

Санкт-Петербургский государственный университет
Математико-Механический факультет
Кафедра прикладной кибернетики

Работа допущена к защите
зав. кафедрой

«_____» _____ 2021 г.

Отчёт по научно-исследовательской работе

**Тема: Сравнительный анализ систем загрузки больших
данных**

Направление: 010400 (01.03.02) – Прикладная математика и информатика

Выполнил студент гр. 323 _____ Мамаев Владислав Викторович

Научный руководитель,

к. ф.-м. н.

_____ Благов Михаил Валерьевич

Отзыв научного руководителя

Тема работы является актуальной. При выполнении работы студент проявил самостоятельность. Работа выполнена на хорошем уровне, отчёт составлен грамотно и достаточно полно отражает результаты работы студента.

Научно-исследовательская работа выполнена полностью и может быть зачтена.

Оценка А

Благов Михаил Валерьевич

Оглавление

1.	Введение	4
2.	Основная часть	5
2.1.	Поддержка различных хранилищ	5
2.2.	Поддержка Spark	6
2.3.	Интеграция с другими инструментами	7
2.4.	Производительность	7
3.	Заключение	7
	Список литературы	11

1. Введение

Базы данных используются повсеместно, при этом системы, для которых они предназначены, можно разделить на 2 класса:

- Online Transaction Processing (OLTP) – системы обработки транзакций в реальном времени. Такие системы используются для операционной деятельности предприятий.
- Online Analytical Processing (OLAP) – системы аналитической обработки в реальном времени. К таким системам относятся системы поддержки принятия решений, инструменты business intelligence, системы анализа данных.

OLAP системы содержат информацию из OLTP систем, при этом OLAP системы обычно функционируют независимо от OLTP систем по следующим причинам:[\[1\]](#)

- Реляционная схема данных, обычно используемая в OLTP системах, не эффективна для OLAP нагрузки.
- OLAP нагрузка на OLTP систему может вызвать проблемы с производительностью транзакций.
- С увеличением размера хранимых данных, возникают ограничения на используемые технологии и существенно увеличивается стоимость хранения данных в системах не предназначенных только для OLAP.
- Необходимость доступа к данным из разных OLAP систем и возможности ограничения доступа к данным

Традиционно, данные попадают в OLAP систему из OLTP системы при помощи Extraction-Transformation-Loading (ETL) программ, запускаемых с некоторой

периодичностью. ETL процессы могут занимать достаточно много времени, и это создает задержку появления данных в OLAP системе.

Для того чтобы уменьшить задержку можно вместо периодичных ETL процессов использовать потоковую обработку. Подобная архитектура интеграции данных описана в [2]. Каждое изменение данных в OLTP системе записывается в очередь сообщений, а некоторая программа непрерывно читает сообщения и обновляет данные в аналитическом хранилище. Использование непрерывных обновлений уменьшает задержку, но требует специальных инструментов для обеспечения целостности данных — традиционно используемые хранилища для большого объема данных, такие как Hadoop, работают по принципу write-once и не поддерживают обновлений.

В данной работе будет проведено сравнение таких инструментов: Apache Hudi, Delta Lake, Apache Iceberg. Сравнение будет произведено по следующим пунктам:

- Качественные критерии: удобство доступа к данным — интеграция с другими инструментами для работы с большими данными и совместимость, возможность использования различных хранилищ.
- Производительность: пропускная способность и время задержки доставки данных

2. Основная часть

2.1. Поддержка различных хранилищ

Для хранения данных аналитических данных необходима технология для хранения файлов. Традиционно в качестве такой технологии используется HDFS. HDFS — Hadoop Distributed File System, распределённая файловая система, часть проекта Hadoop. Альтернативами для HDFS являются облачные файловые системы, такие как Amazon S3, Microsoft Azure Storage, Google Cloud Storage.

Сравниваемые инструменты поддерживают использование HDFS в качестве хранилища данных. Помимо этого Hudi поддерживает совместимые с Hadoop хранилища, включая Amazon S3, Microsoft Azure Storage, Google Cloud Storage, IBM Cloud Object Storage, Alibaba Cloud OSS, Tencent Cloud Object Storage.

Iceberg поддерживает любые хранилища для хранения данных, однако для хранения метаданных необходимо хранилище, поддерживающее атомарное переименование файлов.

