УДК 004.651:623

Еркетаев Н.М. [1], Мукажанов Н.К. [2]

[1,2] Международный университет информационных технологий, Алматы, Казахстан

ЭФФЕКТИВНОЕ ХРАНЕНИЕ НЕСТРУКТУРИРОВАННЫХ ДАННЫХ

Аннотация. В этой статье представлены результаты изучения того, как могут быть использованы различные типы информации для хранения неструктурированных данных в базе данных и классической файловой системе. Также объясняются современные методы хранения данных, предыстория проблемы и ключевые элементы, важные для этого исследования: различные методы хранения неструктурированных данных (преимущества и недостатки), используемые типы данных и экспериментальная среда. Наша основная цель — найти эффективный метод хранения неструктурированных данных.

Ключевые слова: база данных, неструктурированные данные, файловый поток, эффективность, производительность.

Введение

Хорошо известно, что одним из результатов быстрого роста Интернета стало значительное увеличение объема информации, генерируемой и распространяемой организациями практически во всех отраслях и секторах. Проблема не только в генерации данных, но и в их хранении и доступе к ним. Менее известна степень потребности в больших объемах дорогостоящих ресурсов, как человеческих, так и технических, обусловленной происходящим информационным взрывом. Эти потребности, в свою очередь, создали столь же большую, но в значительной степени неудовлетворенную потребность в инструментах, которые можно использовать для управления тем, что мы называем неструктурированными данными. Следует признать, что термин неструктурированные данные может означать разные вещи в разных контекстах. Например, в контексте систем реляционных баз данных это относится к данным, которые нельзя хранить в строках и столбцах. Вместо этого такие данные должны храниться в BLOB (большом двоичном объекте), универсальном типе данных, доступном в большинстве программных систем управления реляционными базами данных (RDBMS). Здесь под неструктурированными данными понимаются файлы электронной почты, текстовые документы, презентации, файлы изображений и видеофайлы.

С другой стороны, стремительный рост объемов данных обусловлен достижениями и серьезными преобразованиями в технологии магнитной записи. Стоимость хранения резко упала с более чем 4 долларов за мегабайт в 1990 году до менее чем 0,001 доллара за мегабайт сегодня.

Согласно исследованию, проведенному IDC, ведущей фирмой по исследованию и анализу рынка информационных технологий, объем данных, которые будут собираться, храниться и воспроизводиться по всему миру, вырастет со 161 эксабайта в 2006 году до 988 эксабайт в 2010 году (1 эксабайт = 1018 байт). Два ключевых вывода этого исследования:

- Основная часть этих данных будет в виде изображений, снятых большим количеством устройств, таких как цифровые камеры, телефоны с камерами, камеры наблюдения и медицинское оборудование для визуализации. Большая часть этих данных должна храниться и управляться централизованными системами внутри организаций. Исследование показывает, что к 2010 году, хотя предприятия будут создавать, собирать и тиражировать только 30% цифровой вселенной [3], им придется хранить более 85% всех данных и управлять ими.
- Более 95% цифровой вселенной — это неструктурированные данные. Согласно этому исследованию, 80% всех хранимых организациями данных не структурированы.

Ожидается, что эта тенденция роста сохранится и в будущем, поскольку потребует эффективных способов хранения, поиска, структурирования и обеспечения безопасности неструктурированных данных. Об аналогичных тенденциях в отношении важности обработки неструктурированных данных также сообщили другие исследовательские и консультационные фирмы, такие как Gartner Group и Butler Group [4].

Очевидно, что неструктурированные данные — это наша реальность, и поиск эффективного метода хранения данных является ключевым аспектом этого исследования и будущих результатов в этой области. В настоящей статье объясняются современные методы хранения данных.

Связанная работа

Существует множество исследований, основанных на тестировании систем баз данных. Одна статья, представляющая собой хорошую отправную точку для нашей работы, особенно интересна: «Набор тестов для обработки неструктурированных данных» [2].

Основной целью авторов было собрать набор рабочих нагрузок, выявить и изучить широкий спектр приложений для обработки неструктурированных данных, выявить их ключевые характеристики обработки и доступа к вводу-выводу, а также составить рабочие нагрузки, воплощающие эти характеристики. Поэтому, чтобы спроектировать системы хранения для этого важного развивающегося класса приложений, они создают набор тестов, который может фиксировать их характеристики обработки и ввода-вывода.

