

Новая модель распределённого библиографического и полнотекстового поиска в J-ИРБИС 2.0

The new model of broadcast bibliographic and full-text search in J-IRBIS 2.0 unit

К.Е. Соколинский

Международная ассоциация ЭБНИТ, Управление информационно-образовательных ресурсов СПбГУТ

Санкт-Петербург, Россия

Kirill Sokolinsky

International Association of Electronic Libraries and New Information Technologies Users and Developers, Department of Information Education Resources, SPbSUT

St. Petersburg, Russia

Анализируется практика использования распределённого поиска и его новая модель, реализованная в модуле САБ ИРБИС – J-ИРБИС 2.0

The practice of using a broadcast search and its new model, implemented in the module of IRBIS ALIS – J-IRBIS 2.0

Актуальность вопроса интеграции библиографических и полнотекстовых ресурсов библиотек подтверждается сегодня возникновением целого ряда библиотечных объединений различных уровней. Вузы одного профиля стремятся к созданию сводных каталогов и объединению электронных коллекций ради минимизации затрат. Муниципальные библиотеки крупных городов и регионов формируют сводные каталоги для внедрения технологии единого читательского билета и консолидации краеведческих ресурсов. На федеральном уровне реализуются сразу несколько интеграционных проектов, среди которых выделяются ИС ЭКБСОН, НЭБ, СКБР, АРБИКОН, ИСК НТЛ.

В эпоху высокоскоростных каналов связи и распространения протоколов обмена метаданными (Z39-50, SRU/SRW, OAI-PMH) было бы закономерно широкое использование технологии интеграции на основе распределённого поиска. Логическое объединение результатов поиска в физически распределённых базах позволяет выполнять поиск исключительно за счёт функциональных возможностей программного обеспечения, без существенных затрат вычислительных ресурсов. Поиск в базах, состоящих из миллионов записей, оказывается возможным средствами малопроизводительных серверов. При этом исчезает потребность в организационных структурах, выполняющих координационные функции, без которых трудно представить создание сводных каталогов и консолидированных поисковых индексов. Использование распределённого поиска обеспечивает гарантированную актуальность результатов и возможность работы со значительными массивами записей серверов-источников. Кроме того, распределённый поиск может выполняться в проприетарных базах данных, так как не предполагает их полное копирование [1].

В то же время, практика использования модели распределённого поиска сталкивается с целым рядом существенных проблем. Данная модель зависима исключительно от скорости и надёжности серверов-источников, стабильности связи между ними и качества реализации коммуникативных протоколов. Как следствие, скорость в значительной степени определяется количеством участвующих в поиске серверов. Добавление каждого сервера-источника увеличивает вероятность задержек. При использовании источников, применяющих различные протоколы или различные профили одного протокола, возникает необходимость ориентироваться на источник, который предоставляет минимум поисковых возможностей. Если какой-то поисковый атрибут не поддерживается одним из серверов, его нецелесообразно применять.

С начала 2000-х годов был опубликован целый ряд статей и научных работ, в которых анализировались пути преодоления ограничений распределённого поиска. Исторически сформировалось два основных подхода к его реализации: с группировкой и без группировки.

В первом случае результатом поиска становится библиографический список, во втором – гиперссылки на списки найденных записей в каждом из источников.

Первый подход во многом ориентирован на использование специфических особенностей протокола Z39-50, в котором разделяются операции поиска и извлечения записей. Поиск является самой простой операцией протокола. Он позволяет определить текущий статус сервера и дать возможность пользователю работать в режиме извлечения записей лишь с теми серверами, которые продемонстрировали соответствие требованиям (например, скорости отклика). Благодаря выводу гиперссылок на результаты поиска сразу после получения отклика от серверов-источников, у пользователя появляется возможность сразу начать просмотр записей. Это допустимо сделать, не дожидаясь полного завершения поискового процесса, который зачастую бывает достаточно продолжительным. Кроме того, за счёт дискретного представления списков записей, полученных от различных серверов, исключается ситуация, при которой сбой одного сервера влияет на список результатов. Если сервер функционирует, пользователь может перейти к просмотру записей. Если нет – пользователь сразу видит это и осуществляет просмотр результатов, полученных от других действующих серверов.

Данный подход широко использовался в таких проектах как “Сигла” и “Корпоративная сеть библиотек Москвы”. Его наиболее современная реализация -- новая поисковая система ГПНТБ России, позволяющая реализовывать поиск не только с помощью Z39-50, но и через специализированные API (в т.ч. Summon).[3]

Второй подход предполагает объединение результатов поиска. Запросы на поиск, извлечение записей следуют один за другим. Результаты поиска по распределённым ресурсам отображаются единым списком. При этом, как правило, они не открываются пользователю до тех пор, пока не выполнен полный опрос всех источников.

[[Расширенный поиск](#)]

CONSENSUS OMNIUM: корпоративная сеть библиотек Урала

Базы данных

Зональная научная библиотека Уральского федерального университета (ул. Тургенева, ул. Куйбышева) ^
Зональная научная библиотека Уральского федерального университета (ул. Мира)
Научная библиотека Уральского государственного экономического университета
Научная библиотека Уральского государственного лесотехнического университета
Научная библиотека Уральского государственного горного университета
Научная библиотека Уральского государственного педагогического университета
Научная библиотека Российского государственного профессионально-педагогического университета
Библиотека им. С. Р. Коган Уральского государственного университета путей сообщения
Библиотека Уральского института РАНХиГС v

Автор	▼	соколинский
И	▼	
Заглавие	▼	
И	▼	
Тематика	▼	
Количество извлекаемых записей	<input type="text" value="20"/>	<input type="button" value="Поиск"/> <input type="button" value="Просмотр"/>

[[Расширенный поиск](#)]

Рисунок 1 Интерфейс OPAC Руслан

Наиболее распространённым решением, использующим этот подход, является OPAC САБ Руслан. Он стал базой для многих региональных библиотечных корпораций. Разработанный в начале 2000-х годов, OPAC реализовывал исключительно широкий для своего времени функционал. Записи из нескольких внешних источников, поддерживающих Z39-50, не только выводились единым списком, но и отображались в виде единообразных

библиографических записей. Работа с источниками выполнялась в асинхронном (параллельном) режиме – запросы одновременно осуществлялись по нескольким источникам.

Для решения проблемы эпизодической недоступности серверов, в дополнение к поисковой системе предусматривался специальный робот, который через определённые промежутки времени анализировал состояние доступных источников и предотвращал соединение со сбойными серверами.

Но начиная с 2000-х гг., интерфейсные и функциональные возможности ОПАС не претерпели существенных изменений. Поэтому, несмотря на широкое распространение, данное решение не отражает современные тенденции развития технологий.

