подходить к проектированию архитектуры системы для достижения соответствия всем требованиям, которые сегодня предъявляются к программному обеспечению подобного рода. Предметом исследования являются системы информационного поиска в корпоративных информационных системах. Работа преследует цели описать основные архитектурные подходы к построению систем информационного поиска, выявить их слабые и сильные стороны, а также сформировать базовые идеи по разработке архитектуры систем информационного поиска на основе очередей сообщений, описать основные составные части таких систем и детализировать ключевые аспекты практической реализации хранилища данных пользовательских поисковых запросов и результатов их обработки. Рассматриваются результаты исследований, касающихся типичных проблем процесса поиска информации сотрудниками организации в корпоративных системах. Дается оценка существующим архитектурным подходам к реализации систем информационного поиска. Осуществляются анализ и сравнение двух наиболее популярных разноплановых брокеров обмена сообщениями. Обосновывается актуальность проблемы поиска информации в корпоративных информационных системах. Приводится описание основных подходов к созданию архитектуры систем информационного поиска, рассматриваются их достоинства и недостатки, а также архитектурные диаграммы компонентов. Описываются микросервисы, из которых состоит система, основанная на очереди сообщений. Обосновывается выбор инструмента Kafka в качестве наиболее подходящего для решения рассматриваемой задачи брокера сообщений. Также приводится графическая схема механизма обработки ошибок, возникающих во время работы поисковых сервисов.

Ключевые слова: информация, система информационного поиска, онтология, брокер сообщений, очередь сообщений, мультиагентные системы, Kafka, RabbitMQ, корпоративные информационные системы, архитектура программного обеспечения, масштабирование, отказоустойчивость.

O.V. Loginovskiy, A.A. Shinkarev, M.E. Koval

South Ural State University, Chelyabinsk, Russian Federation

DEVELOPMENT OF ARCHITECTURE OF MESSAGE QUEUE BASED INFORMATION SEARCH SYSTEMS FOR ENTERPRISE INFORMATION SYSTEMS

Today, information search has become not only an ordinary skill for users of information systems, but also an essential part of business. The amount of data that organizations own is growing day by day. Accordingly, it becomes difficult to find the necessary information in this data, especially if it is allocated in different storages. To facilitate the search in enterprise information systems, information search systems with a single graphical interface and ability to search in different data sources are used. There are ready-made software packages on the market, though some companies opt out to develop their own search engine. In the latter case, however, it is critical to be scrupulous designing the system architecture to comply with all the requirements that are imposed on this kind of software today. Information search systems for enterprise information systems. The purpose of the study was to describe the main architectural approaches to developing information search systems and identify their strong and weak points; to form basic ideas for the development of the architecture of information search systems based on message queues and to describe the main components of such systems; to elaborate on the key aspects of the practical implementation of the data warehouse of user search queries and the results of their processing. The paper considers the results of studies on the typical problems which employees face during information search in enterprise applications. The authors evaluate the existing architectural approaches to information search systems, analyze and compare two of the most popular message brokers. The article substantiates the relevance of the problem of finding information in enterprise information systems. The authors provide a description of the main approaches to the architecture of information search systems, go into their advantages and disadvantages, and provide architectural diagrams of the components. The microservices that make up a message queue-based system are described. Kafka is chosen and substantiated as the most suitable message broker. The authors also give a graphical scheme of handling errors that arise during search services operation.

Keywords: information, information search system, ontology, message broker, message queue, multi-agent systems, Kafka, RabbitMQ, enterprise information systems, software architecture, scalability, resiliency.

Введение

В современном мире многие компании достигли высокого уровня информатизации и автоматизации бизнес-процессов. При этом вырос и объем информации, которой владеет среднестатистическая компания. По подсчетам ученых, объем информации в мире возрастает ежегодно на 30 % [1]. Еще недавно, до массового перехода к накоплению и обработке больших данных при сравнительно небольшом объеме информации, поиск в ней был возможен вручную или с помощью простых инструментов текстового поиска, например с помощью регулярных выражений [2]. Сейчас же с увеличением объемов информации поиск новых сведений и анализ данных становятся более трудоемкими. Именно поэтому появились поисковые системы в ин-

тернете, а также системы информационного поиска внутри корпоративных информационных систем.

Системы информационного поиска должны обладать возможностью агрегировать существующие данные и получать из них новую информацию для предоставления ответов на разнообразные запросы пользователей системы. Также система должна постоянно расширять границы поиска, объемы обрабатываемой информации должны расти для более полного, точного, а главное, полезного поиска. Ввиду этого к таким системам предъявляются довольно высокие требования, касающиеся совершенствования алгоритмов и способов поиска информации, их скорости и точности. Иными словами, системы должны эволюционировать со временем. При этом для пользователя процесс улучшения системы должен оставаться незаметным с точки зрения непрерывной доступности системы во время обновлений.

В связи с развитием компаний и увеличением объема их данных должны совершенствоваться и инструменты, используемые для обработки данных. Многие крупные организации образуют цифровые экосистемы – объединения сервисов и систем, которые могут взаимодействовать друг с другом. Для взаимодействия сервисов между собой в настоящее время активно используются очереди сообщений [3]. Сегодня обмен сообщениями – это один из самых популярных способов взаимодействия сервисов между собой. Однако сам по себе инструмент является лишь способом для связи сервиса информационного поиска и остальных частей корпоративной информационной системы.

В статье приводится обзор существующих подходов к реализации информационного поиска, рассматривается один из возможных методов построения архитектуры системы информационного поиска, а также приводятся ключевые детали реализации.

1. Информационный поиск и для чего он нужен в организации

Информационный поиск – это действия, методы и процедуры, позволяющие осуществлять отбор определенной информации из массива данных (ГОСТ 7.73.96 «Поиск и распространение информации. Термины и определения»). Классический информационный поиск – это поиск документов, удовлетворяющих запросу, в некоторой коллекции документов.

С точки зрения использования компьютерной техники под информационным поиском понимается совокупность логических и технических операций, целью которых является нахождение документов, сведений о них, фактов и данных, которые соответствуют запросу потребителя [4]. К информационному поиску относятся и задачи по навигации пользователей в списке документов и их фильтрации, а также задачи по дальнейшей обработке найденных документов.

Язык, с помощью которого формулируются запросы к поисковым системам, называется информационно-поисковым или языком поисковых запросов [5].

