Сравнение открытых OLAP-систем Big Data: ClickHouse, Druid и Pinot

ClickHouse, Druid и Pinot – три открытых хранилища данных, которые позволяют выполнять аналитические запросы на больших объемах данных с интерактивными задержками. Эта статья - перевод подробного сравнения, выполненного Романом Левентовым.

Источники информации

Подробности реализации **ClickHouse** стали мне известны от Алексея Зателепина, одного из **ключевых разработчиков проекта**. Доступная на английском документация достаточно скудна – наилучшим источником информации служат последние четыре секции данной страницы документации.

Я сам участвую в развитии Druid, но у меня нет личной заинтересованности в этой системе - по правде говоря, скорее всего в ближайшее время я перестану заниматься её разработкой. Поэтому читатели могут рассчитывать на отсутствие какой-либо предвзятости.

Всё, что я буду далее писать про **Pinot**, основывается на странице Архитектура в вики Pinot, а также на других страницах вики в разделе “Проектная документация”. Последний раз они обновлялись в июне 2017 года - больше, чем полгода назад.

Рецензентами оригинальной статьи стали Алексей Зателепин и Виталий Людвиченко (разработчики ClickHouse), Жан Мерлино (самый активный разработчик Druid), Кишор Гопалакришна (архитектор Pinot) и Жан-Франсуа Им (разработчик Pinot). Мы присоединяемся к благодарности автора и полагаем, что это многократно повышает авторитетность статьи.

Предупреждение: статья достаточно большая, поэтому вполне возможно вы захотите ограничиться прочтением раздела “Заключение” в конце.

Сходства между системами

Связанные данные и вычисления

На фундаментальном уровне, ClickHouse, Druid и Pinot похожи, поскольку они хранят данные и выполняют обработку запросов на одних и тех же узлах, уходя от “разъединенной” архитектуры BigQuery. Недавно я уже описывал несколько наследственных проблем со связанной архитектурой в случае Druid [1, 2]. Открытого

эквивалента для BigQuery на данный момент не существует (за исключением, разве что, Drill?) Возможным подходам к построению подобных открытых систем посвящена другая статье в моем блоге.

Отличия от Big Data SQL-систем: индексы и статическое распределение данных

Рассматриваемые в этой статье системы **выполняют запросы быстрее**, чем системы Big Data из семейства класса SQL-on-Hadoop: Hive, Impala, Presto и Spark, даже когда последние получают доступ к данным, хранящимся в колоночном формате - к примеру, Parquet или Kudu. Это происходит потому, что в ClickHouse, Druid и Pinot:

- Имеется **свой собственный формат для хранения данных с индексами**, и они тесно интегрированы с движками обработки запросов. Системы класса SQL-on-Hadoop обычно можно назвать агностиками относительно форматов данных и поэтому они менее “навязчивы” в бэкендах Big Data.
- **Данные распределены относительно “статично”** между узлами, и при распределенном выполнении запроса это можно использовать. Обратная сторона медали при этом в том, что ClickHouse, Druid и Pinot **не поддерживают запросы, которые требуют перемещения большого количества данных** между узлами - к примеру, join между двумя большими таблицами.

Отсутствие точечных обновлений и удалений

Находясь на другой стороне спектра баз данных, ClickHouse, Druid и Pinot **не поддерживают точечные обновления и удаления**, в противоположность колоночным системам вроде Kudu, InfluxDB и Vertica (?). Это даёт ClickHouse, Druid и Pinot возможность производить более эффективное колоночное сжатие и более агрессивные индексы, что означает **большую эффективность использования ресурсов** и быстрое выполнение запросов.

Разработчики ClickHouse в Yandex планируют начать поддерживать обновления и удаления в будущем, но я не уверен, будут ли это “настоящие” точечные запросы или обновления/удаления диапазонов данных.

Поглощение в стиле Big Data

Все три системы поддерживают потоковое поглощение данных из Kafka. Druid и Pinot поддерживают потоковую передачу данных стриминг в Лямбда-стиле и пакетное поглощение одних и тех же данных. ClickHouse поддерживает пакетные вставки напрямую, поэтому ему не требуется отдельная система пакетного поглощения подобная той, что используется в Druid и Pinot. Если вас интересуют подробности, то их вы сможете найти далее.

