Д.Ю. Шенгелия [1], И.А. Бакланов [1], А.М. Воробьёв [1,2], О.А. Иваненко [1]

[1] Тюменский государственный университет, г. Тюмень

[2] Научно-технический университет «Сириус», г.Сочи

УДК 004.62

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕШЕНИЙ ДЛЯ ЗАДАЧ СБОРА И ХРАНЕНИЯ ДАННЫХ О РЕЗУЛЬТАТАХ ОБУЧЕНИЯ СТУДЕНТОВ

Аннотация. В данной статье рассмотрено сравнение нескольких решений для задачи формирования единого хранилища данных, генерируемых в процессе обучения. Было проведено сравнение следующих решений: Apache Cassandra, PostgreSQL, HBase и Filezilla Server. Приведено количественное обоснование выбора конкретного решения для поставленной задачи.

Ключевые слова: хранилище данных, сбор данных, извлечение данных, сравнительный анализ, Apache HBase.

Внедрение индивидуальных образовательных траекторий в Тюменском государственном университете повысило приоритеты задач по автоматизированному анализу данных, генерируемых студентами в процессе обучения [1]. В связи с этим в Институте математики и компьютерных наук ТюмГУ была инициирована собственная разработка по анализу данных об учебном опыте студентов. Важным этапом создания такой системы является построение собственного распределенного хранилища данных, аккумулирующего в себе разнородные данные по студентам, собранные за период их обучения.

Данные, необходимые для сбора и проведения анализа, распределены по нескольким системам, часть из них хранится в виде

отдельных файлов, что затрудняет формирование комплексного взгляда на текущее состояние студента.

Сбор данных осуществляется из источников, ресурсов и систем, которые используются в университете: 1С:Университет, Modeus, образовательный портал на платформе Moodle (elearning.utmn.ru), Microsoft Teams, портал «Вместе» (vmeste.utmn.ru), методический ресурс (op.utmn.ru), результаты опросов студентов, разные файловые хранилища. Причем данные собираются с разной периодичностью:

- тексты учебных программ (УМК) — в начале учебного года;
- тексты студенческих работ (курсовые, отчеты по практике, выпускные квалификационные работы) — раз в семестр;
- видеозаписи с камер в аудиториях — раз в месяц;
- данные о текущей и итоговой успеваемости студентов— раз в месяц и раз в семестр;
- исходные коды программ, написанные студентами в рамках преподаваемых дисциплин и курсовых работ — раз в неделю;
- обратная связь от студентов посредством автоматизированного опроса — раз в 2 недели.

Для консолидации собираемых данных возникла потребность в формировании единого хранилища. Чтобы осуществить обоснованный выбор конкретного хранилища данных, были проведены сравнительные тесты по замеру времени доступа к 50 текстам УМК из таблиц с идентичной структурой, но с различным количеством записей (1000, 5000, 25000 и 100000), предварительно выгруженных в csv-формате и загруженных в тестируемые решения (Apache Cassandra, PostgreSQL, HBase и Filezilla Server, работающий на протоколе SFTP).

Все решения были протестированы на устройстве с аппаратными характеристиками:

- ЦП: AMD Ryzen 7 2700 (8 ядер, 16 потоков)

- ОЗУ: Crucial Ballistix LT 16 ГБ DDR4
- ПЗУ: SSD Plextor M5S 128 ГБ

Для каждого решения был развернут свой узел (количество ядер процессора - 2, ОЗУ - 8 Гб, ПЗУ ~20 Гб).

Получение текстов производилось несколькими способами: SQL-запрос для PostgreSQL, запрос на языке запросов Cassandra (Cassandra Query Language - CQL) для Apache Cassandra, scan-запрос в HBase Shell для Apache HBase, GET-запрос по протоколу SFTP. Сформированные запросы показаны в таблице 1.