Информационно-поисковый язык – это формализованный искусственный язык. Он обычно состоит из словаря (тезауруса) и грамматики различной сложности, а также логических операторов, морфологии языка, регистра слов, возможности учета расстояния между словами и расширенного поиска.

Информационный поиск состоит из нескольких этапов [6]:

1. Уточнение информационной потребности и формулировка запроса.
2. Выбор источников информации, соответствующих запросу пользователя.
3. Извлечение информации из информационных массивов.
4. Оценка результатов поиска.

Поиск давно стал одним из ключевых механизмов доступа пользователей к нужной информации. Однако сегодня многие люди с легкостью решают задачи по поиску нужной информации при работе в открытом и почти бесконечном пространстве интернета и при этом испытывают значительные трудности в рамках своей корпоративной информационной системы или даже на собственном персональном компьютере. Как показало исследование международной ассоциации AIIM (The Association for Intelligent Information Management), около 72 % респондентов считают, что найти корпоративную информацию намного труднее, чем открытую информацию в интернете. В 2010 г. компания IDC (International Data Corporation) провела опрос, результаты которого показали, что 62 % работников тратят не менее двух часов на поиск информации, 57 % говорят о том, что разрозненная информация затрудняет поиск, 84 % хранят важную и ценную информацию на своих рабочих компьютерах, а 41 % не способен справиться с постоянным

увеличением объемов информации [7]. Такая ситуация объясняется тем, что на предприятиях может существовать несколько разных систем и хранилищ данных, которые могут иметь собственный сервис по информационному поиску, тогда как интернет представляет собой более однородную среду, поиск в которой легко выполнять с помощью поисковых систем, таких как Google.

Если говорить о различиях между корпоративным поиском и поиском в интернете, можно отметить, что поиск в интернете охватывает открытую часть веб-сайтов, в то время как на предприятии он охватывает информационные ресурсы с учетом прав доступа. Одним из важных и частых требований является то, что внутри корпоративной системы предприятия должны быть найдены все документы, удовлетворяющие запросу. Кроме того, при корпоративном поиске должна учитываться должность сотрудника, например программист в первую очередь хочет видеть технические документы, а бухгалтер – финансовые и юридические.

Принимая во внимание все вышесказанное, можно выделить следующие требования к средствам внутрикорпоративного поиска:

1. Возможность поиска по ключевым словам.
2. Осуществление поиска по всем хранилищам данных (базы данных, документы и т.д.).
3. Классификация результатов поиска.
4. Возможность добавления новых поисковых механизмов без ущерба для работоспособности.
5. Отказоустойчивость, т.е. способность системы сохранять рабочее состояние даже при отказе одной из ее частей.
6. Доставка результатов поиска пользователю по мере их нахождения.
7. Кеширование результатов поиска, т.е. сохранение результатов поиска в памяти, к примеру в оперативной, для более быстрого ответа на запрос.
8. Определение синонимов для принятых в компании сокращений и терминов.
9. Повышение полезности информации, т.е. исключение избыточности данных.

Таким образом, можно сделать вывод о том, что корпоративная экосистема должна включать в себя сервис по осуществлению инфор-

мационного поиска, который бы мог удовлетворять всем вышеперечисленным требованиям, а также осуществлял бы поиск по всем корпоративным подсистемам и хранилищам данных для того, чтобы упростить работу сотрудников и избавить их от необходимости ручного поиска по множеству разрозненных источников информации. Существуют готовые программные комплексы от IBM, Google, Microsoft, которые можно приобрести и использовать на предприятии.

Приобретение готовой поисковой системы является более простым способом внедрения механизма поиска в корпоративную информационную систему, нежели разработка поискового сервиса с нуля. При разработке системы информационного поиска важно грамотное проектирование архитектуры. На сегодняшний день существует несколько способов организации архитектуры системы информационного поиска. Рассмотрим основные подходы к построению архитектуры информационных систем подобного класса.

2. Обзор подхода к реализации системы информационного поиска с использованием онтологии предметной области

В данном случае в основе архитектуры системы информационного поиска лежит использование онтологий предметной области – структур, которые описывают объекты, составляющие предметную область, и связи между ними. На вход системы информационного поиска подается исходный текст запроса, указанные пользователем онтологии и список элементов онтологии. Во время обработки запроса производится подготовка для его расширения, т.е. обработка для дальнейшего поиска релевантной информации в каждой из выбранных онтологий, затем осуществляется сам поиск и предоставление результатов поиска пользователю.

Анализ архитектуры и изучение систем информационного поиска, которые представлены на рынке, позволили сделать вывод о том, что нет необходимости разрабатывать подсистемы для морфологического разбора, семантического анализа и т.д. Аналогично и с алгоритмами, которые решают задачи непосредственно по поиску информации: можно использовать уже готовые варианты на базе классических методов поиска. В качестве готовых решений для поиска возможно использование таких поисковых серверов, как Sphinx, Apache Lucene, Xapian [8]. Для хранения онтологий предпочтительнее

использовать формат OWL (Ontology Web Language) – это формат, который позволяет хранить онтологии в XML [9].

При использовании подхода на основе онтологии предметной области система реализует трехуровневую архитектуру [10]. При такой архитектуре система включает в себя следующее:

1. Слой пользовательского интерфейса.
2. Слой бизнес-логики.
3. Слой доступа к данным.

На рис. 1 представлена диаграмма компонентов системы, которая соответствует описанной архитектуре.

