

Российский фонд фундаментальных исследований  
«Открытые системы»

# Интеллектуальные системы хранения и обработки информации

Программа и тезисы докладов



## Технологии БАЗ ДАННЫХ



Hadoop  
In-memory  
Open Source

Москва, 2016 год

Открытые системы
2016
ПРАКТИЧЕСКИЕ СЕМИНАРЫ И КОНФЕРЕНЦИИ

**ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ  
ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ**  
www.osp.ru/iz/tbd\_dbms

Москва,  
29-30 ноября 2016 года

УДК 004.65, 004.657, 004.2, 004.3  
ББК 32.973.

При поддержке



Организаторы



Партнеры





# ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ

[www.osp.ru/iz/tbd\\_dbms](http://www.osp.ru/iz/tbd_dbms)

Москва,  
29-30 ноября 2016 года

Оргкомитет конференции «Интеллектуальные системы хранения и обработки информации» выражает признательность за поддержку Российскому фонду фундаментальных исследований (грант 16-07-20891-г), компаниям «TmaxSoft», Oracle и Visiology.

## **Тематика конференции «Интеллектуальные системы хранения и обработки информации»:**

Пленарная сессия. «Архитектуры «эпохи пост-SQL»»

Секция. «Инструменты и архитектуры»

Секция. «Практика систем хранения и анализа»

Секция: «Интеллектуальные средства СУБД»

Секция. «Стендовые доклады»

**Программа и тезисы докладов конференции «Интеллектуальные системы хранения и обработки информации» (Москва, 29-30 ноября 2016 г.). // [Под. ред. Волкова Д.В.]. – М.: «Открытые системы», 2016. – 26 с.**

В сборник трудов включены тезисы докладов конференции «Интеллектуальные системы хранения и обработки информации», прошедшей 29-30 ноября 2016 года в Москве, гостиница «Интурист Коломенское». Цель конференции – обсуждение наиболее актуальных направлений работ в области организации хранения данных; обзор методов и инструментов интеллектуального анализа структурированных и неструктурированных данных; обмен опытом решения исследовательских и прикладных аналитических задач. Материалы сборника тезисов предназначены для представителей федеральных ведомств, научных сотрудников, разработчиков и пользователей, а также преподавателей и аспирантов, интересующихся проблемами создания и эксплуатации СУБД для эффективной обработки экспоненциально растущих объемов данных различной природы. Подробную информацию о конференции можно найти по адресу [www.osp.ru/iz/dbms](http://www.osp.ru/iz/dbms).

**Copyright 2016 ООО «Открытые системы»**

### **«Умные» СУБД**

Еще совсем недавно реляционная модель считалась единственно возможной, способной обеспечить поддержку промышленных решений организации хранения данных и их анализа. Однако появление новых задач работы с Большими Данными, облачными сервисами и мобильными приложениями вскрыло неспособность традиционных СУБД оперативно обрабатывать экспоненциально растущие объемы структурированных и неструктурированных данных. Современным приложениям нужна функциональность, не свойственная реляционным системам, – в частности, требуется возможность изменения схем хранения и поддержки всего многообразия типов и моделей данных. Мало того, для хранения и эффективного анализа данных научных экспериментов и сведений из социальных сетей, создания аналитических комплексов и рекомендательных систем на основе методов машинного обучения одной лишь СУБД недостаточно. Сегодня известны сотни систем управления базами данных, соответствующих не только канонам реляционной алгебры, появились новые алгоритмы работы с данными и накоплен опыт использования инструментов, построенных на пост-реляционной модели. Проблемы больших объемов не обошли стороной и транзакционные системы, данных в которых на порядки меньше, чем в аналитических хранилищах, но и через них сегодня проходят огромные массивы сведений и транзакций, что требует новых подходов, обеспечивающих масштабирование.

Аналитики и лица, принимающие решения, уже не могут довольствоваться заранее подготовленными данными, которые, как правило, описывают прошедшие события, – им требуются сведения реального времени и инструменты, позволяющие мгновенно интегрировать, фильтровать и самостоятельно анализировать разноформатную информацию, поступающую в огромных объемах из различных распределенных в пространстве источников.

Таким образом, завершился период относительного затишья в области СУБД и интеллектуального анализа данных – сегодня наблюдается всплеск интереса к этой сфере знаний, сопровождаемый появлением новых идей, алгоритмов и методов, способных нарушить незыблемость требований ACID: атомарности, согласованности, изолированности и долговечности хранения. Пришло время разобраться в реальных, а не декларируемых возможностях «умных» инструментов класса NoSQL, оценить экспертизу, проанализировать существующие алгоритмы и решения, понять их место в общем наборе средств интеллектуального анализа. Однако, чтобы такие «умные» СУБД достигли такого же уровня надежности и зрелости, как и традиционные, предстоит еще решить немало задач.

**Организационный комитет**

**Организационный комитет:**

Аветисян Арутюн Ишханович  
Волков Дмитрий Владимирович  
Карпенко Анатолий Павлович  
Кореньков Владимир Васильевич

чл.-корр. РАН, директор ИСП РАН, Председатель  
с.н.с., ИПМ им.М.В.Келдыша РАН, гл. редактор «Открытые системы.СУБД», зам. Председателя  
д.ф.-м.н., профессор, МГТУ им.Н.Э.Баумана  
д.т.н., директор ЛИТ ОИЯИ

**Программный комитет:**

Кузнецов Сергей Дмитриевич  
Бунина Елена Игоревна  
Дубова Наталия Аркадиевна  
Кузнецов Сергей Олегович  
Легалов Александр Иванович  
Мальцева Светлана Валентиновна  
Сухомлин Владимир Александрович

д.т.н., профессор МГУ им. М.В.Ломоносова, Председатель  
д.ф.-м.н., директор отделения computer science Школы анализа данных «Яндекс»  
научный редактор, «Открытые системы.СУБД»  
д.ф.-м.н., профессор, НИУ ВШЭ  
д.т.н., профессор, Сибирский федеральный университет  
д.т.н., профессор, НИУ ВШЭ  
д.т.н., профессор, МГУ им. М.В.Ломоносова

29.11.2016

**Открытие конференции**

**Пленарная секция: «Архитектуры "эпохи пост-SQL"»**

- «Коротко» о СУБД ClickHouse  
Башкеев А.А., к.ф.-м.н, Миловидов А.Н., «Яндекс»
- СУБД PostgreSQL: высокоскоростной полнотекстовый поиск  
Бартунов О.С., Коротков А.Е., к.т.н., «Постгресс профессиональный»
- Эволюция репликации в СУБД MySQL и MariaDB  
Петруня С.Ф., **MariaDB Corp. Ab**
- Как обеспечить продажу вашего автомобиля за 30 минут?  
Под «капотом» автомобильного аукциона реального времени: MySQL, Elasticsearch, In-memory  
Буйлов Ю.С., **CarPrice**
- Решение проблемы сборки мусора в СУБД «Ред База Данных»  
Симаков Р.А., к.т.н., «Ред Софт»
- Отладка производительности СУБД MySQL  
Смирнова С.А., **Percona**
- Коммерческая СУБД против нишевых. тенденции развития баз данных  
Ривкин М.Н., **Oracle**
- СУБД: между прошлым и будущим  
Кузнецов С.Д., д.т.н., **МГУ, ИСП РАН**

**Секция: «Инструменты и архитектуры»**

- Хранилища для аналитики – конвергенция Hadoop, СУБД In-memory и SQL  
Борчук Л.Е., к.т.н., «Яндекс»
- Особенности работы с Google Cloud Datastore  
Кива В.В., **Process Insight**
- СУБД Tiberio и комплекс TiberioZetaData  
Морозов О.Е., **TmaxSoft**
- PythonQL: интеграция, чистка и аналитика разнородных данных средствами встроенного языка запросов  
Велихов П.Е., **Finstar Labs**
- Как получать и интерпретировать показатели производительности СУБД?  
Николаенко А.В., **IBS**