Таблица 1. Список запросов для различных решений

Решения/ запросы	До оптимизации	После оптимизации

Apache Cassandra	SELECT umk_text FROM umk LIMIT 50;	// PreparedStatement для предварительной компиляции запроса
		PreparedStatement prepared = session.prepare("SELECT umk_text FROM umk LIMIT 50;")
		session.execute();

PostgreSQL	SELECT umk_text FROM umk LIMIT 50;	CREATE UNIQUE INDEX umk_id_idx ON umk (umk_id); SELECT umk_text FROM umk LIMIT 50;
Apache Hbase	scan 'umk', 'umk:umk_text', {'LIMIT' => 50}	

SFTP	*// Из БД берутся первые 50 уникальных id УМК* SELECT DISTINCT umk_id FROM umk LIMIT 50; *// После этого ищем на SFTP файлы вида "*id УМК*.txt"* var client = new WebClient(); client.Credentials = new NetworkCredential("логин", "пароль"); *// В цикле по i от 0 до 50* var url =$@"ftp://{ip}/texts/{umk_id[i].txt}"; client.DownloadFile(url, "*путь к папке назначения*");

Средний размер одного файла УМК составляет около 100 килобайт. Результат работы является средним значением из 10 замеров и показан в таблице 2.

Таблица 2. Доступ к 50 записям из таблицы с УМК

Кол-во записей в таблице	Время доступа к 50 записям (секунды)			
	Apache Cassandra	PostgreSQL	Apache HBase	SFTP
1000	6,75	6,89	8,25	7,20
5000	21,43	23,32	10,05	9,55
25000	53,98	49,44	15,90	17,60
100000	162,32	193,77	25,20	26,63

С ростом количества записей УМК в случае с Apache Cassandra время доступа к записям нелинейно увеличивается ввиду того, что запрос полностью сканирует кластер и выполняет операцию рассеяния/сборки [2]. Это негативно влияет на производительность запроса при растущем количестве выборки (см. рис.1).

Другое распределенное хранилище данных Apache HBase обработало запрос значительно быстрее, чем Apache Cassandra. Данная

колоночно-ориентированная система хранит данные в распределенной файловой системе (HDFS), что положительно влияет на производительность операции чтения данных [3].

Реляционная БД PostgreSQL показала нелинейное увеличение времени запроса, так как первоначально не была проведена индексация. Увеличение времени запроса к файлам через протокол SFTP близко к линейному, как и в случае с Apache HBase.

Для повышения быстродействия запросов были предварительно проиндексированы столбцы в PostgreSQL. Для Apache Cassandra была выполнена предварительная компиляция запроса [4]. Запросы к Apache HBase и SFTP остались неизменными (см. Таблицу 1). Затем было проведено повторное тестирование по времени чтения данных. Результат работы является средним значением из 10 замеров и показан в таблице 3.

Таблица 3. Доступ к 50 записям из таблицы с УМК после изменений

Кол-во записей в таблице	Время доступа к 50 записям (секунды)			
	Apache Cassandra	PostgreSQL	Apache HBase	SFTP
1000	6,15	5,94	8,25	7,20
5000	10,65	10,10	10,05	9,55
25000	16,44	17,85	15,90	17,60
100000	28,48	27,39	25,20	26,63

