Методы и системы обработки больших данных

10. Обработка потоков данных (Mining Data Streams)

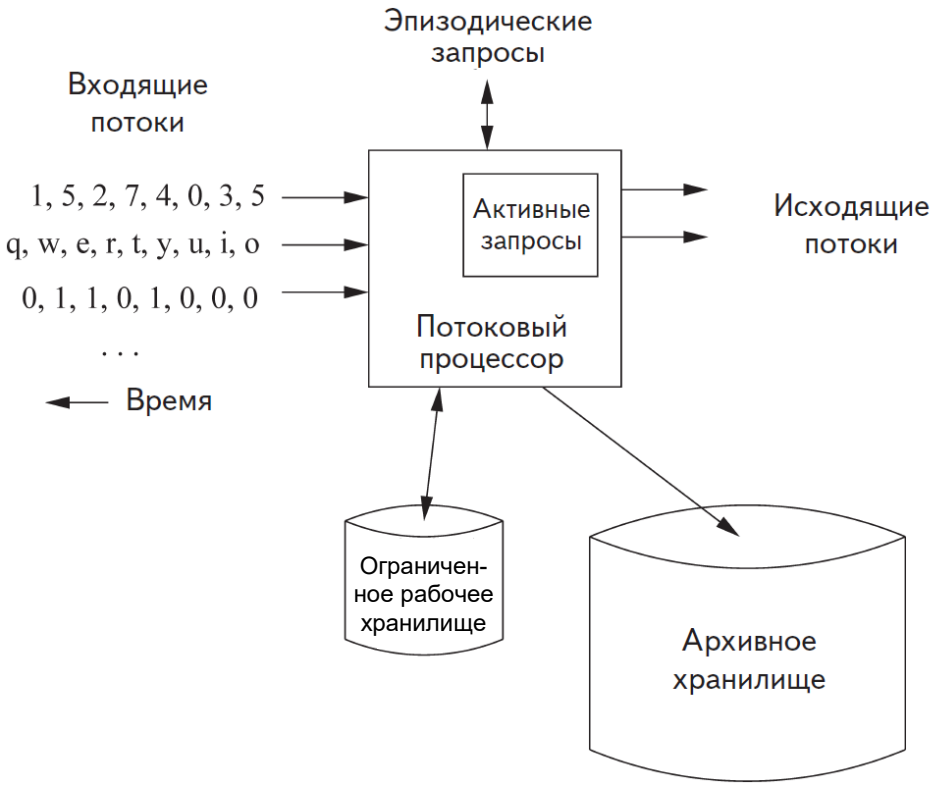
Примеры больших потоков данных

- Прогнозы погоды и предупреждение природных катастроф
  - Датчики, измеряющие состояние мирового океана ежедневно поставляют 3.6 ТБ данных
- Промышленные сенсорные сети
  - Датчики, используемые в транспортных средствах, промышленном оборудовании и сельскохозяйственной технике, отправляют данные в систему управления потоками данных, которая осуществляет мониторинг производительности, предупреждает возникновение возможных дефектов и автоматически заказывает необходимые запасные части для предотвращения простоя оборудования
- Роботрейдинг (алгоритмическая торговля ценными бумагами)
  - Финансовое учреждение отслеживает изменения на фондовых биржах в режиме реального времени, вычисляет рисковую стоимость и автоматически выполняет ребалансировку портфеля ценных бумаг на основании изменений биржевого курса
- Интернет-игры
  - Компания-разработчик интернет-игр выполняет сбор потоковых данных о взаимодействиях игроков с играми и передает эти данные на игровую платформу. Затем выполняется анализ этих данных в режиме реального времени, в результате чего формируются стимулы и динамические эффекты для повышения вовлеченности игроков
- Электронная коммерция
  - Интернет-магазин собирает все записи о потоках кликов из своих онлайн-ресурсов, агрегирует демографические данные о пользователях, оптимизирует размещение контента на своем сайте, обеспечивая релевантность и лучший опыт для клиентов
- Потоковое мультимедиа
  - Мультимедийный издатель осуществляет потоковую передачу миллиардов записей со своих онлайновых ресурсов, выполняет агрегацию и дополнение данных с учетом демографической информации о пользователях и оптимизирует размещение контента на веб-сайте, благодаря чему обеспечивается релевантность контента и повышается качество обслуживания посетителей
- Противодействие терроризму и борьба с преступностью
  - Оперативная обработка данных с уличных видеокамер
  - Мониторинг сотовых сетей
  - Мониторинг электронных браслетов
- Оптимизация автомобильного трафика в мегаполисах
  - Данные с видеокамер
  - Данные с автомобильных навигаторов и мобильных телефонов
- Здравоохранение
  - Данные с умных часов
  - Данные с мобильных диагностических приборов