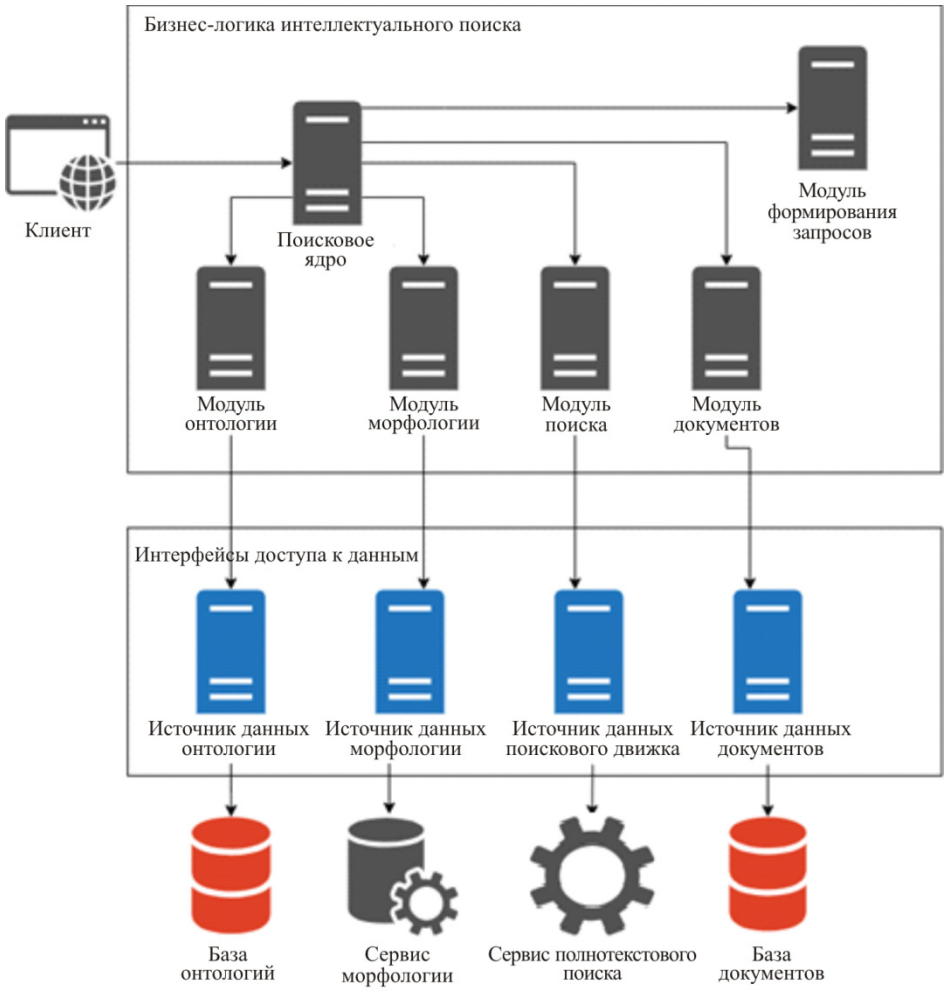

Рис. 1. Архитектура системы информационного поиска с использованием онтологии предметной области

Такая архитектура предоставляет ряд преимуществ:

1. Возможность изменения компонентов или их замены на другие без ущерба для работоспособности всей системы.

2. Открытость системы, которая предоставляет возможность для интегрирования в нее уже реализованных компонентов, например поисковых машин.

С учетом модульной структуры системы каждый из этапов обработки поискового запроса производится конкретным функциональным модулем:

1. Модуль морфологической обработки запроса проводит морфологический разбор, т.е. определяет, к какой части речи относится то или иное слово, какова его синтаксическая роль в предложении и т.д.

2. Модуль онтологической обработки для каждого элемента в запросе осуществляет поиск сходных элементов и формирует матрицу расширенных элементов запроса, т.е. схожих элементов.

3. Модуль формирования расширенных дополнительных запросов на основе расширенных элементов формирует новый запрос, в который включает эти элементы для расширения границ поиска.

4. Модуль поиска осуществляет поиск в хранилище данных по заданным элементам запроса.

5. Модуль обработки результатов поиска обрабатывает запрос и формирует ответ для отправки пользователю.

Архитектура с использованием онтологии является одной из первых, которую применили для создания систем информационного поиска. Такой подход позволяет реализовать довольно гибкую и масштабируемую систему, однако не предлагает решений для обеспечения отказоустойчивости системы, обработки ошибок поиска, повышения скорости работы системы. Исходя из этого, были предложены и другие способы организации систем информационного поиска, например архитектура, основанная на мультиагентном подходе.

3. Обзор подхода к реализации системы информационного поиска с использованием мультиагентной технологии

Интеллектуальный агент – это сервис, который самостоятельно выполняет задание, полученное от пользователя или другого сервиса. В системах информационного поиска агенты представляют собой про-

граммы, предназначенные для автоматического сбора, фильтрации и организации информации [11]. Мультиагентные системы обладают рядом преимуществ по сравнению с другими системами, разработанными с применением иного подхода к архитектуре:

1. Уменьшаются время, стоимость передачи данных и нагрузка на сеть.
2. Вычисления выполняются автономно и асинхронно.
3. Вычисления могут адаптироваться к условиям своего выполнения.
4. В данном случае в основе архитектуры – системы информационного поиска.

Исходя из этого, одним из наиболее эффективных подходов для реализации систем информационного поиска является подход с использованием мультиагентной технологии. В рамках такого подхода система строится как совокупность агентов, например таких, как агент пользователя или агент поиска.

Перед появлением программных агентов разрабатывались открытые системы, что явилось причиной появления архитектуры «клиент – сервер». При создании мультиагентной системы используются две модели, которые предоставляет архитектура «клиент – сервер»: «толстый» клиент (fat client) – «тонкий» сервер (thin server) и «тонкий» клиент (thin client) – «толстый» сервер (fat server) [12]. При этом применяется статический или динамический подход, обеспечивающий также передачу программного кода. Динамический подход опирается на мобильных агентов, которые, в отличие от статических, обладают возможностью перемещаться по сети, т.е. могут покидать клиентский компьютер и перемещаться на удаленный сервер для выполнения своих действий, после чего могут возвращаться обратно [13].

Мультиагентный сервис поиска информации в корпоративной информационной системе имеет архитектуру, которая представлена на рис. 2.

Предложенная архитектура состоит:

– из интерфейсного агента, отвечающего за связь между пользователем и системой;

– агента обработки запросов, необходимого для обработки введенных пользователем запросов;

– агента онтологии, обеспечивающего единообразное представление всех агентов мультиагентной системы о содержащихся онтологиях и хранение всей информации о них;

– агента регистрации, обеспечивающего регистрацию новых агентов и служб в мультиагентной системе;

– координатора рабочих групп, взаимодействующего с агентом регистрации путем получения запроса от него на факт наличия или отсутствия группы, в которой новый агент или служба просит регистрации;

– каталога агентов и каталога служб, необходимых для централизованного хранения информации обо всех присутствующих в системе агентах и службах соответственно, а также для поиска и формирования всех агентов и служб, доступных для взаимодействия;

– агентов-координаторов, являющихся связующим звеном между интерфейсным агентом и группой поисковых агентов, относящихся к той же онтологии, что и сам агент-координатор;

– поисковых агентов;

– агентов обработки результатов, взаимодействующих с агентом-координатором и интерфейсным агентом и служащих для формирования и ранжирования результатов поисковой выдачи по степени соответствия и значимости, а также производящих расчет показателей страниц данных на основе пользовательских действий через интерфейс программы для последующих итераций поиска [14].