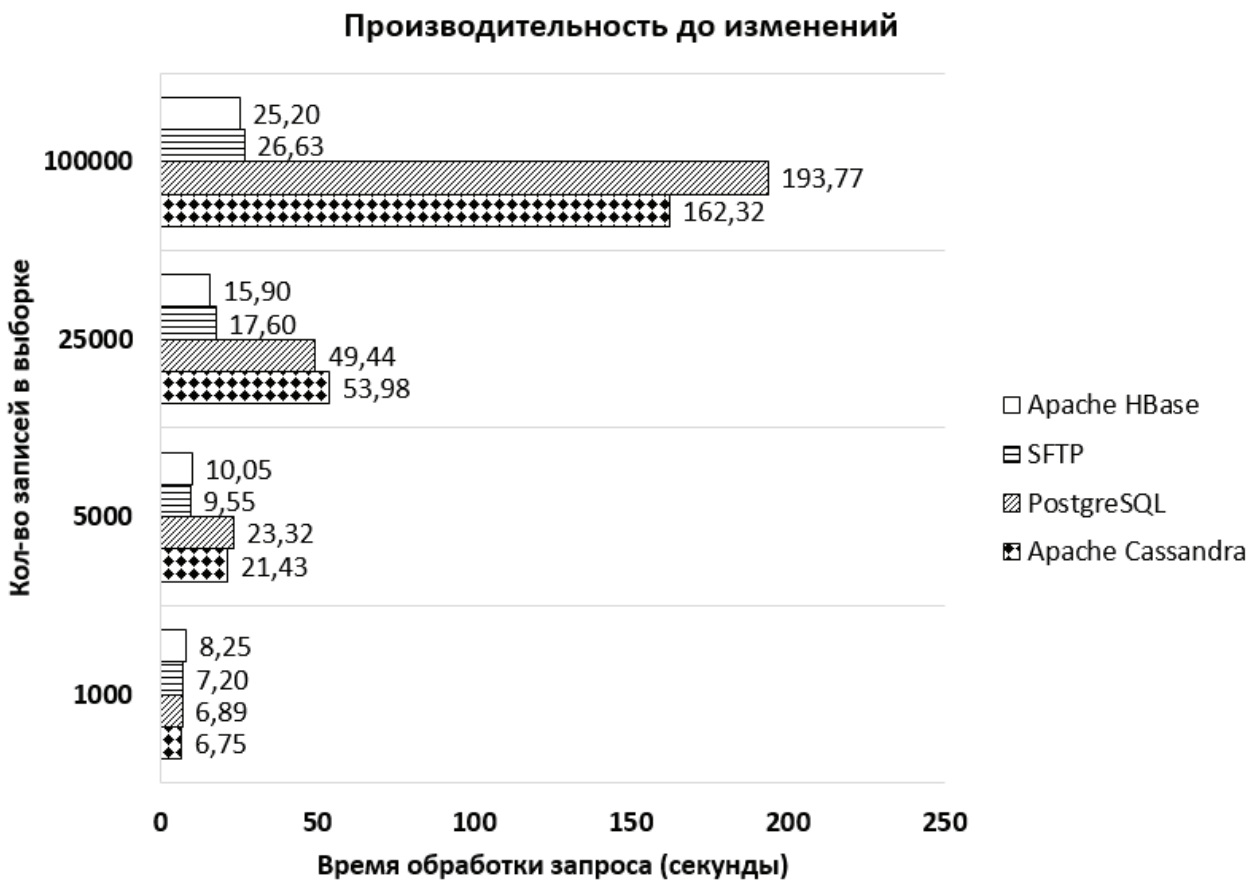

Рис. 1. Диаграммы со временем работы запросов до изменений

После проведения доработок результаты производительности запроса в Apache Cassandra и PostgreSQL стали сопоставимы с показателями Apache HBase и решений на основе SFTP (см. рис. 2).