Пакет Benchmark состоит из четырех рабочих нагрузок:

- Обнаружение края;
- Поиск близости;
- Сканирование данных;
- Слияние данных.

Был сделан вывод, что приложения, использующие неструктурированные данные для бизнес-процессов, очень интенсивны по вводу-выводу и предъявляют высокие требования к системе хранения [2].

Но предварительные исследования оставляют возможность продолжить подобные эксперименты с неструктурированными данными.

История исследований

В следующем разделе будут объяснены предыстория и ключевые элементы, важные для этого исследования: различные методы хранения неструктурированных данных (преимущества и недостатки), используемые типы данных и экспериментальная среда. Наша основная цель — найти эффективный метод хранения неструктурированных данных. В ходе этого процесса мы проведем обширный сравнительный анализ на основе четко определенных параметров.

Исследование основано на следующих методах хранения неструктурированных данных:

- Неструктурированные данные в среде реляционных баз данных;
- Неструктурированные данные вне реляционной среды.

Мы начнем с объяснения этих методов, каждого преимущества и побочных эффектов.

A. Неструктурированные данные в реляционной среде

Спор обычно возникает относительно темы реляционных баз данных и данных больших двоичных объектов (BLOB): интегрировать большие двоичные объекты в базу данных или хранить их в файловой системе? Каждый метод имеет свои преимущества и недостатки.

Хранение неструктурированных данных, таких как изображения, аудиофайлы и исполняемые файлы в базе данных с типичными текстовыми и числовыми данными, позволяет вам хранить вместе всю связанную информацию для данного объекта базы данных

(рис. 1). И этот подход позволяет легко искать и извлекать данные BLOB; вы просто запрашиваете соответствующую текстовую информацию. Однако хранение неструктурированных данных может значительно увеличить размер ваших баз данных.

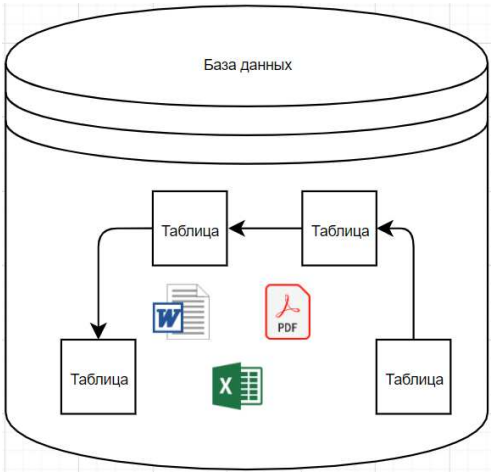

Рисунок 1 - Среда базы данных с неструктурированными данными внутри

Таблица-1. Характеристики базы данных

	Характеристики базы данных				
	Назначение базы данных	Размер с неструктурированными данными	Количество пользователей	Количество BLOB-записей	Время резервного копирования
1.	LMS/CMS	≈ 12 ГБ	≈ 5000 студентов	≈ 7000 файлов размером от 100 Кб до 15 Мб	25 мин.

Основные преимущества использования этого метода:

- Когда данные хранятся в базе данных, резервная копия является согласованной. Нет необходимости в отдельной политике резервного копирования;
- Когда данные хранятся в базе данных, это часть транзакции. Так, например, откат включает все традиционные операции с базой данных вместе с операциями с двоичными данными. Обычно это делает клиентское решение более надежным и с меньшим количеством кода.

Основные недостатки использования этого метода:

- Хранение неструктурированных данных позволяет значительно увеличить размер баз данных;
- Резервное копирование и восстановление может занять много времени;
- Проблемы с производительностью подсистем ввода-вывода;

Б. Неструктурированные данные вне реляционной среды

Распространенной альтернативой вышеописанному методу является хранение двоичных файлов вне базы данных с последующим включением в качестве данных в базу данных пути к файлу или URL-адреса объекта.

Этот отдельный метод хранения имеет несколько преимуществ по сравнению с интеграцией данных BLOB в базе данных. Это несколько быстрее, потому что чтение данных из файловой системы требует немного меньше накладных расходов, чем чтение

данных из базы данных. А без больших двоичных объектов базы данных, как правило, меньше.

