JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 30 (2024) 1-8
Received: 10 July 2023
Accepted: 3 November 2023

Analysis of data processing efficiency with use of Apache Hive and Apache Pig in Hadoop environment

Analiza efektywności przetwarzania danych z użyciem Apache Hive i Apache Pig w środowisku Hadoop

Mikołaj Skrzypczyński*, Piotr Muryjas

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
The aim of this paper is the analysis of data processing efficiency with use of Apache Hive and Apache Pig in Hadoop environment. The analysis was based on comparison between both mentioned tools with use of large data set, represented by 28 million records. Research was provided with use of scripts and queries destined for Apache Hive and Apache Pig, and then executed 10 times on environment brought by created virtual machine. Those methods were performed on the same data sets for 16 times according to previously prepared research scenarios. As the conclusion, authors had observed that Apache Hive is more efficient tool, than Apache Pig.

Keywords: data processing; Apache Hive; Apache Pig, Hadoop

Streszczenie
Celem niniejszej pracy jest analiza efektywności przetwarzania danych z użyciem Apache Hive i Apache Pig w środowisku Hadoop. Analiza polegała na porównaniu pomiędzy obydwoma wspomnianymi narzędziami z użyciem dużych zbiorów danych, w formie 28 milionów rekordów. Badanie zostało przeprowadzone z użyciem skryptów i zapytań przeznaczonych dla Apache Hive oraz Apache Pig, a następnie wykonanie dziesięciokrotnie na środowisku dostarczonym dzięki utworzonej maszynie wirtualnej. Wymienione metody zostały uskutecznione na tych samych zbiorach danych 16 razy, zgodnie z uprzednio przygotowanymi scenariuszami badawczymi. W rezultacie autorzy zaobserwowali, iż Apache Hive jest bardziej efektywnym narzędziem, niż Apache Pig.

Słowa kluczowe: przetwarzanie danych; Apache Hive; Apache Pig; Hadoop

*Corresponding author
Email address: mikolaj.skrzypczynski@pollub.edu.pl (M. Skrzypczyński)

1. Wstęp

Ilość danych na świecie wzrasta z dnia na dzień, co wiąże się z coraz większym skomplikowaniem procesów związanych z ich przetwarzaniem. Istotnym aspektem w dostępie do danych jest efektywność korzystania z nich, w celu uzyskania zakładanych efektów przy określonej złożoności obliczeniowej, a co za tym idzie również czasowej. Odpowiedzią na te zagadnienia jest rozwiązanie Big Data, które to nastawione jest na wydajną pracę z dużymi zbiorami danych.

Celem niniejszej pracy jest analiza efektywności przetwarzania danych z użyciem Apache Hive i Apache Pig w środowisku Hadoop. Apache Hadoop jest środowiskiem mającym na względzie umożliwienie pracy na klastrach komputerów, w celu procesowania wielkich zestawów danych z wykorzystaniem złożonych narzędzi takich jak Apache Hive czy też Apache Pig.

Bazując na rosnącej popularności wyżej wymienionych rozwiązań, autorzy niniejszej pracy postanowili zbadać ich efektywność. Jest to kluczowy czynnik wpływający na wydajność korzystania z tych narzędzi. W efekcie badany aspekt jest istotną składową, w kwestii rozważania, której technologii użyć poza jej składnią zapytań czy też łatwością konfiguracji.

Do przeprowadzenia badań: dokonano przeglądu literatury, skonfigurowano środowisko, przygotowano scenariusze badawcze, a następnie ich implementację z wykorzystaniem skryptów Apache Pig i zapytań Apache Hive, podejmując zarówno aspekt teoretyczny jak i praktyczny zagadnienia.

2. Przegląd literatury

Niniejszy rozdział zawiera analizę literatury obejmującej tematykę narzędzi Big Data: Apache Hadoop, Apache Pig i Apache Hive.

W pracy pt. „*Analyzing Performance of Apache Pig and Apache Hive with Hadoop*” autorzy porównali narzędzia Apache Hive i Apache Pig pod względem wydajności i złożoności zapytań. Jako dane wejściowe dla badań wybrano duży zbiór danych zawierający informacje z ośrodków lekarskich w Stanach Zjednoczonych dotyczących złych praktyk wykonywanych przez lekarzy. Na ich podstawie autorzy próbowali dowiedzieć się, w jakich okolicznościach powinno się używać narzędzia Apache Pig, a w jakich Apache Hive. Narzędzia zostały porównane w wymiarach sposobu przechowywania danych, ekstrakcji i przekształcania danych oraz opłacalności użycia. Autorzy wykazali, że pod względem opłacalności użycia oba narzędzia są na tym samym poziomie. Natomiast w kwestii przechowywania danych Hive jest lepszym wyborem, a w przypadku ekstrakcji i przekształcania danych

lepszy jest Pig. Narzędzia zostały porównane także pod względami czasów wykonania oraz liczby linijek kodu dla każdego ze zdefiniowanych zapytań. Ostateczne wyniki przeprowadzonych eksperymentów pokazały, że Hive jest wydajniejszy, wygodniejszy i mniej skomplikowany w użyciu niż Pig [1].

Autorzy pracy „*Performance Analysis of ECG Big Data using Apache Hive and Apache Pig*" porównali wydajność Apache Pig i Apache Hive w kontekście analizy danych EKG pacjentów pochodzących ze szpitali. Celem badania było wykazanie, które narzędzie jest lepsze pod względem szybkości przetwarzania i optymalizacji czasu zapytań. Wynik badań miał za zadanie ułatwić koncernom medycznym mającym styczność z danymi EKG wybór narzędzia używanego do analizy danych. Zbadane zostały dwa aspekty, które mogą mieć wpływ na wynik badań, którymi są kolejno rozmiar zestawu danych oraz czas wykonywania zapytania w celu pozyskania określonej informacji. Przeprowadzone badania wykazały zależność między zbiorem danych, a czasem przetwarzania. Wraz ze wzrostem rozmiaru danych rośnie czas ich przetwarzania. Do porównania narzędzi Apache Pig i Apache Hive wybrano dwie próbki danych zawierających odpowiednio 5076 oraz 15230 rekordów. Czasy wykonania podobnych zapytań na mniejszym zbiorze prezentowały się następująco: dla Pig było to 3,00 sekundy, natomiast dla Hive 46,66 sekund. Natomiast dla większego zbioru danych było to odpowiednio 9,00 sekund oraz 55,69 sekund. Autorzy, wykazali, że czas wykonywania zapytań przez Apache Hive jest zdecydowanie wyższy w porównaniu do Apache Pig. Ostateczny werdykt określił Pig znacznie wydajniejszym narzędziem [2].