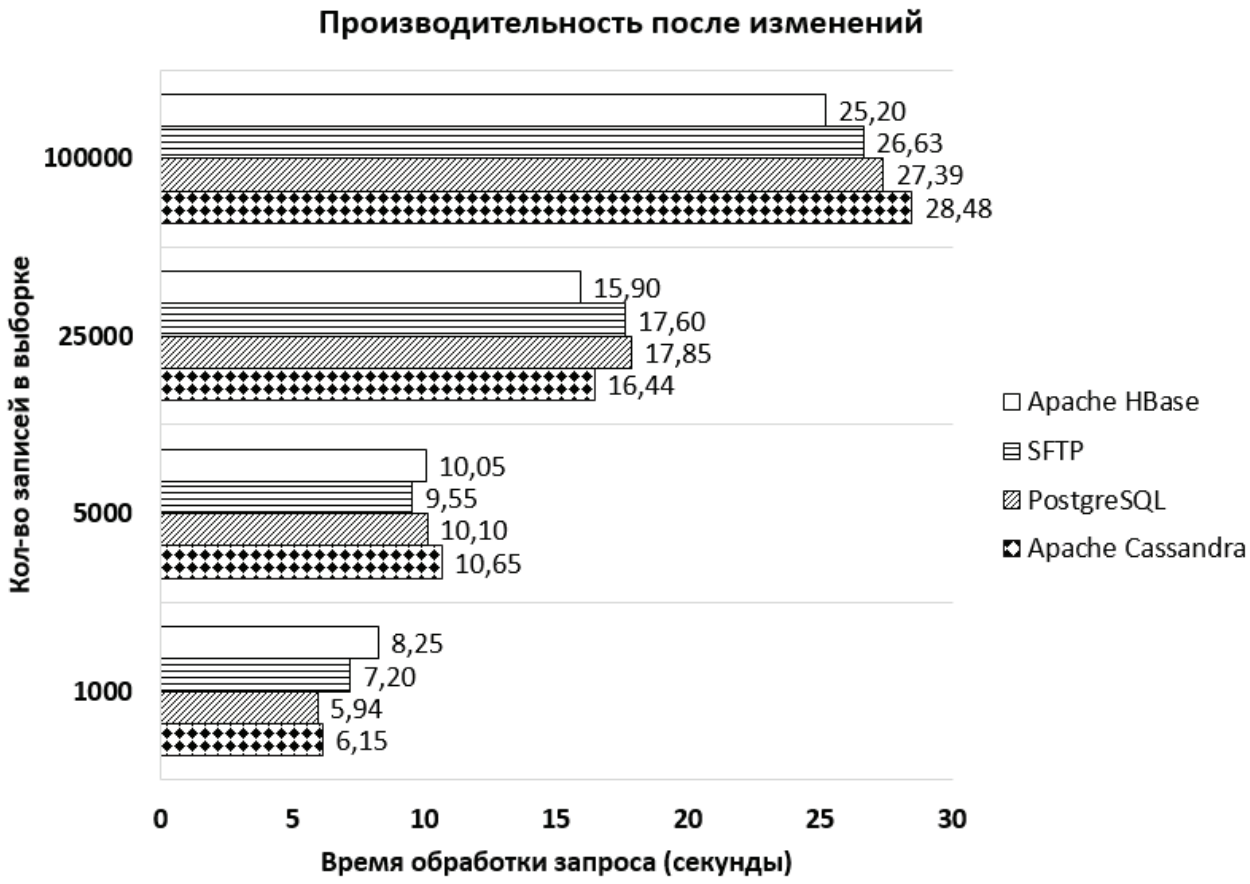

Рис. 2. Диаграммы со временем работы запросов после изменений

После сравнения времени доступа к текстовым файлам было проведено схожее тестирование, но с данными видеоряда. В СУБД и хранилищах данных (Apache Cassandra, Apache HBase, PostgreSQL) видеозаписи хранятся в формате массива двоичных данных (BLOB) [5], в решениях на основе SFTP – в виде различных видеоформатов (.mp4, .avi, .mkv и т. д.).

Для сравнения времени доступа были взяты две видеозаписи со следующими характеристиками:

- Запись №1 (видеозапись с камеры проведения ЕГЭ в аудитории): размер 150 мегабайт, разрешение видео 640x480 пикселей, битрейт 150 кбит/с, длительность видео 2 часа, формат mp4.

- Запись №2 (видеозапись лекции МФТИ по Python 3 с YouTube): размер 500 мегабайт, разрешение видео 1920x1080 пикселей, битрейт 750 кбит/с, длительность видео 1 час 15 минут, формат mp4.

Ввиду неравномерной скорости загрузки больших файлов было проведено 3 замера и взято среднее время доступа к видеозаписи. Результаты показаны в таблице 4, визуализация - на рисунке 3.

Таблица 4. Сравнительная таблица времени доступа к видеозаписи

№ видеозаписи	Время доступа к видеозаписи (в мм:сс)			
	Apache Cassandra	PostgreSQL	Apache HBase	SFTP
1	02:42	02:19	01:21	00:51
2	06:24	05:48	03:57	03:11

