JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 28 (2023) 210–216
Received: 24 May 2023
Accepted: 17 June 2023

A comparative analysis of the performance of the relational database and the Hadoop environment in the context of analytical data processing

Analiza porównawcza wydajności relacyjnej bazy danych i środowiska Hadoop w kontekście analitycznego przetwarzania danych

Michał Zadrąg*

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

Abstract
The article presents a detailed comparative analysis of the performance of a Microsoft SQL Server relational database and an Apache Hadoop environment in the context of analytical data processing. The study was carried out by executing more than a dozen research scenarios with different queries on datasets of varying sizes. For each research scenario, the average query execution time on different datasets was compared. Based on the results, it was found that the average execution time of queries from the presented scenarios is significantly shorter in MS SQL Server than in Apache Hadoop.

Keywords: Apache Hadoop; SQL Server; relational database; OLAP

Streszczenie
Artykuł przedstawia szczegółową analizę porównawczą wydajności relacyjnej bazy danych Microsoft SQL Server i środowiska Apache Hadoop w kontekście analitycznego przetwarzania danych. Badanie zostało przeprowadzone poprzez wykonanie kilkunastu scenariuszy badawczych, w których zastosowano różne zapytania na zbiorach danych o zróżnicowanej wielkości. Dla każdego scenariusza badawczego porównywano średni czas wykonania zapytania na różnych zbiorach danych. Na podstawie otrzymanych wyników stwierdzono, że średni czas wykonania zapytań z przedstawionych scenariuszy jest znacznie krótszy w MS SQL Server niż w Apache Hadoop.

Słowa kluczowe: Apache Hadoop; SQL Server; relacyjna baza danych; OLAP

*Corresponding author
Email address: michal.zadrag@pollub.edu.pl (M. Zadrąg)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W dzisiejszych czasach dane są najcenniejszym zasobem. Sposób, w jaki są one przechowywane oraz przetwarzane jest bardzo istotny, ponieważ może on znacząco zwiększyć przewagę konkurencyjną danej firmy na rynku. Zgromadzone dane można poddać analitycznemu przetwarzaniu i dzięki temu uzyskać informacje pozwalające wyciągnąć wnioski na interesujący nas temat. Wraz z rozwojem technologii a w szczególności Internetu, wolumen danych generowanych oraz przetwarzanych przez systemy informatyczne znacznie się powiększył.

Analityczne przetwarzanie danych [1] (OLAP, Online Analytical Processing) to technologia pozwalająca na złożoną analizę dużego wolumenu danych. Systemy OLAP zostały zaprojektowane w celu ułatwienia wyodrębniania informacji analizy biznesowej z danych w wysoce wydajny sposób. Narzędzia OLAP umożliwiają tworzenie zaawansowanych raportów i analiz, co pozwala na lepsze zrozumienie danych i lepsze podejmowanie decyzji.

Celem niniejszego artykułu jest przeprowadzenie analizy porównawczej wydajności relacyjnej bazy danych Microsoft SQL Server i środowiska Apache Hadoop w kontekście analitycznego przetwarzania danych. Badanie to ma na celu zidentyfikowanie mocnych stron i słabych stron obu podejść, aby pomóc organizacjom w wyborze odpowiedniego narzędzia w zależności od specyfiki ich danych i wymagań analitycznych. Przyjętą hipotezą badawczą jest: Środowisko Hadoop jest wydajniejszym rozwiązaniem w analitycznym przetwarzaniu danych przy dużych wolumenach danych.

W rozdziale drugim przedstawiony zostanie przegląd literatury wykorzystany podczas tworzenia niniejszej pracy. Rozdział trzeci artykułu przedstawia szczegółowy opis relacyjnej bazy danych SQL Server i środowiska Apache Hadoop, omawia różnice w sposobie przechowywania i przetwarzania danych. Opis środowiska badawczego zawarty będzie w rozdziale numer cztery. Rozdział piąty przedstawi metody badawcze oraz zaprezentuje wybrane scenariusze badawcze, wykorzystane podczas badań. Wyniki otrzymane po wykonaniu scenariuszy badawczych, zostaną przedstawione w rozdziale szóstym. Na podstawie wyników tych eksperymentów dokonana zostanie analiza. Rozdział siódmy zawierał będzie podsumowanie oraz wyciągnięte wnioski.

2. Przegląd literatury

Autorzy w artykule [2] opisują czym według nich jest Big Data. Według nich jest to termin określający olbrzymie, różnorodne i posiadające złożoną strukturę dane. Problemem wynikającym z ilości tych danych jest

zdecydowanie sposób ich przechowywania, bardzo istotne jest, aby wybrać optymalny sposób. Autorzy zwracają również uwagę na to, że aby Big Data była użyteczna należy poddać ją szczegółowej analizie w celu wyciągnięcia wniosków, które pozwolą przekuć zdobytą wiedzę w konkretne działania. Na wyszczególnienie zasługuje przytoczona przez autorów statystyka obrazująca znaczący wzrost liczby generowanych danych: Do roku 2003 ludzkość wygenerowała 5 eksabajtów danych. W 2012 taki wolumen danych był generowany w ciągu dwóch dni. Na te dane składa się cała aktywność w internecie. Tradycyjna analiza danych nie jest w stanie przetworzyć danych z Big Data, dlatego wymagała ona kroku naprzód. Krok ten musiał uwzględniać trzy główne cechy wyróżniające Big Data od zwykłego zbioru danych. Autorzy artykułu zwracają uwagę na fakt, że Big Data spowodowała powstanie zupełnie nowej architektury jaką jest architektura MapReduce. Jako przykład implementacji przytaczają Apache Hadoop.