**Секция: «Практика систем хранения и анализа»**

- Секреты онлайн-рекламы: Redis, MongoDB, Aerospike и Hbase  
Ведерников А.В., **Data-Centric Alliance**
- СУБД In-memory на практике в Сургутнефтегаз  
Юношев А.С., «Сургутнефтегаз»
- Как доставить пиццу быстрее прохождения транзакции в СУБД? Платформа Bringo для поддержки краудсорсинга в логистике  
Корнев М.А., **Bringo**
- Переход документооборота Правительства Московской области с СУБД Oracle на PostgreSQL  
Алания Т., **Министерство государственного управления, ИТ и связи Московской области**  
Панченко И.Е., к.ф.-м.н, «Постгресс профессиональный»
- Как повысить качество телекоммуникационной сети: СУБД MongoDB, Redis и HBase?  
Смирнов А.Н., к.т.н., **Motorola Solutions**  
Егоров А.В., **BeKitzur**

30.11.2016

**Секция: «Интеллектуальные средства СУБД»**

- Разработка базы знаний современного научного эксперимента  
Григорьева М.А., к.т.н., Аулов В.А., Голосова М.В., **НИЦ «Курчатовский институт»**, Губин М.Ю., **Томский политехнический университет**, Климентов А.А., к.ф.-м.н., **Брукхэвенская национальная лаборатория**
- Верификация потоков работ интеграции данных  
Ступников С.А., к.т.н., **ИПИ РАН**
- Обнаружения знаний в базах данных с помощью мультимодальной кластеризации  
Игнатов Д.И., к.т.н., **НИУ ВШЭ**
- Алгоритм бустинга с динамическим преобразованием признаков  
Китов В.В., к.ф.-м.н., Гой А.С., **МГУ**
- Кредитный скоринг на основе анализа формальных понятий: персональные модели  
Масютин А.А., **НИУ ВШЭ**
- Графовые СУБД и deep learning в системе «аромовидения»  
Крылов В.В., д.т.н., **НИУ ВШЭ**, Кучменко Т.А., д.х.н., **ВГУИТ**
- Метод обработки запросов к распределенной базе данных с использованием фильтра Блума на платформе Spark  
Григорьев Ю.А., д.т.н., Пролетарская В.А., **МГТУ им. Н.Э. Баумана**
- Ускорение выполнения SQL-запросов в СУБД PostgreSQL с использованием динамической компиляции  
Мельник Д.М., **ИСП РАН**
- Концепция реализации произвольных запросов к базам данных на естественном языке  
Сергиевский Г.М., к.т.н., **НИЯУ МИФИ**
- Infobright: оптимизация SQL запросов с помощью приближений теории неточных множеств  
Кашницкий Ю.С., **НИУ ВШЭ**
- СУБД для вычислительных комплексов семейства "Эльбрус"  
Королев С.Д., **МЦСТ**

**Секция: Стендовые доклады**

- Метрики качества для бикластеризации  
Кузнецов С.О., д.ф.-м.н., Махалова Т.П., **НИУ ВШЭ**
- Поиск закономерностей в базе данных демографических последовательностей на основе узорных структур  
Гиздатуллин Д.К., **НИУ ВШЭ**
- Хранилище системы сбора статей электронных библиотек  
Незнанов А.А., Паринов А.А., **НИУ ВШЭ**

**Закрытие конференции**

## Пленарная сессия. Архитектуры «эпохи пост-SQL»

### Коротко о СУБД ClickHouse

Башкеев А.А., к.ф.-м.н, Миловидов А.Н. – «Яндекс»

ClickHouse – распределенная аналитическая СУБД с открытым исходным кодом. В докладе рассмотрена история разработки этой СУБД, основные принципы и особенности ее реализации, а также разобраны ситуации, в которых оправдано использование ClickHouse. Особое внимание уделено примерам использования СУБД ClickHouse для поддержки сервиса «Яндекс.Метрика» и ряде других, предлагаемых компанией «Яндекс».

### СУБД PostgreSQL 9.6: высокоскоростной полнотекстовый поиск

Бартунов О.С., Коротков А.Е., к.т.н. – «Постгресс профессиональный»

Полнотекстовый поиск присутствует в СУБД PostgreSQL больше 10 лет. За это время он прочно обосновался в ядре и стал весьма популярен, что, однако не исключает необходимости в его улучшении. В докладе представлены новые возможности, полнотекстового поиска, внесенные в версию PostgreSQL 9.6, в которой появился давно ожидаемый поиск по фразам и новые функции для работы с полнотекстовыми документами. Кроме этого, в PostgreSQL 9.6 появилась возможность добавлять новые индексные методы доступа в расширениях. В докладе рассмотрены новые возможности полнотекстового поиска, а также разобран новый индексный метод RUM (Really Useful Method), позволяющий в разы сократить время выполнения многих запросов полнотекстового поиска.

### Эволюция репликации в СУБД MySQL и MariaDB

Петруня С.Ф. – MariaDB Corp. Ab

В новых версиях СУБД MySQL и MariaDB были существенно усовершенствованы технологии репликации – появилась параллельная репликация, Global Transaction Ids, binlog server, поддержка multi-master и т.д. Естественно, что между новыми функциями имеется взаимосвязь, однако в ряде случаев она не очевидна, а порой и нетривиальна, не говоря уже о том, что реализации этих функций в СУБД MySQL и MariaDB различаются. Доклад посвящен обсуждению особенностей эволюции репликации, разбору зависимостей функций ее реализации и принципов их построения, а также возможностей и ограничений функций для пользователя. Кроме этого, в докладе проведено сравнение СУБД MySQL и MariaDB, а также проанализированы дополнительные возможности последней.

## **Как обеспечить продажу вашего автомобиля за 30 минут? Под «капотом» автомобильного аукциона реального времени: MySQL, Elasticsearch, In-memory**

Буйлов Ю.С. – CarPrice

Онлайн аукцион CarPrice – это вершина айсберга, скрывающая множество онлайн- и оффлайн-сервисов, позволяющих автовладельцам комфортно, безопасно и по максимально возможной цене быстро продать свою машину. Разные задачи – это разные требования и разные технологии. В докладе показано: как устроен автомобильный онлайн-аукцион реального времени; как поддерживается консистентность данных на стыке онлайн- и оффлайн-сервисов; как управлять потоками данных в реальном времени.

## **Решение проблемы сборки мусора в СУБД «Ред База Данных»**

Симаков Р.А., к.т.н. – «Ред Софт»

Технология «единого окна» и повышение количества услуг, предоставляемых федеральными ведомствами постоянно приводят к повышению нагрузки на информационные системы и конечно на их СУБД. Поддержка обработки постоянно высокого потока запросов требует от создателей СУБД новых алгоритмов и инструментов, позволяющих масштабировать конфигурацию под новые нагрузки. В докладе рассмотрены особенности построенной на платформе СУБД с открытым исходным кодом Firebird СУБД Ред База Данных, позволяющие обеспечить ее промышленную эксплуатацию в ведущих отечественных федеральных ведомствах.

## **Отладка производительности СУБД MySQL**

Смирнова С.А. – Percona

СУБД MySQL работает медленнее, чем раньше? Запросы «висят»? Клиенты ждут ответа сервера дольше, чем ожидалось? Что делать? В докладе показан порядок действий, помогающий выяснить причину возникновения той или иной проблемы с производительностью. Что необходимо предпринять для выявления самых медленных запросов и как заставить их работать быстрее? Кроме этого в докладе изложена схема работы системы блокировок и разобраны факторы их влияния на эффективность высоко- конкурентной среды, а также показано как оборудование и настройки влияют на работу сервера MySQL.



## **Коммерческая СУБД против нишевых: тенденции развития баз данных**

Ривкин М.Н. – Oracle

Доклад посвящен анализу развития коммерческих баз данных на примере СУБД Oracle – разбираются применяемые в этой системе решения и методы обеспечения надежности, безопасности, масштабируемости, производительности и управляемости. В процессе своего развития СУБД Oracle интегрировала такие наиболее актуальные и востребованные пользователями функции, как шардинг, обработка в памяти, управление жизненным циклом данных, мультиарендность -- консолидация и управление множеством баз данных как одной, машина баз данных, команды SQL в процессоре и другие по-отдельности встречающиеся в тех или иных нишевых СУБД. Особое внимание в докладе уделено современным тенденциям развития СУБД в целом: облака и DbaaS, обработка в памяти, поддержка средств работы с Большими Данными, «СУБД в процессоре», инженерные системы и Cloud Machine, машинное обучение.