W artykule pt. „*Processing performance on Apache Pig, Apache Hive and MySQL cluster*" badacze porównują czas przetwarzania danych z użyciem narzędzi Hive, Pig i MySQL Cluster przy zbiorze danych charakteryzującym się prostą strukturą. Celem badania było określenie, które z narzędzi jest najwydajniejsze. Autorzy wykazali, że na niedrogim sprzęcie Hive jest znacznie szybszy od MySQL Cluster oraz jest w stanie obsługiwać obszerne zbiory danych. Natomiast dla wybranego zestawu danych Apache Pig wydaje się nieodpowiednim narzędziem. Według autorów Apache Pig znajduje lepsze zastosowanie dla złożonych zapytań i jeszcze większych zestawów danych niż użyty w badaniach. Eksperyment pokazał, że zapytania MySQL Cluster są szybsze niż zapytania Hive i Pig do pewnego momentu. Jednakże, kiedy liczba rekordów zestawów danych zwiększa się, to narzędzie Apache Hive staje się bardziej niezawodne. Jedną z przeszkód klastra MySQL jest to, że wraz ze wzrostem rozmiaru danych wymaga on więcej pamięci RAM do ich przetwarzania. Indeksowane kolumny są zawsze przechowywane w pamięci. Im większy rozmiar danych, tym więcej pamięci lub maszyny będzie potrzebować. Wyniki eksperymentu pokazały, że Apache Hive pokonuje wydajność MySQL Cluster i Apache Pig. Jednakże autorzy pracy zaznaczają, że wydajność narzędzi może ulec zmianie w zależności od klasy używanego sprzętu. Przy większej ilości pamięci RAM wydajność klastra MySQL mogłaby być większa i przewyższać inne narzędzia [3].

Wyniki każdej z wymienionych prac świadczą o tym, że nie ma ostatecznego zwycięzcy pośród badanej dwójki narzędzi, którymi są Pig i Hive. Wydajność narzędzi może być odmienna zależnie od dostępnego środowiska, wielkości zbiorów danych oraz złożoności tworzonych zapytań. Niniejsza praca oraz zawarte w niej wyniki badań mogą stanowić pomoc przy decyzji w wyborze odpowiedniego narzędzia do przetwarzania i analizy danych.

3. Narzędzia Big Data w ekosystemie Hadoop

W niniejszym rozdziale przedstawiono technologie i narzędzia Big Data powiązane z tematem badań, a dokładniej: środowisko Apache Hadoop, technologię MapReduce oraz narzędzia Pig i Hive.

3.1. Środowisko Apache Hadoop

Apache Hadoop jest otwarto-źródłowym szkieletem aplikacyjnym rozwijanym przez organizację Apache Software Foundation, której przeznaczeniem jest przetwarzanie i przechowywanie dużych zbiorów danych. Szkielet udostępnia możliwości rozproszonego przetwarzania Big Data w klastrach komputerów przy użyciu prostych modeli programowania. Zaprojektowano go w taki sposób, aby liczba serwerów stosowana w implementowanym rozwiązaniu mogła być skalowalna od pojedynczej do nawet tysięcy maszyn [4].

Projekt Apache Hadoop składa się z kilku modułów, z których każdy ma inne zastosowanie. Tymi modułami są: HDFS – rozproszony system plików, YARN – zarządca zasobów i zadań oraz MapReduce – model programowania służący do przetwarzania dużej ilości danych w sposób równoległy na którym został oparte narzędzia Apache Pig i Apache Hive [4].

Otwarto-źródłowość projektu przyczyniła się do powstania społeczności miłośników tej technologii oraz zwiększenia liczby osób zaangażowanych w rozwój projektu. Wokół projektu zaczęło pojawiać się wiele innych projektów i narzędzi powiązanych ze szkieletem aplikacyjnym Hadoop. Znaczna część powstałych rozwiązań na stałe zagościła w projekcie Hadoop i stała się jego integralną częścią. Zbiór narzędzi i projektów zbudowanych dookoła opisywanego środowiska nazywa się Hadoop Ekosystem. Wymieniony zbiór daje możliwość wykonywania jeszcze bardziej zaawansowanej analizy danych i pozwala na lepsze wykorzystanie potencjału samego środowiska Hadoop [5].

3.2. Apache MapReduce

Apache Hadoop MapReduce jest szkieletem aplikacyjnym umożliwiającym tworzenie aplikacji, które przetwarzają równolegle ogromne zbiory danych na dużych klastrach sprzętu komputerowego w sposób niezawodny i odporny na błędy.

Hadoop MapReduce zazwyczaj dzieli zbiór danych wejściowych na niezależne fragmenty, które są potem przetwarzane przez zadania mapowania w sposób

równoległy, następnie wyniki mapowania są sortowane i wprowadzane do zadań redukcji. Najczęściej zarówno dane wejściowe jak i wyjściowe są przechowywane w systemie plików HDFS [6].

Szkielet MapReduce zajmuje się planowaniem i monitorowaniem zadań oraz ponownym wykonywaniem, tych które się nie powiodły. MapReduce i HDFS działają na tych samych zestawach węzłów, co oznacza, że węzły przeznaczone do obliczeń i węzły magazynowe są tymi samymi. Taka konfiguracja pozwala na efektywne zaplanowanie zadań na węzłach, na których już znajdują się dane, co skutkuje bardzo wysoką łączną przepustowością w całym klastrze [6].