Однако нам необходимо вручную создать и поддерживать связь между базой данных и файлами внешней файловой системы, которые могут рассинхронизироваться. Кроме того, обычно нам требуется уникальная схема именования или хранения файлов ОС, чтобы четко идентифицировать потенциально сотни или даже тысячи файлов BLOB [1].

Хранение данных BLOB в базе данных устраняет эти проблемы, позволяя хранить данные BLOB вместе с соответствующими реляционными данными (рис. 2).

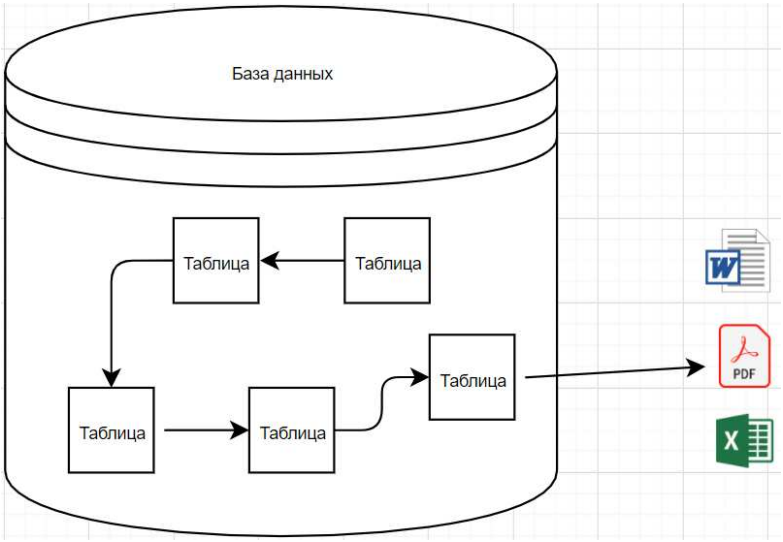

Рисунок 2 - Среда базы данных без неструктурированных данных внутри

В таблице 2 у нас такая же среда базы данных, как и в таблице 1. Единственное отличие состоит в том, что мы исключаем пару таблиц, которые содержат неструктурированные данные внутри физического файла базы данных. Разница более чем очевидна: размер базы данных при одинаковом количестве пользователей составляет половину от первоначального. Давайте подробнее рассмотрим преимущества этого метода.

Таблица-2. Характеристики базы данных

	Характеристики базы данных				
	Назначение базы данных	Размер с неструктурированными данными	Количество пользователей	Количество BLOB-записей	Время резервного копирования
1.	LMS/CMS	≈ 6 ГБ	≈ 5000 студентов	0	11 мин.

Основные недостатки использования этого метода:

- Когда данные хранятся вне базы данных, резервная копия не согласована;
- Неструктурированные данные не являются частью транзакции.

Основные преимущества использования этого метода:

- Хранение неструктурированных данных за пределами базы данных может снизить размер баз данных и снизить пропускную способность ввода-вывода;

- Резервное копирование и время восстановления могут занять меньше времени;

С. Гибридный способ хранения неструктурированных данных

Проблема с первыми двумя методами заключается в том, что мы не знаем, как фактический размер данных влияет на производительность базы данных. Результаты пока не говорят нам, есть ли разница в хранении BLOBS: 10 КБ, 1 МБ, 100 МБ и т. д.

Основные поставщики баз данных теперь поддерживают гибридный способ хранения неструктурированных данных. В этом случае данные находятся «вне» среды базы данных. Но главное преимущество заключается в том, что BLOB объекты находятся в условиях транзакционной согласованности базы данных (рис. 3).