Artykuł [3] autorstwa Umeshwar Dayal oraz Surajit Chaudhuri, skupia się na tym, czym jest według autorów analityczne przetwarzanie danych OLAP (ang. Online analytical processing) w kontekście hurtowni danych. OLAP to metoda przetwarzania danych która pozwala na wielopoziomową i wielowymiarową analizę ogromnych wolumenów danych, dostarczając przy tym możliwość wizualizacji zbioru danych z różnych perspektyw. Głównymi operacjami w OLAP są: drill-down, umożliwia użytkownikom nawigację z wyższego poziomu szczegółowości do niższego poziomu szczegółowości danych; Roll-up, pozwala użytkownikom na nawigację z niższego poziomu szczegółowości do wyższego poziomu szczegółowości danych; Slice-and-dice, pozwala użytkownikom na selektywne przeglądanie podzbioru danych w oparciu o określone kryteria; Pivot, pozwala użytkownikom obracać dane, aby zobaczyć je z innej perspektywy; Drill-through, pozwala użytkownikom na dostęp do szczegółowych danych dla konkretnego punktu danych; Data Mining, pozwala użytkownikowi na wydobycie wiedzy z danych poprzez zastosowanie technik statystycznych, algorytmów uczenia maszynowego i innych metod; Agregacja, umożliwia użytkownikowi wykonywanie obliczeń na danych, takich jak suma, liczba, średnia itp. Autorzy podkreślają to, że zarządzanie hurtowniami danych oraz OLAP stawia nowe wyzwania przez użytkownikami, nadal nie są to technologie bez wad i istotne jest, aby cały czas pracować nad ich doskonaleniem.

Artykuł o tytule "HaoLap: a Hadoop based OLAP System for Big Data" [4] opisuje autorski system OLAP, z wykorzystaniem Apache Hadoop, dla Big Data. Autorzy opisują w artykule cały proces tworzenia systemu od opracowania architektury, aż po samą implementacje. Istotnym czynnikiem jest fakt, że nowo utworzony system został porównany z istniejącymi już systemami: Hive, HadoopDB, HBaseLattice oraz Olap4Cloud. Całe porównanie opierało się na porównaniu, ładowania, funkcji cube oraz roll–up i przechowywania w kontekście 13 klastrów. Na potrzeby eksperymentu autorzy stworzyli zbiór danych "OceanCube" który zawierał dane oceanograficzne. Wyniki porównania ładowania wykazały, że autorski HaoLap jest o 15 razy szybszy od Olap4Cloud i 16 razy szybszy od HBaseLattice. Drugim kryterium było całkowite zużycie czasu na funkcję cube, zgodnie z wynikami HaoLap wypadł lepiej od wszystkich systemów za wyjątkiem HBaseLattice. Najgorszy w zestawieniu okazało się rozwiązanie HadoopDB. Kolejnym testem była operacja roll–up, wyniki były tożsame z tymi z operacji kostkowania tj. HaoLao wypadł lepiej od wszystkich systemów za wyjątkiem HBaseLattice. Ostatnim testem była wielkość jaka jest potrzebna do przechowywania danych przez dane rozwiązanie. HaoLap i Hive wypadły w tym teście najlepiej, ponieważ oba te rozwiązania nie korzystają z żadnych metadanych. Olap4Cloud natomiast bazuje na HBase który musi przetrzymywać indeksy i metadane. HadoopDB z kolei opiera się na relacyjnej bazie danych.

3. Porównywane technologie

3.1. Microsoft SQL Server

SQL Server [5] jest relacyjnym systemem zarządzania bazą danych firmy Microsoft. Pierwsza wersja systemu pojawiła się w 1993 roku i obsługiwała ówczesny standard języka SQL (SQL – 89). Produkt firmy Microsoft korzysta zarówno z tradycyjnego języka zapytań SQL, jak i z rozszerzenia tego języka o nazwę Transact-SQL. T-SQL rozszerza możliwości zwykłego SQL o takie konstrukcje jak pętle, zmienne oraz instrukcje warunkowe. Dodatkowo istnieje możliwość tworzenia własnych funkcji, procedur składowanych, wyzwalaczy oraz kursorów. Według strony https://db-engines.com/ Microsoft SQL Server jest jednym z najpopularniejszych systemów zarządzania relacyjnymi bazami danych.

3.2. Apache Hadoop

Apache Hadoop [6] został stworzony w celu przetwarzania ogromnych zbiorów danych, nie jest to odpowiednie rozwiązanie dla aplikacji czasu rzeczywistego. Cały ekosystem składa się z kilku głównych modułów, które razem tworzą platformę rozproszonego przetwarzania danych:

- Hadoop Distributed File System (HDFS)
- Hadoop MapReduce
- Hadoop YARN (Yet Another Resource Negotiator)
- Hadoop Common

Hadoop Distributed File System [7] został zaprojektowany do niezawodnego przechowywania bardzo dużych wolumenów danych. Opiera się na zasadzie przechowywania plików na wielu maszynach. Hadoop MapReduce [8] jest implementacja paradygmatu MapReduce do przetwarzania dużych ilości danych. Hadoop YARN [9] to system zarządzania zasobami dla Hadoop, odpowiedzialny za zarządzanie zasobami klastra Hadoop i harmonogramowanie zadań do wykonania na tych zasobach. Hadoop Common jest zbiorem bibliotek będących podstawą dla innych modułów z ekosystemu Hadoopa, zapewnia ich podstawową funkcjonalność

oraz wsparcie co pomaga w zarządzaniu omawianym rozwiązaniem.