Проблемы обработки больших потоков данных

- Большой объем
  - Размер базы данных слишком велик для управления с помощью обычной SQL-СУБД
- Большая скорость
  - Данные поступают слишком быстро, чтобы обычная СУБД успевала их обработать («нельзя напиться из пожарного шланга»)
- Большое разнообразие
  - Данные поступают из слишком большого количества различных источников, и существует огромная проблема их интеграции

Система управления потоками данных (СУПД)



- Несколько входных потоков данных
  - Различная скорость передачи
  - Различные типы данных

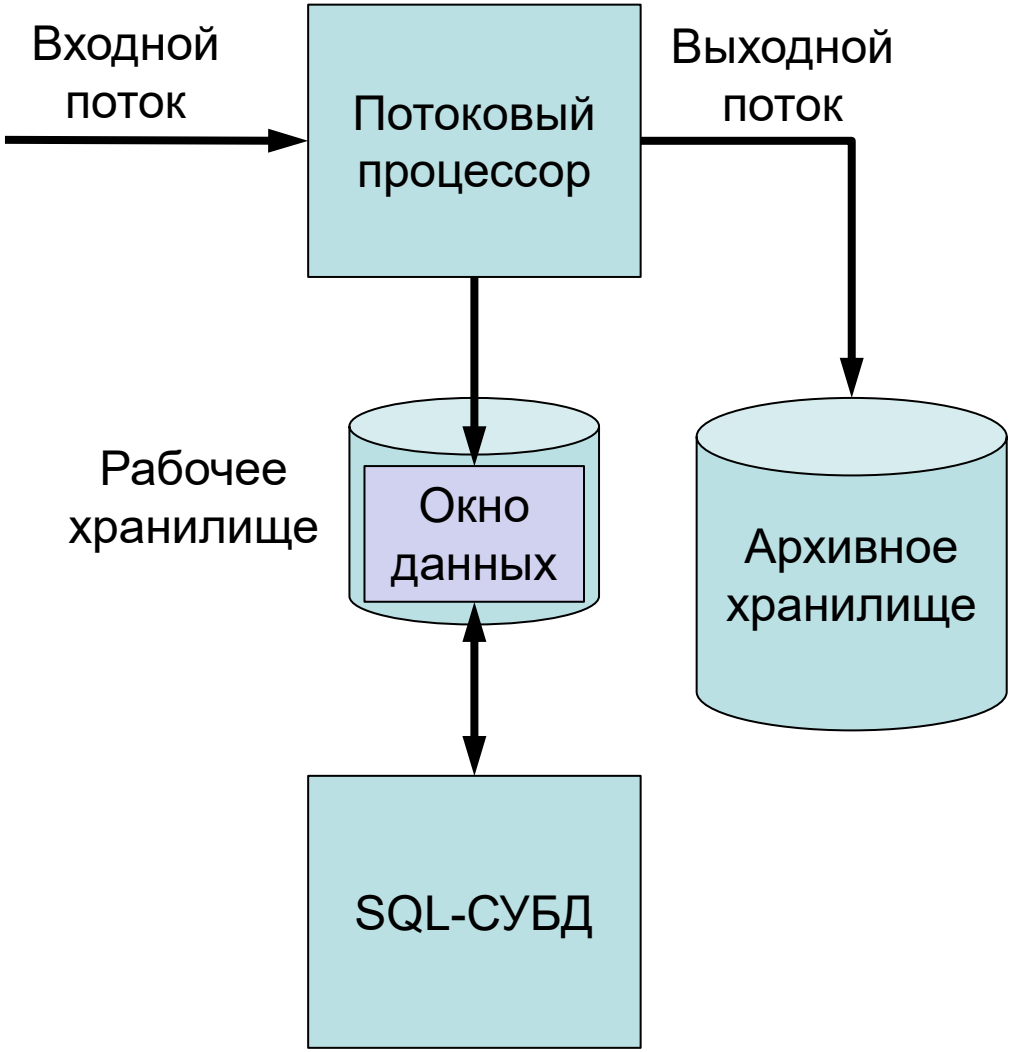
Архивное хранилище
  - Хранит сырые данные
  - Невозможна оперативная обработка в SQL-СУБД

Ограниченное рабочее хранилище
  - Агрегированные данные
  - Окна данных
  - Возможна оперативная обработка в SQL-СУБД

