Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Informatyki

Studia Podyplomowe
Big Data - przetwarzanie i analiza dużych zbiorów danych

PRACA KOŃCOWA

Kamil Gontarz

Zaprojektowanie i wykonanie systemu do składowania, przetwarzania oraz analizy danych o aktywności użytkowników strony internetowej.

Opiekun pracy
mgr Patryk Pilarski

Warszawa, 2022

Spis treści

1. Kontekst biznesowy

Klientem jest informacyjny portal internetowy, którego głównym źródłem dochodu są reklamy pokazywane użytkownikom. Do tej pory główny reklamodawca rozliczał się z portalem na zasadzie ryczałtu, ustalonej kwoty przelewanej na konto klienta w miesięcznych interwałach. W ostatnim czasie odstąpił od takiego modelu finansowania swoich kampanii reklamowych na rzecz ustalonej kwoty za każde 1000 odsłon stron zawierającyh materiały reklamowe. Przy takim sposobie rozliczania klient stracił znaczną część swojego głównego źródła dochodu. **Głównym zadaniem jest zdefiniowanie co wpływa na poziom ruchu na stronie oraz jakie zadania należy wykonać aby zwiększyć liczbę odsłon**. Jak się okazało klient nie dysponuje żadnym dedykowanym do tego typu zadań działem analitycznym ani działem wyspecjalizowanym działem technicznym. Techniczne kwestie związane z serwowaniem treści zostały oddelegowane do zewnętrznego podmiotu, który nie oferuje usług w zakresie doradztwa analitycznego.

Po konsultacjach z klientem oprócz głównego celu zdefiniowano również cel dodatkowy jakim jest przygotowanie testowego środowiska dla przyszłego działu analitycznego którego zadaniem będzie utrzymanie/zwiększenie efektywności dochodowej protalu. Środowisko te ma pozwalać na:

- przechowywanie surowych danych (obecnie dane ze skryptów na stronie nie są przechowywane a jedynie agregowane ilościowo w kubełkach godzinnych co nie pozwala na inną niż ilościową analizę historycznych danych)
- przetwarzanie danych w celu tworzenia cyklicznych, automatycznie powstających raportów
- prezentację danych w formie tabel, wykresów oraz tablic agregujących różne wyniki (dashboard)
- wykonywanie na surowych danych doraźnych, niestandardowych danych

Z założenia środowisko ma być testowe(aby w pierwszym okresie nie poświecać czasu na uprodukcyjnianie rozwiązania badawczego) ale użyte technologie powinny być:

- łatwo skalowalne - klient zakłada w przyszłości wzrost generowanego ruchu
- powszechnie używane w środowisku - co zapewnia dostęp do specjalistow znających daną technologię oraz potwierdza jej przydatność i możliwość zastosowania w realnych przypadkach
- w miare możliwości open source z rozwiniętą społecznością wokół technologii - podejście to nie generuje kosztów licencyjnych a jednocześnie pozwala na znalezienie rozwiązań wielu problemów w ogólnodostępnych tematycznych forach internetowych
- modułowe - system podzielony na kilka mniejszych modułów jest łatwiejszy w zrozumieniu, utrzymaniu, rozwijaniu czy znajdowaniu błędów
- integrowalne z wieloma rozwiazaniami - brak potrzeby manualnej integracji przy każdorazowej zmianie/dodaniu rozwiązania znacząco przyspiesze rozwój oraz zmniejsza ilość potencjalncyh błędów

- zastępowalne - dzięki identycznym lub podobnym interfejsom w następnych fazach projektu będzie można zastąpić wybrane rozwiązanie innym o bardziej porządanych parametrach czy cechach bez przebudowywania dużej części systemu.

Klient zdaje sobie sprawę, że w pierwszym kroku analizy mogą nie przynieść oczekiwanych efektów jednocześnie jest świadomy tego, że aby przedsiębiorstwo mogło pracować w oparciu o dane należy zaszczepić w nim kulturę pracy z tymi danymi, a do tego niezbędne jest odpowiednie środowisko techniczne.

Dodatkowe wymaganie pochodzi z konsultacji klienta z wewnętrznym działem prawnym. Adres IP jest jedną z informacji przysyłanych przez skrypty umieszczone na stronie portalu. Adres ten w pewnych przypadkach może być uznawany za daną osobową.[1] Dane osobowe muszą być specjalnie przechowywane co zwiększa potencjalne koszty oraz komplikuje system. W związku z tym podjęto decyzję o anonimizowaniu adresu IP w przechowywanych permanentnie danych.