3.3. Apache Hive

Apache Hive to projekt, którego celem jest uproszczenie przetwarzania dużych zbiorów danych. Jest skierowany w szczególności dla użytkowników korzystających ze środowiska Hadoop oraz narzędzi typu business intelligence. Projekt dostarcza interfejs zbliżony do języka SQL dla technologii Hadoop MapReduce [7].

Rozwiązanie to wspiera właściwości ACID (ang. atomicity, consistency, isolation, durability) oraz indeksowanie, jednakże nie jest przeznaczone do zadań typu OLTP (ang. Online Transaction Processing). Język zapytań dostarczany z projektem nosi nazwę HQL (ang. Hive Query Language). Jego semantyka i funkcje przypominają standard języka SQL dla relacyjnych baz danych [7].

Struktura metadanych Hive pozwala na konstruowanie wysokopoziomowych struktur na szczycie systemu plików HDFS, złudnie podobnych do tabel w typowych rozwiązaniach magazynowych tabele odpowiadają katalogom danych w HDFS [7].

Praca z Hive nie wymaga wcześniejszego tworzenia schematów bazodanowych przed umieszczeniem danych w magazynie. Schematy metadanych Hive są zazwyczaj tworzone w sposób ad-hoc przy wczytywaniu zbiorów danych. Rozwiązanie zapewnia elastyczność i wydajność w pracy z danymi, a także dostarcza gotowy do użycia, zoptymalizowany i prosty w obsłudze model zapytań nie wymagający złożonych operacji programistycznych jak w przypadku MapReduce [7].

3.4. Apache Pig

Apache Pig jest otwarto-źródłową platformą przeznaczoną do analizy i przekształcania zbiorów danych. Na platformę składają się proceduralny język skryptowy Pig Latin, oraz środowisko uruchomieniowe. Narzędzie posiada nowatorskie środowisko do debugowania, którego przydatność może być szczególnie doceniona przy pracy z ogromnymi zbiorami danych [8].

Pig Latin zapewnia: wsparcie dla elastycznych zagnieżdżonych modeli danych, wsparcie dla funkcji definiowanych przez użytkownika oraz możliwość przeprowadzania operacji na plikach bez informacji o strukturze danych. Język ten został zaprojektowany tak, aby mógł obsługiwać wiele najpopularniejszych prostych typów danych takich jak ciągi znaków, liczby całkowite i zmiennoprzecinkowe oraz złożonych typów danych takich jak krotki, kolekcje krotek, mapy. Jego ideą jest umożliwienie intuicyjnego i łatwego modelowania przepływów danych i ich przekształcania [8].

Program napisany w Pig Latin składa się z sekwencji instrukcji opisujących operacje przetwarzania zbioru danych w sposób potokowy. Operacje te z pomocą kompilatora po uprzedniej analizie składni i poprawności kodu, są optymalizowane i konwertowane na zadania MapReduce, które następnie są uruchamiane. Pig umożliwia równoległe przetwarzanie ogromnych zbiorów danych oraz wykonywanie operacji ETL na danych pochodzących z różnych źródeł. Dane wynikowe zapisywane są w HDFS [2].

4. Środowisko badawcze

W celu oceny efektywności przetwarzania danych z użyciem Apache Hive i Apache Pig w środowisku Hadoop do przeprowadzenia badań wykorzystano jeden z dostępnych na rynku wariantów oprogramowania implementujących środowisko Hadoop – Cloudera. Środowisko Cloudera może być zainstalowane, uruchomione i konfigurowane na fizycznej stacji roboczej lub w postaci gotowej do użycia maszyny wirtualnej. Do badań wykorzystano maszynę wirtualną, którą uruchomiono w środowisku VMWare Workstation 17 Player [9]. Konfigurację stacji roboczej przedstawiono w Tabeli 1.

Tabela 1: Konfiguracja fizycznej maszyny

Właściwość	Opis
Model laptopa	Lenovo Thinkpad T14 Gen 3
System operacyjny	Windows 11 Pro 22H2
Środowisko wirtualizacji	VMware Workstation 17 Player
Procesor	Intel Core i7-1260P 2.10 GHz, 12 procesorów, 16 wątków
Pamięć RAM	2x16 GB, Kingston, DDR4, 3200 MHz
Dysk twardy	SK Hynix PC711 512GB SSD, PCIe GEN 3 x4.0 NVMe

Aby zmaksymalizować wydajność maszyny wirtualnej do jej konfiguracji przydzielono największą możliwą ilość zasobów dostępnych w maszynie hosta umożliwiających stabilną pracę. Cloudera Distributed Hadoop (CDH) to w pełni otwarto-źródłowa dystrybucja platformy Cloudera zawierająca środowisko Apache Hadoop oraz ważne narzędzia powiązane ze środowiskiem Hadoop w których skład wchodzą m.in. Apache Pig i Apache Hive [10]. Parametry maszyny wirtualnej uruchamiającej środowisko Cloudera Quickstart CDH VM 5.3 przedstawiono w Tabeli 2.

Tabela 2: Konfiguracja maszyny wirtualnej Cloudera Quickstart CDH VM 5.3

Właściwość	Opis
System operacyjny	CentOS 6.4 z zainstalowanym środowiskiem Cloudera CDH
Procesor	8 procesorów logicznych
Pamięć RAM	26 GB
Dysk	128 GB
System operacyjny	CentOS 6.4 z zainstalowanym środowiskiem Cloudera CDH
Procesor	8 procesorów logicznych

Na potrzeby badań wykorzystani darmowy zbiór danych „NYC Taxi Trips" udostępniany w portalu Maven Analytics [11] zawierający ponad 28 milionów rekordów z danymi przejazdów wszystkich zielonych taksówek w Nowym Jorku w latach od 2017 do 2020. Rekordy zawierają m.in. informacje dotyczące dat i godzin rozpoczęcia i zakończenia przejazdów, lokalizacji początkowej i końcowej, długości przejazdów, liczbie pasażerów i kosztach przejazdu.