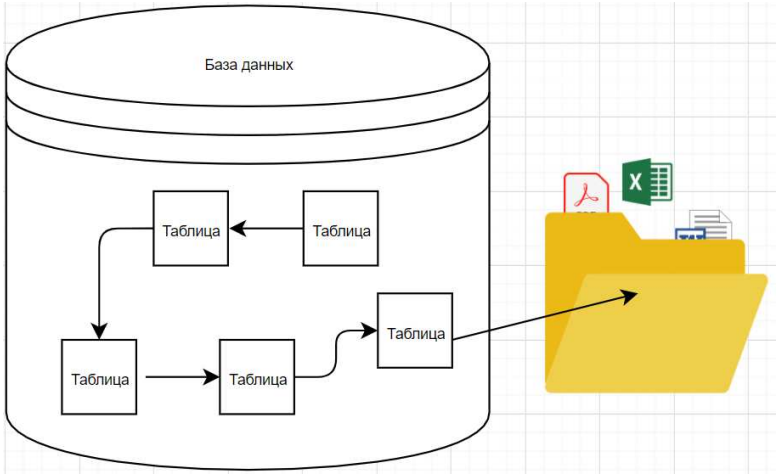

Рисунок 3 - Гибридный способ хранения неструктурированных данных

В нашем случае используются те типы данных, которые предлагаются в новой версии SQL Server 2008. Использование этой новой технологии фактически делает возможным гибридный метод хранения неструктурированных данных. Ядро базы данных поддерживает новый тип данных с именем filestream.

Ключевым аспектом исследования является выяснение эффективности этого метода хранения данных.

Экспериментальная среда

Для этого мы настраиваем тестовую среду со следующими компонентами:

- Процессор: AMD X2 3.0 Ghz 4 ГБ;
- Физическая память;
- Файлы базы данных на C;
- Компонент файлового потока на диске E;
- Диски C: и E: на отдельных физических дисках SATA;
- SQL Server 2008 R2 и клиентское приложение для настраиваемого тестирования находятся на одном компьютере;

На следующих диаграммах показано среднее время загрузки для следующих условий:

- Файл размером 10 КБ, повторяется 3 раза для каждого измерения (рис. 4);
- Файл размером 1 МБ, повторяется 3 раза для каждого измерения (рис. 5);
- Файл размером 10 МБ, повторяется 3 раза для каждого измерения (рис. 6);

Результаты для файла размером 10 КБ

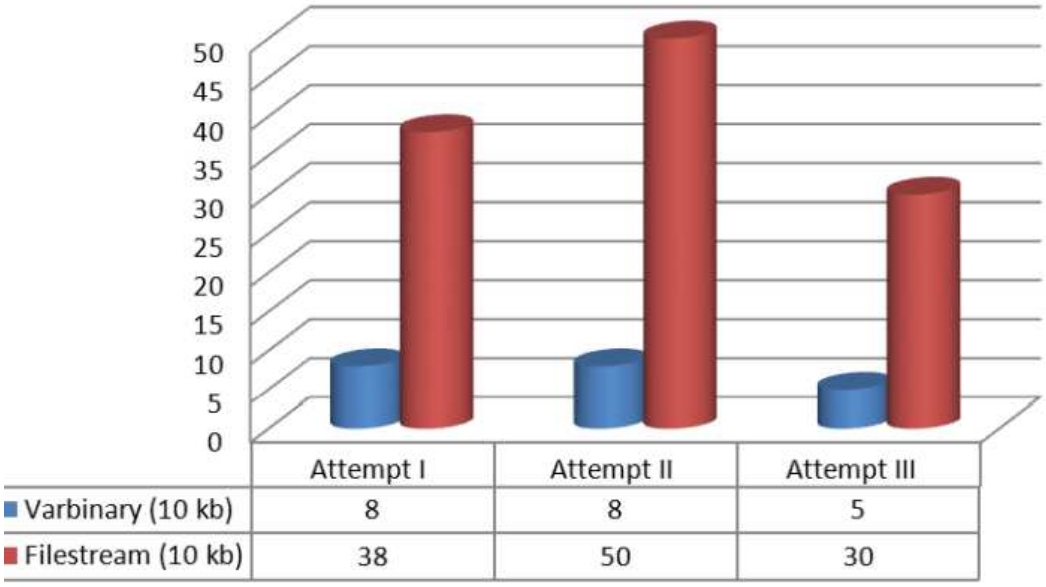

Рисунок 4 - Результаты сохранения файла размером 10 КБ внутри базы данных и файловой системы.