Для корректной работы Delta Lake файловое хранилище должно обеспечивать: атомарную видимость файлов, атомарное создание и переименование файлов, согласованную операцию чтения списка файлов — после записи все последующие чтения списка должны возвращать записанный файл. Встроенная поддержка обеспечена для Amazon S3 (без гарантии отсутствия потерь данных в случае конкурентной записи), Microsoft Azure storage, Google Cloud Storage, IBM Cloud Object Storage, Oracle Cloud Infrastructure.

2.2. Поддержка Spark

Наиболее распространённым инструментом для ETL процессов является Apache Spark. Apache Spark - это система для обработки больших данных.^[3] Delta Lake версии 0.7.0 и выше поддерживает работу со Spark 3.x.x. Hudi поддерживает 2 и 3 версии Spark. Iceberg предоставляет дистрибутивы для работы со Spark 2 и 3. Однако, при использовании Spark 2, некоторые функции недоступны. Одна из таких функций — это SQL merge into, операция, позволяющая обновить таблицу используя новые данные, добавив строку либо обновив существующую, в зависимости от того выполнено ли условие. Ввиду отсутствия поддержки данной функции, Iceberg будет исключён из сравнения производительности.

2.3. Интеграция с другими инструментами

Hive — это система, которая упрощает чтение, запись и управление большими наборами данных, находящимися в распределенном хранилище, с помощью Hive Query Language. Может работать с данными, уже находящимися в хранилище. Для подключения пользователей к Hive предоставляются инструмент командной строки и драйвер JDBC.[4] Все инструменты поддерживают работу с Apache Hive 2.x.

Apache Flink - это инструмент распределённой обработки для вычислений с отслеживанием состояния в неограниченных и ограниченных потоках данных.[5] Hudi работает с Flink 1.12.x. Iceberg работает с Apache Flink 1.11.x. Delta Lake не поддерживает Flink

2.4. Производительность

В качестве тестовых данных использовались сгенерированные записи событий с датчиков. Такие данные были выбраны так, как часто возникают в задаче интеграции OLTP и OLAP систем для интернета вещей.[6] В качестве входных данных используется такой набор данных в котором каждая запись уже содержится в таблице.

На рисунках 1,2,3 приведены зависимости времени работы ETL программ с использованием различных технологий от размера входных данных при различных конфигурациях ресурсов кластера Spark и различном количестве записей в таблице.

3. Заключение

В рамках работы были описаны качественные различия трёх систем загрузки больших данных и проведено сравнение производительности.