Поиск информации с помощью системы, построенной по такой архитектуре, требует всего одного запроса через пользовательский интерфейс. Затем запрос перенаправляется с целью определения из текущей онтологии агентов-координаторов нужного агента. И уже после этого отправляется поисковым агентам для выполнения поиска. Полученные данные возвращаются агентам-координаторам, затем агенту обработки результатов с последующей передачей пользователю [15].

Использование мультиагентного подхода предоставляет следующие преимущества:

1. Снижение нагрузки на каналы передачи данных.
2. Автоматизация и интеллектуализация обработки информации.
3. Более быстрая обработка информации.
4. Сокращение времени сотрудника на поиск информации.

Несмотря на перечисленные плюсы, системы, основывающиеся на мультиагентном подходе, имеют и свои недостатки. В частности, такие системы довольно сложны в разработке, поскольку используют алгоритмы машинного обучения, например нейронные сети, которые зачастую сложно интерпретируются. Не для всех компаний такая сложность может быть оправданна.

Ввиду этих причин для разработки системы информационного поиска возможно использование более простого в реализации, но не менее гибкого и эффективного подхода к построению архитектуры на основе очередей сообщений. Более подробное описание данной архитектуры приводится далее.

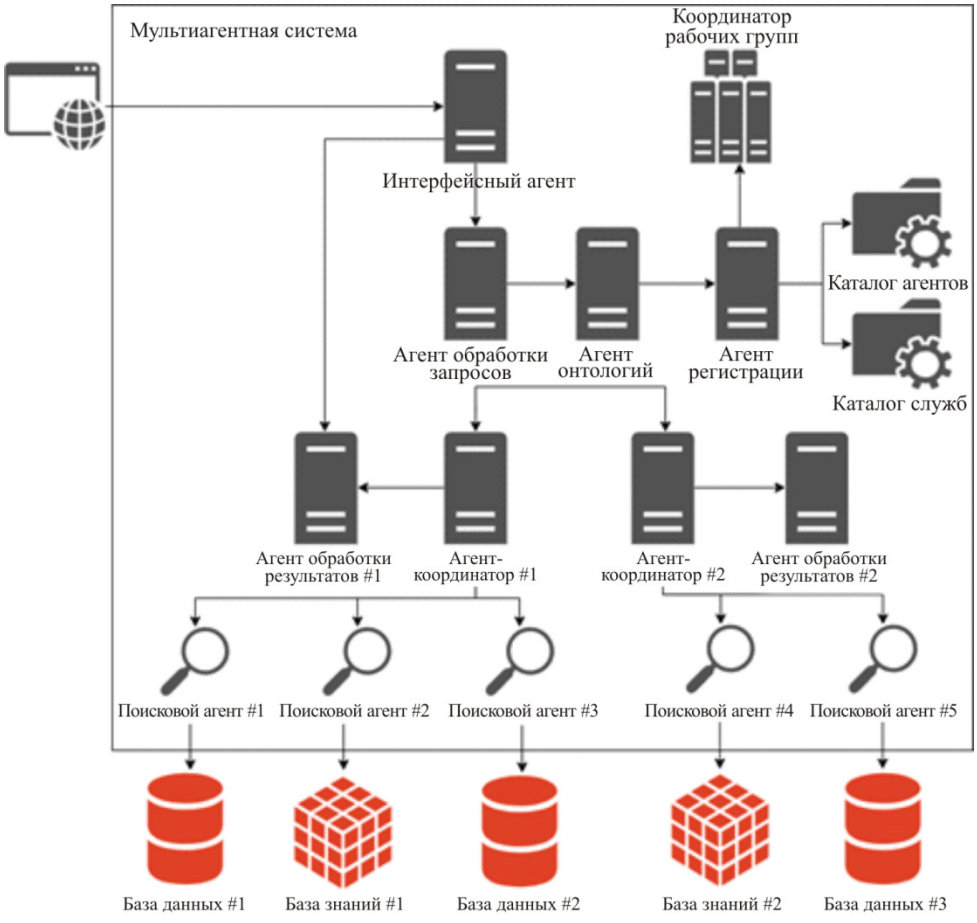

Рис. 2. Архитектура системы информационного поиска, реализованная с помощью мультиагентного подхода

4. Разработка архитектуры информационного поиска с использованием технологии обмена сообщениями

В этом подразделе приводится описание подхода к реализации системы информационного поиска в корпоративной информационной системе предприятия, в основе которой лежит архитектура с использованием очередей сообщений, а также раскрываются некоторые детали реализации. Постановка задачи и часть деталей реализации приводятся из первой части исследования [16], посвященного данной теме, но изложенного в более краткой форме без поддержки иллюстративным материалом, содержащим архитектурные диаграммы.

Для реализации системы информационного поиска на основе очереди сообщений предлагается использовать микросервисный подход [17–19]. Такой подход позволяет реализовать гибкую инфраструктуру, которая сможет обеспечить возможность расширения системы, т.е. добавления новых механизмов расчетов, использования разных языков программирования, а также обеспечит слабую связанность между уровнем представления и уровнем расчетов.

Для организации взаимодействия между частями системы предлагается использовать брокер обмена сообщениями [20]. Из решений с открытым исходным кодом, которые широко используются сегодня, можно выделить RabbitMQ и Kafka [21]. RabbitMQ представляет собой классический брокер обмена сообщениями, который не хранит их в очереди после успешной обработки. Но, в отличие от RabbitMQ, брокер Kafka позволяет хранить сообщения и после их обработки, в том числе предоставляет возможность новым потребителям получить всю предыдущую историю запросов. Такой функционал реализуется такими режимами потребления, как самый ранний (earliest) и последний (latest) соответственно [22].

Ввиду того, что Kafka позволяет хранить историю сообщений, предлагается использовать его в качестве брокера обмена сообщениями. Это решение позволит удовлетворить предъявляемым требованиям к расширяемости логики расчетов с учетом или без учета истории запросов. Также взаимодействие сервисов системы между собой через брокер сообщений позволяет реализовать возможность обработки ситуаций повторной доставки и разрывов соединения. Брокер сообщений полезен и в тех случаях, когда поисковой сервис не смог обработать сообщение запроса. В таком случае сервис отправляет сообщение о не-

удаче в сервис перезапуска поиска, который в свою очередь с определенным интервалом отправляет сообщения запроса в сервис поиска. На рис. 3 изображена диаграмма, демонстрирующая процесс перезапуска поиска при сбое обработки запроса поисковым сервисом.