## **СУБД: между прошлым и будущим**

Кузнецов С.Д., д.т.н. – МГУ, ИСП РАН

Раз в несколько лет ведущие представители исследовательского сообщества баз данных проводят двухдневные встречи, на которых обсуждается и оценивается состояние дел в области СУБД и формулируются наиболее актуальные в ближайшие годы темы исследований. По результатам таких встреч готовятся и публикуются отчеты, традиционно пользующиеся высоким авторитетом в сообществе баз данных и оказывающие, в итоге, серьезное влияние на развитие исследований и разработок в этой области. В докладе обсуждаются самые интересные прогнозы, содержащиеся в отчетах о встречах предыдущих лет, излагаются наиболее существенные моменты отчета о последней по встрече, состоявшейся осенью 2013 года в Калифорнии.

## Секция. Инструменты и архитектуры

### Хранилища для аналитики – конвергенция Hadoop, СУБД In-memory и SQL

Борчук Л.Е., к.т.н. – «Яндекс»

Еще вчера четко была видна граница между областями использования каждой конкретной СУБД – начиная с определенного объема данных система либо просто переставала работать, либо резко возрастало время выполнения задачи. Для хранилищ емкостью в сотни терабайт данных единственным разумным вариантом был и остается Hadoop, однако лучшее, на что приходилось рассчитывать – минуты-часы ожидания ответа на запрос. Для получения более оперативного ответа (секунды-минуты) необходимо было устанавливать традиционную СУБД и настраивать сложную схему обмена данными. Наблюдаемые сегодня изменения: смена архитектуры с существенным улучшением производительности плюс более тесная интеграция различных СУБД, значительно меняют ландшафт, размывая границы между отдельными инструментами и позволяя строить гибкие системы. Однако, это означает, что старые приемы и методы перестают работать. В докладе обсуждаются происходящие сегодня изменения в технологиях и архитектурах, и их влияние на построение аналитических хранилищ.

### Особенности работы с Google Cloud Datastore

Кива В.В. – Process Insight

Использование Google Cloud Datastore в промышленных системах позволяет разработчикам сфокусироваться на реализации бизнес-задач, практически исключив рутинное администрирование, в частности, перед ними открываются интересные возможности при использовании архитектуры schemaless. Однако, простота и удобство предполагает ряд существенных ограничений, без учета которых сложно спроектировать эффективную систему. В докладе изложен опыт работы с СУБД NoSQL Google Cloud Datastore при эксплуатации сервиса Process Insight для решения задач управления финансами предприятия.

### СУБД Tiberio и комплекс TiberioZetaData

Морозов О.Е. – TmaxSoft

Первая версия СУБД Tiberio была выпущена в 2003 году с целью предоставить Южной Корее решение, альтернативное зарубежным поставщикам. СУБД Tiberio – единственная база максимально совместимая с Oracle, имеющая аналогичные механизмы обработки транзакций на уровне блокировок. Tiberio предоставляет множество идентичных механизмов по разработке, управлению, администрированию и оптимизации. Все это сделано для упрощения ее освоения и использования администраторами, а также разработчиками, имеющими опыт работы с СУБД Oracle. Максимальное сходство словаря данных, оптимизатора запросов, синтаксиса PL/SQL, DCL, DML, DDL позволяет с минимальными затратами выполнять миграцию из Oracle в Tiberio, сохраняя структуру и логику работы кода приложения. Возможность миграции приложения обеспечивается за счет поддержки ряда интерфейсов ODBC, JDBC, а также OCI, что позволяет подключать приложение к Tiberio без изменения исходного кода.

В докладе изложены возможности СУБД Tiberio 6, поддерживающей кластеризацию (Active Cluster), кластерную файловую систему (Active Storage), технологию Standby, включающую средства восстановления (flashback query), а также инструменты flashback database и усовершенствованную технологию компрессии данных. Особое внимание в докладе уделено описанию направлений дальнейшего развития СУБД Tiberio 7: in memory, средства интеграции с облаками и др.

## **PythonQL: интеграция, очистка и аналитика разнородных данных средствами встроенного языка запросов**

Велихов П.Е. – Finstar Labs

Ежедневно количество разнородных СУБД растет – наряду с традиционными привычными реляционными СУБД появились системы с другими моделями: XML, JSON, key-value, многомерные, графовые и т.п. Одновременно с этим, специалисты по данным (data scientists) все реже используют языки запросов конкретных СУБД, да и сами базы данных стараются применять по-минимуму. Одна из основных причин -- это высокий барьер освоения каждого конкретного языка запросов, как правил предусматривающего изучение всевозможных библиотек работы с данными. Язык PythonQL ([www.pythonql.org](http://www.pythonql.org)) – расширение популярного языка Python — дает возможность пользоваться всеми преимуществами этого мощного языка запросов не заботясь о специфике различных СУБД, обращаясь к каждой из них через один языковой интерфейс. В докладе, кроме описания PythonQL, рассмотрены популярные сценарии работы с данными, эффективно реализуемые в PythonQL и применяемые в практике специалистов по работе с данными.

## **Как получать и интерпретировать показатели производительности СУБД?**

Николаенко А.В. – IBS

Совет по оценке производительности обработки транзакций (TPC.org) с 1988 года создает эталонные тесты для измерения производительности СУБД, однако уже с середины нулевых новые оценки не публикуются, а в технических материалах все реже фигурируют миллионы транзакций в секунду и запросы в час, но зато появились новые метрики: время на сортировку терабайта данных, число SQL-операций ввода-вывода в единицу времени, объемы загрузки в единицу времени, время отклика на запрос. Снижение интереса к оценкам от TPC в большей степени связано с появлением NoSQL-систем, распространением СУБД In-memory, прогрессом в оборудовании, сдвигающим «бутылочное горло» от сети к устройствам хранения и обратно, а также общим смещением фокуса внимания от классических приложений с их длинными транзакциями к сверхмногопользовательским и супермасштабируемым приложениям, от классической запросно-ответной аналитики над базами к задачам Больших Данных. В докладе рассказывается про актуальные подходы и методики измерения показателей производительности СУБД, раскрываются секреты проведения тестирования и озвучены наиболее дискуссионные вопросы, волнующие сегодня пользователей СУБД, поставщиков и разработчиков СУБД, а также производителей оборудования.

## Секция. Практика систем хранения и анализа

### **Секреты онлайн-рекламы: Redis, MongoDB, Aerospike и Hbase**

Ведерников А.В. – Data-Centric Alliance

В основе любой programmatic-экосистемы, позволяющей рекламодателям управлять целевыми аудиториями рекламных кампаний, показывая на любом устройстве рекламу только тем пользователям, кому она интересна и только тогда, когда она действительно нужна, лежит платформа обработки аудиторных данных (DMP, Data Management Platform). Платформа Facetz DMP позволяет обрабатывать данные о более чем 600 млн анонимных пользователей Интернета, используя для этого четыре NoSQL СУБД, каждая из которых предпочтительна для реализации конкретных сценариев доступа к данным. В докладе обсуждаются особенности функционирования DMP, разбираются типы обрабатываемых такими платформами данных, разбираются способы организации эффективного хранения и обеспечения доступа к данным. Кроме этого, анализируется целесообразность и особенности использования систем Redis, MongoDB, Aerospike и HBase внутри одной DMP.

### **СУБД In-memory на практике в Сургутнефтегаз**

Юношев А.С. – «Сургутнефтегаз»

SAP HANA – платформа для обработки данных в реальном времени, включающая в себя поколоночную СУБД, а также средства создания баз данных и приложений. В Сургутнефтегаз эта платформа используется с 2010 года и применяется для решения аналитических задач, формирования отчетности, а также для ускорения выполнения «тяжелых» запросов ERP-систем. В докладе рассказано об опыте применения СУБД In-memory в Сургутнефтегаз и особенностях эксплуатации платформы в реальных условиях. Кроме этого рассмотрены вопросы развертывания инфраструктуры, наиболее оптимальной для конкретных условий эксплуатации, а также проанализированы возможные архитектуры приложений и решений, построенных на платформе SAP HANA.