Современным инструментом распределённого поиска второго типа стал “Мета-сервер”, разработанный для Томского библиотечного консорциума. Это решение интересно своей программной платформой в виде REST API, который может использоваться другими разработчиками. В нём реализуется работа с группами Z39-50 и SRU/SRW серверов, отслеживание их работоспособности с помощью робота, слияние и ранжирование полученных в итоге записей. [1]

Оригинальным решением, сочетающим в себе черты двух подходов (с выводом записей и выводом ссылок), является “Виртуальный каталог” Технологического института Карлсруэ (Karlsruher Institut für Technologie). В рамках проекта реализован общедоступный интерфейс, позволяющий выполнять поиск не только по немецкоязычным ресурсам, но и по целому ряду крупнейших сводных каталогов (в т.ч. OCLC) и каталогов национальных библиотек. Наряду с библиотеками, в качестве источников представлены издательства и электронные архивы открытого доступа (DOAJ, Google Books и др.). Записи здесь выводятся по мере получения и не группируются, а объединяются под гиперссылками на источники. Чтобы просмотреть полный список результатов каждого источника, требуется осуществить переход к источнику, а записи отображаются в интерфейсе поисковой системы в том библиографическом формате, который способен генерировать источник. Поэтому формат отображения результатов для разных источников существенно отличается. Такой сервис оказался примером успешной реализации распределённого поиска и ещё в начале 2000-х годов обрабатывал до 750 тыс. запросов на немецком языке в месяц.

Тем не менее, несмотря на успешные примеры реализации сервисов распределённого поиска за рубежом, интерес к технологии снижается. Количество публикаций по данной теме по сравнению с серединой 2000-х годов существенно уменьшилось. Большинство федеральных проектов создаются на основе технологий консолидации данных. На региональном уровне и на уровне университетских объединений при реализации интегративного поиска отдаётся предпочтение сводным каталогам. Широкое распространение морально устаревших решений в России породило в библиотечном сообществе скептическое отношение к распределённому поиску, возникло недоверие к его эффективности, надёжности и эргономичности.

Можно выделить целый ряд проблем, которые существенно снижают привлекательность технологии:

1. **Низкая скорость.** Распределённый поиск обычно проигрывает в скорости поиску в консолидированном массиве. Это обусловлено наличием таких факторов, как использование каналов связи интернет и работа с большим количеством соединений.

2. **Ограниченные возможности.** Применение протокола Z39.50 для реализации распределённого поиска существенно ограничивает возможности поискового инструментария. И чем большее количество серверов с различным программным обеспечением участвует в поиске, тем более вероятно, что возникнет необходимость отказаться от отдельных поисковых возможностей.

3. **Нестабильность распространённого ПО.** Большинство Z39.50 серверов, используемых для распределённого поиска в России, не отличаются стабильностью работы [2]. Они являются частью трёхзвенной архитектуры и взаимодействуют с СУБД. Это определяет их зависимость от стабильности СУБД, САБ в целом и процессов конверсии данных. В то же время, более современные протоколы SRU/SRW и OAI-PMH ещё не получили широкого распространения.

4. **Необходимость ручной настройки системы поиска.** Для приемлемого качества работы с серверами Z39.50 требуется эмпирически отслеживать их возможности по поддержке форматов, кодировок и представлений данных. Один и тот же универсальный

сервер (например, ZooPark или Руслан) предоставляет разные возможности при его использовании с различными САБ.

5. Проблемы разграничения прав доступа к электронным версиям. Доступ к электронным документам в удалённых источниках возможен лишь в том случае, если ссылки на них являются открытыми. Реализацию какого-либо авторизованного доступа стандартные протоколы, применяемые при распределённом поиске (Z39-50, SRU/SRW, OAI-PMH), не предусматривают. А специализированные протоколы (SAML) требуют для своего внедрения преодоления высокого технологического барьера.

6. Отсутствие универсальных технологий полнотекстового поиска. Z39.50 и все протоколы, на нём основанные, предусматривают полнотекстовый поиск, но информация о поддержке этих возможностей в конкретных программных реализациях российских Z39.50 серверов отсутствует. Работа сервера, который бы обеспечивал не только выборку данных, но и отображение контекста, требует преодоления высоких технологических барьеров. И, несмотря на то, что многие решения (Т-Libra, Библиопортал, J-ИРБИС 2.0) реализуют распределённый полнотекстовый поиск, в библиотечном сообществе пока отсутствуют его общепризнанные стандарты.

Таким образом, вопрос эффективности применения распределённого поиска может быть решен исходя из перспективы преодоления связанных с ним проблем. Допустимо утверждать, что ключевые из них были устранены в таком проекте как ИРБИС-корпорация. Это решение стало третьим по количеству библиотек-источников корпоративным объединением в России. Ежедневно она обслуживает до 7 тыс. запросов, а среднее время вывода результата составляет всего несколько секунд. На сегодняшний день, ИРБИС-корпорацию применяют более чем 15 000 сотрудников библиотек. Успешное функционирование этой системы позволяет опровергнуть многие негативные стереотипы, связанные с распределённым поиском.

Критерии эффективности распределённого поиска

Критерии эргономики интерфейса

Анализ успешного опыта работы распределённых поисковых систем позволяет выделить условия востребованности, требования к интерфейсу и архитектуре.

В первую очередь, это соответствие привычным для пользователя интерфейсным стандартам. Как показывает статистика, работая с библиотечными поисковыми системами, типичный пользователь желает видеть поисковый интерфейс идентичный Google и Yandex – простой и понятный. Продвинутый пользователь хочет получить более широкий перечень функций, но и он не готов считаться с проблемами поисковой системы, обусловленными её технической структурой. Идеальная система распределённого поиска должна быть частью привычного ОРАС, а не автономной системой.

Если типичные пользователи системы -- студенты, условием востребованности распределённого поиска будет его включение в интерфейс каталога или электронной библиотеки. Если пользователи – читатели районной ЦБС, то поисковая система должна быть включена в интерфейс сайта конкретной ЦБС. Пользователь должен в единообразной форме получать данные из локальных баз и внешних систем.

Учитывая, что поисковые системы в интернете возвращают результаты в течение нескольких десятых долей секунды, выполнение запроса, более чем, в течение 60 секунд расценивается пользователем как зависание[2]. Поэтому исключительное значение приобретает реалистичная индикация прогресса, позволяющая заключить, в какой степени запрос был выполнен и сколько ещё предстоит ждать. В противном случае нетерпеливый пользователь будет перезагружать страницу. «Неопределённая» индикация процесса, сообщающая лишь о том, что система занята, также не вызывает доверия.

Google даёт пример того, как результаты поиска могут отображаться по ходу ввода запроса пользователем (далее такой принцип вывода будет называться здесь перманентным поиском). И хотя эта технология пока не получила большого распространения в библиотечных системах, у неё есть большой потенциал. Перманентный поиск сводит к минимуму объём вводимых данных и позволяет легко выявить ошибки при запросе.

Технические критерии эффективности

Можно выделить ряд технических характеристик системы, позволяющих ей обеспечивать приемлемый уровень надёжности и скорости. Учитывая разнообразие коммуникативных протоколов (Z39.50, SRU/SRW, OAI-PMH) и различных возможностей каждого сервера-источника записей, требуется обеспечить работу не только с предпочтительным для источника протоколом, но и с конкретным набором атрибутов, корректно поддерживаемым источником. Распределённая поисковая система должна обеспечивать поддержку различных кодировок и форматов метаданных. Например, система должна запрашивать записи в том формате, который является базовым для удалённой системы.