В этом измерении вы можете четко увидеть накладные расходы и влияние на производительность, вызванные использованием файлового потока на “маленькие” файлы. Время при хранении внутри базы данных было каждый раз меньше 10 миллисекунд. С другой стороны, время хранения с использованием файлового потока в 4-5 раз больше.

Результаты для файла размером 1 МБ

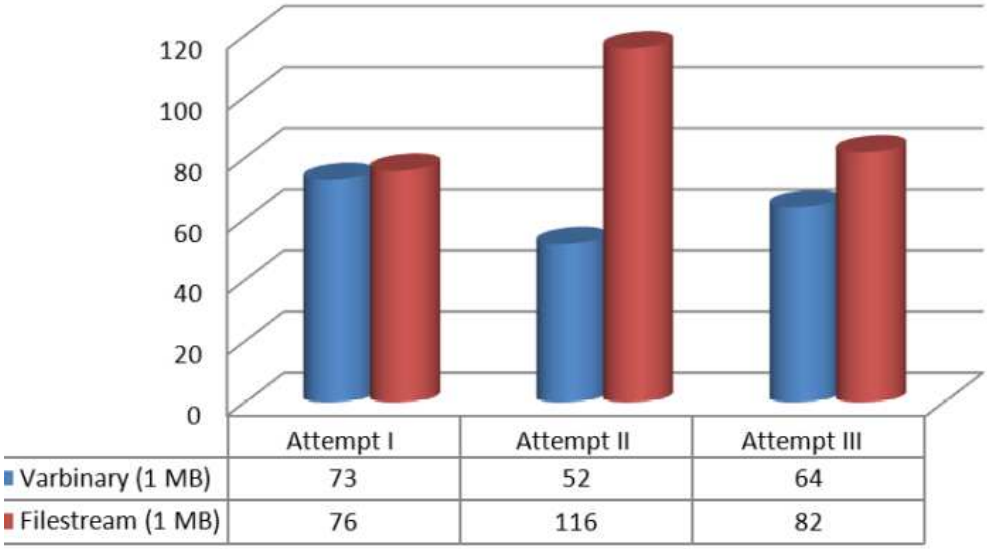

Рисунок 5 - Результаты сохранения файла размером 1 МБ внутри базы данных и файловой системы.

При объеме данных 1 МБ традиционный двоичный файл и файловый поток действуют одинаково, и разница между ними — максимум на один раз быстрее.

Результаты для файла размером 10 МБ

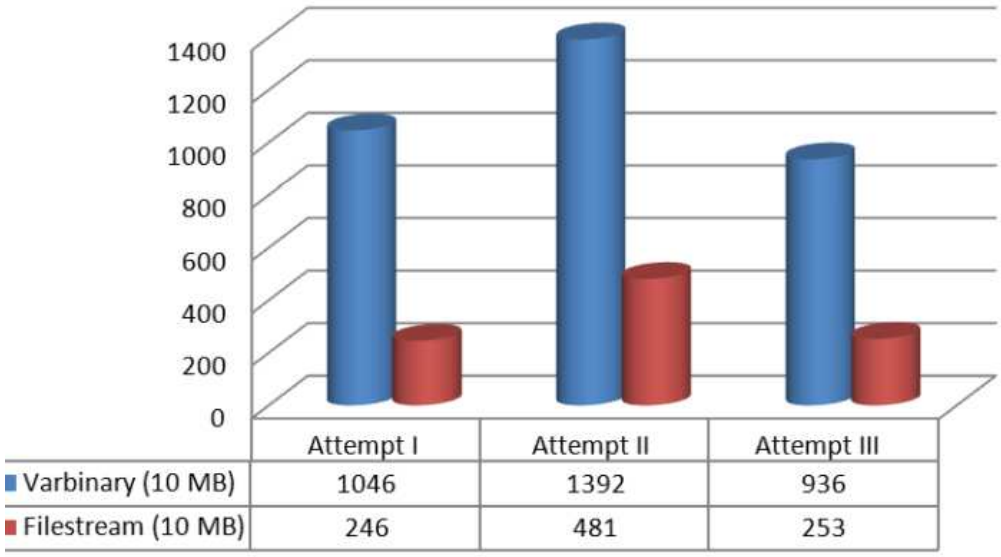

Рисунок 6 - Результаты сохранения файла размером 10 МБ внутри базы данных и файловой системы.