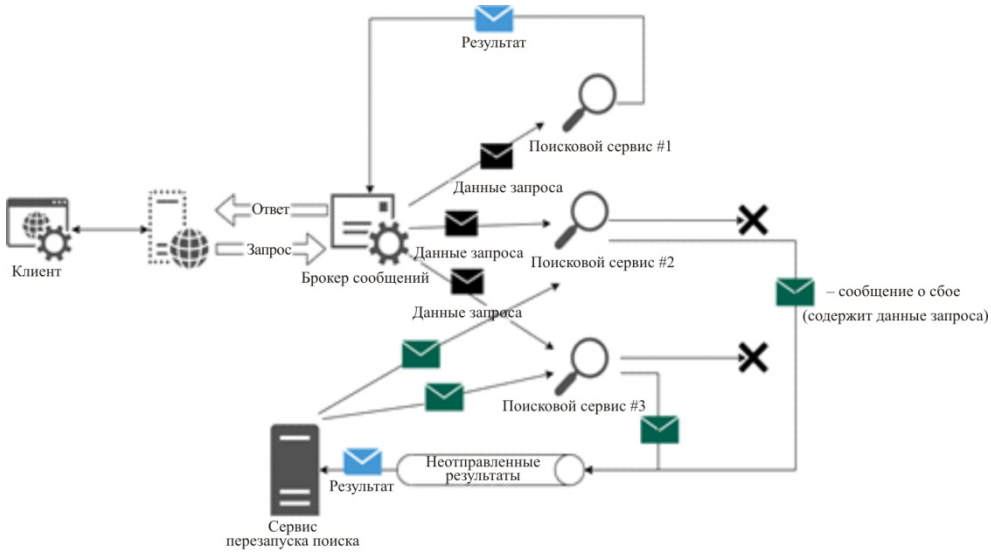

Рис. 3. Диаграмма процесса перезапуска поиска
при сбое обработки запроса поисковым сервисом

Рассмотрев основные архитектурные решения, такие как применение микросервисной архитектуры и выбор Kafka в качестве брокера обмена сообщениями, можно представить следующие составные части системы:

1. Пользователи – инициаторы первоначальных запросов поиска информации.

2. Web Application – веб-приложение, доступное с помощью веб-браузера, в котором происходит заполнение формы при создании запроса пользователями.

3. Web API – веб-сервер, в который приходят запросы пользователей на поиск информации и который сохраняет их в базу данных запросов и результатов.

4. База данных запросов и результатов – база данных, которая хранит в себе данные пользователей, исходные данные запросов и полученные результаты. Возможно разделение этого хранилища на несколько отдельных баз данных, причем возможно применение как SQL-, так и NoSQL-решений [23, 24].

5. Брокер обмена сообщениями – связующее звено между частями системы, которое обеспечивает хранение сообщений и реализует механизм их обработки потребителями.

6. Сервисы обсчета – части системы, где происходит поиск информации по заданным критериям из изначального запроса и по специализации каждой конкретной группы сервисов.

7. Сервисом расчета теоретически может являться как человек, сидящий за компьютером и реализующий определенный поиск, так и умный алгоритм поиска и анализа данных. Время обработки конкретного запроса пользователя может сильно отличаться от одной группы сервисов расчета к другой.

8. Сервис обработки результатов – сохраняет результаты работы сервисов обсчета в базу данных запросов и результатов.

9. Сервис мониторинга – обеспечивает наблюдение за состоянием и непротиворечивостью данных системы. Сигнализирует в случае неполноты данных, превышения ожидаемого времени обработки и передачи данных между частями системы, отказа частей системы.

Диаграмма компонентов системы изображена на рис. 4.

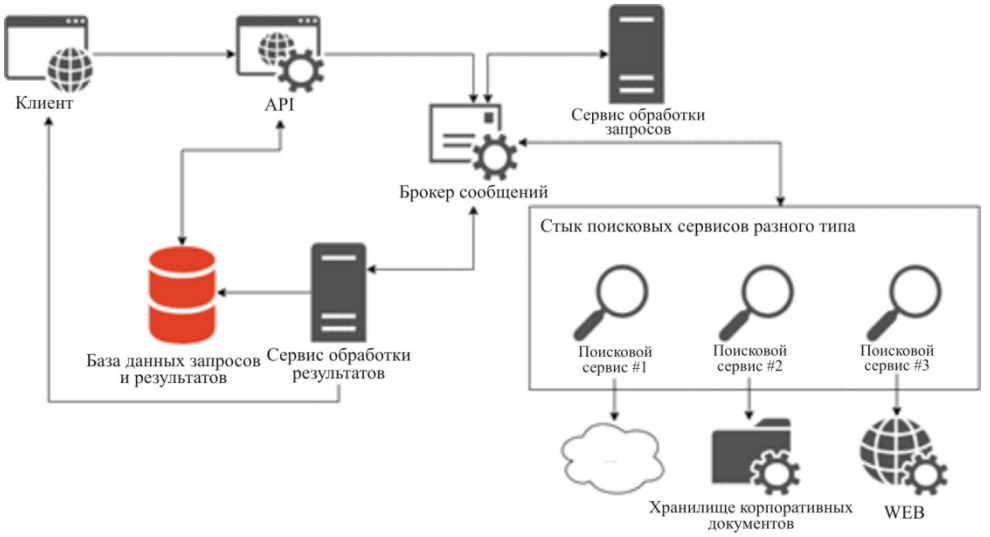

Рис. 4. Диаграмма компонентов системы информационного поиска на основе очереди сообщений

Очередь первоначальных запросов не должна очищаться и терять историю сообщений. Для этого она должна быть реализована не как

классическая fire and forget-очередь, когда после обработки сообщение удаляется, а наоборот, должна представлять собой хранимый лог сообщений (persistent messages log). Очередь результирующих сообщений может не хранить их всегда, а иметь период очистки (retention period) и удалять сообщения, например, по прошествии недели.

Как только часть результата получена и сохранена в базе данных, ей присваивается уникальный идентификатор и она может быть кеширована (cached) до тех пор, пока не будет произведен принудительный пересчет результатов по набору входных параметров.