Помимо поддержки стандартизированных протоколов важна поддержка оригинальных протоколов/API интегрируемых систем (Summon, EBSCO, САБ ИРБИС). По своим возможностям эти протоколы, как правило, значительно превосходят универсальные в аспекте надёжности, скорости и качества предоставляемых данных. Исключение потребности в преобразованиях, интеграции с чужеродными системами, взаимодействие по оригинальным (native) протоколам всегда даёт значительные преимущества. Исчезает и крайне актуальная для библиотечного сообщества проблема соблюдения коммуникативных форматов, их различной интерпретации.

Одной из важных характеристик системы распределённого поиска является корректная обработка ошибок при поиске. Сбой на одном из серверов не должен приводить к искаженному отображению результатов поиска. В то же время, пользователь должен получать информацию о том, какие источники не участвовали в поиске.

Описанные выше требования позволяют разработать универсальную модель распределённого поиска, отвечающую всем указанным выше характеристикам.

Новая модель распределённого поиска и её элементы

Предлагаемая модель основывается на предположении, что поиск является лишь одной из функций используемой библиотекой САБ. Соответственно, при поиске через ОПАС существует потребность объединения результатов в локальных базах и внешних источниках. Допускается, что у пользователя есть возможность выполнять одновременный поиск не более чем в 100 источниках и эти источники доступны через различные коммуникативные протоколы. Мы исходим из того, что в качестве базовой системы автоматизации применяется одна из распространённых российских САБ (ИРБИС, MARC-SQL, Руслан), которые используют для формирования библиографических описаний по ГОСТ 7.1-2003 интерпретируемый язык.

1) Оценка оптимального размера группы запросов. Часть операций (п. 6-12), связанных с поиском и обработкой результатов поиска, выполняется в параллельных потоках. Это позволяет более эффективно использовать ресурсы современных многопроцессорных серверов и одновременно обращаться к целому ряду источников. Но в то же время, важно чтобы суммарное количество поисковых процессов согласовывалось с вычислительными возможностями сервера.

Наиболее ресурсоёмкой составляющей процесса обработки является процесс конвертирования записей (п.7) и формирования библиографических описаний -- расформатирование (п.11). В большинстве САБ расформатирование реализуется средствами интерпретируемых (скриптовых языков), поэтому требует значительного количества процессорного времени. В каждом конкретном случае длительность расформатирования определяет сложность описания. Например, полное библиографическое описание в соответствии с ГОСТ 7.1-2003 формируется дольше, чем библиографической ссылки в соответствии с ГОСТ 7.0.5-2008. Кроме того, если библиографическое описание формируется из нескольких составных элементов (например, предметные рубрики, полочный индекс и т.п.), ресурсоёмкость его создания будет определяться совокупностью ресурсоёмкости для всех элементов.

2) Сортировка источников Сортировка источников определяет порядок выполнения запросов к ним. При условии, что скорость отклика источников одинакова и ранжирование не применяется, сортировка целиком определяет порядок отображения записей из разных источников.

Сортировку допустимо выполнять исходя из полезности источника. Тогда чем более высокий рейтинг у источника, тем больше вероятность, что пользователь получит результат от него первым. Учитывая, что обычно источники неравнозначны (например, РГБ для российского читателя более доступна, чем Библиотека конгресса), это позволяет отображать результаты самых предпочтительных источников первыми.

Сортировка по скорости отклика ориентирует поиск на максимизацию скорости вывода. В случае приоритетного опроса самых быстрых источников, появляется возможность предоставить первые результаты как можно более оперативно.

3) Группировка источников. Запросы должны выполняться асинхронно, так это обеспечивает максимизацию скорости. Чтобы избежать перегрузок сервера, требуется разбиения запросов на последовательно (синхронно) выполняемые группы.

Запросы должны группироваться, исходя из порядка сортировки. Размер группы определяется ресурсоёмкостью запроса. В рамках группы порядок запросов, за счёт асинхронного выполнения, не отражается на итоговом результате. Но номер группы, в которой находится запрос, полностью определяет положение результатов его выполнения в общем результате поиска.

Практика ИРБИС-корпорации показала, что группировка 50 запросов позволяет избежать перегрузок системы, и при этом поиск выполняется достаточно быстро.

4) Трансляция запроса. Синтаксис запросов к удаленным источникам, как правило, отличается от базового синтаксиса запросов САБ. Поэтому требуется трансляция (конвертация) сформированного ОПАС запроса.

Трансляция всегда должна выполняться индивидуально для каждого удалённого сервера-источника, так как даже при использовании одного и того же протокола для всех источников, набор поддерживаемых атрибутов может существенно отличаться. Кроме ограничения по атрибутам поиска, требуется учитывать архитектурные ограничения серверов-источников. К таким неявным ограничениям относится, например, количество в запросе терминов, объединённых оператором “ИЛИ”. К ним же можно отнести и недопустимость определённых сочетаний атрибутов. Таким образом, трансляция не всегда позволяет обеспечить полное семантическое соответствие конвертированного запроса исходному, но должна к этому стремиться.

Ограничением выступает также конкретная реализация протокола или синтаксис протокола в целом. Например, несмотря на развитый синтаксис, протокол Z39.50 в принципе не предполагает возможности поиска по ключевым словам. Поэтому данный вид поиска приходится заменять на поиск по всем терминам записи.

5) Выполнение группы запросов на поиск и извлечение записей. Из группы запросов, определённой в рамках предыдущих этапов, формируются независимые процессы поиска и обработки полученных данных. Количество этих процессов соответствует количеству запросов в группе. Каждый запрос независимо от других обращается к серверу, преобразует и кэширует полученные в результате поиска данные. Количество полученных записей и ошибки поиска фиксируются в поисковой сессии.

6) Блокировка неактивных серверов-источников. Эпизодическая недоступность отдельных серверов в результате их неработоспособности, перегрузки или проблемы с каналами связи – неизбежное явление при распределённом поиске. Типовое решение этих проблем с помощью периодического запуска сканеров-роботов существенно увеличивает время, в течение которого пользователи будут предпринимать попытки обращения поисковой системы к проблемным серверам. В то же время, как показывают многочисленные исследования, именно недоступность серверов является главным фактором, провоцирующим задержки при распределённом поиске.

Как правило, поисковая система ожидает отклика от сервера-источника максимально возможное время, отведённое на запрос (timeout). В результате один сбойный сервер обычно является причиной существенного замедления выполнения группы запросов. Поэтому блокировка сервера-источника оправдана немедленно после обнаружения первой проблемы. Поскольку многие проблемы доступности сервера носят эпизодический характер, блокировка должна быть временной. После истечения определённого срока диагностика сервера-источника должна повторяться и при повторном неудовлетворительном результате блокировка устанавливается уже на больший промежуток времени. Учитывая, что недоступность многих серверов бывает крайне длительной и продолжается месяцы,

оправданным является экспоненциальное увеличение времени блокировки при каждом новом обращении к серверу.