4. Opis środowiska

Badania zostały przeprowadzone na wirtualnych maszynach o takich samych parametrach, w celu wyeliminowania zniekształcenia wyników wynikającego z różnych parametrów maszyn, na których przeprowadzano test. Tabela 1 przedstawia specyfikację komputera, na którym zostały uruchomione omawiane środowiska wirtualne.

Tabela 1: Specyfikacja komputera do maszyn wirtualnych

Procesor	12th Gen Intel(R) Core(TM) i5-12500H 1.70 GHz
System operacyjny	Windows 10 Pro
Dysk	Kioxia 512GB SSD
RAM	32 GB

Dla każdego środowiska uruchomiona została osobna maszyna wirtualna przy pomocy Oracle VirtualBox w wersji 6.1. Każda instancja opierała się na dystrybucji systemu operacyjnego Linux, Red Hat i posiadała 16 GB pamięci RAM. Na jednej z maszyn wirtualnych została zainstalowana relacyjna baza danych Microsoft SQL Server, natomiast druga korzysta z gotowej dystrybucji Apache Hadoop o nazwie "Cloudera Quickstart VM".

5. Metody badawcze

W celu porównania wydajności, badanie przeprowadzone zostało w formie wykonania zapytań bazodanowych na odpowiednio przygotowanych bazach danych. Baza składa się z tabel wymiarów (Dim) i faktów (Fact), które są powiązane relacjami. Tabele wymiarów zawierają atrybuty służące do analiz. Tabele faktów natomiast, zawierają metryki poddawane analizie, np. dane o sprzedaży. Baza danych liczy jeden milion rekordów. Po każdym wykonaniu zapytania został zmierzony czas jego wykonania. Na podstawie czasów wykonania poszczególnych zapytań zostanie wyciągnięty średni czas wykonania zapytania dla danego zbioru danych. Każde zapytanie, zostało wykonane na wolumenie danych:

- 250 tysięcy rekordów (Zestaw danych numer 1),
- 500 tysięcy rekordów (Zestaw danych numer 2),
- 750 tysięcy rekordów (Zestaw danych numer 3),
- 1 milion rekordów (Zestaw danych numer 4).

Wielokrotne wykonanie zapytań pozwalało na poddanie wyników obróbce statystycznej. Natomiast różna ilość danych pozwala stwierdzić w jaki sposób zachowują się omawiane rozwiązania zależnie od ilości danych. Przygotowane zostały cztery scenariusze badawcze, na podstawie których zostały przeprowadzone badania. Scenariusze badawcze zostały zdefiniowane w ten sposób, aby sprawdzały możliwie wszystkie sposoby przetwarzania danych przez te rozwiązania. Scenariusz pierwszy zawierał definicje zapytania wykorzystującego polecenie ROLLUP. Omawiane zapytanie zwraca wyniki sprzedaży według kraju i roku, a następnie sumuje je na poziomie kraju i na całkowitym poziomie. Kod zapytania ze scenariusza pierwszego został umieszczony na Listingu 1.

Listing 1: Kod zapytania ze scenariusza pierwszego

```
SELECT
    COALESCE(CONVERT(varchar, g.COUNTRYNAME), 'All Countries') AS Country,
    COALESCE(CONVERT(varchar, dt.CALENDARYEAR), 'All Years') AS Year,
    COUNT(DISTINCT c.CUSTOMERKEY) AS NumCustomers,
    ROUND(SUM(os.TRANSACTIONPRICE * os.QUANTITY),2) AS Sales
FROM
    FactOnlineSales os
INNER JOIN
    DimCustomer c ON os.CUSTOMERKEY = c.CUSTOMERKEY
INNER JOIN
    DimDate dt ON os.ORDERDATEKEY = dt.DATEKEY
INNER JOIN
    DimGeography g on c.GEOGRAPHYKEY = g.GEOGRAPHYKEY
GROUP BY
     g.COUNTRYNAME,
     dt.CALENDARYEAR
WITH ROLLUP;
```

Scenariusz drugi zawierał definicje zapytania które zwraca liczbę zamówień złożonych w każdym regionie sprzedaży w styczniu 2020 roku. Ten przykład badawczy sprawdzał wpływ zawężenia wyników zapytania poprzez polecenie WHERE, na średni czas wykonania zapytania. Treść zapytania została przedstawiona na Listingu 2.

Listing 2: Kod zapytania ze scenariusza drugiego

```
SELECT
    st.SALESTERRITORYNAME,
    COUNT(*) AS 'Number of orders'
FROM
    dbo.FactOnlineSales os
JOIN
    dbo.DimSalesterritory st
ON
    os.SALESTERRITORYKEY = st.SALESTERRITORYKEY
JOIN
    dbo.DimDate dt
ON
    os.ORDERDATEKEY = dt.DATEKEY
WHERE
    dt.MONTHNUMBEROFYEAR = 1
    AND dt.CALENDARYEAR = 2020
GROUP BY
    st.SALESTERRITORYNAME
```

Scenariusz trzeci zawierał definicje zapytania które wykorzystywało podzapytanie. Taki test pozwolił określić wpływ wykorzystania podzapytania skorelowanego na średni czas wykonania zapytania. Treść zapytania została przedstawiona na Listingu 2.