Если рассматривать реляционное хранилище для результатов, полученных сервисами обсчета на поисковые запросы пользователей, то предлагается хранить их в таблице со следующим набором обязательных атрибутов:

1. result_hash – хранит MD5-хеш, полученный по объекту результата расчета, что позволяет считать контрольную сумму по результатам, в том числе для быстрого поиска одинаковых результатов.
2. result_type – хранит тип полученного результата.
3. schema_version – хранит версию контракта данных результата, что позволяет хранить в одной таблице сериализованный объект результата разных версий.
4. processor_type – уникальный в пределах системы человекочитаемый идентификатор типа сервиса обсчета. Может существовать несколько экземпляров сервиса с одним и тем же значением этого атрибута. Позволяет строить на его основе имя группы потребителей для брокера обмена сообщениями.
5. processing_occurrence_number – хранит в себе порядковый номер повторной обработки исходного запроса, который привел к созданию этого результата.
6. processing_session_id – уникальный идентификатор сессии запуска конкретного экземпляра сервиса обсчета, который получается за счет объединения текущей временной метки и UUID, что позволяет упорядочивать этот атрибут хронологически, что необходимо для поиска наиболее актуальных результатов и отфильтровывания устаревших.
7. message_id – уникальный идентификатор сообщения с результатом расчета, которое генерируется каждый раз, когда мы пытаемся послать сообщение брокеру, позволяет на его основе строить логику обработки повторной доставки одного и того же результата.

В любой системе информационного поиска может возникнуть ситуация, при которой будут найдены одинаковые результаты. Для решения этой проблемы предлагается использовать операцию SQL Merge для вставки результатов, с гарантией при этом отсутствия дублей [25].

Для того чтобы предлагаемый механизм работал ожидаемым образом, необходимо определить ограничения на уникальность комбинаций атрибутов, описанных ранее. Необходимо создать уникальный индекс на атрибут message_id, чтобы предотвратить образование дублирующихся записей при повторной доставке одного и того же сообщения от брокера обмена сообщениями.

Также необходим уникальный индекс по следующим атрибутам: parameters_id, result_type и result_hash. Этот индекс гарантирует, что для одной и той же комбинации входных параметров поиска мы не добавим одинаковые результаты одного и того же типа более чем один раз, даже если у них будут разные message_id. Например, сервис обсчета создает три результата и успешно отправляет только первые два из них, потом аварийно завершается и заново пробует обработать и отправить те же три результата, на этот раз успешно. Таким образом, нам необходимо сохранить два результата от первой попытки и только один результат от второй.

Операция SQL Merge используется, чтобы реализовать групповое потребление и сохранение результатов (batch consumption), с сохранением при этом гарантии транзакций ACID [26]. Это позволяет достигать большей пропускной способности обработки результатов по сравнению с обработкой и сохранением результатов по одному за раз с созданием транзакции на каждый отдельный результат. Также использование операции SQL Merge позволяет «элегантно» обрабатывать случай попытки вставить уже существующие данные и делать эту проверку средствами, встроенными в СУБД.

Заключение

В связи с ростом объемов электронных данных, которыми владеют организации, происходит и закономерное развитие систем, позволяющих обрабатывать эти данные и получать из них новую информацию. Естественно, что чем больше данных организация использует для осуществления своей деятельности, тем сложнее сотрудникам искать информацию внутри корпоративной информационной системы. Для облегчения поиска информации сотрудники пользуются специальными поисковыми системами.

Некоторые организации используют готовые решения поисковых систем, другие предпочитают разрабатывать их самостоятельно для удовлетворения всех потребностей компании. Важно правильно подходить к процессу проектирования и реализации подобного рода систем. В статье были приведены описания подходов к построению архитектуры систем информационного поиска: это архитектура на основе онтологии и архитектура с использованием мультиагентного подхода. На текущий момент одним из самых популярных методов построения микросервисных систем является архитектура с использованием очередей сообщений. Такой подход позволяет бороться со сложностью, делает компоненты инфраструктуры слабосвязанными, а также позволяет осуществлять горизонтальную масштабируемость системы, что немаловажно при разработке систем информационного поиска. В частности, рассматриваемый вариант построения архитектуры позволяет естественным образом добавлять новые сервисы поиска без изменения остальной инфраструктуры и кода других приложений.

Как было описано во втором подразделе статьи, для добавления новых поисковых сервисов с учетом истории предыдущих запросов поисковая система должна позволять хранить обработанные сообщения в очереди. Такой функционал предоставляет брокер сообщений Kafka. Использование данного брокера сообщений позволяет получить поисковую систему, которая реализует такие важные механизмы, как хранимый лог сообщений, групповая подписка на новые сообщения, режимы потребления «последний» и «самый ранний», конкурентное потребление. Также Kafka рассчитан на использование в высоконагруженных системах, что позволит удовлетворить требования большинства организаций к быстродействию и отказоустойчивости такого типа систем. Стоит отметить, что Kafka – продукт с открытой лицензией, что является плюсом в современном мире разработки программного обеспечения и позволяет сократить бюджет на разработку собственной поисковой системы.

Предлагаемый в статье подход для построения архитектуры информационного поиска на основе очередей сообщений может оказаться адекватным вариантом решения задач создания систем обновляемого информационного поиска для компаний, которые только планируют заняться решением подобной задачи или уже имеют готовые решения, но хотят найти им достойную замену. Рассматриваемый подход позволяет сделать систему гибкой и открытой к расширению, а также дает

возможность использовать для написания сервисов языки программирования, которые наиболее подходят для решения конкретной задачи поиска в определенной базе знаний.

Список литературы

1. Степунина О.А. Характерные черты и опасные тенденции информационного общества // Молодой ученый. – 2017. – № 22 (155). – С. 54–56.

2. Шокин Ю.И., Федотов А.М., Барахнин В.Б. Проблемы поиска информации. – Новосибирск: Наука, 2010. – 195 с.

3. Message Bus. – URL: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBus.html (accessed 05 February 2020).

4. Алешин Л.И., Максимов Н.В. Информационные технологии / Моск. финанс.-промыш. акад. – М., 2004. – 512 с.

5. Язык запросов [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/RequestLanguage (дата обращения: 07.02.2020).

6. Витухновская А.А. Обучение технологии и стратегии информационного поиска на основе дифференциальных признаков информационно-поисковых систем // Информационное общество. – 2013. – № 1-2. – С. 69–79.

7. Шмайлов Д. «Умный» поисковик для корпоративных систем и web // E. Dok / Э. Док. – 2011. – № 06 (06). – С. 8–9.