Снятие блокировки, в отличие от установки, целесообразно выполнять средствами программы-робота. В этом случае тестирование источника не отражается на выполнении запросов пользователей.

7) Конверсия записей. Конверсия обеспечивает унификацию формата и представления записей. Это позволяет выполнить расформатирование с помощью единого для всех записей алгоритма. Распределённые поисковые системы, как правило, работают с одним из двух представлений записи – ISO 2709 и XML. Наиболее распространёнными форматами являются RUSMARC, UNIMARC, USMARC и Dublin Core. Но в ряде случаев сервера-источники возвращают записи в собственных, оригинальных форматах (например, специальные упрощённые форматы, используемые в САБ ИРБИС).

Преобразование записей в системе, применяющей для этого скриптовые языки, является продолжительным и ресурсозатратным процессом, выполнение которого целесообразно осуществлять в режиме распараллеливания.

8) Формально-логический контроль конвертированных записей. В библиотечных каталогах часто появляются записи, не соответствующие минимальным требованиям качества. Это могут быть, например, записи в кодировке отличной от кодировки каталога в целом, пустые записи, которые ещё не завершены сотрудниками и др. Они должны фильтроваться, так как они не представляют интереса для пользователей.

9) Исключение неактуальных элементов. Полученные из удалённых источников записи, как правило, бывают избыточными с точки зрения состава имеющихся в них данных. Сведения об экземплярах, данные об учебном назначении, признак “автор-сотрудник” и другие второстепенные элементы не представляют интереса для пользователей корпоративных проектов, поэтому оправдано их удаление.

10) Дополнение необходимыми данными. Правильное отображение записи часто требует корректных данных в кодированных полях (маркер записи, поля 1XX MARC форматов). Но в тех организациях, где САБ не контролирует правильность заполнения полей, они очень часто вводятся с ошибками. Поэтому в ходе унификации записей оправдана корректировка этих данных или их искусственное формирование.

Кроме того, в записи желательно указание сиглы хранения и, если это необходимо, обратной ссылки на каталог, в котором находится оригинальная версия записи. Поэтому имеет смысл динамическое формирование отдельных полей.

11) Формирование библиографических описаний. Автоматическая трансформация набора библиографических данных в библиографическое описание (расформатирование) по ГОСТ 7.1-2003 является ресурсозатратным и часто сопровождается ошибками. Любая ошибка при заполнении полей записи может негативно сказаться на процессе. Кроме того, формирование качественного библиографического описания пока вызывает сложности у многих разработчиков САБ. Поэтому данный этап допустимо заменить простой генерацией набора подписей к полям.

Тем не менее, если сегодня библиотеками заполняется до 1000 элементов библиографической записи, предусмотренных коммуникативными форматами, скрытие их от пользователя и ограничение формата отображения 15 элементами Dublin Core не целесообразно. Кроме того, многие пользователи электронных каталогов и библиотек применяют поисковые системы в научных и учебных целях, поэтому они нуждаются в полных библиографических описаниях.

12) Кэширование и агрегация в реляционной БД. Кэширование или временное хранение наиболее часто востребованных данных реализуется в информационных системах тогда, когда требуется увеличить скорость получения данных. В описываемой модели распределённого поиска кэш выполняет ещё и функцию агрегации результатов, полученных от отдельных поисковых процессов. Поэтому кэширование является неотъемлемым элементом системы.

Стандартное кэширование индифферентно к содержимому кэшируемой информации. Любая совокупность данных порождает новую дискретную единицу в хранилище кэширующего сервера, доступ к которой реализуется по уникальному ключу. Библиографические записи могут кэшироваться таким образом, но при этом неизбежно перманентное увеличение хранилища и снижение его эффективности. Это особенно вероятно

при выполнении множества родственных запросов, в которых результаты обычно отличаются на 10-20%.

Поэтому, оправданным является кэширование в реляционную структуру, где дублирование данных сводится к минимуму, а дополнительными затратами ресурсов на работу автономного SQL сервера допустимо пренебречь. Они оказываются несущественными по сравнению с общим временем поиска. Кэширование в реляционную базу легко решает проблему дедубликации записей и получения связанных каким-либо признаком групп записей. Кроме того, оно позволяет отбирать результаты по близким запросам, а не только по идентичным. Широкая формулировка запроса (например, “Пушкин”) всегда включает целый ряд более узких (например, “Пушкин Евгений Онегин”). Поэтому востребованность данных кэша значительно выше, чем при стандартном алгоритме кэширования. Это свойство реляционного кэша особенно важно с точки зрения эффективности поиска, выполняемого перманентно при вводе запроса. По мере ввода запроса пользователь формирует не один, а целый ряд похожих запросов.

Объектом кэширования целесообразно делать как библиографические записи, так и сформированные из них библиографические описания. Процесс их формирования описаний во многих случаях сопоставим по своей длительности с процессом поиска, поэтому кэширование обеспечивает значительную экономию ресурсов.

Ещё одним фактором, определяющим целесообразность использования внешней реляционной базы является масштабирование системы, которое может быть реализовано путём выделения независимых аппаратных мощностей для SQL сервера. Кэширование в реляционную БД, позволяет выполнять нестандартные выборки из кэша. Например, возможность выделения из результатов группового запроса результатов поиска, связанных с конкретным сервером-источником или его базой. Таким образом, если первый пользователь выполнял запрос по базам А и Б, а второй пользователь решил выполнить запрос только в базе А, то он получит результат значительно быстрее за счёт кэширования.

Неявной задачей кэша является ускорение перехода по страницам с результатами поиска. При распределённом поиске кэширование, как правило, носит опережающий характер по отношению к потребностям в выводе данных. Т.е. кэширование предполагает хранение не одной, а нескольких страниц результатов поиска. Это связано с тем, что первичный запрос на получение и формирование библиографических описаний обычно приводит к получению большего числа записей, чем требуется для отображения на одной странице. Благодаря этому работа с результатами ускоряется за счёт отображения данных из кэша.

Объектом кэширования могут выступать не только записи, но и технические сведения о результатах поиска. Они позволяют после однократного распределённого поиска по всем источникам избегать его в дальнейшем и обращаться к источникам последовательно. Этот фактор существенно влияет на скорость перехода по страницам с результатами.

13) Ранжирование результатов для целей отображения. Сортировка или ранжирование является опциональным элементом процесса (по умолчанию порядок записей определяется скоростью получения и обработки партий записей).

Возможности сортировки на уровне приложения клиента (в роли которого выступает система распределённого поиска) являются достаточно ограниченными. Так как любая сортировка предполагает выгрузку всего сортируемого массива в полном объёме, то это фактически означает поиск, преобразование и кэширование всех записей, найденных по запросу. Поэтому объёмы сортировки целесообразно ограничить таким количеством записей, которое предположительно просматривается пользователем. Вероятность просмотра пользователем 1000 страниц результатов существует, но она крайне мала. Статистика свидетельствует, что большинство пользователей ограничиваются первыми тремя страницами результатов. Поэтому чтобы сохранить вычислительные мощности и уменьшить до приемлемого уровня время вывода результатов, целесообразно использование сортировки лишь на небольших объёмах записей (например, до 200 записей).