Listing 3: Kod zapytania ze scenariusza trzeciego

```
SELECT
    c.CustomerKey AS [Customer ID],
    CONCAT(c.LASTNAME, ' ', c.FIRSTNAME) AS [Customer Name],
    dt.FULLDATE AS [Order Date]
FROM
    FactOnlineSales os
    JOIN DimDate dt ON os.ORDERDATEKEY = dt.DATEKEY
    JOIN DimCustomer c ON os.CUSTOMERKEY = c.CUSTOMERKEY
WHERE
    dt.MONTHNUMBEROFYEAR = 12 AND dt.CALENDARYEAR = 2019 AND
    NOT EXISTS (
        SELECT
            1
        FROM
            FactOnlineSales os2
            JOIN DimDate dt2 ON os2.ORDERDATEKEY = dt2.DATEKEY
        WHERE
            os2.CUSTOMERKEY = os.CUSTOMERKEY AND
            dt2.MONTHNUMBEROFYEAR = 1 AND dt2.CALENDARYEAR = 2020
    )
ORDER BY
    c.LASTNAME;
```

Scenariusz czwarty zawierał definicje zapytania, które korzystało z polecenia Zapytanie korzysta z operatora IN, które pozwala na sprawdzanie czy wartość zawiera się w podanym zbiorze. Zapytanie grupuje rezultaty oraz je sortuje. Listing 4 przedstawia treść zapytania omawianego w scenariuszu czwartym.

Listing 4: Kod zapytania ze scenariusza czwartego

```
SELECT
    dt.CALENDARYEAR AS [Year],
    p.PRODUCTSUBCATEGORYNAME AS [Product],
    COUNT(DISTINCT os.CUSTOMERKEY) AS [#Customers],
    ROUND(SUM(os.TRANSACTIONPRICE * os.QUANTITY), 2) AS [Total Value]
FROM
    FactOnlineSales os
    INNER JOIN DimProduct p ON os.PRODUCTKEY = p.PRODUCTKEY
    INNER JOIN DimDate dt ON os.ORDERDATEKEY = dt.DATEKEY
WHERE
    p.PRODUCTSUBCATEGORYNAME IN ('Mountain Frames', 'Road Frames', 'Touring Frames')
GROUP BY
    dt.CALENDARYEAR, p.PRODUCTSUBCATEGORYNAME
ORDER BY
    [Year], [Product]
```

Scenariusz piąty zawierał definicje zapytania, które korzystało z klauzuli GROUP BY oraz funkcji arytmetycznej SUM oraz ROUND. Zapytanie dodatkowo grupuje rezultaty oraz następnie je sortuje. Listing 5 przedstawia treść zapytania omawianego w scenariuszu piątym.

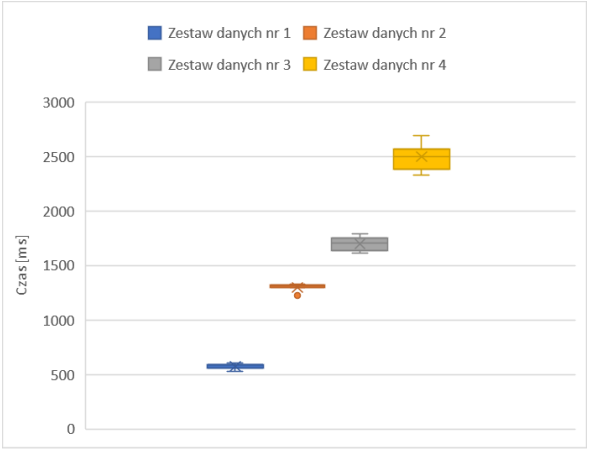
Listing 5: Kod zapytania ze scenariusza piątego

```
SELECT
    dt.CALENDARYEAR AS Year,
    ROUND(SUM(os.TRANSACTIONPRICE * os.QUANTITY), 2) AS 'OrderValue'
FROM
    FactOnlineSales os
    JOIN DimDate dt ON os.ORDERDATEKEY = dt.DATEKEY
GROUP BY
    dt.CALENDARYEAR
ORDER BY
    dt.CALENDARYEAR ASC
```