Когда используются файлы размером 10 МБ, хранение данных в традиционном файле базы данных происходит намного медленнее. Основываясь на этих измерениях, можно сказать, что более эффективно использовать файловый поток, когда типичный размер файла составляет около 1 МБ или более. Если файлы имеют небольшой размер (явно менее 1 МБ), традиционное хранилище работает лучше. При измерении времени мы обнаружили, что удаление строк на основе файлового потока происходит намного быстрее, чем при хранении внутри таблицы.

Заключение

В исследовании представлена гипотеза о неэффективности существующих методов хранения и извлечения неструктурированных данных.

Новый способ использования файловой системы при согласованности базы данных был хорошей возможностью опробовать новую технологию в разных областях хранения данных. Примеры тестов сохранения файлов размерами 10 КБ, 1 МБ и 10 МБ в реальных средах информационных систем могут относиться к фотографиям пользователей, файлам резюме, небольшим офисным документам, видео - и аудиофайлам. Результаты исследования могут использоваться для моделирования системы и четкого определения потребностей в хранении данных. Преимущество заключается в получении максимальной производительности на основе аппаратной и программной инфраструктуры. Также в системах, в которых проблемы с производительностью уже существуют, эта модель может помочь выявить узкие места и найти способ их улучшить.

Результаты исследования позволяют создать модель тестирования системы для хранения неструктурированных данных. Дальнейшие шаги по улучшению — реализация этой модели в реальных средах, где важны неструктурированные данные (платформы электронного обучения, социальные сети, порталы хранения видео и т. д.). После этого настоящее исследование может стать официальной методологией анализа системы с потребностями в неструктурированных данных. Современные тенденции [3] показывают нам, что www становится глобальным хранилищем для всех видов данных, где преобладают неструктурированные данные.

СПИСОК ЛИТЕРАТУРЫ

1. Hitachi Global Storage Technologies - Обзорные схемы технологии HDDT. [Электронный ресурс] URL:http://www.hitachigst.com/hdd/technolo/overview/stor agetechchart.html. (дата обращения: 10.02.2021)
2. Набор тестов для обработки неструктурированных данных Smullen, C.W.; Tarapore, S.R.; Gurumurthi, S.; Архитектура сети хранения данных и Параллельный ввод-вывод, 2007г. SNAPI. Международный семинар 24 сентября. 2007 Страниц: 79 – 83.
3. J. Gantzandetal. Расширение цифровой вселенной - прогноз роста мировой информации до 2010 г., март 2007 г. Белая книга IDC.
4. C.White. Консолидация, доступ и анализ неструктурированных данных, O'Reilly Media, 2005 г. Страницы 118–124.
5. Р. Чемберлен, М. Франклин. Использование реконфигурируемости для текстового поиска. В материалах семинара по высокопроизводительным встроенным вычислениям (HPEC), O'Reilly Media, 2006 г. Страницы 1178–1181.
6. Повышение доступности данных с помощью файловых сетей Geer, B; Компьютер. O'Reilly Media, 2017 г. Страницы 1356–1359.
7. Основы классификации неструктурированных данных Островски, Д.А.; Семантические вычисления, 2009. ICSC '09. Международная конференция IEEE, 14–16 сентября 2009 г. Страницы: 373–377.
8. Анализ неструктурированных данных и оптимизация их хранения. [Электронный ресурс] URL: https://habr.com/ru/company/hpe/blog/265499. (дата обращения: 19.02.2021)
9. Преобразование неструктурированных данных из разрозненных источников в знания Plejic, B.; Vujnovic, B.; Penco, R.; Семинар по приобретению знаний и моделированию, 2008 г. Семинар по KAM, 2008 г. Международный симпозиум IEEE 21–22 декабря 2008 г. Страницы: 924 – 927.
10. G.Fountainand и S.Drager. Высокопроизводительная архитектура слияния в реальном времени. O'Reilly Media, 2002 г. Страницы 1478–1485.