Реляционные БД позволяют выполнять сложную сортировку. В том случае, если требуется сортировка с использованием составного ключа (например, автор-заглавие, с исключением неинформативных символов), такой ключ имеет смысл генерировать на этапе формирования библиографических описаний средствами скриптового языка. В дальнейшем набор кэшированных ключей сортировки может многократно использоваться для сортировки различных наборов записей.

При условии поддержки сервером-источником индекса релевантности, существуют условия для выполнения сортировки (ранжирования) по этим индексам с использованием алгоритма Борда [1].

14) Отображение результатов. Отображение результатов и индикация прогресса при отключённом режиме сортировки должны осуществляться по мере получения новых данных. В случае включения пользователем режима сортировки отображение записей можно выполнять только при полном завершении сканирования всех серверов-источников.

При отключении сортировки отображаемые данные должны вначале соответствовать данным ассимилированного кэша, затем первым полученным результатам поиска. Пользователь должен иметь возможность начать работать с записями, не дожидаясь завершения самого длительного процесса - полного опроса источников.

Полное обследование источников, несмотря на длительность, необходимо для формирования у пользователя адекватного представления о количестве результатов поиска. Например, при наличии 100 записей их просмотр имеет смысл, а при наличии 10 тыс. записей целесообразно уточнение запроса. Кроме того, как уже отмечалось выше, полный опрос источников позволяет реализовать превентивное кэширование и избежать при переходе по страницам результатов распределённого поиска, многократно ускоряя, таким образом, процесс навигации.

Реализация распределённого поиска в J-ИРБИС 2.0

Общая характеристика

J-ИРБИС 2.0 – это модуль САБ ИРБИС 64, предназначенный для построения библиотечных порталов и электронно-библиотечных систем¹ с функциями WEB 2.0. В последние годы он использовался как в небольших библиотеках, так и в федеральных проектах – ИСК НТИ и ЭКБСОН (макет). Модуль предоставляет широкие возможности: поиск в стиле Google, автоматическое извлечение обложек из Интернет, настройку отображения результатов, специальный интерфейс для мобильных устройств, широкий набор статистических инструментов, функции комментирования и оценки документов и многое другое.

The screenshot displays the 'Интегрированный Сводный Каталог научно-технической информации (ИСК НТИ)' interface. At the top, there is a search bar with the text '10 ключевым словам' and a date '29 - 05 - 2016'. Below the search bar, the 'Федеративный поиск' section is active, showing a search for 'соколинский' under the 'Простой поиск' tab. The results section shows one document: 'Друцкой-Соколинский, Андреевич (князь)'. The document details include: 'На службе Отечеству [Текст] / Андреевич Друцкой-Соколинский. - Орел : РКФ"3 июля":РИГ"Невероят.мир", 1994. - 404 с. : ил. + 21см. - ISBN 5868430018 : 1500.00 р. Кл. слова (ненормированные): россия история -- до 1917 -- воспоминания'. There are buttons for 'Вид документа' and 'Лицензия'. On the right side, there is a sidebar with 'Организации' (listing ИСК НТИ, ЭБС Книгафонд, НБ МГУ, Библиотека для слепых (СПб.), Российская государственная библиотека, Библиотека конгресса) and 'Фед. поисковые режимы' (Поиск по ключевым словам, Расширенный поиск). At the bottom right, there is a 'Корзина' section showing 'Выбрано 0 записей'.

Рисунок 2 Пример распределённого поиска на сайте ИСК НТИ

С помощью J-ИРБИС 2.0 может быть также обеспечен доступ к корпоративной коллекции полнотекстовых или мультимедийных документов. При этом, в зависимости от корпоративной

¹ На 19.05.16 – 201 пользователь.

политики, права на выгрузку документов предоставляются только определённым категориям пользователей или ограничиваются страничным (защищенным) просмотром в браузере.

Особенностью J-ИРБИС 2.0 является поддержка не только физических, но и логических (“виртуальных”) баз, которые отображаются в интерфейсе как физические, но при этом включают логически выделенные из внутренних и (или) внешних источниках массивы записей. Например, виртуальная база “Электронная библиотека” может фактически обеспечивать выборку всех записей нескольких локальных и удалённых баз с подключёнными к ним электронными версиями.

J-ИРБИС 2.0 поддерживает различные модели организации поиска. Он может применяться для полностью автоматического создания сводных каталогов [5] и в то же время использоваться для реализации распределённого поиска. Распределённый поиск в J-ИРБИС 2.0 является функцией ядра поисковой системы. Область применения модуля достаточно широка – от создания корпоративных порталов до интеграции данных внутри организации. Например, поиска в каталогах библиотек-филиалов университета или филиалах ЦБС.

Несмотря на широкие функциональные возможности J-ИРБИС 2.0, для его настройки не обязательно привлечение технических специалистов. Достаточно навыков опытного пользователя ПК. Это является уникальной характеристикой данного продукта на российском рынке.

Программная платформа

Технически J-ИРБИС 2.0 – это программный пакет, включающий Apache, MySQL, PHP и CMS Joomla с набором расширений. Специфические компоненты, ориентированные на потребности библиотек, представлены в виде компонентов, плагинов и модулей Joomla. Тем не менее, поисковая система относительно независима и выступает в роли автономного программного компонента.

Её интерфейс построен на основе технологии AJAX. Результаты поиска загружаются динамически без перезагрузки страницы. Это позволяет увеличить скорость поиска и уменьшить нагрузку на каналы связи.

При работе поисковой системы широко применяется распараллеливание процессов. И в отличие от большинства систем распределённого поиска, оптимизация средствами распараллеливания реализуется не только для распределённого поиска, но и для поиска в одном источнике. При наличии свободных системных ресурсов, требуемая пользователю совокупность записей извлекается за один или несколько запросов.

В системе предусмотрены четыре типа провайдеров для подключения к различным источникам (WEB ИРБИС, ИРБИС-ТС/IP, Z39-50 серверу). При получении провайдером формата отличного от базового (формат САБ ИРБИС базируется на UNIMARC), записи унифицируются и приводятся к базовому формату. Конвертирование выполняется как путём внутренней программной обработки (если формат является достаточно простым), так и с помощью дополнительного запроса к ИРБИС ТСР/IP серверу.

Интерфейс

Интерфейс J-ИРБИС 2.0 выступает не просто средством отображения результатов поисковой системы, он детерминирует архитектуру системы поиска. Средствами front-end программирования (Java Script) во многом контролируется логика поискового процесса.

Технология вывода результатов по мере ввода запроса (далее перманентный поиск) обусловила специфику технологии кэширования. Потребность в информировании читателя о ходе выполнения запроса определила необходимость достоверной индикации прогресса. Представление записей как гибкого (кастомизируемого) набора элементов потребовало изменения традиционного взгляда на библиографическое описание как на унитарный элемент.