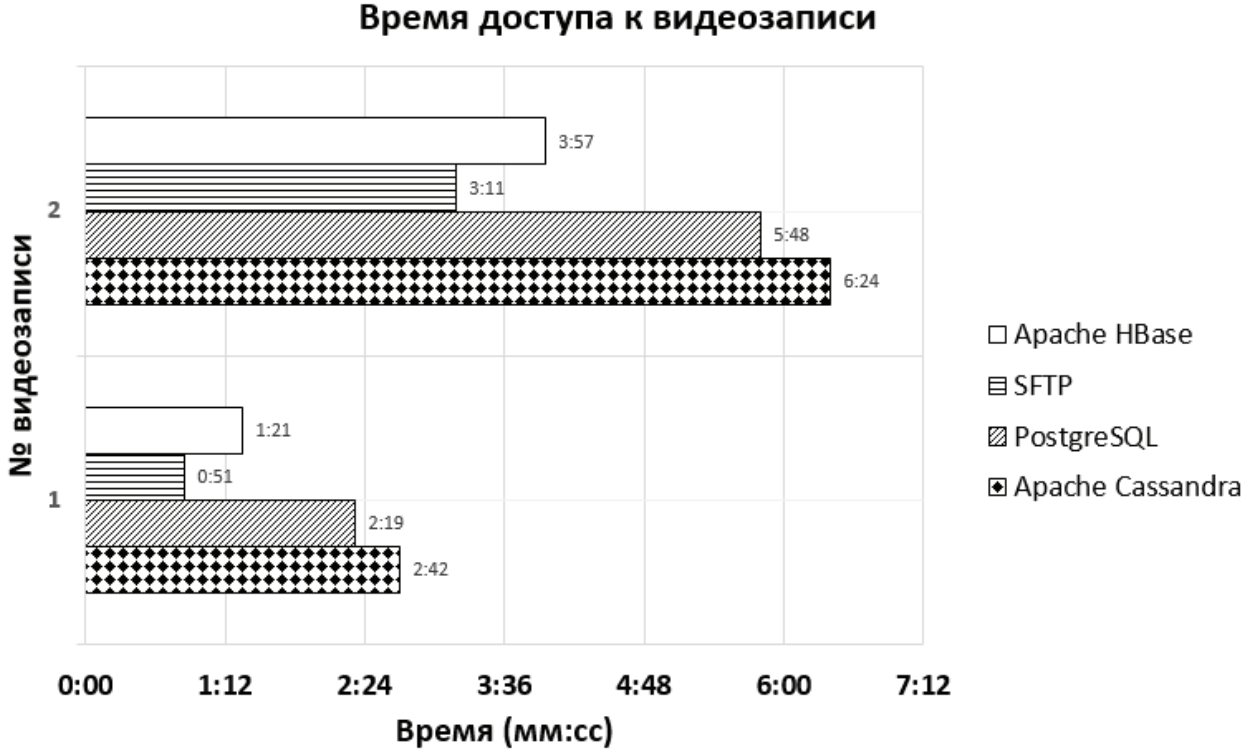

Рис. 3. Диаграмма времени доступа к видеозаписи различных решений

В данном тестировании SFTP преобладает над остальными решениями. Данное решение, используемое для хранения видеофайлов,

основано на сетевом протоколе SSH, который обеспечивает высокий уровень безопасности и более высокую производительность в отличие от решений, работающих по прикладным протоколам FTP и FTPS [6]. Apache HBase и SFTP показали сопоставимое время чтения данных.

Необходимо учитывать то, что каждое из решений проверялось на одном своем узле, в то время как Apache HBase рассчитан на параллельную работу множества узлов. В таком случае получение видеозаписи через Apache HBase может оказаться быстрее, чем по SFTP-протоколу.

Проведенное тестирование показало, что для хранения неструктурированных текстовых данных стоит использовать Apache HBase, а для больших файлов видеозаписей — решения на основе SFTP.

В результате исследования было решено организовать двухкомпонентное хранилище, консолидирующее разнородные данные. Для структурированных данных было решено использовать PostgreSQL, а для текстов студенческих работ, текстов УМК и исходных кодов программ — хранилище Apache HBase.

Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-37-51028.

СПИСОК ЛИТЕРАТУРЫ

1. Захарова И. Г. Big Data и управление образовательным процессом//Вестн. Тюмен. гос. ун-та. Серия: Гуманитарные исследования. Humanitates. 2017. № 1. С. 210-219.
2. Anti-patterns | DSE Planning guide - DataStax Docs [Электронный ресурс]. URL: https://docs.datastax.com/en/dse-

planning/doc/planning/planningAntiPatterns.html#planningAntiPatterns_
_AntiPatMultiGet (дата обращения: 10.04.2020).
3. Марк Гровер, Hadoop Application Architectures: Designing Real-World Big Data Applications, 2015. С. 27.
4. Count rows in Cassandra Table – GitHub [Электронный ресурс]. URL: https://github.com/brianmhess/cassandra-count (дата обращения: 18.04.2020).
5. Blob type | CQL for Cassandra 3.x - DataStax Docs [Электронный ресурс]. URL: https://docs.datastax.com/en/cql-oss/3.x/cql/cql_reference/blob_r.html (дата обращения: 03.05.2020)
6. The Difference Between FTP, FTPS, and SFTP [Электронный ресурс]. URL: https://medium.com/@ExaVault/the-difference-between-ftp-ftps-and-sftp-5f80a33a7838 (дата обращения: 11.05.2020)