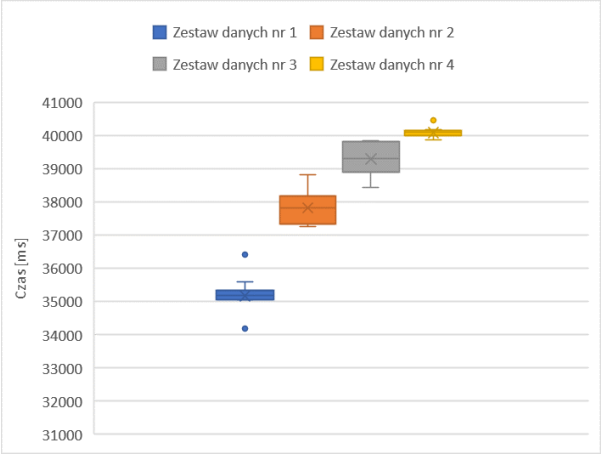
6. Wyniki

Każde zapytanie zostało wykonane sto razy na omawianym w rozdziale dotyczącym metodyki badań wolumenie danych. Czas wykonania zapytania został zmierzony przy użyciu SSMS (SQL Server Management Studio) w przypadku SQL Server, oraz przy użyciu Apache Hue w przypadku Apache Hadoop. Następnie z otrzymanych wyników została wyznaczona średnia oraz utworzone odpowiednie wykresy prezentujące otrzymane wyniki. W celu ujednolicenia wyników oraz ograniczenia wpływu zewnętrznego na wyniki, całe zbędne oprogramowanie zostało wyłączone na czas przeprowadzania badań. Dodatkowo został odrzucony pierwszy wynik, ponieważ czas wykonania tego zapytania był znacznie dłuższy od kolejnych, ponieważ silnik bazodanowy konstruował plan zapytania. Do przedstawienia wyników wykorzystano wykres pudełkowy, ponieważ wykres ten jest użytecznym narzędziem do analizy rozkładu danych, identyfikowania wartości odstających oraz porównywania rozkładów między różnymi grupami lub kategoriami. Dodatkowo wykorzystano wykresy kolumnowe do przedstawienia średnich czasów wykonania dla poszczególnych scenariuszy badawczych.

Wykres zaprezentowany na Rysunku 1 przedstawia rozkład wyników badania, na podstawie scenariusza pierwszego dla technologii SQL Server. Wykres przedstawiony na Rysunku 2 przedstawia rozkład wyników badania na podstawie scenariusza pierwszego dla technologii Apache Hadoop.

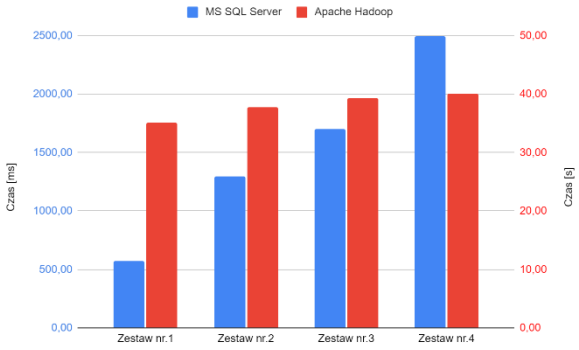


Rysunek 1: Wykres pudełkowy dla scenariusza pierwszego dla SQL Server.



Rysunek 2: Wykres pudełkowy dla scenariusza pierwszego dla Apache Hadoop.

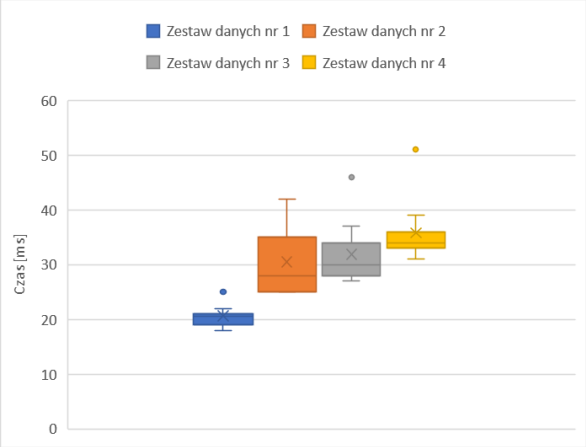
Rysunek 3 przedstawia wykres prezentujący średnie czasy wykonania zapytania, które zostało zdefiniowane i omówione w scenariuszu pierwszym na obu technologiach.



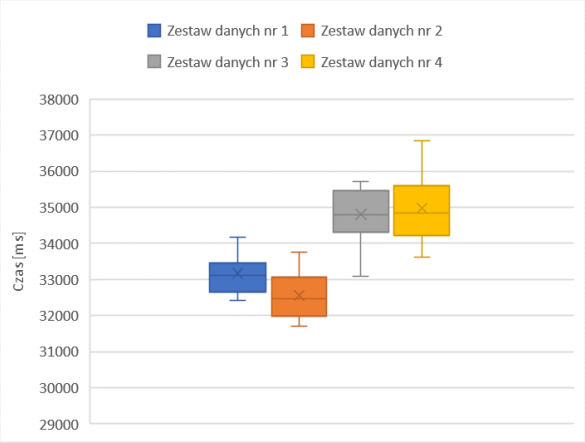
Rysunek 3: Średnie czasy wykonania scenariusza pierwszego.

Wykres przedstawiony na Rysunku 4 przedstawia rozkład wyników badania dla technologii Microsoft SQL Server. Wykres zaprezentowany na Rysunku 5 przedstawia rozkład wyników badania na podstawie

scenariusza drugiego dla technologii Apache Hadoop od firmy Oracle.

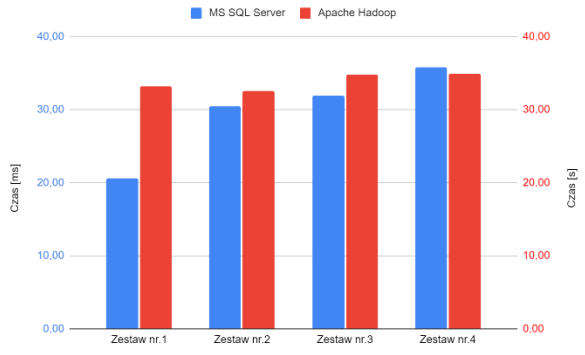


Rysunek 4: Wykres pudełkowy dla scenariusza drugiego dla SQL Server.