### **Как доставить пиццу быстрее прохождения транзакции в СУБД? Платформа Bringo для поддержки краудсорсинга в логистике**

Корнев М.А. – Bringo

Bringo – молодой проект, ставший за несколько лет ведущим в России краудсорсинговым логистическим сервисом, помогающим таким компаниям как IL Патио, Philip Morris, «Азбука вкуса», Yota, DPD и Pony Express обслуживать своих клиентов. Платформа Bringo объединяет всех, кому нужна быстрая и качественная доставка с ее исполнителями – краудсорсерами-курьерами или брингерами. В докладе рассказано об особенностях организации краудсорсинга в отечественных условиях, сопутствующих проблемах в области хранения данных (сопровождаемой пиковыми всплесками нагрузок, поддержкой консистентности в разных источниках и т.п.) и способах их решения, в частности, средствами СУБД PostgreSQL.

## **Переход документооборота Правительства Московской области с СУБД Oracle на PostgreSQL**

Алания Т., Министерство государственного управления, ИТ и связи Московской области

Панченко И.Е., к.ф.-м.н. – «Постгресс профессиональный»

В начале 2016 года Министерство государственного управления, информационных технологий и связи Московской области (МГУИТ МО) завершило работы по переносу базы данных Межведомственной системы электронного документооборота Московской области (МСЭД) с СУБД Oracle на свободно-распространяемую объектно-реляционную СУБД PostgreSQL. На сегодняшний день в МСЭД зарегистрировано более 42 тыс пользователей, а количество одновременно работающих пользователей превышает 12 тыс в сутки. В докладе рассказано о причинах и целях миграции базы, насчитывающей около 18 млн документов, ожидавшихся препятствиях и реальных трудностях, возникших на пути переноса, а также об эффекте, который был достигнут в Правительстве Московской области от эксплуатации системы документооборота в новой среде.

## **Как повысить качество телекоммуникационной сети: СУБД MongoDB, Redis и HBase?**

Смирнов А.Н., к.т.н. – Motorola Solutions, Егоров А.В. – BeKitzur

Телекоммуникационная сеть -- это сложная техническая система, состоящая из множества компонентов, работающих с разнообразной информацией, к которой предъявляются различные, как правило, повышенные требования по сроку хранения данных, организации, скорости доступа и т. д. Выбор наиболее подходящего для каждого конкретного случая хранилища на базе конкретного решения NoSQL оказывает существенное влияние на качество работы телекоммуникационной сети. В докладе рассматриваются примеры использования СУБД MongoDB, Redis и HBase для построения хранилища данных телекоммуникационной сети, а также разбираются критерии выбора того или иного инструмента.

### Секция: «Интеллектуальные средства СУБД»

#### **Разработка базы знаний современного научного эксперимента**

Григорьева М.А., к.т.н., Аулов В.А., Голосова М.В. – НИЦ «Курчатовский институт», Губин М.Ю. – Томский политехнический университет, Климентов А.А. – к.ф.-м.н. – Брукхэвенская национальная лаборатория

Современные научные эксперименты с интенсивной обработкой данных имеют длительный жизненный цикл, сложную инфраструктуру, в которой хранятся сотни петабайт и обрабатываются экзабайты данных. Все стадии жизненного цикла эксперимента сопровождаются вспомогательными метаданными. В большинстве научных сообществ метаданные, описывающие цепочки анализа и обработки данных, и метаданные о публикации научных результатов, существуют независимо друг от друга. Кроме того, чтобы воспроизвести или подтвердить результаты уже проведенного эксперимента, ученым бывает необходимо провести исследования при тех же условиях, проверить результаты обработки наборов данных новой версией программного обеспечения, или опробовать новые алгоритмы. Описанная в работе база научных знаний (Data Knowledge Base - DKB) обеспечивает хранение и быстрый доступ к релевантной научной и вспомогательной метаинформации. Архитектура DKB имеет два уровня хранения данных: хранилище Hadoop, в котором данные от различных источников метаданных интегрируются, агрегируются и обрабатываются, и онтологическое хранилище Virtuoso, в котором сохраняются все извлеченные данные. Агенты DKB автоматически обрабатывают и агрегируют метаданные из систем управления и обработки данных, избавляя ученых от необходимости подробно аннотировать каждый компонент эксперимента.

#### **Верификация потоков работ интеграции данных**

Ступников С.А., к.т.н. – ИПИ РАН

Разработка методов и средств интеграции данных становится все более актуальной в связи со значительным ростом объемов и разнообразия данных. Различные по семантике и структурированности данные могут быть необходимы при решении одной задачи в науке или промышленности. Увеличение объемов данных требует применения масштабируемых платформ распределенной параллельной обработки данных, таких, как Apache Hadoop. При этом все составляющие процесса интеграции данных должны быть реализованы в виде программ над соответствующей платформой. На практике, процесс интеграции собственно данных представляет собой набор операций трансформации данных, представленных в SQL или подобном ему языке, причем важным является порядок исполнения операций. Обычно интеграция данных организуется в виде потоков работ. Процессы интеграции данных являются достаточно важными и сложными, и в течение всего периода развития методов и средств интеграции возникали вопросы определения их формальной семантики и верификации, т.е. формальной проверки на соответствие заданным требованиям.

Данная работа нацелена на развитие методов определения формальной семантики и верификации процессов интеграции данных. Рассматривается интеграция структурированных данных, т.е. данных, конформных некоторой схеме, определяющей типы (структуры) данных. Предполагается также, что процесс интеграции данных представляется в виде набора операций трансформации данных на SQL-подобном языке. При этом свойства процесса интеграции в целом не выражаются явно, определяются лишь правила преобразования данных в конкретных операциях.

В процессе интеграции данных обычно выделяют различные этапы: трансформация данных, разрешение сущностей (entity resolution)[1] и слияние данных (data fusion)[2]. Трансформация данных подразумевает их преобразование из исходной схемы (схемы коллекции - источника данных) в целевую (единую интегрированную схему). Под разрешением сущностей обычно понимают выделение и связывание информации об одной и той же сущности реального мира из разных коллекций данных. Под слиянием сущностей понимают комбинацию различных представлений одной и той же сущности реального мира в единое представление. Этапы интеграции данных могут состоять из большого количества операций, и не обязательно идут в указанном порядке. Поэтому более правильно говорить, что процесс интеграции данных представляет собой поток работ, деятельности которого представляют собой отдельные операции трансформации структурированных (типизированных) данных, разрешения или слияния сущностей.

Наряду с платформами распределенной параллельной обработки данных разрабатываются высокоуровневые языки программирования, которые могут быть использованы (или прямо предназначены) для трансформации данных, разрешения и слияния сущностей в рамках этих платформ.

Идея подхода, предлагаемого в докладе, состоит в том, чтобы сообщить высокоуровневым программам интеграции данных семантику в некотором языке спецификаций, поддержанном средствами формального автоматического и/или интерактивного доказательства: для языка интеграции данных строится его отображение в язык спецификаций. Свойства программ, подлежащие проверке, представляются в виде выражений выбранного языка спецификаций. Затем, с использованием формальных средств доказательства, спецификация, выражающая семантику конкретного потока работ интеграции данных, проверяется на соответствие необходимым свойствам.

Для иллюстрации подхода в качестве языка интеграции данных выбран HIL [3] – язык, разработанный IBM, поставляемый в составе Hadoop-решения BigInsights, и используемый, например, в проектах интеграции финансовых и социальных данных. В качестве языка спецификаций выбран язык Нотация абстрактных машин (AMN) [4], поддержанный средствами формального доказательства.

