Анализ логов web-сервера Apache 2.4

Выполнил: Классен Роман Константинович

Задача

Разработать скрипт формирования витрины на основе логов web-сайта следующего содержания:

1. Суррогатный ключ устройства
2. Название устройства
3. Количество пользователей
4. Доля пользователей данного устройства от общего числа пользователей.
5. Количество совершенных действий для данного устройства
6. Доля совершенных действий с данного устройства, относительно других устройств
7. Список из 5 самых популярных браузеров, используемых на данном устройстве различными пользователями, с указанием доли использования для данного браузера относительно остальных браузеров.
8. Количество ответов сервера отличных от 200 на данном устройстве
9. Для каждого из ответов сервера, отличных от 200, сформировать поле, в котором будет содержаться количество ответов данного типа

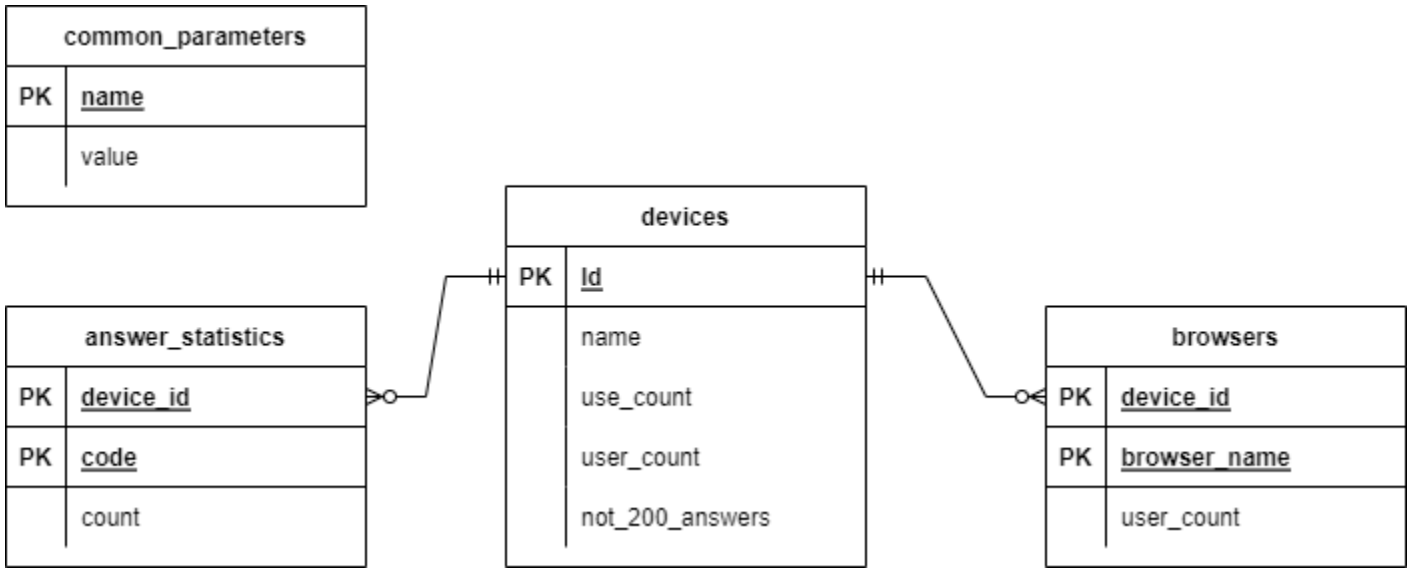
План реализации

1. Ознакомится с файлом исходных данных для анализа
   - Выбрать несколько случайных строк из разных мест файла
   - Определить формат записи
2. Разработать модуль разбора строки из файла
3. Разработать модуль анализа данных
   - Проверка и очистка данных
   - Подсчет агрегатов
   - Формирование промежуточного файла для загрузки в СУБД
4. Разработать схему хранения данных в СУБД
5. Разработать загрузчик данных в СУБД

Используемые технологии

- Python
  - Распространяется по умолчанию в множестве Linux дистрибутивов.
  - Интерпретируемый язык. Не требует дополнительных средств при изменении алгоритма работы.
  - Имеет обширный набор подключаемых модулей.
    - user_agents – для разбора записей User Agent
    - psycopg2-binary – для подключения к СУБД PostgreSQL
  - Имеет все необходимые встроенные функции для обработки текста и анализа
    - Работа с регулярными выражениям.
    - Работа с файлами JSON.
    - Работа с текстовыми файлами построчно.
- SQL
  - Индустриальный стандарт для доступа к реляционным данным.
  - Любят аналитики.
  - Используем PostgreSQL, т.к. есть возможность использования в РФ как Postgres Pro

Схема данных витрины



- common_parameters – дополнительные, несвязанные параметры, например, количество строк в файле, количество уникальных пользователей
- devices – агрегированная информация о устройстве
- browsers – информация о браузерах с разбивкой по устройствам
- answer_statistics - информация о ответах сервера с разбивкой по устройствам

Результаты разработки

- Разработан скрипт разбора файла лога и агрегирования необходимой информации
- Разработан скрипт загрузки агрегированных данных в СУБД
- Итоговый вид витрины:

id	name	use_count	use_share	user_count	user_share	not_200_answers	browsers	answers
1	Spider	1109418	0	3522	0	393645	{"BingPreview 1.0","AhrefsBot 6.1","bingbot 2.0","Baiduspider 2.0","Googlebot 2.1"}	{"304: 283460","302: 52357","301: 41733","404: 13940","499: 987","403: 810","502: 250","500: 45","504: 37","400: 26"}
2	ALE-L21	18627	0	1670	0	448	{"Android 6.0","Chrome Mobile 50.0.2661","Android 5.0.1","Chrome Mobile 46.0.2490","Chrome Mobile 66.0.3359"}	{"302: 181","499: 106","304: 101","403: 43","301: 17"}
3	Windows 8	119296	0	1032	0	1587	{"Firefox 16.0","Chrome 71.0.3578","Firefox 64.0","IE 10.0","Opera 57.0.3098"}	{"301: 556","302: 538","499: 213","304: 162","404: 70","403: 18","502: 16","500: 14"}

Заключение

- Поставленная задача выпалена в полном объеме.
- Получение данных для витрины возможно из файла JSON или СУБД.
- Ускорение работы скрипта возможно с помощью использования параллельной обработки на одном вычислительном узле.
  - Дополнительное ускорение работы можно получить уменьшив размерность разбираемых данных (например, не разбирать, а пропускать значение в логе, если оно не участвует в дальнейшей обработке).
- Исходный код: https://github.com/rozh1/DE_Sprint/tree/main/final