[1] https://archiwum.giodo.gov.pl/pl/319/2258

2. Dostępne dane

Jak było wspomniane w poprzednim rozdziale surowe dane ze skryptów monitorujących ruch na portalu nie są nigdzie przechowywane. Jednak są one doskonałą bazą wykonania niezbędnych w realizacji celu analiz. Z danych agregowanych godzinowych (przekazanych przez obecną firmę obsługującą portal od strony technicznej serwowania treści) na temat aktywności użytkowników portalu można wywnioskować, że maksymalne zaobserwowane obciążenie godzinowe wynosi około 1 000 000 odsłon co daje średnią około 300 odsłon/sekundę. Na razie nie da się stwierdzić na ile maksymalne wartości sekundowe będą większe od wyliczonych średnich.

2.1 Struktura

Poniżej przedstawiono listę dostępnych danych ze skryptów wraz z ich typem oraz przykładowymi wartościami:

- time: timestamp(UTC) - 1654002960
- cookieID: String - 426657233775674784687905101944228963337
- userAgent: String - “Mozilla/5.0 (X11; Linux i686; rv:1.9.7.20) Gecko/2430-08-14 01:01:49 Firefox/13.0”
- url: String - “example.com/category/main/tags?testVersion=a”
- referer: String - “google.com”
- ip: String - “60.21.16.140”

Poza kolumną time która jest typu timestamp reszta to kolumny tekstowe częściowo ustrukturyzowane

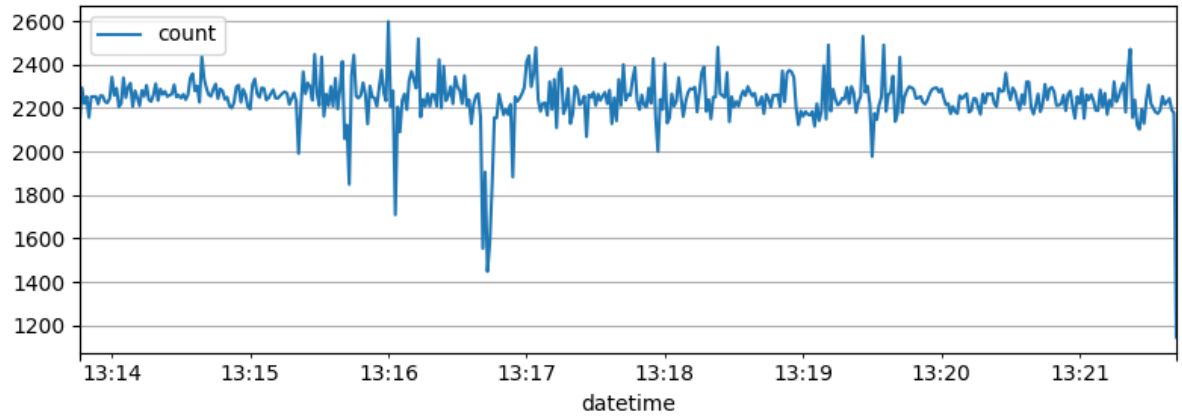
2.3 Statystyki

Podsumowanie na podstawie około jednego miliona testowych rekordów.

2.3.1 Zdarzenia w czasie.

Opis wykresu:

- oś y - liczba zdarzeń
- oś x - kolejne sekundy

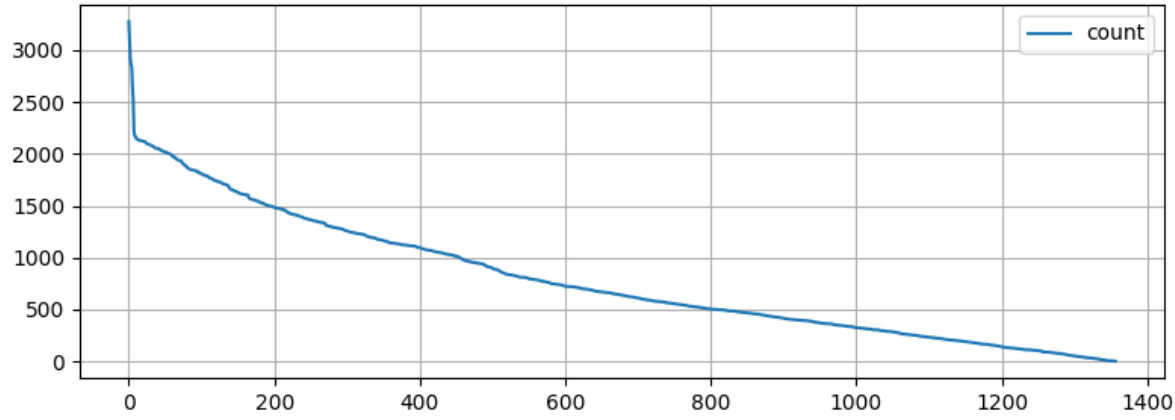


Rysunek 1: Liczba zdarzeń w czasie.