### Литература

1. Getoor L., Machanavajjhala A. Entity resolution for big data // KDD'13: 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining Proceedings, 2013. P. 1527–1527.
2. Luna Dong X., Naumann F. Data fusion — resolving data conflicts in integration // Proc. VLDB Endowment, 2009. Vol. 2. Iss. 2. P. 1654–1655.
3. Hernandez M., Koutrika G., Krishnamurthy R., Popa L., Wisnesky R. HIL: A high-level scripting language for entity integration // EDBT'13: 16th Conference (International) on Extending Database Technology Proceedings, 2013. P. 549–560.
4. Abrial J.-R. The B-Book: Assigning Programs to Meanings. Cambridge: Cambridge University Press, 1996.
5. С. А. Ступников. Формальная семантика языка разрешения сущностей и слияния данных и ее применение для верификации потоков работ интеграции данных // Аналитика и управление данными в областях с интенсивным использованием данных: XVIII Международная конференция DAMDID/RCDL'2016. – М.: ФИЦ ИУ РАН, 2016. С. 247-255



### **Обнаружения знаний в базах данных с помощью мультимодальной кластеризации**

Игнатов Д.И, к.т.н. – НИУ ВШЭ

Анализ формальных понятий (АФП) возник в начале 1980-х гг. как математическая теория, опирающаяся на математическую формализацию понятия и понятийного мышления, а впоследствии стал популярным подходом к алгебраическому анализу данных. Иерархии понятий извлекаются из объектно-признаковых данных, а их диаграммы порядка используются для визуализации и последующего анализа. Каждое формальное понятие можно рассматривать как абсолютно плотный бикластер для исходных объектно-признаковых данных. В докладе представлена попытка поиска «оптимальных» паттернов в триадических данных при наличии дополнительного по отношению к объектам и признакам измерения – условий. Типичным примером таких данных являются фолксономии в системах совместного доступа к ресурсам, когда имеются множества пользователей, ресурсов (например, документов) и тегов, которые пользователи присвоили конкретным ресурсам. В докладе обсуждаются результаты сравнения пяти методов трикластеризации по ресурсной эффективности, шумоустойчивости и четырем критериям качества и будет показано, что задача поиска оптимального трикластерного покрытия NP весьма нетривиальна, а подсчет таких покрытий относится к классу #P-полных задач. Кроме этого обсуждается применимость рассмотренных методов к задачам анализа реляционных данных в различных областях.

### **Алгоритм бустинга с динамическим преобразованием признаков**

Китов В.В., к.ф.-м.н., Гой А.С. – МГУ

Алгоритм бустинга является одним из наиболее популярных в машинном обучении методов прогнозирования, применяемом как для задач регрессии, так и задач классификации. Он получил широкую популярность начиная с работы Френда и Шапиро в 1997 году благодаря высокой точности и сравнительно несложной оптимизации. В алгоритме градиентного бустинга строится ансамбль из базовых прогнозирующих моделей. Для повышения вычислительной эффективности базовые модели строятся последовательно, без перенастройки ранее созданных моделей. Прогноз строится с помощью суммы базовых моделей, где каждая последующая модель пытается исправить ошибки прогноза предшествующих базовых моделей. Как правило, в качестве базовых моделей используются решающие деревья, которые имеют следующие преимущества: они быстро настраиваются, самостоятельно отбирают признаки и способны органично учитывать как дискретные, так и непрерывные признаки. Недостатком стандартных реализаций решающих деревьев (CART, C4.5) является то, что они могут строить разбиения признакового пространства лишь параллельно осям координат – в каждом узле тестируется условие, выше или ниже заданного признака определенного порогового значения. Это накладывает ограничения на моделируемую зависимость, которая, в общем случае, этим ограничениям не удовлетворяет: в случае классификации границы между классами могут идти наклонно, а не параллельно осям координат, а в случае задачи изо-линии моделируемой функции также, в общем случае, будет не параллелен ни одной из осей координат. Из-за такого модельного ограничения снижается точность аппроксимации целевой величины и, как следствие, точность прогнозов. Для компенсации этого ограничения необходим большой объем обучающих данных, чтобы аппроксимировать зависимости

общего вида избыточным количеством разбиений вдоль осей.

Поскольку каждое решающее дерево обладает описанным ограничением, то и ансамбль решающих деревьев, полученный в результате процедуры бустинга, также будет обладать указанными ограничениями. В докладе предлагается описание подхода для решения описанного ограничения за счет динамических поворотов признакового пространства на каждом этапе бустинга перед построением новой базовой модели. Повороты переводят признаки к новым осям, которые соответствуют либо главным компонентам, либо осям, которые наилучшим образом дискриминируют ошибочные прогнозы от корректных в задаче классификации. Указанные повороты пересчитываются на каждом шаге бустинга, чтобы помочь решающим деревьям аппроксимировать целевую функцию более точно за меньшее число шагов. В указанной процедуре по построению решающие деревья получают возможность производить любые наклонные разбиения. Кроме этого в докладе приводятся данные численных экспериментов, демонстрирующие преимущества указанного подхода.

## **Кредитный скоринг на основе анализа формальных понятий: персональные модели**

Масютин А.А. – НИУ ВШЭ

Коммерческие банки работают в условиях жесткой конкуренции, следствием которой является снижение маржи. При росте клиентской базы кредиты начинают предоставляться заемщикам с более низкой кредитоспособностью, что чревато обострением проблемы неблагоприятного отбора – ростом кредитных рисков. С другой стороны, растут затраты банков на привлечение и удержание клиентов. Для решения этих задач необходимо структурировать, обрабатывать огромные потоки данных о профиле клиента, его транзакциях, поведении, интересах для извлечения закономерностей («knowledge extraction», «rule induction»). Вместе с тем, классические методы, основанные на логистической регрессии, демонстрируют инфериорную точность прогноза по отношению к методам типа «black-box». В докладе показано как анализ формальных понятий позволяет применить методы, показывающие более высокую точность по сравнению с классическими подходами, сохраняя при этом интерпретируемость алгоритма.

## **Графовые СУБД и deep learning в системе «аромовидения»**

Крылов В.В., д.т.н. – НИУ ВШЭ, Кучменко Т.А., д.х.н. – ВГУИТ

Исследования в области искусственных органов чувств направлены на создание полноценных систем искусственного интеллекта, способных, подобно живым организмам, воспринимать окружающий мир. Рассматриваемая в докладе система «аромовидения» умеет «чувствовать» запахи с качеством, соизмеримым с обонянием собаки, используя запаховые пространственные образы, в частности, для навигации в пространстве. В качестве системы первичных сенсоров используется массив разноизбирательных нановзвешивателей, с регистрацией динамических процессов в каждом сенсоре, называемых «электронный нос», а обработка данных, поступающих от «носа» производится на основе триплетного анализа, состоящего в попарном сравнении трех запахов. Такой анализ производится на нейронной сети глубокого обучения, совместно с самоорганизующейся структурой данных. Самоорганизующаяся структура имплементируется с помощью графовой СУБД Neo4J средствами языка Cypher и Java, нейронная сеть пос-

троена и обучена с использованием Keros Tensor Flow Python, а ее предобученная модель реализована в Java. В докладе приводится архитектура системы и особенности реализации прототипа.

### **Метод обработки запросов к распределенной базе данных с использованием фильтра Блума на платформе Spark**

Григорьев Ю.А., д.т.н., Пролетарская В.А. – МГТУ им. Н.Э. Баумана

В последние несколько десятилетий в области обработки данных доминировали реляционные СУБД. В таких системах данные хранятся в виде таблиц, они также предполагают наличие схемы базы данных. Но при работе с Большими Данными разработчики стали испытывать значительные затруднения, выбирая реляционные СУБД: 1) усложнилась процедура агрегирования данных, т.к. это требует чтения записей из большого числа связанных таблиц (возникла проблема потери соответствия), 2) возникло противоречие между необходимостью хранения больших объемов неструктурированных данных и необходимостью их как-то структурировать посредством разработки схемы базы данных, 3) для хранения больших объемов информации необходимо покупать дорогие специализированные аппаратно-программные комплексы параллельных систем баз данных (Teradata, Sun Oracle Database Machine и др.), 4) при наличии большого числа узлов возникает проблема обеспечения требуемой отказоустойчивости системы. Как попытка решить накопившиеся проблемы реляционных баз данных появились альтернативные средства хранения и обработки данных: базы данных NoSQL, платформы параллельных вычислений Hadoop, Spark и др.