8. Полнотекстовый поиск в веб-проектах: Sphinx, Apache Lucene, Xapian [Электронный ресурс]. – URL: https://habr.com/ru/post/30594/ (дата обращения: 15.02.2021).

9. Спецификация языка веб-онтологий OWL [Электронный ресурс]. – URL: http://www.w3.org/TR/owl-features/ (дата обращения: 17.02.2021).

10. Savolainen J., Mlrneimi V. Layered architecture revisited – comparison of research and practice // Joint Working IEEE/IFIP Conference on Software Architecture 2009 and European Conference on Software Architecture 2009, WICSA/ECSA 2009, Cambridge, UK, 14–17 September 2009 / IEEE Computer Society. – Cambridge, UK, 2009. – P. 317–320.

11. Пономаренко Л.А., Филатов В.А., Цыбульник Е.Е. Агентные технологии в задачах поиска информации и принятия решений // Управляющие системы и машины. – 2003. – № 1. – С. 36–41.

12. Таненбаум Э., М. ванн Стеен. Распределенные системы. Принципы и парадигмы / пер. с англ. В. Горбункова. – СПб.: Питер, 2003. – 877 с.

13. Зобнин Б., Вожегов А. Мультиагентные системы. – Саарбрюккене: LAP Lambert Academic Publishing, 2014. – 156 с.

14. Федотов В.Б. Построение распределенной системы доступа к информационным ресурсам на основе многоагентной архитектуры // Современные технологии в информационном обеспечении науки: сб. науч. тр. / под ред. Н.Е. Каленова. – М.: Научный мир, 2003. – С. 63–73.

15. Алгулиев Р., Хайрахимова М. Некоторые аспекты организации и реализации мультиагентной системы поиска информации в распределенной информационной среде // Proceedings of the International Scientific Conference “Problems of Cybernetics and Informatics”, Baku, Azerbaijan, 24–26 October 2006 / Ин-т информ. техн. НАН Азербайджана. – Баку, 2006. – С. 31–34.

16. Шинкарев А.А. Об одном подходе к реализации информационной инфраструктуры обновляемого информационного поиска // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2021. – Т. 21, № 1. – С. 5–11. DOI: 10.14529/ctcr210101

17. Kalske M., Mäkitalo N., Mikkonen T. Challenges when moving from monolith to microservice architecture // Lecture Notes in Computer Science. – 2018. – Vol. 10544. – P. 32–47. DOI: 10.1007/978-3-319-74433-9_3

18. Namiot D., Sneps-Sneppe M. On micro-services architecture // International Journal of Open Information Technologies. – 2014. – Vol. 2, iss. 9. – P. 24–27.

19. Microservices. – URL: https://martinfowler.com/ articles/microservices.html (accessed 21 February 2020).

20. Ferreira D.R. Message brokers // Enterprise Systems Integration. – Berlin: Springer-Verlag Berlin Heidelberg, 2013. – P. 75–92.

21. Ayanoglu E., Aytas Y., Nahum D. Mastering RabbitMQ. – Packt Publishing, 2016. – 286 c.

22. Dobbelaere P., Esmaili K.S. Kafka versus RabbitMQ: A comparative study of two industry reference publish /subscribe implementations // Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems (DEBS 2017), Barcelona, Spain, 19–23 June 2017. – ACM, 2017. – P. 227–238.

23. Alshafie Gafaar Mhmoud Mohmmed, Saife Eldin Fatoh Osman SQL vs NoSQL // Journal of Multidisciplinary Engineering Science Studies (JMESS). – 2017. – Vol. 3, iss. 5. – P. 1790–11792.

24. Gessert F., Wingerath W., Ritter N. Polyglot persistence in data management // Fast and Scalable Cloud Data Management. – Springer, Cham, 2020. – P. 149–174.

25. Badia A. More SQL // SQL for Data Science. – Springer International Publishing, 2020. – P. 221–242.

26. Ba S., Yang X. Transaction Systems // The Rise of New Brokerages and the Restructuring of Real Estate Value Chain. – Singapore: Springer Singapore, 2018. – P. 43–93.

References

1. Stepunina A.O. Harakternye cherty i opasnye tendencii informacionnogo obshchestva [Characteristics and Dangerous Trends of the Information Society]. *Young Scientist*, 2017, vol.155, no.22, pp.54–56.

2. Shokin Y.I., Fedotov A.M., Barahnin V.B. Problemy poiska informacii [Information search problems]. Novosibirsk, Nauka, 2010, 195 p.

3. Message Bus, available at: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBus.html (Accessed 05 February 2020).

4. Aleshin L.I., Maksimov N.V. Informacionnye tekhnologii [Information Technology]. Moscow, Moskovskaya finansovo-promyshlennaya akademiya, 2004, 512 p.

5. Query language, available at: https://ru.wikipedia.org/wiki/Request Language (Accessed 07 February 2020).

6. Vitukhnovskaia A.A. Obuchenie tekhnologii i strategii informatsionnogo poiska na osnove differentsial'nykh priznakov informatsionno-poiskovykh sistem [Teaching information retrieval technology and strategy based on differential features of information retrieval systems]. *Informatsionnoe obshchestvo*, 2013, no. 1-2, pp.69–17.

7. Shmailov D. «Umnyi» poiskovik dlia korporativnykh sistem i web [«Smart» search engine for corporate systems and web]. *E.Dok*, 2011, no.06 (06), pp.8–9.

8. Full-text search in web projects: Sphinx, Apache Lucene, Xapian, available at: https://habr.com/ru/post/30594/ (Accessed 15 February 2021).

9. OWL Web Ontology Language Specification, available at: http://www.w3.org/TR/owl-features/ (Accessed 17 February 2021).

10. Savolainen J., Mlrneimi V. Layered Architecture Revisited – Comparison of Research and Practice. Joint Working IEEE/IFIP Conference on Software Architecture 2009 and European Conference on Software Architecture 2009, WICSA/ECSA 2009, IEEE Computer Society, 2009, pp. 317–320.

11. Ponomarenko L.A., Filatov V.A., Tsybul'nik E.E. Agentnye tekhnologii v zadachakh poiska informatsii i priniatiia reshenii [Agent technologies for information retrieval and decision-making]. *Upravliaiushchie sistemy i mashiny*, 2003, no.1, pp. 36–41.