Przed wykonaniem badań należało uprzednio dostosować środowisko Cloudera w maszynie wirtualnej do potrzeb eksperymentów. Uruchomienie maszyny wirtualnej z oprogramowaniem CDH nie uruchamia automatycznie interfejsu Cloudera Manager. W celu korzystania z jego funkcjonalności, należało z poziomu pulpitu użytkownika systemu wykonać skrypt uruchamiający interfejs. W celu dalszego przygotowania wirtualnej maszyny do przeprowadzenia badań należy wyłączyć narzędzia, które nie będą wymagane do zaimplementowania scenariuszy. Wyłączone zostały następujące technologie: Apache Flume, Apache HBase, Apache Hue, Apache Impala, Apache Oozie, ks_indexer, Apache Sentry, Apache Solr, Apache Spark, Apache Sqoop. Natomiast technologie, które pozostawiono uruchomionymi to: HDFS, Apache Hive, Apache Zookeeper, Apache Yarn. Zarządzanie stanem wyżej wymienionych usług odgrywa istotną rolę w wykorzystaniu zasobów, stąd zalecane jest ograniczenie liczby włączonych usług w celu zwiększenia wydajności ich działania.

5. Metoda badań

Badania zostały przeprowadzone w uprzednio przygotowanym oraz skonfigurowanym środowisku badawczym. Po czynnościach związanych z dostosowaniem fizycznej stacji roboczej i maszyny wirtualnej, do środowiska został wczytany zestaw danych. który posłużył do realizacji szesnastu scenariuszy badawczych zdefiniowanych dla obu testowanych narzędzi. Scenariusze badawcze zostały zaimplementowane w postaci zapytań HQL i skryptów Pig Latin. Przed wykonaniem skryptów założono, że istnieją już utworzone tabele z danymi w narzędziu Apache Hive.

Za pomocą metody badań zmierzone zostały czasy wykonania (w sekundach) skryptów zdefiniowanych dla treści scenariusza. Mierzono odcinek czasu od przekazania pików z implementacją scenariuszy do testowanych narzędzi, aż do zakończenia ich działania i zapisania wyników przetwarzania do plików.

Na potrzeby eksperymentu przygotowano katalog zawierający podkatalogi odpowiadające konkretnym scenariuszom badawczym. W każdym z podkatalogów umieszczono pliki z implementacjami scenariuszy narzędzi oraz skrypt uruchamiający je dziesięciokrotnie i wykonujący pomiary czasu. Wyniki każdej iteracji zapisywano do plików w podkatalogach odpowiadającym uruchomionemu narzędziu dla danego scenariusza. Po wykonaniu zaplanowanej sekwencji działań informacje o czasach wykonania poszczególnych prób zapisywano do pliku tekstowego w katalogu scenariusza. Zebrane z eksperymentów umieszczano ręcznie w pliku arkusza kalkulacyjnego, w którym wyliczano średni czas wykonania każdego ze scenariuszy badawczych. W kolejnym kroku wynik wyliczeń umieszczano w tabeli grupującej ostateczne wyniki dla scenariuszy.

6. Scenariusze badawcze

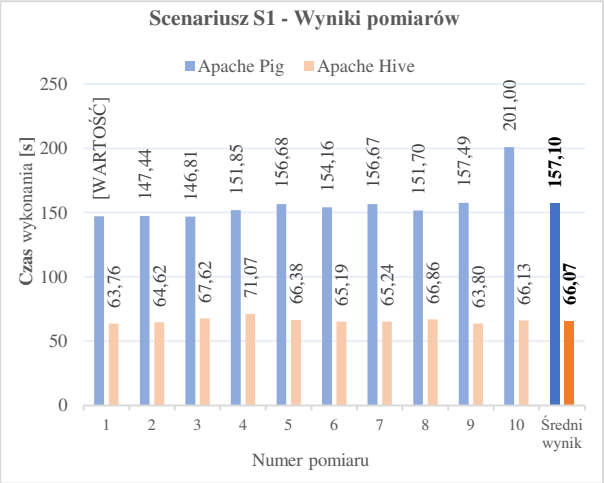
Na potrzeby przeprowadzenia badań w celu oceny efektywności przetwarzania danych z użyciem narzędzi Apache Hive i Apache Pig w środowisku Hadoop na podstawie wybranego zbioru danych przygotowano następujące scenariusze badawcze:

1. Wyznaczenie całkowitej liczby przejazdów taksówek dla poszczególnych miesięcy dla każdego roku (S1).
2. Wyznaczenie całkowitej kwoty płatności kartą kredytową dla każdego roku (S2).
3. Wyznaczenie całkowitej liczby przejazdów dla każdego rodzaju przejazdu, które były opłacone kartą kredytową (S3).
4. Wyznaczenie średniej liczby pasażerów w danym miesiącu w roku 2018 (S4).
5. Wyznaczenie średniej liczby pasażerów dla każdej godziny w ciągu doby w dniach roboczych w maju 2019 (S5).
6. Wyznaczenie pięciu dzielnic, w których kończyło się najwięcej przejazdów taksówką w weekendy w roku 2020 (S6).
7. Wyznaczenie średnich czasów podróży dla kwartałów w latach 2017-2020 (S7).
8. Wyznaczenie średniego kosztu przejazdu dla godzin dziennych (6.00- 21.59) i nocnych (22.00-5.59) dla każdego roku (S8).
9. Wyznaczenie trzech dni w roku 2017, w których średnie koszty przejazdów taksówką rozpoczętych w godzinach między godziną 16.00 a 17.00 były najmniejsze (S9).
10. Wyznaczenie całkowitej liczby przejazdów taksówką w latach 2017-2020 (S10).
11. Wyznaczenie maksymalnej i minimalnej kwoty napiwku dla każdego roku (S11).
12. Wyznaczenie średniej kwoty opłaty za przejazdu dla przejazdów z napiwkiem i bez napiwku w roku 2017 (S12).
13. Wyznaczenie średniej kwoty napiwku dla przejazdów w zależności od liczby pasażerów w godzinach dziennych 2020 roku (S13).
14. Wyznaczenie dla każdego roku średniej prędkości przejazdu taksówką na najpopularniejszej trasie przejazdu w każdym roku dla każdego roku i końcową ze wszystkich lat (S14).
15. Wyznaczenie średniej długości trasy przejazdu taksówką dla przejazdów opłaconych kartą kredytową i gotówką dla każdego roku (S15).
16. Wyznaczenie trzech dzielnic miasta, które mają największą różnicę w średnim koszcie przejazdu taksówką w ciągu dnia między godzinami szczytu (14-17), a godzinami poza szczytem (6-14 i 17-22) (S16).