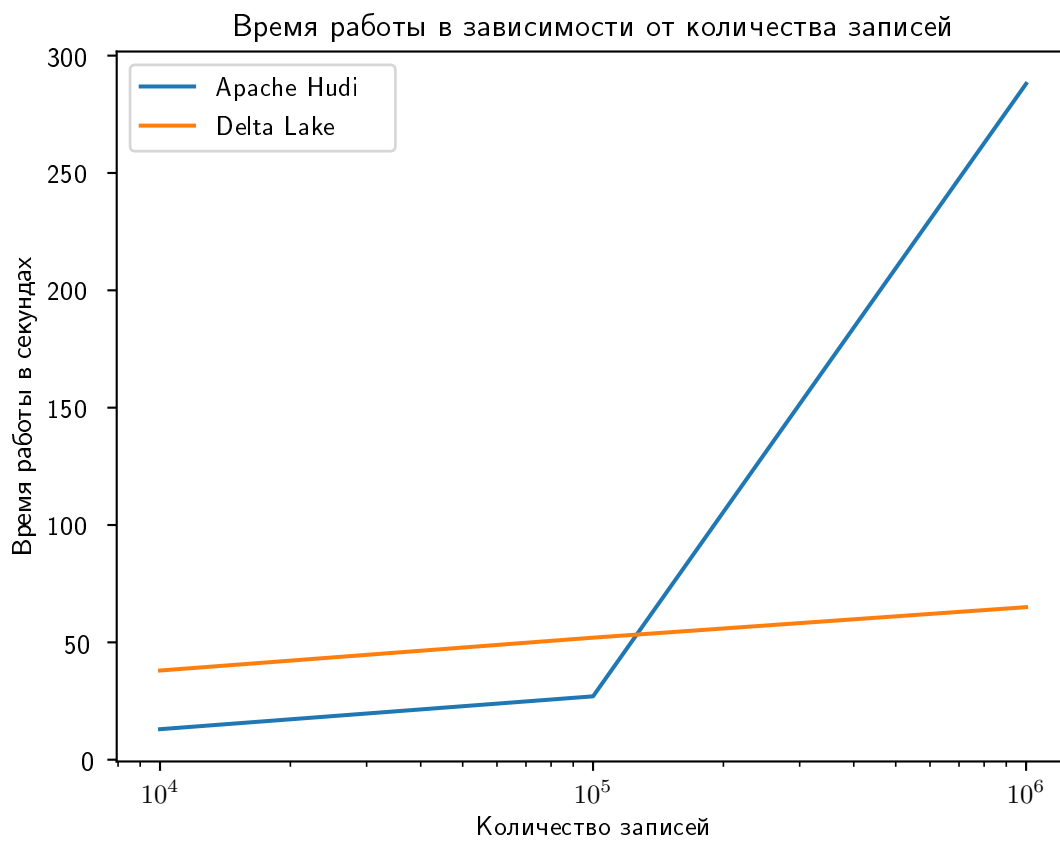


Рис. 1. Зависимость времени работы от количества записей, 1 executor, 12 ядер и 10Gb памяти, 10^6 записей в таблице

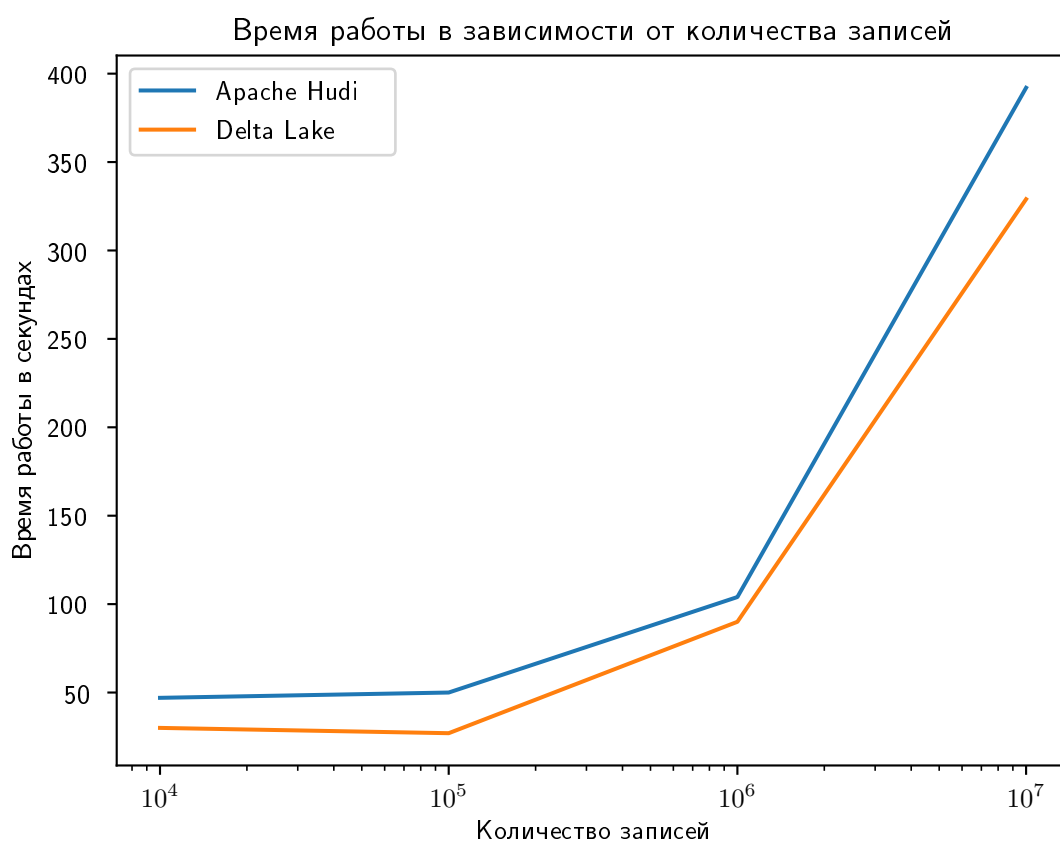


Рис. 2. Зависимость времени работы от количества записей, 2 executors, 3 ядра и 6Gb памяти каждый, 10^7 записей в таблице