В интерфейсе предусмотрена реализация трёх индикаторов прогресса. Два индикатора в виде блоков находятся непосредственно под поисковой формой и в форме вывода на печать. Третий, представленный в виде вращающегося круга, располагается рядом с индикатором поиска. Справа от каждого индикатора расположена точная информация о степени завершения поиска в процентах.

Основные индикаторы отражают количество полученных и обработанных порций записей. Дополнительный индикатор показывает, что в текущий момент выполняется HTTP запрос на получение данных.

Отображение записей параметрируется пользователем и администратором системы на уровне отдельно взятой записи и на уровне результатов поиска в целом.

Здесь, как и во многих ОРАС, предусмотрена возможность выбора пользователем форматов отображения записей (полный, краткий, с подписями). Но при этом, по умолчанию часть объёмных элементов каталожной карточки (аннотация, предметные рубрики, ключевые слова и др.) не отображается. Они добавляются в библиографическое описание при по запросу пользователя (при нажатии кнопки в библиографическом описании). В то же время, если пользователь нуждается в детализированных библиографических описаниях постоянно, он может настроить для собственных нужд профиль отображения записей. Требуемый набор данных фиксируется системой и начнёт использоваться при каждом обращении пользователя к сайту.

Реализация функций кастомизированного отображения результатов потребовала технического представления библиографического описания как набора независимых элементов. Полочный шифр, предметные рубрики, систематические индексы, аннотация, сведения об экземплярах и другие элементы формируются исключительно при условии, что они предусмотрены в выбранном пользователем формате отображения.

Если пользователю А потребовалось представить набор элементов А1, а пользователю Б представить набор элементов Б1, то дельта между этими наборами будет получена в результате преобразования кэшированной ранее записи. Запись извлекается из кэша, система анализирует, какие элементы описания требуется дополнительно сформировать, формирует их, добавляет элементы в кэш, а затем запись выводится пользователю.

В системе предусмотрены разнообразные опции профилирования форматов отображения записей и поисковых форм для отдельных БД. Существует возможность применения различных алгоритмов расформатирования (преобразования к библиографическому описанию) для выбранных пользователем БД. При поиске в нескольких БД, результаты поиска, если необходимо, формируются по различным алгоритмам. Для каждой БД может быть предусмотрен специфический набор форм поиска с индивидуальным перечнем полей. Потребность в этом обусловлена частым разделением БД по типу содержащегося в них контента (например, журналы, диссертации, статьи и т.п.). Если пользователем выбраны несколько разнородных баз, будут использоваться универсальные формы, если база с конкретным контентом, форма, предназначенная специально для неё. Например, в базе книг поиск по названию журнала не имеет смысла, а в базе журналов, напротив, необходим. Поэтому, если выбрана база книг, поле не будет отображаться. А если выбраны как база книг и база журналов, поле будет отображаться.

Реализация полнотекстового поиска

Особый интерес представляет реализованная в J-ИРБИС 2.0 технология полнотекстового поиска. Синтаксис полнотекстовых запросов здесь является подмножеством синтаксиса запросов библиографического поиска. Это нивелирует различия между поисками и позволяет выполнять их как независимо, так и вместе.

Поскольку не существует унифицированных протоколов распределённого поиска [4], J-ИРБИС 2.0 реализует эту технологию лишь для серверов на базе САБ ИРБИС 64. ИРБИС 64 сервер выполняет традиционные для полнотекстового поиска функции, включая ранжирование текстов и страниц по релевантности, выделение условного контекста страницы, связывание с библиографическими описаниями и многие другие.

Наиболее сложный вопрос защищенного использования (без права копирования) полных текстов может решаться в J-ИРБИС 2.0 различными способами. Здесь, как и в других системах, предусматривается простейшая технология перенаправления пользователя по ссылке, предоставленной владельцем ресурса. Но если в корпорации библиотек, работающих в САБ ИРБИС64 действуют единые правила, разрешающие доступ к документам как локальным, так и удалённым пользователям, существует возможность реализации защищенного просмотра средствами сервера, выполняющего распределённый полнотекстовый поиск.

Условием доступа к ресурсам ограниченного доступа является аутентификация пользователя по локальной БД читателей (т.е. по БД на сервере распределённого поиска). При этом определяется категория пользователя и соответствующие ей права доступа. В случае, если условия доступа в локальной системе и удалённой системе(сервере-источнике) идентичны, пользователь получает доступ к удалённому документу. Если категория

пользователя не совпадает ни с одной из тех, которые указаны на удалённом сервере, в качестве условия доступа к документу, доступ к документу запрещается.

В случае предоставления прав на чтение документа, найденный пользователем файл извлекается средствами удалённого ИРБИС TCP/IP сервера и копируется на локальный сервер. Здесь формируется его временная копия, которая в режиме on-line преобразуется к защищенному виду, в котором и отображается пользователю. При этом ресурсоёмкая операция формирования полного защищенного образа документа не требуется.

Аналогичный режим может использоваться при выполнении поиска в локальных базах и при работе с полнотекстовыми документами, подключёнными к записям.

Реализация распределённого поиска

Модель распределённого поиска в J-ИРБИС 2.0 основана на работе трёх асинхронных процессов: процесса отображения (далее «O» (Output)), процесса распределённого поиска и контроля (далее «BS» (Broadcast Search)), процесса поиска в конкретном источнике и кэширования (далее «SS» (Single Search)). Интегрирующим звеном между процессами выступает поисковая сессия (далее S (Session)), которая в отличие от обычной PHP сессии, блокируется лишь на время записи, а в остальное время работа с ней ведётся в режиме чтения (см. Рисунок 3).

BS (Broadcast Search) - процесс, который запускается сразу после инициализации поиска и завершается после опроса всех серверов-источников. При включённом режиме перманентного поиска (поиска, инициализируемого автоматически по мере ввода запроса пользователем) это происходит через незначительные промежутки времени (по умолчанию – одна секунда), после того, как пользователь прекращает ввод запроса и делает паузу. Если пользователь начинает ввод нового запроса, не дождавшись завершения предыдущего, AJAX запрос прерывается, но процесс продолжает выполняться на сервере, осуществляя кэширование данных, которые могут быть востребованы в дальнейшем.

Первой операцией системы является формулировка запроса на языке запросов САБ ИРБИС 64. Этот запрос в дальнейшем, если необходимо, транслируется провайдерами поиска в любой другой синтаксис. Формирование запроса на языке САБ ИРБИС выполняется посредством конфигурационного массива данных, определяющего правила обработки каждого поля поисковой формы по его имени. Этот массив доступен для коррекции и дополнения библиотекой-пользователем. Правила обработки полей определяются как внутри массива, так и в отдельных элементах управления поисковой формы (например, необходимость усечения определяется пользователем с помощью соответствующего переключателя).