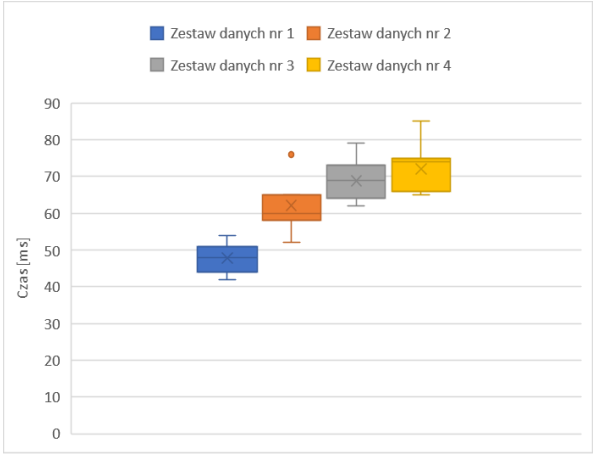
Rysunek 5: Wykres pudełkowy dla scenariusza drugiego dla Apache Hadoop.

Rysunek 6 przedstawia wykres prezentujący średnie czasy wykonania zapytania zdefiniowanego i omówionego w scenariuszu drugim, na obu technologiach.

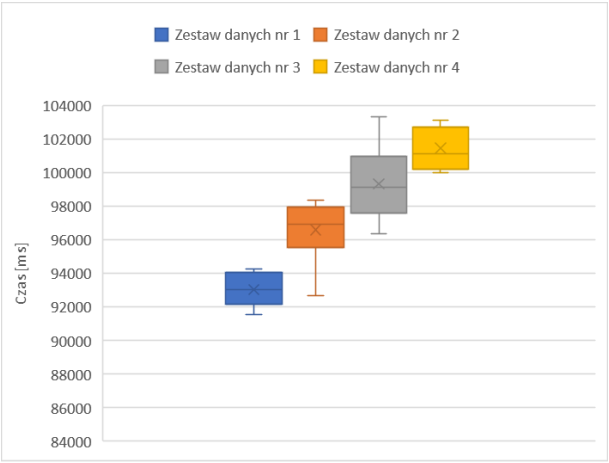


Rysunek 6: Średnie czasy wykonania scenariusza drugiego.

Wykres przedstawiony na Rysunku 7 przedstawia rozkład wyników badania na podstawie scenariusza trzeciego dla technologii Microsoft SQL Server. Wykres zaprezentowany na Rysunku 8 przedstawia rozkład wyników badania na podstawie scenariusza trzeciego dla technologii Apache Hadoop.

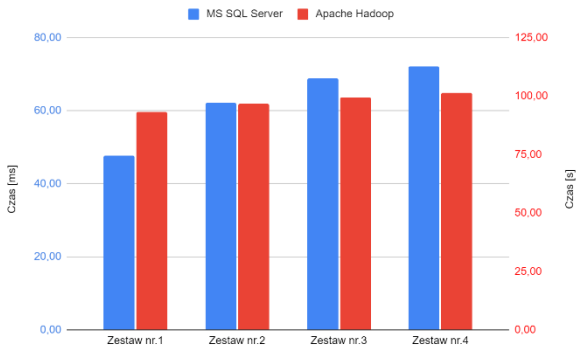


Rysunek 7: Wykres pudełkowy dla scenariusza trzeciego dla SQL Server.



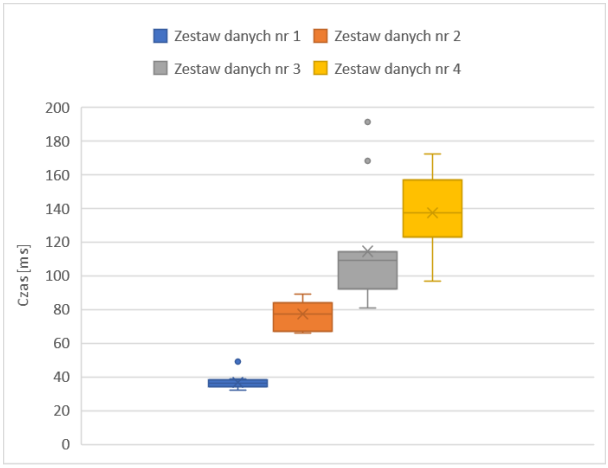
Rysunek 8: Wykres pudełkowy dla scenariusza trzeciego dla Apache Hadoop.

Rysunek 9 przedstawia wykres prezentujący średnie czasy wykonania zapytania ze scenariusza trzeciego.

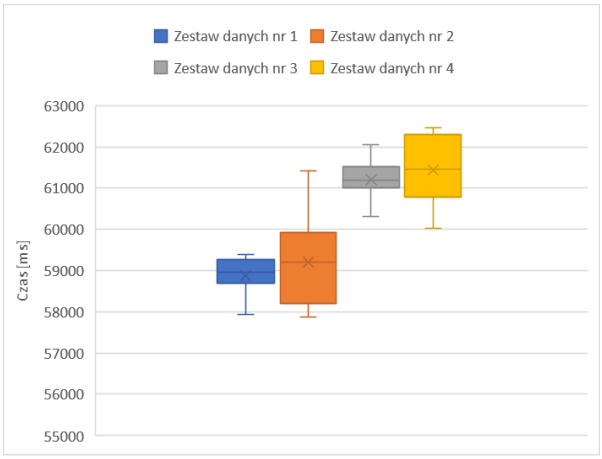


Rysunek 9: Średnie czasy wykonania scenariusza trzeciego.