Проверено на крупном масштабе

Все три системы проверены на работоспособность в крупных масштабах: в Yandex.Metrica работает кластер ClickHouse, состоящий из примерно десятка тысяч ядер CPU. В Metamarkets используется кластер Druid аналогичного размера. Один кластер Pinot в LinkedIn включает в себя “тысячи машин”.

Незрелость

Все рассматриваемые в статье системы являются **незрелыми по меркам открытых enterprise-систем Big Data**. Однако, скорее всего они незрелы не более, чем среднестатистическая открытая система Big Data - но это совсем другая история. В ClickHouse, Druid и Pinot недостает некоторых очевидных оптимизаций и функциональности, и они кишат багами (насчет ClickHouse и Pinot я не уверен на все 100%, но не вижу причин, по которым они в этом плане были бы лучше Druid).

Это плавно подводит нас к следующему важному разделу.

Про сравнение производительности и выбор системы

Я регулярно вижу в сети, как некоторые проводят сравнения систем больших данных: они берут набор своих данных, каким-либо образом “скармливают” его оцениваемой системе, а затем немедленно пытаются измерить производительность - сколько памяти или дискового пространства было занято, и насколько быстро выполнялись запросы. Причем понимание того, как устроены изнутри испытываемые ими системы, у них отсутствует. Затем, используя лишь подобные специфичные данные о производительности - иногда вместе со списками функциональности, которая им нужна и которая есть в системе *на настоящий момент*, - они в итоге делают свой выбор или, что еще хуже, выбирают написать свою собственную, “лучшую” систему с нуля.

Такой подход мне кажется неправильным, по крайней мере он неприменим в отношении открытых OLAP-систем для Big Data. Задача создания системы Bid Data OLAP, которая смогла бы работать эффективно в большинстве сценариев использования и содержала бы все необходимые функции настолько велика, что я оцениваю ее реализацию как минимум в **100 человеко-лет**.

На сегодня, ClickHouse, Druid и Pinot оптимизированы *только* для конкретных сценариев использования, которые требуются их разработчиком - и содержат по большей части лишь те функции, в которых нуждаются сами разработчики. Я могу гарантировать, что ваш случай обязательно “упрется” в те узкие места, с которыми разработчики рассматриваемых OLAP-систем еще не сталкивались - или же в те места, что их не интересуют.

Не говоря уже о том, что упомянутый выше подход “забросить данные в систему, о которой вы ничего не знаете, и затем измерить её эффективность” весьма вероятно даст искаженный результат из-за серьезных “узких” мест, которые на самом деле могли бы быть исправлены **простым изменением конфигурации**, схемы данных или другим построением запроса.

CloudFlare: ClickHouse против Druid

Одним таким примером, хорошо иллюстрирующим описанную выше проблему, является пост Марека Вавруша о выборе между ClickHouse и Druid в Cloudflare. Им потребовалось 4 сервера ClickHouse (которые со временем превратились в 9), и по их оценкам, для разворачивания аналогичной установки Druid им бы потребовались “сотни узлов”. Пусть Марек и признает, что **сравнение является нечестным**, поскольку Druid недостаёт “сортировки по первичному ключу”, он возможно даже не осознает, что достичь примерно того же самого эффекта в Druid возможно просто установив правильный порядок измерений в “*ingestion spec*” и произведя простую подготовку данных: обрезать значение колонки `__time` в Druid до некоей грубой детализации (к примеру, один час) и опционально добавить другую “длинно-типовую” колонку “precise_time”, если для некоторых запросов требуются более точные временные рамки. Да, это хак, но, как мы только что выяснили, и в Druid можно сортировать данные по какому-либо измерению перед `__time`, и это достаточно просто реализовать.