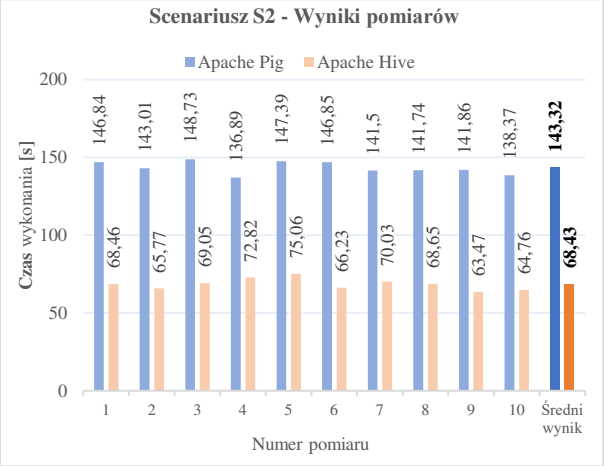
7. Wyniki badań

W niniejszym rozdziale zaprezentowano wyniki badań uzyskane na podstawie zastosowanej metody badań dla zdefiniowanych scenariuszy. Wyniki badań zostały zaprezentowane w postaci wykresów słupkowych zawierających informacje o czasach wykonywania zapytania odpowiednio dla każdego z badanych narzędzi.

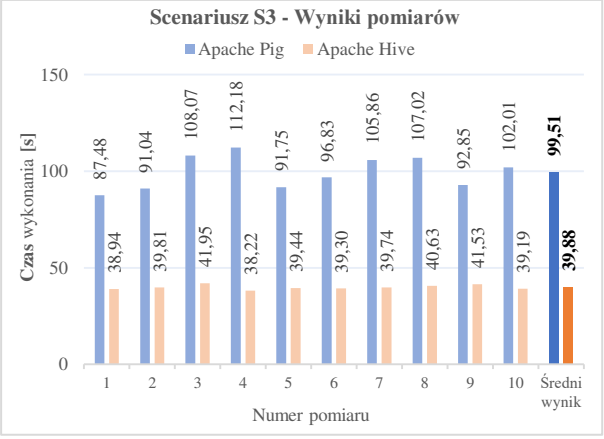
Rysunki 1-16 przedstawiają wyniki pomiarów i ich uśrednioną wartość dla scenariuszy S1-S16.



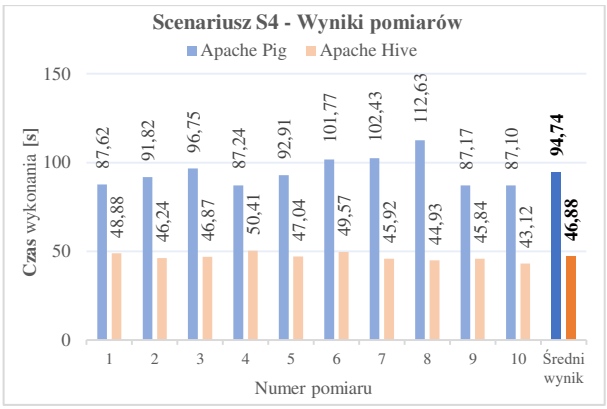
Rysunek 1: Scenariusz S1 – Wyniki pomiarów.



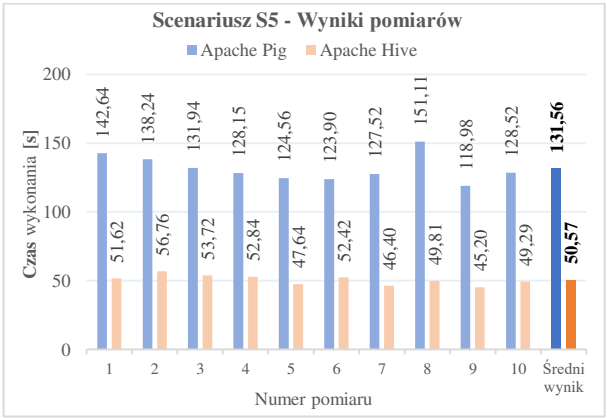
Rysunek 2: Scenariusz S2 – Wyniki pomiarów.



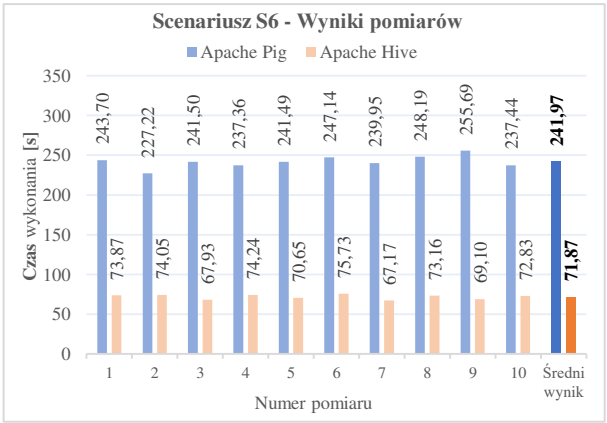
Rysunek 3: Scenariusz S3 – Wyniki pomiarów.



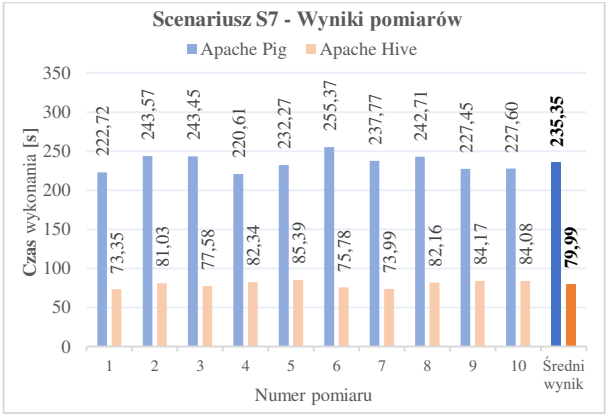
Rysunek 4: Scenariusz S4 – Wyniki pomiarów.