Wykres zaprezentowany na Rysunku 10 przedstawia rozkład wyników badania na podstawie scenariusza czwartego dla technologii Microsoft SQL Server. Wykres przedstawiony na Rysunku 11 przedstawia rozkład wyników badania na podstawie scenariusza czwartego dla technologii Apache Hadoop.

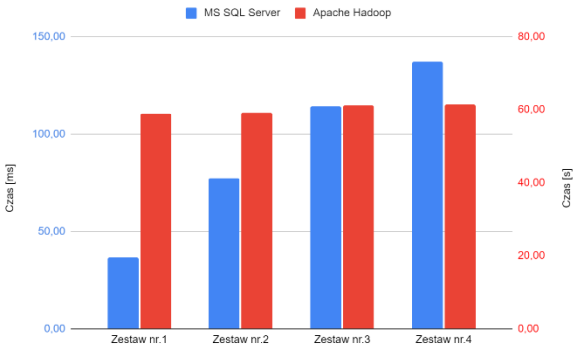


Rysunek 10: Wykres pudełkowy dla scenariusza czwartego dla SQL Server.



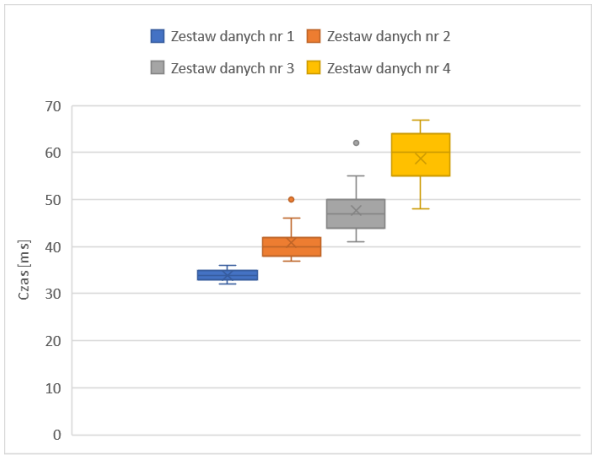
Rysunek 11: Wykres pudełkowy dla scenariusza czwartego dla Apache Hadoop.

Rysunek 12 przedstawia wykres prezentujący średnie czasy wykonania scenariusza czwartego na relacyjnej bazie danych SQL Server oraz technologii Apache Hadoop.

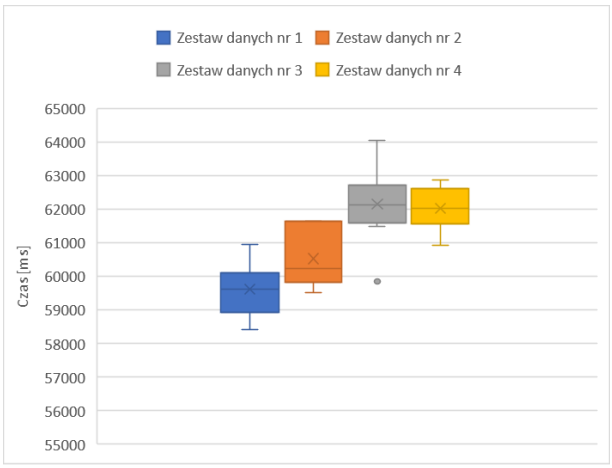


Rysunek 12: Średnie czasy wykonania scenariusza czwartego.

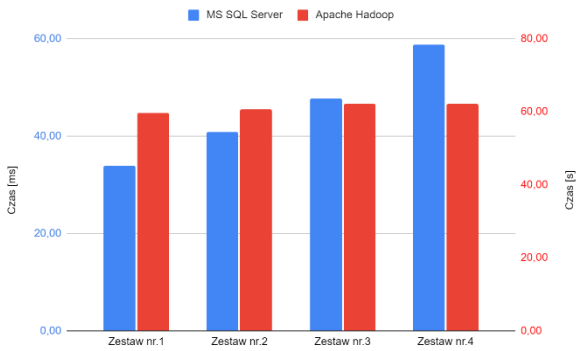
Wykres na Rysunku 13 przedstawia rozkład wyników badania na podstawie scenariusza czwartego dla technologii Microsoft SQL Server. Wykres na Rysunku 14 przedstawia rozkład wyników badania na podstawie scenariusza czwartego dla technologii Apache Hadoop Rysunek 15 przedstawia wykres prezentujący średnie czasy wykonania zapytania zdefiniowanego w scenariuszu piątym.



Rysunek 13: Wykres pudełkowy dla scenariusza piątego dla SQL Server.



Rysunek 14: Wykres pudełkowy dla scenariusza piątego dla Apache Hadoop.



Rysunek 15: Średnie czasy wykonania scenariusza piątego.