Впрочем, я не стану спорить с их итоговым решением выбрать ClickHouse, поскольку на масштабе примерно в 10 узлов и для их нужд ClickHouse мне тоже кажется лучшим выбором, чем Druid. Но сделанное ими заключение о том, что ClickHouse как минимум на порядок эффективнее (по меркам стоимости инфраструктуры), чем Druid - это серьезное заблуждение. На самом деле, из рассматриваемых нами сегодня систем, **Druid предлагает наилучшую возможность для реально дешевых установок** (смотрите раздел “Уровни узлов обработки запросов в Druid ” ниже).

> Когда вы выбираете систему OLAP Big Data, не сравнивайте то, насколько они сейчас хорошо подходят для вашего случая. Сейчас они все субоптимальны. Вместо этого, сравните, насколько быстро ваша компания способна заставить двигаться эти системы в том направлении, которое нужно именно вам.

В силу своей фундаментальной архитектурной схожести, ClickHouse, Druid и Pinot имеют примерно один и тот же “предел” эффективности и оптимизации производительности. Здесь нет “волшебной таблетки”, которая позволила бы какой-либо из этих систем быть быстрее, чем остальные. Не позволяйте запутать себя тем фактом, что *в своем текущем состоянии* системы показывают себя очень по-разному в различных бенчмарках.

Допустим, Druid не поддерживает “сортировку по первичному ключу” настолько хорошо, насколько это умеет ClickHouse - а ClickHouse в свою очередь не поддерживает “инвертированные индексы” столь же хорошо, как Druid, что дает данным системам преимущества с той или иной нагрузкой. **Упущенные оптимизации**

могут быть реализованы в выбранной системе при помощи не таких уж и больших усилий, если у вас есть намерение и возможность решиться на подобный шаг.

- В вашей организации должны быть инженеры, способные прочитать, понять и модифицировать исходный код выбранной системы, к тому же у них должно быть на это время. Заметьте, что ClickHouse написан на C++, а Druid и Pinot — на Java.

- Или же ваша организация должна подписать контракт с компанией, которая оказывает поддержку выбранной системы. Это будут Altinity для ClickHouse, Imply и Hortonworks для Druid. Для Pinot таких компаний в данный момент нет.

Другие сведения о разработке систем, которые вам стоит принять во внимание:

- Авторы ClickHouse, работающие в Yandex, утверждают, что они тратят 50% своего времени на создание функциональности, которая требуется им внутри компании, и другие 50% уходят на функции, который набирают большинство “голосов сообщества”. Однако, чтобы вы получили от этого факта преимущество, требуется, чтобы **функции, которые нужны вам, были и наиболее востребованы сообществом** ClickHouse.
- Разработчики Druid из Imply мотивированы работать над широко используемыми функциями, поскольку это позволит им максимально увеличить объем охвата своего бизнеса в будущем.
- Процесс разработки Druid сильно напоминает модель Apache, когда ПО несколько лет разрабатывается несколькими компаниями, у каждой из который достаточно своеобразные и различные приоритеты, и среди них нет ведущей компании. ClickHouse и Pinot пока еще далеки от подобного этапа, поскольку ими занимаются соответственно лишь Yandex и Linkedin. Сторонний вклад в развитие Druid имеет минимальный шанс быть отклоненным в силу того, что он расходится с видением основного разработчика - ведь **в Druid нет “основной” компании-разработчика**.
- Druid поддерживает “API разработчика”, который позволяет привносить собственные типы колонок, механизмы агрегации, возможные варианты для «глубокого хранения» и пр., причем все это вы можете держать в кодовой базе, отдельной от самого ядра Druid. Данное API документировано разработчиками Druid, и они следят за его совместимостью с предыдущими версиями. Однако, оно недостаточно “взрослое”, и ломается практически с каждым новым релизом Druid. Насколько мне известно, в ClickHouse и Pinot схожие API не поддерживаются.
- Согласно Github, **над Pinot работает наибольшее число людей** - по всей видимости, лишь за прошлый год в Pinot было вложено не менее 10 человеко-лет. Для ClickHouse эта цифра составляет примерно 6 человеко-лет, а для Druid - 7. В теории, это должно означать, что Pinot улучшается быстрее всех остальных систем, которые мы рассматриваем.