REFERENCES

1. Hitachi Global Storage Technologies – Overview diagrams of HDDT technology. [Electronic resource] URL:http://www.hitachigst.com/hdd/technolo/overview/stor agetechchart.html. (date of the application: 10.02.2021)
2. A Benchmark Suite for Unstructured Data Processing Smullen, C.W.; Tarapore, S.R.; Gurumurthi, S.; Storage Network Architecture and Parallel I/Os, 2007. SNAPI.International Workshop on 24 Sept. 2007 Page(s): 79 – 83
3. J.Gantzandetal. The Expanding Digital Universe – A Forecast of Worldwide Information Growth Through 2010, March 2007. IDC Whitepaper
4. C.White. Consolidating, Accessing, and Analyzing Unstructured Data, O'Reilly Media, 2005 Page(s): 118 – 124
5. R. Chamberlain, M. Franklin and R. Index. Using reconfigurability for text search. In High Performance Embedded Computing (HPEC) Workshop Proceedings. O'Reilly Media, 2006 Page(s): 1178 – 1181
6. Increasing data availability with Geer, D. File networks; Computer. O'Reilly Media, 2017 Page(s): 1356 – 1359
7. Fundamentals of unstructured data classification Ostrowski, D.A.; Semantic Computing, 2009. ICSC '09. IEEE International Conference, September 14-16, 2009, Pages: 373-377
8. Analysis of unstructured data and optimization of their storage. [Electronic resource] URL: https://habr.com/ru/company/hpe/blog/265499. (Date accessed: 19.02.2021)
9. Converting unstructured data from disparate sources to knowledge Plejic, B.; Vujnovich, B.; Penco, R .; Knowledge Acquisition and Modeling Workshop 2008 KAM Workshop 2008 IEEE International Symposium December 21-22, 2008 Pages: 924 - 927

10. J.Fountain and S.Drager. High-performance real-time merge architecture. O'Reilly Media, 2002 Page(s): 1478 – 1485

Н.М. Еркетаев, Н.К. Мұқажанов
Құрылымсыз деректерді тиімді сақтау

Аңдатпа. Бұл мақалада құрылымсыз деректерді мәлімет базасында және классикалық файлдық жүйеде сақтау үшін сан түрлі типтегі деректерді пайдалану бойынша зерттеу нәтижелерін ұсынамыз. Ол сонымен қатар деректерді сақтаудың заманауи әдістерін, фондық тарихты және негізгі элементтерді түсіндіреді. Осы зерттеуге қатысты: құрылымдалмаған мәліметді сақтаудың әртүрлі әдістері (артықшылықтары мен кемшіліктері), қолданылатын мәлімет типтері және эксперименттік орта. Біздің басты мақсат – құрылымдалмаған деректерді сақтаудың тиімді әдісін табу.

Түйінді сөздер: мәлімет қоры, құрылымдалмаған мәлімет, файлдар ағыны, тиімділік, өнімділік.

N.M. Yerketayev, N.K. Mukazhanov
Efficient storage of unstructured data

Abstract. In this article we present the research findings on the use of different types of unstructured data stored in a database and the classic file system. It also explains modern data storage methods, background history and key elements. The issues relevant to this research: different methods of storing unstructured data (advantages and disadvantages), the used data types and the experimental environment. Our main goal is to find an efficient method for storing unstructured data.

Keywords: database, unstructured data, file stream, efficiency, performance.

Авторлар туралы мәлімет:

Еркетаев Нұрзат Мейірханұлы, «Компьютерлік инженерия және ақпараттық қауіпсіздік» кафедрасының магистранті, Халықаралық ақпараттық технологиялар университеті.

Мұқажанов Нуржан Какенұлы, PhD, «Компьютерлік инженерия және ақпараттық қауіпсіздік» кафедрасының ассистент-профессоры, Халықаралық ақпараттық технологиялар университеті.

Сведения об авторах:

Еркетаев Нұрзат Мейірханұлы, магистрант кафедры «Компьютерная инженерия и информационная безопасность», Международный университет информационных технологий.

Мукажанов Нуржан Какенович, PhD, ассистент-профессор кафедры «Компьютерная инженерия и информационная безопасность», Международный университет информационных технологий.

About the authors:

Nurzat M. Yerketayev, master student, Department of Computer Engineering and Information Security, International Information Technology University.

Nurzhan K. Mukazhanov, PhD, Assistant-Professor, Department of Computer Engineering and Information Security, International Information Technology University.