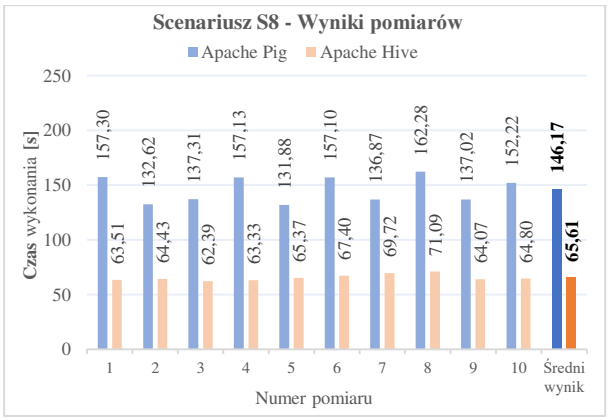
Rysunek 5: Scenariusz S5 – Wyniki pomiarów.



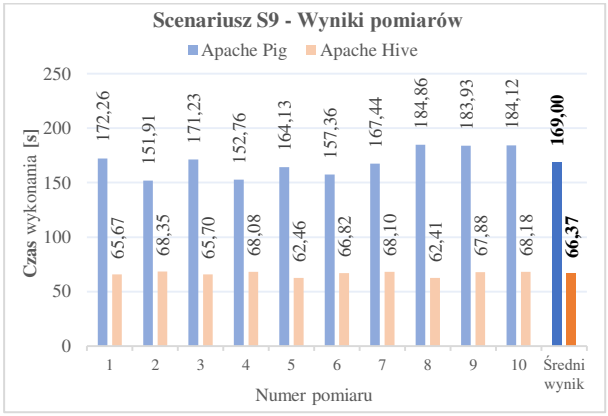
Rysunek 6: Scenariusz S6 – Wyniki pomiarów.



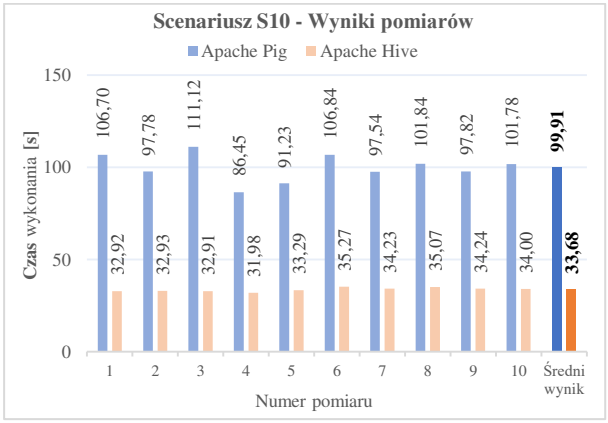
Rysunek 7: Scenariusz S7 – Wyniki pomiarów.



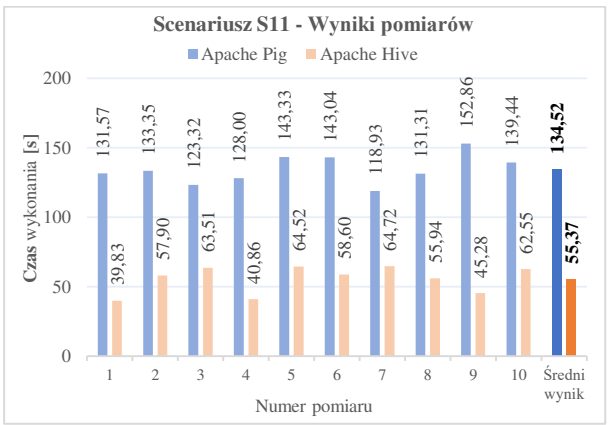
Rysunek 8: Scenariusz S8 – Wyniki pomiarów.



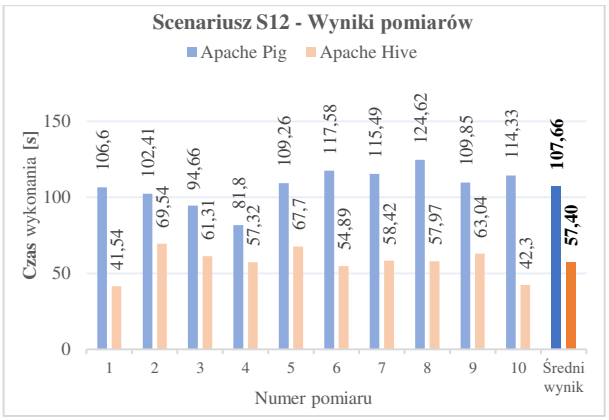
Rysunek 9: Scenariusz S9 – Wyniki pomiarów.



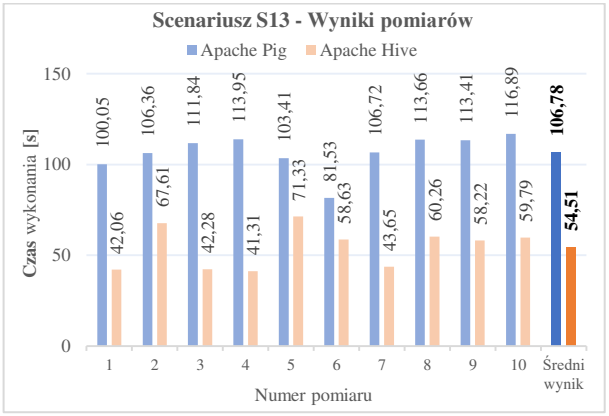
Rysunek 10: Scenariusz S10 – Wyniki pomiarów



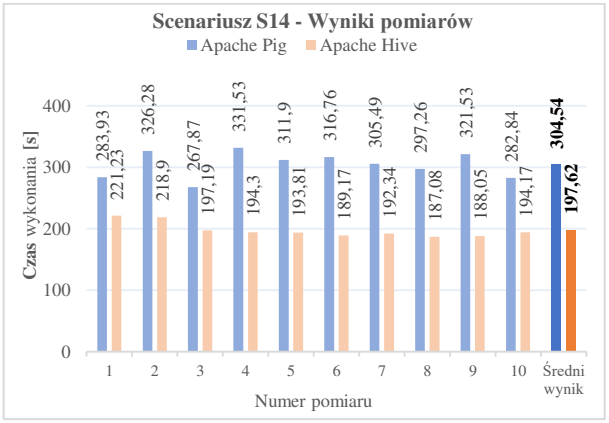
Rysunek 11: Scenariusz S11 – Wyniki pomiarów.