В докладе рассмотрено использование системы Spark на примере разработки биллинговой системы, осуществляющей учет объема потребляемых абонентами услуг, расчет и списание денежных средств в соответствии с тарифами компании ([http://citforum.ru/operating\\_systems/linux/billing/](http://citforum.ru/operating_systems/linux/billing/)). Работа такой системы организуется примерно так. От коллекторов поступает поток данных, которые сохраняются в детальной базе данных. Далее информация обобщается и сохраняется в агрегированном хранилище данных. К нему могут обращаться операторы для получения текущих данных или с аналитическими запросами. В настоящее время при разработке новых и использовании существующих биллинговых систем возникают проблемы:

1. Существенный рост трафика. Например, в Facebook наблюдается более 1 млрд активных пользователей в день (<http://lpgenerator.ru/blog/2016/06/10/socseti-novosti-trendy-i-5-interesnejshih-pokazatelej>) и это число неуклонно растет, что влияет и на нагрузку биллинговых систем.
2. Большой объем хранимых данных. Например, ЦОД Facebook насчитывают порядка 200 тыс. серверов, а объем сохраняемых данных увеличивается на 500 Тбайт в день. Растет и объем данных, сохраняемых в биллинговых системах.
3. Отсутствуют развитые системы поддержки принятия управленческих решений по результатам анализа хранилищ данных биллинговых систем, включая анализ стратегических конкурентных преимуществ.

В 2014 году консультанты компании Accenture провели опрос менеджеров высшего звена крупных коммуникационных и медийных компаний Африки, Европы, Латинской Америки и Ближнего Востока – большинство из них были не удовлетворены работой своих биллинговых систем. Основные претензии предъявлялись к низкой масштабируемости, к невозможности работы в режиме реального времени, к низкой производительности биллинга. Все это побуждает к инвестированию в разработку биллинговых систем нового поколения на платформе параллельных вычислений.

Рассмотрим пример реализации биллинговой системы на платформе Spark. Поток данных поступает от коммутаторов в Spark Streaming, где он преобразуется в блоки RDD (Resilient Distributed Datasets). Эти блоки равномерно распределяются по всем узлам кластера и там параллельно обрабатываются. Учитывая, что входной поток данных содержит большой объем служебной информации, программы Map1 выделяют из блоков требуемые поля по специальным шаблонам. Эти данные сохраняются в детальной базе данных. Далее эта информация параллельно интегрируется и сохраняется в хранилище данных, которое равномерно распределено по узлам кластера. Каждый запрос к хранилищу обрабатывается параллельно на разных узлах кластера.

Система Spark отлично масштабируется, обработка на каждом этапе ведется параллельно на многих серверах. Это позволяет обеспечить высокую эффективность обработки входного потока данных и запросов операторов.

В докладе предлагаются решения, позволяющие существенно уменьшить время выполнения сложных аналитических запросов к биллинговой системе – одно из таких решений основано на применении фильтра Блума.

Фильтр Блума – это специально построенный массив битов на основе хеширования значений ключей таблиц измерений. Внешние ключи записи таблицы фактов обрабатываются фильтром Блума. Этот фильтр дает «осечку» с вероятностью  $(1/2)^{K}$ , где K – параметр фильтра Блума. При достаточно большом значении K можно добиться, чтобы вероятность ложноположительного срабатывания фильтра была бы очень небольшой. В этом случае уменьшается число записей, участвующих в соединении, а в некоторых случаях вообще не требуется выполнять соединение таблицы измерений с таблицей фактов. Это позволяет существенно уменьшить время обработки запросов.

Фильтр Блума уже применяется на практике, например в хранилище данных типа «звезда» [1] – метод (SBFCJ, Spark Bloom-Filtered Cascade Join) выигрывает по времени реакции и объему данных, пересылаемых между узлами (shuffled bytes) и дополнительно сохраняемых на диске (disk spill) по сравнению с методами, реализованными в MapReduce (MR). Ему конкуренцию составляет метод SBJ (Spark Broadcast Join), в котором все таблицы измерений целиком кэшируются в оперативной памяти на каждом узле. Но при использовании метода SBJ наблюдаются сильные флуктуации объемов используемой оперативной памяти при достаточно больших размерах хранилища и небольших объемах ОП [1].

Авторами доклада был разработан метод параллельной обработки запроса к хранилищу данных с произвольной структурой с Каскадным Использованием Фильтра Блума (КИФБ). Этот метод, как и в случае SBFCJ и SBJ, основан на использовании широковещательных переменных Spark (broadcast). В Hadoop для широковещательной рассылки используется объект DistributedCache. Число перемещений равно произведению количества блоков (split) таблиц измерений на число задач Map, где обрабатываются эти таблицы измерений. В Spark широковещательная рассылка организована

более рационально – эта платформа поддерживает следующие механизмы [2]:

1. Centralized HDFS Broadcast (CHB). Широковещательная переменная (ШП) сохраняется в файловой системе HDFS. Если ее размер большой, то ШП делится на блоки (по умолчанию размер блока равен 128 Мбайтов) и они распределяются по узлам кластера. Причем для каждого блока сохраняются несколько реплик (по умолчанию 3). Задача (worker) читает блоки ШП, которые располагаются ближе к станции, выполняющей задачу.
2. Chained Streaming Broadcast (CSB). ШП длиной  $Z$  делится на небольшие блоки размером  $ZB$  (`spark.broadcast.blockSize=4` Мбайта). Узел-драйвер последовательно передает блоки какому-либо slave-узлу, запросившему ШП. Другие slave-узлы могут подключаться или к узлу-драйверу, или к slave-узлам, и читать поступившие туда текущие блоки. Slave-узел сохраняет прочитанные блоки в «куче» узла (одна ШП на все задачи узла). Таким образом, slave-узлы образуют дерево, в корне которого располагается узел-драйвер. Это дерево образует конвейер, по которому блоки параллельно передаются от узла к узлу до тех пор, пока не будет передана вся широковещательная переменная всем slave-узлам.
3. BitTorrent Broadcast (BTB). Seed-узел (узел-драйвер, где уже полностью хранится ШП) раздает «куски» (256 Кбайтов) широковещательной переменной получателям (leecher-узлам). Те leecher-узлы, которые продолжают раздавать данные, становятся peer-узлами. Другой leecher-узел может подключиться к нескольким peer-узлам. Он может сформировать какой-либо «кусочек», параллельно читая небольшие блоки (по 16 Кбайтов) из этих peer-узлов. Leecher-узел становится peer-узлом, если он полностью сформирует «кусочек» (256 Кбайтов) и проверит его целостность.

В [2] экспериментально показано, что второй вариант по производительности лучше. Например, для  $N=40$  узлов и при размере широковещательной переменной, равной 1 Гб, метод CHB начинает «проседать», при  $N=60$  время CHB в 5.5 раз хуже, чем время CSB. Было выполнено сравнение 4-х вариантов выполнения запроса Q3 из теста TPC-H для коэффициента масштабируемости  $SF=2$ : через консоль Hive (в среде Hive); через консоль `ruspark` с помощью стандартного `SQLContext` (Spark SQL); через консоль `ruspark` с каскадным использованием фильтра Блума (КИФБ) – разработанный метод; через консоль `ruspark` без использования фильтра Блума (БИФБ) – таблицы измерений и таблица фактов соединялись напрямую.

Эксперименты проведены на одном узле (без shuffle): двудерный процессор, операционная система CentOS 7.2 x64, оперативная память - 4Гбайт, два диска 200 Гбайт. Выполнение запроса повторялось несколько раз для каждого варианта. Результаты: Hive - 100-120 с, Spark SQL – 25-27 с, КИФБ – 30-32 с, БИФБ – 46-47 с. Таим образом, результаты для КИФБ и Spark SQL оказались минимальными и почти совпали.