Активные запросы

- *Активный запрос* обрабатывает текущие данные одного или нескольких потоков
- *Примеры* активных запросов для потока данных, генерируемых датчиками на поверхности мирового океана
  - Уведомление, когда температура превышает 25 градусов
  - Среднее арифметическое последних 25 измерений
  - Максимум температуры за всю историю наблюдений

Окна данных



- Для заданного потока СУПД хранит и поддерживает в актуальном состоянии *окно данных*, содержащее фиксированное количество последних элементов потока
- Окно данных может обрабатываться SQL-СУБД

Пример запроса к окну данных

- Окно данных *Logins(IP_address,time)* содержит все обращения к сайту за последние 30 дней
- Выдать количество уникальных пользователей веб-сайта за последние 30 дней

```
SELECT COUNT(DISTINCT(IP_address))
FROM Logins
```

Эпизодические запросы

- Эпизодические запросы – запросы, семантика которых заранее неизвестна
  - Адресуются к одному или нескольким потокам
  - Поступают нерегулярно и непредсказуемо
- Необходимо поддерживать окна данных для всех потоков, к которым возможны эпизодические запросы

Проблема обработки потоков

- Имеется запрос Q пользователя к потоку R
- Нет возможности оперативно выполнить запрос применительно ко всем данным потока R над архивным хранилищем
- Решение: сформировать репрезентативную выборку из потока R для выполнения запроса Q

Методы оперативной обработки потоков данных

- Репрезентативная выборка данных из потока (скетчинг данных)
- Фильтрация потоков
- Подсчет различных элементов в потоке

Методы оперативной обработки потоков данных

Репрезентативная выборка данных из потока (скетчинг данных)

Репрезентативная выборка данных из потока: поясняющий пример

- **Вычислить долю повторных запросов пользователей к поисковой системе за последний месяц**
- Выборка содержит 1/10 элементов из потока запросов
- Пользователь сделал s одинарных запросов и d повторных запросов
- Количество одинарных запросов в выборке «1/10»: ?
- Вероятность попадания в выборку «1/10» повторных запросов:?
- Количество повторных запросов в выборке:?
- Вероятность того, что первый из пары одинаковых запросов попадет в выборку «1/10», а второй не попадет:?
- Вероятность того, что первый из пары одинаковых запросов не попадет в выборку «1/10», а второй попадет:?
- Вероятность того, что один из двух одинаковых запросов не попадет в выборку «1/10», а другой попадет:?
- Доля повторных запросов в выборке:?
- Правильный ответ: $\boldsymbol{d/(s+d)}$

Репрезентативная выборка данных из потока: решение проблемы

- Нельзя получить репрезентативную выборку из потока поисковых запросов всех пользователей
- Необходимо отобрать 1/10 пользователей и включить все их запросы в выборку
  - Сохраняем список всех пользователей с признаком «Входит/не входит в выборку»
  - При поступлении запроса от пользователя из списка с признаком «Входит в выборку», добавляем запрос в выборку
  - При поступлении запроса от нового пользователя добавляем его в список и с вероятностью 1/10:
    - устанавливаем признак «Входит в выборку»
    - Добавляем запрос в выборку
- Недостаток метода: необходимо хранить и оперативно обрабатывать список всех пользователей, что не всегда возможно

Репрезентативная выборка данных из потока: замена списка пользователей на хеш-функцию

- $f(192.168.1.1) = 19216811$
- $h(IP) = f(IP) \bmod 10$
- Пользователь с адресом IP сделал поисковый запрос:
  - Если $h(IP) = 0$, то добавляем запрос в выборку, иначе не добавляем

Методы оперативной обработки потоков данных

Фильтрация потоков

Фильтрация потоков: поясняющий пример (Google Chrome)

- Имеется список S, содержащий URL-адреса вредоносных сайтов
- Для любого URL, введенного в адресную строку браузера Chrome, необходимо очень быстро проверить, не является ли указанный сайт вредоносным
- Проверка одного URL занимает 10^{-5} сек.
- $3.3 \cdot 10^9$ человек используют Chrome
- Одновременная проверка $0.033 \cdot 10^9$ URL займет 330 сек. (5.5 мин.)