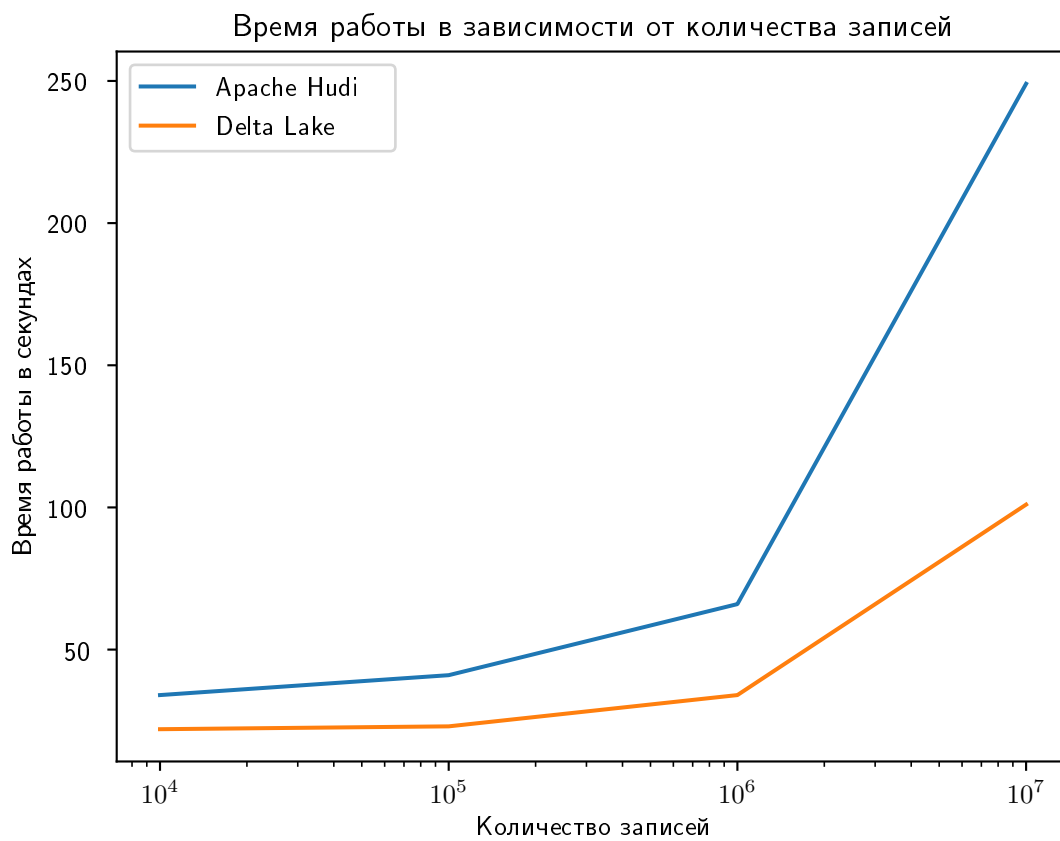


Рис. 3. Зависимость времени работы от количества записей, 4 executors, 3 ядра и 6Gb памяти каждый, 10^7 записей в таблице

Список литературы

1. *Conn S. S.* OLTP and OLAP data integration: a review of feasible implementation methods and architectures for real time data analysis // Proceedings. IEEE SoutheastCon, 2005. — 2005. — С. 515—520. — DOI: [10.1109/SECON.2005.1423297](https://doi.org/10.1109/SECON.2005.1423297).
2. *Tho M. N., Tjoa A. M.* Zero-latency data warehousing for heterogeneous data sources and continuous data streams // 5th International Conference on Information Integration and Web-based Applications Services. — 2003. — С. 55—64.
3. Официальный сайт Spark. — URL: <https://spark.apache.org/>.
4. Официальный сайт Hive. — URL: <https://hive.apache.org/>.
5. Официальный сайт Flink. — URL: <https://flink.apache.org/>.
6. Data Ingestion for the Connected World. / J. Meehan [и др.] // CIDR. — 2017.