Эксперименты будут продолжены на кластере из нескольких узлов с целью оценки влияния «перетасовки» (shuffle) на время выполнения запросов для каждого варианта.

### Литература

1. Jaqueline Joice Brito, Thiago Mosqueiro, Ricardo Rodrigues Ciferri, and Cristina Dutra de Aguiar Ciferri. Faster cloud Star Joins with reduced disk spill and network communication. ICCS 2016. The International Conference on Computational Science, Volume 80, 2016, Pages 74–85.
2. Chowdhury M. Performance and Scalability of Broadcast in Spark //2014-10-08]. <http://www.cs.berkeley.edu/~agearh/cs267.sp10/files/mosharaf-spark-bc-report-spring10.pdf>. – 2014.

### Полезные ссылки

1. Palla K. A comparative analysis of join algorithms using the hadoop map/reduce framework //Master of science thesis. School of informatics, University of Edinburgh. – 2009.
2. Григорьев Ю.А., Плутенко А.Д., Плужников В.Л., Ермаков Е.Ю., Цвященко Е.В., Пролетарская В.А. Теория и практика анализа параллельных систем баз данных. – Владивосток: Дальнаука, 2015. – 336 с.
3. Dharmapurikar, Sarang Krishnamurthy, Praveen Taylor, David E. Longest Prefix Matching Using Bloom Filters // SIGCOMM'03, August 25–29, 2003, Karlsruhe, Germany.
4. Zhou, G. Zhu, Y. Wang, G.Cache Conscious Star-Join in MapReduce Environments. Cloud-I '13 Proceedings of the and International Workshop on Cloud Intelligence, August 26 2013.

### **Ускорение выполнения SQL-запросов в СУБД PostgreSQL с использованием динамической компиляции**

Д. М. Мельник, Р. А. Жуйков, Р.А. Буцацкий, Е. Ю. Шарыгин – ИСП РАН

Сегодня в СУБД PostgreSQL для исполнения SQL-запросов применяется интерпретатор, реализующий модель итераторов (Volcano-модель), однако можно добиться существенного ускорения, выполняя динамическую компиляцию запроса «на лету». В этом случае появляется возможность генерации кода специально для конкретного SQL-запроса, а также применения оптимизации при компиляции, учитывая, что во время выполнения уже известна структура используемых таблиц и типы данных. Такой подход особенно актуален для сложных запросов, скорость выполнения которых ограничена производительностью процессора. В работе рассматривается использование динамической компиляции для ускорения различных этапов выполнения SQL-запросов с использованием компиляторной инфраструктуры LLVM (Low Level Virtual Machine), а также разобраны необходимые для эффективной работы динамического компилятора изменения архитектуры СУБД PostgreSQL.

На первом этапе была реализована динамическая компиляция выражений в предикате WHERE, что позволило получить двукратное ускорение для простых запросов. Однако, для существенного улучшения производительности на сложных запросах необходимо также реализовать динамическую компиляцию не только для фазы фильтрации, но и для остальных этапов выполнения запроса: сканирование, агрегация, сортировка, объединение. Кроме того, особенности модели итераторов PostgreSQL (виртуальные вызовы и сохранение внутреннего состояния в узлах дерева плана) не позволяют эффективно оптимизировать код при динамической компиляции с помощью LLVM. Поэтому для того, чтобы воспользоваться всеми преимуществами такого подхода, необходимо перейти к модели явной итерации ("push"-модели). Особое внимание уделяется автоматической кодогенерации во время динамической компиляции кода встроенных функций СУБД, используемых в запросах, что позволяет использовать один и тот же исходный код как для динамического компилятора, так и для имеющегося интерпретатора.

В докладе приводятся результаты применения предлагаемого подхода. Для разработанного динамического компилятора выражений (код опубликован в открытом доступе) ускорение на тестах TPC-H составляет 20%. В то же время, разработка расширения для СУБД, реализующая динамическую компиляцию на всех этапах выполнения запроса позволяет увеличить производительность до пяти раз на том же наборе тестов.

### **Концепция реализации произвольных запросов к базам данных на естественном языке**

Сергиевский Г.М., к.т.н. – НИЯУ МИФИ

Проблема реализации запросов к базе данных на естественном языке является одной из старейших задач, исследуемых как в среде специалистов ИИ, так и в среде разработчиков СУБД. Неудача в ее решении до настоящего времени объясняется многими причинами: нерешенность проблемы выбора «языка представления знаний» для формального представления содержания текстов запросов, а также сложность создания парсера естественного языка из-за несовпадения синтаксической (поверхностной) и семантической (глубинной) структур предложений. В докладе излагается концепция, позволяющая преодолеть указанные проблемы. В качестве языка представления знаний предлагается использовать разработанный автором абстракт-

ный язык управления базой данных с условным названием NLQRL, для которого глубинная структура предложения является одновременно скриптом запроса на DML NLQRL. Реализация парсера основана на прямом построении глубинной структуры запроса (минуя определение поверхностной структуры) с помощью методов поиска решений в пространстве состояний. Генерация скрипта на DML целевой СУБД, выполняется с помощью компилятора с DML NLQRL в DML целевой СУБД. В докладе отмечено, что при генерации кода компилятор использует предварительно откомпилированные определения (построенные средствами NLQRL) всех производных понятий, присутствующих в запросе.

### **Infobright: оптимизация SQL запросов с помощью приближений теории неточных множеств**

Кашницкий Ю.С. – НИУ ВШЭ

В большинстве бизнес-задач, связанных с применением алгоритмов машинного обучения, возникают проблемы с интерпретируемостью результатов. В докладе представлены результаты исследования подходов к локальной интерпретации алгоритмов классификации для объяснения того, почему конкретный тестовый пример был классифицирован так или иначе. Известно, что именно из-за сложности с объяснением отнесения тестового примера к тому или иному классу, многие алгоритмы типа нейронных сетей, SVM и композиций деревьев не находят широкого применения в таких задачах, как прогнозирование оттока клиентов и кредитного дефолта клиентов банка, медицинская диагностика и т.д. Хотя, как правило, за счет применения именно этих алгоритмов можно достичь высокого качества классификации применительно к конкретной задаче.

В докладе рассматриваются подходы к интерпретации композиций деревьев (случайный лес и градиентный бустинг над деревьями) и произвольных моделей классификации, а также предлагается алгоритм построения точных и локально интерпретируемых классифицирующих правил.

### **СУБД для вычислительных комплексов семейства «Эльбрус»**

Королев С.Д. – МЦСТ

Процессоры семейства «Эльбрус», разработанные российской компанией «МЦСТ» построены на принципах широкого командного слова (VLIW) и для этой архитектуры был разработан дистрибутив открытого ПО (ОПО «Эльбрус») на базе ОС Debian, включающий на сегодняшний день более четырех тысяч пакетов. На данный момент на вычислительный комплекс портированы такие популярные СУБД, как PostgreSQL-9.5.2, MySQL-5.5.49 и MongoDB-2.4.10. Однако, опыт практического применения и тестирования показывает, что лишь простого портирования требуемых программ на конкретную аппаратную платформу недостаточно для достижения максимальной производительности – требуется адаптация программного кода, предусматривающая также обеспечение надежности и защищенности данных в используемых СУБД.

В докладе изложен опыт работы и адаптации различных программ к особенностям архитектуры «Эльбрус», в частности, приведены результаты исследования производительности на стандартных тестах и конкретных прикладных задачах. На основании исследований была проведена оптимизация работы СУБД. Кроме того, к защищенной операционной среде «Эльбрус» предъявляются требования соответствия руководящим документам по защите



информации, что потребовало реализовать механизм мандатного разграничения доступа в СУБД PostgreSQL. Для повышения надежности и скорости работы СУБД был создан специализированный сервер «Эльбрус-4.4 БД», включающий в себя аппаратную реализацию RAID-контроллера. Проведено портирование и отладка распределенной системы хранения данных Ceph. Также в ОПО «Эльбрус» добавлены системы Postgres-XL и Greenplum, основанные на технологиях PostgreSQL и специально разработанные для эксплуатации в кластерных системах. Кроме этого, в составе ОПО «Эльбрус» имеется реализация Nadoor и портирована СУБД ModgoDB.