12. Tanenbaum A.S., van Steen M. Distributed systems: Principles and Paradigms, New Jersey, Prentice hall, 1996, 803 p.

13. Zobnin B., Vozhegov A. Mul'tiagentnye sistemy [Multi-agent systems], Saarbrücken, LAP Lambert Academic Publishing, 2014, 156 p.

14. Fedotov V.B. Postroenie raspredelennoi sistemy dostupa k informatsionnym resursam na osnove mnogoagentnoi arkhitektury [Building a distributed system of access to information resources based on a multi-agent architecture]. Sovremennye tekhnologii v informatsionnom obespechenii nauki: cbornik nauchnykh trudov. Moscow, Nauchnyi mir, 2003, pp. 63–73.

15. Alguliev R., Khairakhimova M. Nekotorye aspekty organizatsii i realizatsii mul'tiagentnoi sistemy poiska informatsii v raspredelennoi informatsionnoi srede [Some aspects of the organization and implementation of a multi-agent information retrieval system in a distributed information environment]. Proceedings of the International Scientific Conference “Problems of Cybernetics and Informatics", Baku, Azerbaijan, Institute of information technologies ANAS, 2006, pp. 31–34.

16. Shinkarev A.A. Ob odnom podkhode k realizatsii informatsionnoi infrastruktury obnovliaemogo informatsionnogo poiska [On One Approach to Imple-

mentation of Information Infrastructure for Renewable Information Search]. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2021, vol. 21, no. 1, pp. 5–11. DOI: 10.14529/ctcr210101.

17. Kalske M., Mäkitalo N., Mikkonen T. Challenges When Moving from Monolith to Microservice Architecture. *Lecture Notes in Computer Science*, 2018, Vol. 10544, pp. 32-47. DOI 10.1007/978-3-319-74433-9_3

18. Namiot D., Sneps-Sneppe M. On micro-services architecture. *International Journal of Open Information Technologies*, 2014, vol. 2, iss. 9, pp. 24–27.

19. Microservices, available at: https://martinfowler.com/articles/microservices.html (Accessed 21 February 2021).

20. Ferreira D.R. Message Brokers. in: Enterprise Systems Integration. Berlin, Springer-Verlag Berlin Heidelberg, 2013, pp. 75–92.

21. Ayanoglu E., Aytas Y., Nahum D. Mastering RabbitMQ, Packt Publishing, 2016, 286 p.

22. Dobbelaere P. and Esmaili K.S. Kafka versus RabbitMQ: A comparative study of two industry reference publish / subscribe implementations. *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*. ACM, 2017, pp. 227–238.

23. Alshafie Gafaar Mhmoud Mohmmed, Saife Eldin Fatoh Osman SQL vs NoSQL. *Journal of Multidisciplinary Engineering Science Studies (JMESS)*, 2017, vol. 3, iss. 5, pp. 1790–11792.

24. Gessert F., Wingerath W., Ritter N. Polyglot Persistence in Data Management. In: Fast and Scalable Cloud Data Management, Springer, Cham, 2020, pp.149–174.

25. Badia A. More SQL. In: SQL for Data Science, Springer International Publishing, 2020, pp. 221–242.

26. Ba S., Yang, X. Transaction Systems. In: The Rise of New Brokerages and the Restructuring of Real Estate Value Chain, Singapore, Springer Singapore, 2018, pp.43–93.

Статья получена: 28.02.2021

Статья принята: 09.03.2021

Сведения об авторах

Логиновский Олег Витальевич (Челябинск, Россия) – доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий кафедрой «Информационно-аналитическое обеспечение управления в социальных и экономических системах», Южно-Уральский государственный университет (454080, Челябинск, пр. Ленина, 76, e-mail: loginovskiyo@mail.ru).

Шинкарев Александр Андреевич (Челябинск, Россия) – кандидат технических наук, докторант кафедры «Информационно-аналитическое обеспечение управления в социальных и экономических системах», Южно-Уральский государственный университет (454080, Челябинск, пр. Ленина, 76, e-mail: sania.kill@mail.ru).

Коваль Максим Евгеньевич (Челябинск, Россия) – магистрант кафедры «Информационно-аналитическое обеспечение управления в социальных и экономических системах», Южно-Уральский государственный университет (454080, Челябинск, пр. Ленина, 76, e-mail: kovalmax06@gmail.com).

About the authors

Oleg V. Loginovskiy (Chelyabinsk, Russian Federation) – Dr. Habil. in Engineering, Professor, Honored Scientist of the Russian Federation, Head of the Department of Informational and Analytical Support of Management in Social and Economic Systems, South Ural State University (76, Lenin av., Chelyabinsk, 454080, e-mail: loginovskiyo@mail.ru).

Alexander A. Shinkarev (Chelyabinsk, Russian Federation) – Ph.D. in Engineering, Doctoral Student, Department of Informational and Analytical Support of Management in Social and Economic Systems, South Ural State University (76, Lenin av., Chelyabinsk, 454080, e-mail: sania.kill@mail.ru).

Maxim E. Koval (Chelyabinsk, Russian Federation) – Master, Department of Informational and Analytical Support of Management in Social and Economic Systems, South Ural State University (76, Lenin av., Chelyabinsk, 454080, e-mail: kovalmax06@gmail.com).

Библиографическое описание статьи согласно ГОСТ Р 7.0.100–2018:

Логиновский, О.В. Разработка архитектуры систем информационного поиска на основе очередей сообщений в корпоративных информационных системах / О. В. Логиновский, А. А. Шинкарев, М. Е. Коваль. – текст : непосредственный. – DOI: 10.15593/2499-9873/2021.1.07 // Прикладная математика и вопросы управления = Applied Mathematics and Control Sciences. – 2021. – № 1. – С. 119–140.

Цитирование статьи в изданиях РИНЦ:

Логиновский О.В., Шинкарев А.А., Коваль М.Е. Разработка архитектуры систем информационного поиска на основе очередей сообщений в корпоративных информационных системах // Прикладная математика и вопросы управления. – 2021. – № 1. – С. 119–140. – DOI: 10.15593/2499-9873/2021.1.07

Цитирование статьи в references и международных изданиях:

Cite this article as:

Loginovskiy O.V., Shinkarev A.A., Koval M.E. Development of architecture of message queue based information search systems for enterprise information systems. *Applied Mathematics and Control Sciences*, 2021, no. 1, pp. 119–140. DOI: 10.15593/2499-9873/2021.1.07 (*in Russian*)