J-ИРБИС 2.0 добавляет к стандартным возможностям языка запросов САБ ИРБИС 64 (усечение, объединение логическими операторами и др.) целый ряд дополнительных опций. Например, опцию морфологического расширения запроса, при которой запрос дополняется всеми предусмотренными в русском и английском языках словоформами, а также опцию нормализации данных об авторах, опцию разбиения фраз на термины и исключения неинформативных элементов (предлоги, союзы и т.д.).

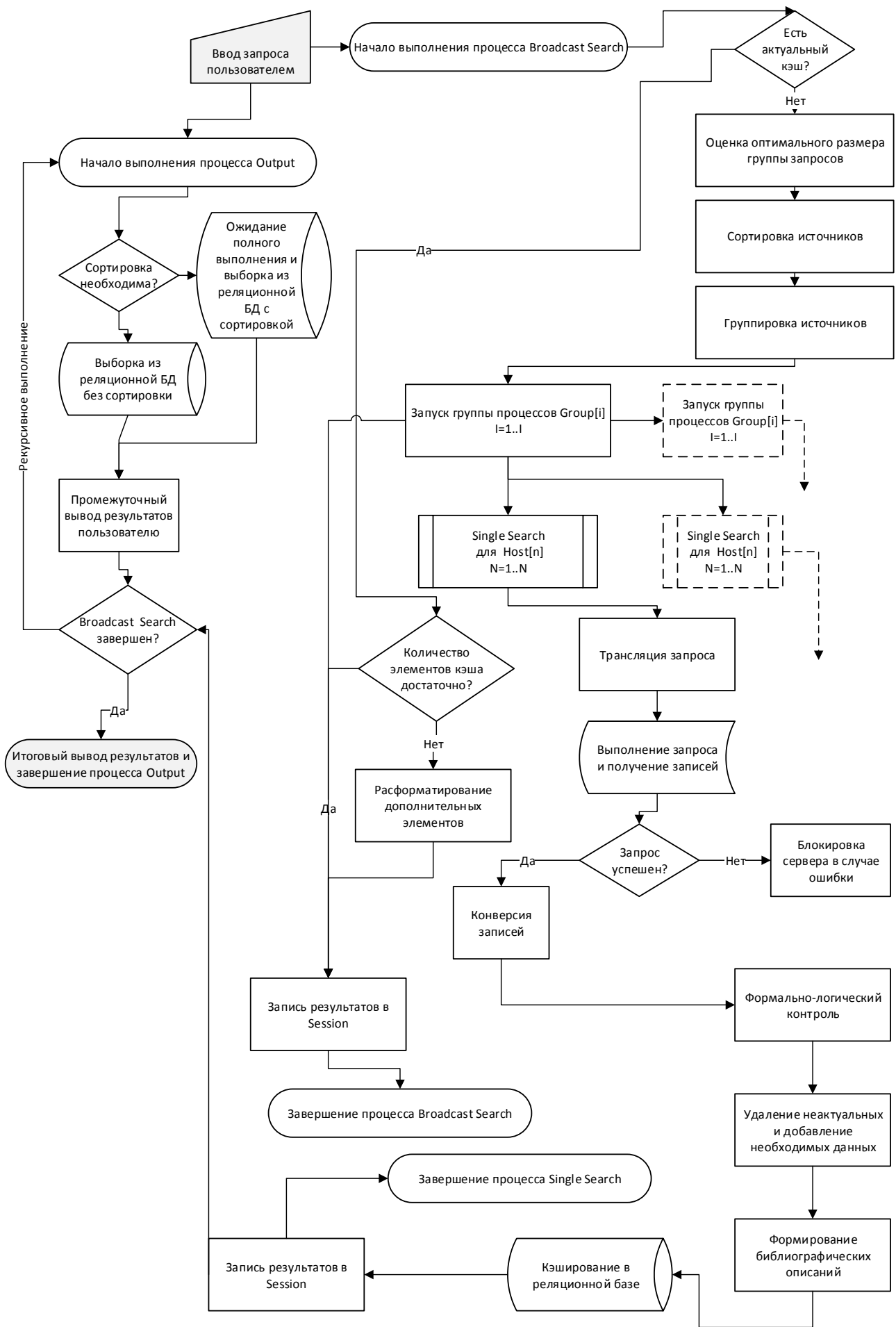


Рисунок 3 Схема распределённого поиска в J-ИРБИС 2.0

Адресные сведения доступных серверов-источников и представленных на них БД хранятся в двух таблицах реляционной БД. Ещё три таблицы используются для целей кэширования записей и хранения технической информации о результатах выполнения запросов. В специальной таблице фиксируются сведения об источнике, запросе, общем количестве найденных записей по запросу и количестве извлечённых записей. В ходе первого этапа выполнения поиска система определяет источники исходя из структуры виртуальных баз, текущего статуса (доступности), наличия кэша по текущему запросу и актуальности этого кэша. Соответствующие всем перечисленным критериям данные источников извлекаются с помощью одного SQL запроса. Таким образом, посредством реляционной БД решается сразу несколько задач разного типа.

Анализ типа виртуальных баз, группирующих или выделяющих данные из физических баз, позволяет сформировать массив адресных данных для распределённого запроса.

Если полностью соответствующий запросу кэш не найден, предпринимается попытка выборки родственных запросов. Например, при поиске по термину “история”, такими запросами будут “история России”, “история Франции” и т.п. Кэш родственных запросов предоставляется пользователю с помощью процесса O, в первую очередь. Это позволяет сразу работать с результатами поиска или уточнить запрос. Поисковый процесс на время получения кэша по родственным запросам (далее ассимилированного кэша) не прерывается, так как презюмируется, что пользователя заинтересуют полные результаты поиска.

Эффективность кэширования в реляционную БД выразительно демонстрирует приведённая ниже диаграмма (см. Рисунок 4)

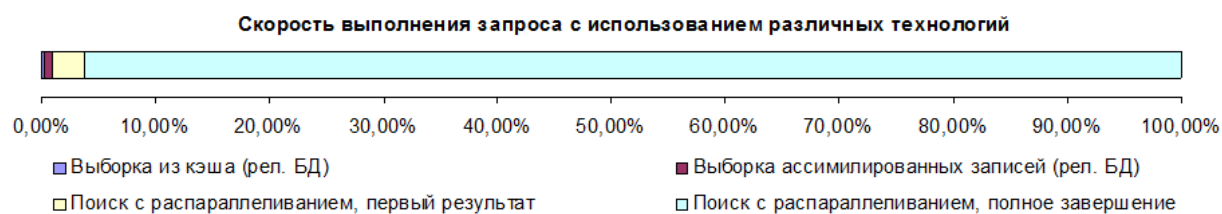


Рисунок 4 Скорость выполнения запроса при различных технологиях поиска

Если полученные данные кэша не включают нужные элементы библиографической записи или должны быть сформированы по другим алгоритмам, записи извлекаются из кэша и расформируются недостающие элементы. Расформатирование для целей распараллеливания осуществляется путём запуска группы процессов, близких по принципам своей работы к процессу SS.