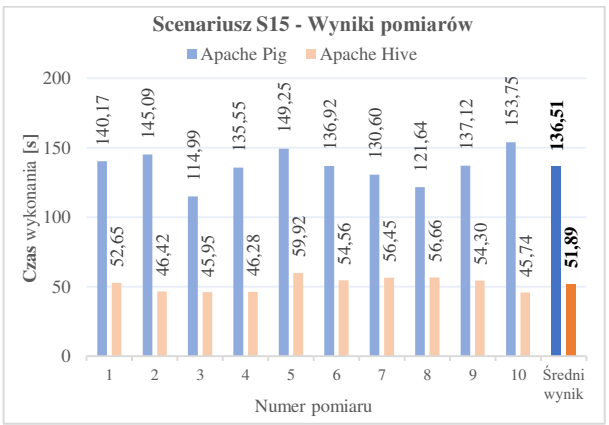
Rysunek 12: Scenariusz S12 – Wyniki pomiarów.



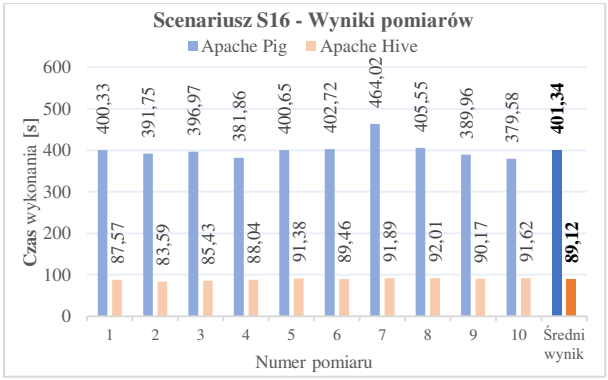
Rysunek 13: Scenariusz S13 – Wyniki pomiarów.



Rysunek 14: Scenariusz S14 – Wyniki pomiarów

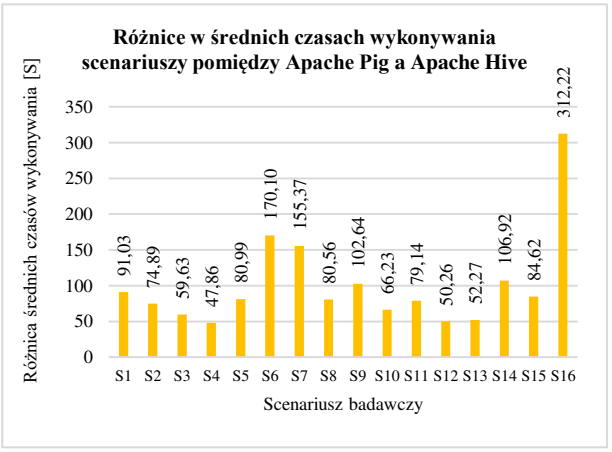


Rysunek 15: Scenariusz S15 – Wyniki pomiarów.



Rysunek 16: Scenariusz S16 – Wyniki pomiarów.

Rysunek 17 przedstawia różnice średnich czasów wykonywania scenariuszy pomiędzy narzędziami Apache Pig a Apache Hive.



Rysunek 17: Różnice w średnich czasach wykonywania scenariuszy pomiędzy Apache Pig a Apache Hive.

8. Dyskusja wyników

Analizując wyniki badań można zauważyć, że zdecydowaną przewagę w efektywnym przetwarzaniu danych w środowisku Apache Hadoop ma narzędzie Apache Hive, którego wyniki w każdym ze scenariuszy prezentują się znacznie lepiej niż dla Apache Pig. Dla większości badanych scenariuszy średni czas wykonywania scenariusza Pig był około dwu- lub trzykrotnie większy w porównaniu do czasu drugiego narzędzia.

Najmniejsza różnica w średnich czasach przetwarzania scenariuszy wyniosła 47,86 s (scenariusz S4), natomiast największa 312,22 s (scenariusz S16). Największa wartość współczynnika średniego czasu przetwarzania Apache Pig do drugiego narzędzia wyniosła 4,5 w scenariuszu S16, natomiast najmniejszy stosunek czasów wystąpił w scenariuszu S14, którego wartość była równa 1,5.

Najdłuższy średni czas wykonywania zapytania dla Hive wyniósł 197,62 s (scenariusz S14), a najkrótszy 33,68 s (scenariusz S10), w przypadku drugiego narzędzia czasy te wyniosły odpowiednio 401,34 s (scenariusz S16) oraz 94,74 s (scenariusz S4).

Pomimo tego, że narzędzie Pig nie okazało się być tak efektywnym jak Hive to istnieje prawdopodobieństwo, że możliwe jest osiągnięcie lepszych wyników z jego użyciem. Wpływ na efektywność przetwarzania może mieć konfiguracja środowiska badawczego tj. m.in.: ilość przydzielonych zasobów takich jak pamięć RAM i liczba procesorów, przepustowość sieci, prędkość odczytu dysków twardych lub liczba maszyn w klastrze biorąca udział w obliczeniach.

Z powodu braku dostępu do bardziej zaawansowanych technologicznie stacji roboczych badania zostały przeprowadzone na pojedynczej maszynie. Środowisko Cloudera w interfejsie Cloudera Manager komunikuje, że dla usług Hive, Yarn czy HDFS powinny istnieć co najmniej jeszcze dwa inne węzły w klastrze. Dodatkowym czynnikiem, który może mieć wpływ na wyniki jest m.in. wielkość zestawu danych.