Фильтр Блума (Bloom filtering)

- Используем для фильтра битовый массив B длиной n (адресация начинается с нуля)
- Используем набор хеш-функций $h_1, \dots, h_k$:
$$h_i\colon \mathrm{URL} \to \{0, \dots, n-1\}$$
- Первоначально заполняем B нулями
- Для каждого s из S:
$$B[h_1(s)] \coloneqq 1; \dots; B[h_k(s)] \coloneqq 1$$
- Для проверяемого URL x:
$$B[h_1(x)] = 0 \vee \dots \vee B[h_k(x)] = 0$$
$$\Rightarrow \text{не вредоносный}$$

Пример построения фильтра Блума

- URL адреса имеют вид: mbXX.com (вместо X может стоять любая цифра)
- $h_1(mb\text{XX}.\text{com}) = (3 \cdot \text{XX} + 5) \bmod 6$
- $h_2(mb\text{XX}.\text{com}) = (11 \cdot \text{XX} + 4) \bmod 6$

Вредоносные URL	h_1	h_2
$mb30.com$	5	4
$mb35.com$	2	5

- Фильтр Блума:

0	1	2	3	4	5
0	0	1	0	1	1

Пример использования фильтра Блума

- $h_1(mb\text{XX}.\text{com}) = (3 \cdot \text{XX} + 5) \bmod 6$
- $h_2(mb\text{XX}.\text{com}) = (11 \cdot \text{XX} + 4) \bmod 6$

- Фильтр Блума:

0	1	2	3	4	5
0	0	1	0	1	1

URL	h_1	h_2	0	1	2	3	4	5	Прошел
$mb45.com$	2	1	0	1	1	0	0	0	1 (да)
$mb60.com$	5	4	0	0	0	0	1	1	0 (нет)
$mb55.com$	2	3	0	0	1	1	0	0	1 (да)
$mb76.com$	5	0	1	0	0	0	0	1	1 (да)
$mb19.com$	2	3	0	0	1	1	0	0	1 (да)

Ответ для теста: 10111

Методы оперативной обработки потоков данных

Подсчет количества различных элементов в потоке

Проблема Count-Distinct: поясняющий пример

- Для каждой страницы сайта Ozon необходимо подсчитать количество уникальных пользователей, заходивших на эту страницу за весь период существования сайта до настоящего времени (идентифицировать пользователя необходимо по IP-адресу)
- Ozon торгует примерно 10 млн товарных наименований
- Сайт посетили 4 миллиона уникальных пользователей
- IP-адрес требует для хранения 4 байта
- Для хранения IP-адресов уникальных пользователей для всех страниц требуется 10000000*4000000*4= 160000000000000 байт=160 Тбайт
- При каждом посещении каждой страницы требуется быстро проверить IP-адрес на уникальность и запомнить его, если это новый посетитель
- В пиковые нагрузки количество одновременно заходящих на сайт пользователей может доходить до миллиона

Алгоритм Флажоле-Мартена

- Используем битовый массив $\mathbb{B}$ длиной 64 (свой для каждой страницы сайта)
- Используем множество хеш-функций $h_1, \dots, h_k$:
$$h_i(x) = a_i x + b_i \ \mathbf{mod}\ 64$$
- $R := 0$ – максимальное количество нулей в хвосте $\mathbb{B}$
- Для каждого посещения с IP-адреса x:

  $\mathbb{B}[0 \dots 63] := \mathbf{0}$

  for $i = 1 \dots k$

  $\quad \mathbb{B}[h_i(x)] := 1$

  end for

  $r :=$ количество нулей в хвосте $\mathbb{B}$

  $R := \max(R, r)$
- Данную страницу посетило примерно 2^R уникальных пользователей
- На практике можно использовать миллион хеш-функций
- Обоснование оценки и совершенствование метода: Лесковец Ю., Раджараман А., Ульман Д.Д. Анализ больших наборов данных. Москва: ДМК Пресс, 2016. 498 с.

Пример применения алгоритма Флажоле-Мартена

- Поток состоит из чисел 2, 4, 1, 4, 1, 1, 4, 2, 7
- $h(x) = (11x + 7) \bmod 32$
- Результат хеш-функции следует рассматривать как 5-разрядное двоичное число

x	$h(x)$	$\lfloor h/16 \rfloor$	$\lfloor (h \bmod 16)/8 \rfloor$	$\lfloor (h \bmod 8)/4 \rfloor$	$\lfloor (h \bmod 4)/2 \rfloor$	$h \bmod 2$	**0-хвост**
2	29	1	1	1	0	1	0
4	19	1	0	0	1	1	0
1	18	1	0	0	1	0	1
4							
1							
1							
4							
2							
7	20	1	0	1	0	0	2
						max	**2**

$2^2 = 4$

- В последовательности …… различных элемента

Конец лекции 10