Распределённый поиск реализуется путём запуска группы параллельных запросов к локальному WEB серверу средствами расширения CURL. Каждый из процессов получает в качестве исходных данных адресную информацию определённого физического сервера, а также начальный и конечный номер записи, которые требуется выгрузить. Процессы возвращают BS в качестве результата лишь информацию о количестве полученных данных или возникших при выполнении запроса ошибках. Записи, являющиеся результатами поиска, кэшируются SS автономно от BS.

SS (Single Search) – процесс, инициирующийся BS для выполнения наиболее ресурсоёмких операций поиска, извлечения, конвертирования и кэширования записей. Процесс предполагает работу только с одним источником, поэтому его выполнение начинается с выбора провайдера данных.

Все провайдеры данных обеспечивают полное единообразие программного интерфейса методов и структуры возвращаемых данных. Провайдер при необходимости выполняет не только поиск записей в сервере-источнике, но и реализует такие функции как конвертирование записей и проверку на соответствие минимально приемлемым требованиям. В зависимости от источника провайдером решается и вопрос относительно способа формирования библиографических описаний. В одном случае он может использовать для построения библиографических описаний средства RHP, в другом будет взаимодействовать для этих целей с ИРБИС TCP/IP сервером. Для всех операций преобразования записей

используются команды группового расформатирования, которое позволяет выполнить преобразование нескольких записей по одному алгоритму.

Результатом работы любого провайдера является массив объектов записей, включающих как набор полей, так и определённые профилем пользователя типы библиографических описаний и их элементы (полочный шифр, аннотация, предметные рубрики и т.д.) – объект типа `jRecord`. Процесс только кэширует эти результаты. При необходимости, чтобы уменьшить объём кэша, из записей извлекаются интегрированные версии обложек. Они сохраняются в виде файлов в специальном хранилище.

Процесс не только фиксирует ошибки, но и предпринимает попытку их корректной обработки. Ошибки подразделяются на два типа: критические и эпизодические. В случае возникновения ряда эпизодических ошибок, обусловленных, например, перегрузкой сервера, запрос повторяется в исходном виде. После этого, если результат так и не был получен, а ошибка повторилась, сервер блокируется. Критические ошибки приводят к блокировке сервера. Если сервер оказался доступен, а счётчик блокировок имеет ненулевое значение, он обнуляется.

Информация о статусе сервера представлена двумя полями таблицы баз – полем “Датаокончания блокировки” и полем счётчиком “Количество неудачных обращений”. Каждое неудачное обращение увеличивает значение счётчика. Дата окончания блокировки увеличивается экспоненциально, исходя из количества неудачных обращений. Таким образом, решается как вопрос временных сбоев, так и вопрос продолжительной неработоспособности сервера.

О (Output)- процесс, инициируемый на уровне поискового интерфейса в браузере и выполняющийся рекурсивно с момента запуска BS и до его завершения. Роль данного процесса заключается в выводе кэшированных в реляционную базу данных записей и текущего статуса выполнения запроса исходя из данных S.

Основная функция О – генерация искусственных задержек, позволяющих обеспечить компромисс между скоростью вывода данных, оперативностью индикации прогресса и нагрузкой на БД.

В зависимости от требований клиента в ходе выполнения О записи могут выводиться в порядке их кэширования или в режиме сортировки. Тем не менее, сортировка реализуется только в ситуации, когда это оказывается целесообразно с точки зрения эффективности поиска. По умолчанию сортировка выполняется не более чем для 200 записей в режиме просмотра и не более чем для 2 тыс. записей в режиме печати.

Заключение

Таким образом, реализованная в J-ИРБИС 2.0 модель распределённого поиска — это попытка преодоления существующих технологических ограничений ради построения нового эргономичного поискового интерфейса. Она решает целый ряд практических задач:

1. Обеспечивает реализацию интерфейса поиска с выводом результатов по мере ввода запроса (перманентного поиска);
2. Даёт возможность начать работу с результатами поиска немедленно после их получения;
3. Предусматривает отображение результатов единым списком с возможностью сортировки и ранжирования;
4. Предоставляет инструменты для кастомизации формата отображения результатов поиска;
5. Позволяет более эффективно использовать аппаратные ресурсы за счёт распараллеливания.

В то же время необходимо отметить и определённые недостатки модели:

1. Ограниченные возможности сортировки, ранжирования и дедубликации записей;
2. Высокая зависимость от производительности SQL сервера, сервера САБ и в целом аппаратных ресурсов сервера.

Предложенная технология распределённого поиска может совершенствоваться и легко адаптироваться для конкретных задач. Оптимальной сферой её эффективного применения можно считать реализацию корпоративных и отраслевых электронных библиотек, служб электронной доставки документов, а также интеграцию с агрегаторами (например EBSCO).

Список литературы

- 1 Колобов О. С. Исследование принципов организации, функционирование и разработка распределенного электронного каталога библиотечного консорциума [Текст]: дис. канд. техн. наук: 05.25.05 / О. С. Колобов. - Томск, 2006. – 131 с.
- 2 Колосов К. А. Создание и применение в библиотечной практике корпоративной технологии на базе протокола Z39.50 [Текст]: дис. канд. техн. наук: 05.25.05 / О. С. Колосов. - Москва, 2007. – 154 с.
- 3 Колосов К.А. Электронная библиотека ГПНТБ России: тенденции развития, взаимодействие сНЭБ, перспективы / К.А. Колосов // Научные и технические библиотеки. - 2015. - N 11. - С. 101-105
- 4 Ляпин С.Х. Кластеры полнотекстового поиска в распределенной информационной среде: технология и проекты [Электронный ресурс] / Р. Т. Усманов, А. А. Кузнецов // Библиотеки и информационные ресурсы в современном мире науки, культуры, образования и бизнеса: Материалы конф. (Судак, Автоном. Респ. Крым, Россия, 6—14 июня 2015 г.). – Электрон. текстовые дан. — М.:ГПНТБ, 2015 . – Режим доступа: <http://www.gpntb.ru/win/inter-events/crimea2015/disk/003.pdf>. – Загл. с экрана (дата обращения: 07.02.2016).
- 5 Соколинский К.Е. Новая технология создания сводных каталогов и корпоративных электронных библиотек в J-ИРБИС 2.0 / К. Е. Соколинский // Научные и технические библиотеки. - 2015. - N 11. - С. 83-100
- 6 Соколинский, К. Е. Новые подходы к каталогизации заимствованием в ИРБИС-корпорации [Текст] / К. Е. Соколинский // Научные и технические библиотеки. - 2010. - N 11. - С. 96-102
- 7 Усманов Р. Т. Оценка эффективности работы распределенной сети Z39.50 серверов АРБИКОН [Электронный ресурс] / Р. Т. Усманов, А. А. Кузнецов // Библиотеки и информационные ресурсы в современном мире науки, культуры, образования и бизнеса: Материалы конф. (Судак, Автоном. Респ. Крым, Украина, 4—12 июня 2005 г.). – Электрон. текстовые дан. — М.:ГПНТБ, 2005 . – Режим доступа: <http://www.gpntb.ru/win/inter-events/crimea2005/disk/217.pdf>. – Загл. с экрана (дата обращения: 07.02.2016).