Сегодня СУБД на платформе «Эльбрус» используются, в частности, в ФГУП НИИ «Восход» для решения задачи изготовления, оформления и контроля заграничных паспортов, в Пенсионном фонде России для поддержки сервисов информирования застрахованных лиц, а также в МО РФ. Дистрибутив ОПО «Эльбрус» в дальнейшем будет расширен нереляционными СУБД. Кроме этого будут продолжены работы по тестированию с целью повышения надежности и эффективности средств работы с данными, а также расширения спектра пользовательских задач.

### Секция. Стендовые доклады

#### **Метрики качества для бикластеризации**

Кузнецов С.О., д.ф.-м.н., Махалова Т.П. – НИУ ВШЭ

Формальные понятия или замкнутые множества признаков играют важную роль во многих задачах анализа данных, поскольку используются для построения как ассоциативных правил, так и таксономий предметных областей. Полученные в результате анализа иерархические структуры, как правило, содержат большое число элементов, количество которых увеличивается экспоненциально при добавлении шума в данные. Одним из наиболее распространенных способов отбора содержательных бикластеров является их оценка с помощью индексов. Подобный подход позволяет не только выделить наиболее интересные для эксперта группы объектов, но и нивелировать влияние шума в данных.

В докладе приводятся результаты сравнительного анализа индексов в контексте таких задач как отбор наиболее содержательных замкнутых множеств признаков и восстановление структуры решетки, построенной на основе зашумленных данных. Выделенные в результате проведенных вычислительных экспериментов группы коррелирующих индексов могут использоваться для снижения вычислительной сложности процесса анализа данных посредством выбора индексов с наименьшей вычислительной сложностью в группе попарно коррелирующих. Определенную в результате экспериментов способность каждого индекса устранять шум в данных (т.е. степень устойчивости к случайно добавленному шуму) следует учитывать при анализе зашумленных данных.

#### **Поиск закономерностей в базе данных демографических последовательностей на основе узорных структур**

Гиздатуллин Д.К. – НИУ ВШЭ

Анализ демографических последовательностей – популярное и многообещающее направление развития демографии. Жизнь людей можно рассматривать как последовательность происходящих событий – исследователям в области демографии интересен переход от анализа отдельных событий и их взаимосвязей к анализу полных последовательностей событий.

Демографическое поведение может сильно различаться среди людей разных поколений, разного пола, уровня образования, и т. д., однако с помощью специально изобретенных техник могут быть найдены и обобщены скрытые сходства. И хотя уже изобретено множество методов для решения этой

задачи она все еще далека от того, чтобы решаться стандартными методами анализа последовательностей – использование методов майнинга данных открывает для демографов новые возможности. Однако, как показано в работе, некоторые стандартные методы, которые используются в традиционном анализе последовательностей, не могут быть использованы напрямую и требуют специальной адаптации под нужды исследователей из других областей.

В работе представлены результаты экспериментов применения узорных структур на последовательностях демографических данных в России. Используются данные об одиннадцати поколениях в период с 1930 по 1984 годы для трех демографических волн, имевших место в 2004, 2007 и 2011 годах. Основная задача состояла в поиске закономерностей, которые являются (замкнутыми) частыми префиксами без “разрывов”. Эти ограничения – естественное требование демографов, необходимое для изучения первых событий на этапе взросления. Для решения этой задачи использованы узорные структуры неразрывных последовательностей и модифицированные FP-деревья. Наилучшие результаты в терминах TPR-FPR были получены при больших значениях параметра роста (с некоторым числом отказов от классификации).

#### **Хранилище системы сбора статей электронных библиотек**

Незнанов А.А., Паринов А.А. – НИУ ВШЭ

Поиск и анализ научных статей и других материалов, основанных на естественном языке, занимает значительное время в работе современного ученого и инженера – количество научных статей настолько велико, что актуальным является развитие средств, собирающих данные об опубликованных материалах и рекомендующей наиболее релевантные и/или значимые. Учитывая огромный объем информации, важной частью системы рекомендаций является хранилище аннотаций статей, текстов, ключевых слов и ссылок. В докладе рассмотрена оригинальная структура хранилища сервиса сбора и предобработки материалов электронных библиотек, позволяющей использовать методы анализа формальных понятий и рекомендательных систем для эффективного интерактивного поиска многоязычных материалов на основе автоматического порождения понятий и анализе предшествующих обращений к сервису. Также рассмотрена проблема дополнения результатов поиска информацией из внешних источников.

## Партнеры конференции

### «Интеллектуальные системы хранения и обработки информации»



**Oracle** упрощает ИТ и ускоряет инновации в бизнесе. Исключительная производительность и эффективность технологий, простая ИТ-инфраструктура и сокращение затрат – это то, что создает Oracle, помогая более 400 000 заказчиков в 145 странах мира развивать инновации и улучшать качество обслуживания клиентов.

Подробнее: <http://www.oracle.com/ru>



Южнокорейская компания **TmaxSoft** основана в 1997 году и в данный момент является глобальной софтверной компанией с лидирующими технологиями (№1 в Корее на рынке WAS). Основа компании – высокая доля R&D-активов, отлаженная техническая поддержка, инновационные технологии и уникальное видение будущего ИТ-индустрии.

Штаб-квартира TmaxSoft расположена в Сеуле, в 2013 году были открыты региональные подразделения в России, Великобритании, Бразилии, Японии, Китае, Сингапуре и США. Представительство TmaxSoft в Москве имеет в своем составе квалифицированных инженеров, способных оказывать техническую поддержку, принимать участие в проектах миграции, а также выполнять бесплатные тестирования.

Флагманский продукт TmaxSoft Tibero – высокопроизводительная СУБД с поддержкой Oracle PL/SQL, кластеризации, со своими средствами отказоустойчивости, репликации и резервного копирования.

Компания также предлагает решения для организации онлайн-репликации данных (с функцией ETL), сервер веб-приложений и веб-сервер, средства мониторинга и оптимизации производительности СУБД и серверов приложений.

Кроме того, компания предлагает решение OpenFrame для рехостинга мейнфреймов IBM – миграции приложений без изменения исходного кода и их запуска на открытых платформах Unix или Linux.

Программные продукты TmaxSoft успешно заменяют решения ведущих игроков рынка и позволяют заказчикам сэкономить до 60% на лицензионных отчислениях и успешно реализовывать проекты миграции с сохранением требуемого уровня надежности, функционала и производительности.

Подробнее: <http://www.tmaxsoft.com/>

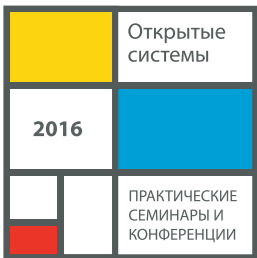


**Visiology** – российская ИТ-компания. Мы разрабатываем программное обеспечение для анализа и визуализации данных, а также осуществляем внедрение своих разработок в коммерческих компаниях и государственных организациях. Технологии Visiology позволяют нашим клиентам лучше работать с данными, принимать более обоснованные и правильные управленческие решения – это особенно важно сегодня, когда объемы накопленных данных растут во взрывном темпе, и преимущество получают те организации, которые могут справиться с этим информационным потоком и извлечь из него выгоду.

Основной продукт компании Visiology – одноименная аналитическая платформа, преимуществами которой являются продвинутая визуализация, поддержка современных технологий анализа данных (включая Big Data и Predictive Analytics) и гибкость настройки. Мы также разрабатываем на основе платформы аналитические решения для различных отраслей, таких как строительство, фармацевтика, дистрибуция, и многих других.

Многие внедрения мы осуществляем совместно с нашим партнером – системным интегратором Polymedia. Многолетний опыт специалистов Polymedia в области управления проектами, построения ситуационных центров и внедрения информационных систем позволяет нам успешно реализовывать проекты любой сложности.

Подробнее: <http://ru.visiology.su/>



Программа и тезисы докладов конференции  
**Интеллектуальные системы хранения и обработки информации**

**Москва, 2016**  
©2016 ООО «Открытые системы»