2.3.2 Zdarzenia per Cookie.

Opis wykresu:

- oś y - liczba zdarzeń
- oś x - kolejne cookiesy, posortowane od najbardziej aktywnego

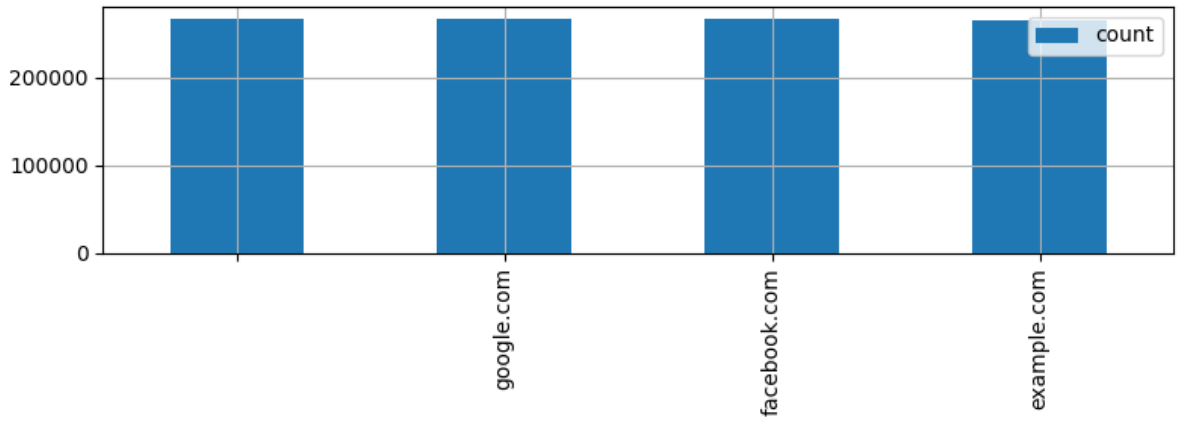


Rysunek 2: Rozkład liczby zdarzeń per cookie.

2.3.3 Zdarzenia per referer.

Opis wykresu:

- oś y - liczba zdarzeń
- oś x - kolejne odsyłacze

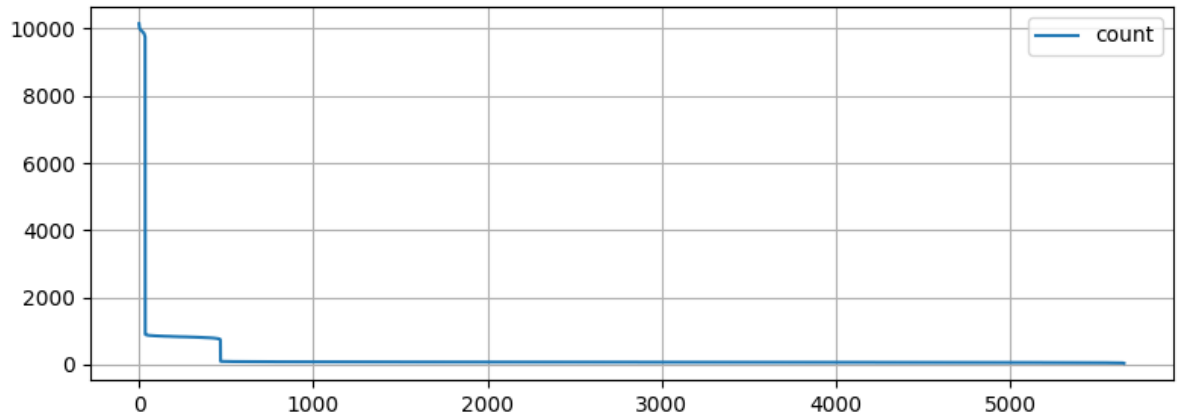


Rysunek 3: Liczba zdarzeń per referer

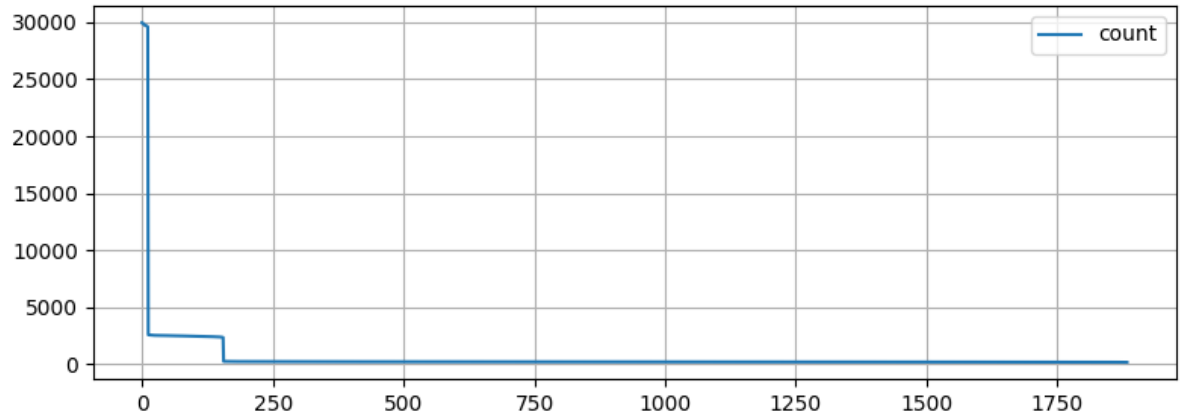
2.3.4 Zdarzenia per url.

Opis wykresów:

- oś y - liczba zdarzeń
- oś x - kolejne urle, posortowane od tego z największą liczbą zdarzeń



Rysunek 4: Liczba zdarzeń per url.

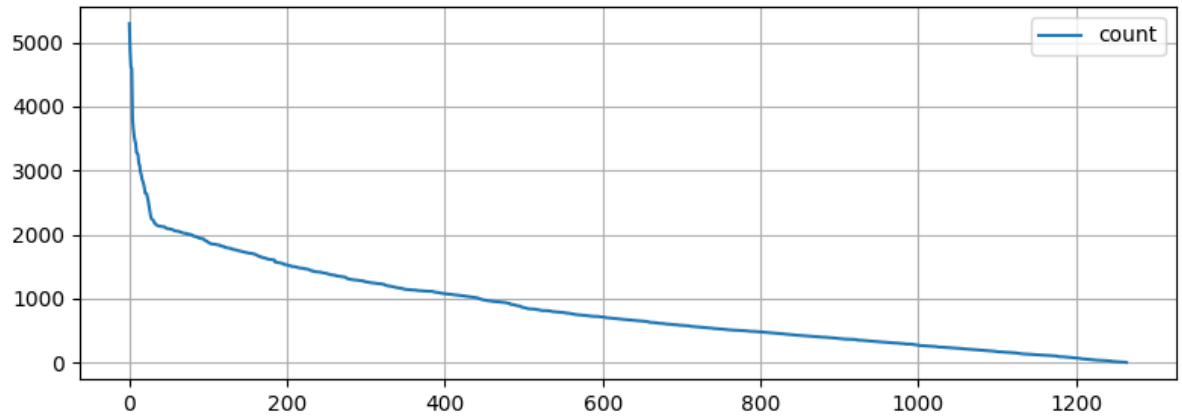


Rysunek 5: Liczba zdarzeń per url bez dodatkowych parametrów.

2.3.5 Zdarzenia per user agent.

Opis wykresu:

- oś y - liczba zdarzeń
- oś x - kolejne user agenty posortowane od najbardziej popularnego w ramach zdarzeń.



Rysunek 6: Liczba zdarzeń per user agent.

2.3.5 Zdarzenia per adres IP.

Odnotowano 1 068 649 unikalnych adresów IP na 1 068 807 zdarzeń. Co oznacza, że w zasadzie przy każdym zdarzeniu wygenerowany został nowy adres IP. To punkt do dalszej analizy, ponieważ jest to wysoce nie prawdopodobne, że IP jest zmieniane co każde zdarzenie

2.3.6 Podsumowanie

Z powyższych testowych statystyk widać, że:

- liczba zdarzeń jest względnie stała w czasie
- cookiesy posiadają różną liczbę odsłon choć o podobnym rzędzie wielkości
- referery podzielone są na 4 kategorie (brak , facebook.com, google.com , wewnetrzny) które są w zasadzie równoliczne.
- liczba urli wynosi niecałe 6 000. Jednak nie usuwano z nich parametrów. Po usunięciu dodatkowych parametrów zostało niecałe 2000 unikalnych urli
- liczba user agentów jest trochę mniejsza liczba cookiesów co oznacza, że tylko niewielka część się powtarza
- liczba adresów IP do zbadania

Do analiz należy dane posegmentować na kilka zbiorów. Przy obecnym rozkładzie statystyk zdarzeń jedynym kandydatem do segmentowania są referery choć są one równoliczne (co może być przypadkiem testowych danych) i mogą być zbyt mało charakterystyczne. Wytypowano pole useragent jako to z którego można wydobyć dodatkowe bardziej ogólne informacje takie jak: typ urządzenia (PC, mobile itp), system operacyjny, rodzaj przeglądarki.

2.4 Generator

3. Rozwiązanie koncepcyjne problemu biznsowego

- zaproponować co można zmienić aby zrealizować cel główny
- zwiększenie ilości i zaangażowania użytkowników strony
- odsłony
- sesje
- czas
- podział na segmenty technicze (os, deviceType, Browser)
- próba znalezienia odpowiedzi na pytanie jak zwiększyć liczbę użytkowników oraz ich zaangażowanie

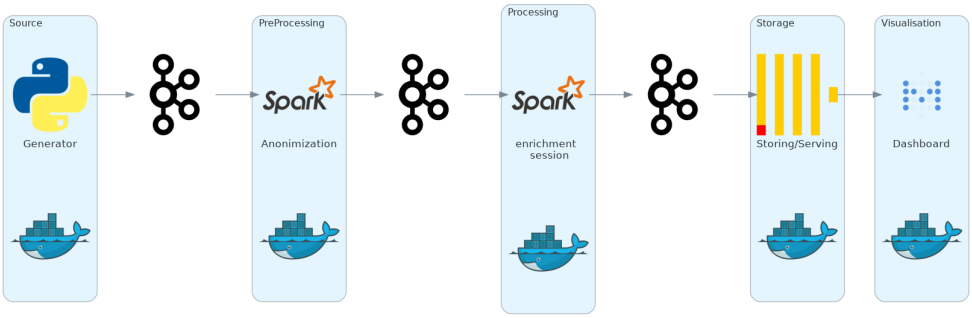
4. Model danych

- tabela eventów
- tabela sesji

5. Opis modułow

5.1 Ogólny schemat rozwiązania

opis



Rysunek 7: Przepływ danych między modułami.

dalszy opis

5.2 Preprocessing

5.3 Processing

5.4 Storage

5.5 Wizualizacja

6. Napotkane problemy oraz ich rozwiązanie

7. Możliwe drogi rozwoju systemu

8. Wnioski

9. Bibliografia

- https://altinity.com/blog/2020/5/21/clickhouse-kafka-engine-tutorial – virtual columns
- https://hub.docker.com/r/clickhouse/clickhouse-server/ - clickhouse docker
- https://github.com/enqueue/metabase-clickhouse-driver – metabase clickhouse community driver
- https://clickhouse.com/docs/en/integrations/kafka/kafka-table-engine/
- https://www.metabase.com/docs/latest/operations-guide/running-metabase-on-docker.html
- https://www.metabase.com/docs/latest/administration-guide/01-managing-databases.html#database-sync-and-analysis
- https://vincent.doba.fr/posts/20211004_spark_data_description_language_for_defining_spark_schema/
- https://towardsdatascience.com/spark-3-2-session-windowing-feature-for-streaming-data-e404d92e267
- https://pypi.org/project/user-agents/
- https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/96810098
- https://stackoverflow.com/questions/70991571/stream-data-from-one-kafka-topic-to-another-using-pyspark
- https://stackoverflow.com/questions/2013124/regex-matching-up-to-the-first-occurrence-of-a-character
- https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#window-operations-on-event-time
- https://github.com/ykursadkaya/pyspark-Docker/blob/master/Dockerfile
- https://stackoverflow.com/questions/37132559/add-jar-files-to-a-spark-job-spark-submit
- https://stackoverflow.com/questions/50217212/how-do-i-write-to-kafka-using-pyspark
- https://faker.readthedocs.io/en/master/
- https://github.com/joke2k/faker
- https://docs.docker.com/develop/develop-images/dockerfile_best-practices/ - multiline env
- https://support.google.com/analytics/answer/2731565?hl=en#overview&zippy=%2Cin-this-article – definition of visits