Kolejnym, ale również ważnym elementem wpływającym na efektywność przetwarzania jest skomplikowanie uruchamianych zapytań i skryptów. Wyniki badań wykazują, że narzędzie Apache Pig przetwarzało dane najwolniej w przypadku bardziej skomplikowanych scenariuszy, w których pojawiały się operacje na danych takie jak: tworzenie podzapytania, więcej niż jedno złączenie tabel, filtrowanie danych czy sortowanie (scenariusze S6, S7, S16). Natomiast dobrze radziło sobie z prostymi zapytaniami, w których występowało grupowanie lub filtrowanie (scenariusze S4, S13). W przypadku Apache Hive czasy wykonania wszystkich scenariuszy oprócz najbardziej skomplikowanego (scenariusz S14) wyniosły mniej niż 90 sekund. Wyniki badań jednoznacznie wykazują, że Apache Hive jest narzędziem efektywniejszym w przetwarzaniu danych niż Apache Pig w skonfigurowanym środowisku badawczym z ekosystemem Apache Hadoop.

9. Wnioski

Celem niniejszej pracy była analiza efektywności przetwarzania danych za pomocą narzędzi Apache Hive i Apache Pig w środowisku Hadoop.

Wstępne badania narzędzi w obszarze Big Data wyróżniły środowisko Hadoop jako przodujące w rozwiązaniach przetwarzania i analizy dużych zbiorów danych. Wspomniane środowisko wyróżnia się szerokim zakresem różnorodnych narzędzi gotowych do użycia, otwarto-źródłowością oraz prężnie działającą społecznością rozwijającą oprogramowanie. Przegląd literatury ugruntował stwierdzenie iż, środowisko Hadoop oraz narzędzia Pig i Hive są flagowymi rozwiązaniami w przetwarzaniu wielkich ilości danych

Kolejnym etapem było merytoryczne przygotowanie środowiska badawczego w postaci wyboru próbki zbioru danych, zdefiniowania scenariuszy badawczych oraz predykcji optymalnej konfiguracji stacji roboczej, maszyny wirtualnej i oprogramowania Hadoop, biorąc pod uwagę sprzęt dostępny autorowi. Środowisko Cloudera CDH implementując Hadoop umożliwiło na sprawne stworzenie warunków koniecznych do przeprowadzenia badań.

Główną metodą badań było wykonanie zapytań HQL i skryptów Pig Latin dla każdego zdefiniowanego scenariusza badawczego na ówcześnie przygotowanym, tym samym zestawie danych. Celem metody badań było

zaobserwowanie wpływu wyboru narzędzia na wydajność procesowania w takich samych warunkach. Scenariusze badawcze cechowały się różnym stopniem złożoności, by przetestować możliwości obu narzędzi.

Wpływ na wiarygodność wyników odgrywało środowisko badawcze. Należy zaznaczyć, że z powodu braku dostępu do zaawansowanych stacji roboczych autor wykorzystał jedynie jedną maszynę. W przypadku wykorzystania klastra obciążenie obliczeniowe prezentowałoby się w odmienny sposób.

Wyniki badań jednogłośnie przemawiają na korzyść Apache Hive. Flagowymi scenariuszami uwydatniającymi różnice przetwarzania są scenariusze S4 i S16. Zauważono iż Hive, może być efektywniejszy od Pig nawet czterokrotnie. W pierwszym skrajnym scenariuszu stosunek długości średniego czasu wykonywania Pig do Hive uzyskał wartość 1.5, natomiast w drugim 4.5. Dodatkowo warto zaznaczyć, że średni czas wykonywania we wszystkich scenariuszach Hive, oprócz scenariusza S14, był mniejszy niż najmniejszy średni czas wykonywania ze wszystkich scenariuszy w Pig.

Podczas badań zmierzono się z wyzwaniami jakie postawiły: konfiguracja środowiska, czy też przechowywanie wyników mających wpływ na dalsze badania.

Działania podjęte w ramach pracy na polu teoretycznym jak i praktycznym umożliwiły osiągnięcie postawionego celu. Wyniki badań jednoznacznie wskazują Apache Hive jako bardziej efektywne narzędzie do przetwarzania dużej ilości danych w środowisku Hadoop.

Literatura

[1] K. Bansal, P. Chawla, P. Kurle, Analyzing Performance of Apache Pig and Apache Hive with Hadoop, International Conference On Engineering Vibration Communication and Information Processing (ICoEVCI), (2018) 41-51, https://doi.org/10.1007/978-981-13-1642-5_4

[2] M. Ahmad, S. Kanwal, M. Cheema, M. A. Habib, Performance Analysis of ECG Big Data using Apache Hive and Apache Pig, 2019 8th International Conference on Information and Communication Technologies (ICICT), (2019) 2-7, https://doi.org/10.1109/ICICT47744.2019.9001287

[3] A. Fuad, A. Erwin, H. P. Ipung, Processing performance on Apache Pig, Apache Hive and MySQL cluster, Proceedings of International Conference on Information, Communication Technology and System (ICTS), (2014) 297-302, https://doi.org/10.1109/ICTS.2014.7010600

[4] Dokumentacja techniczna technologii Apache Hadoop https://hadoop.apache.org/, [10.07.2023]

[5] K. Sitto, M. Presser, Field Guide to Hadoop: An Introduction to Hadoop, Its Ecosystem, and Aligned Technologies, O'Reilly Media, 2015

[6] Dokumentacja techniczna technologii MapReduce https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Overview, [10.07.2023]

[7] D Dayong., Apache Hive Essentials Second Edition, Packt Publishing, 2015

[8] C. Swarna, Z. Ansari, Apache Pig-a data flow framework based on Hadoop Map Reduce. International Journal of Engineering Trends and Technology (IJETT), 50 (5) (2017) 271-275 https://doi.org/10.14445/22315381/IJETT-V50P244

[9] Środowisko wirtualizacji VMware Workstation 17 Player https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html, [10.07.2023]

[10] Komponenty składowe środowiska Cloudera CDH https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html, [10.07.2023]

[11] Zbiór danych testowych „NYC Taxi Trips Dataset" https://maven-datasets.s3.amazonaws.com/Taxi+Trips/NYC_Taxi_Trips.zip, [10.07.2023]