7. Wnioski

Na podstawie wyników uzyskanych ze scenariusza pierwszego można wywnioskować, że średni czas wykonania zapytania rośnie wraz z liczbą rekordów w zestawie danych dla obu rozwiązań. SQL Server był znacząco szybszy, ponieważ czasy wykonania zostały wyrażone w milisekundach, a nie w sekundach, jak w przypadku Apache Hadoop. W SQL Server średni

czas wykonania zapytania na najmniejszym wolumenie danych był ponad 4-krotnie mniejszy od czasu wykonania zapytania na największym wolumenie danych. Natomiast w przypadku Apache Hadoop wzrost czasu wykonania zapytania pomiędzy pierwszym a ostatnim zestawem danych wyniósł 14%, co jest znacznie lepszym rezultatem niż ten uzyskany przez rozwiązanie firmy Microsoft. Na podstawie wyników dla scenariusza pierwszego, można zauważyć, że dla zestawów danych nr 1 do nr 4, czas przetwarzania w Apache Hadoop jest znacznie dłuższy niż w MS SQL Server. Zatem, w kontekście tego scenariusza, MS SQL Server wykazuje wyższą wydajność i szybkość przetwarzania niż Apache Hadoop.

Wyniki uzyskane po wykonaniu scenariusza drugiego pokazują, iż w przypadku SQL Server, procentowa zależność między wzrostem liczby rekordów a średnim czasem wykonania zapytania prezentuje się następująco: dla zestawów danych 1 i 2 wynosi 48,06%, dla zestawów danych 2 i 3 wynosi 4,59%, natomiast dla zestawów danych 3 i 4 wynosi 12,22%. Te same zależności w przypadku Apache Hadoop prezentują się następująco: Zestaw 1 do zestaw 2: -1,8%, zestaw 2 do zestaw 3: 6,9% oraz zestaw 3 do zestaw 4: 0,5%. Wyniki pokazują, że procentowa różnica czasowa między zestawem 1 i 2 jest niewielka, natomiast różnica między zestawami 2 i 3 jest znaczna (6,9%). Różnica między zestawami 3 i 4 jest minimalna (0,5%). Takie wyniki pokazują, że Apache Hadoop jest mniej podatny na wzrost wolumenu danych niż SQL Server. Wyniki pokazują, że średni czas wykonania zapytania w MS SQL Server jest znacznie krótszy, dla wszystkich zestawów danych, niż w Apache Hadoop. Różnice pomiędzy średnimi czasami wykonania są bardzo duże.

Na podstawie wyników otrzymanych po wykonaniu scenariusza czwartego dla SQL Server, widać, że procentowy wzrost czasu wykonania zapytania między kolejnymi zestawami danych wynosi: zestaw 1 do zestaw 2: 29,9%, zestaw 2 do zestaw 3: 10,7% oraz zestaw 3 do zestaw 4: 4,8%. Natomiast w przypadku Apache Hadoop można zaobserwować, że różnice czasowe w średnim czasie wykonania wynoszą: zestaw 1 do zestaw 2: 3.85%, zestaw 2 do zestaw 3: 2,82% oraz zestaw 3 do zestaw 4: 2,14%. Wyniki pokazują, że średni czas wykonania zapytania ze scenariusza trzeciego w MS SQL Server jest trzy rzędy wielkości krótszy od średniego czasu wykonania w Apache Hadoop. Różnica jest zachowana dla każdego zbioru danych, co wskazuje, że SQL Server jest rozwiązaniem wydajniejszym pod względem czasu wykonania zapytania.

Na podstawie wyników otrzymanych po wykonaniu scenariusza czwartego, można zaobserwować, że średni czas wykonania zapytania w relacyjnej bazie danych SQL Server jest znacznie krótszy niż w Apache Hadoop. Ta różnica utrzymuje się dla każdego zestawu danych, co sugeruje, że SQL Server jest bardziej wydajnym rozwiązaniem pod względem czasu wykonania zapytania.

Na podstawie wyników z badania scenariusza piątego można zobaczyć, iż procentowa różnica między średnim czasem wykonania zapytania dla obu technologii wynosi kolejno: 175763,7%, 147888,5%, 130214,4% oraz 105555,9%. Różnice są bardzo duże, jednak zauważalna jest tendencja spadkowa.

Rozwiązanie SQL Server charakteryzowało się dużo mniejszym wpływem wielkości wolumenu danych na średni czas wykonania zapytania. Dodatkową cechą tej technologii były mniejsze wahania czasów wykonania między próbami. Jednak, jeśli chodzi o główny czynnik, czyli czas wykonania zapytania, to rozwiązanie SQL Server uzyskało znacznie korzystniejsze rezultaty.

Literatura

[1] P. O. Queiroz-Sousa, A. C. Salgado, A review on OLAP technologies applied to information networks, ACM Transactions on Knowledge Discovery from Data (TKDD) 14(1) (2019) 1–25, https://doi.org/10.1145/3370912.

[2] S. Sagiroglu, D. Sinanc, Big data: A review, Proceedings of the 2013 international conference on collaboration technologies and systems (CTS), IEEE, San Diego, CA, USA (2013) 42–47, https://doi.org/10.1109/CTS.2013.6567202.

[3] S. Chaudhuri, U. Dayal, An overview of data warehousing and OLAP technology, ACM Sigmod record 26(1) (1997) 65–74, https://doi.org/10.1145/248603.248616.

[4] J. Song, C. Guo, Z. Wang, Y. Zhang, G. Yu, J. M. Pierson, HaoLap: A Hadoop based OLAP system for big data, Journal of Systems and Software 102 (2015) 167–181, https://doi.org/10.1016/j.jss.2014.09.024.

[5] R. Stanek, Microsoft SQL Server. Brno: Computer Press, 2013. [12.02.2023]

[6] R. Kumar, B. B. Parashar, S. Gupta, Y. Sharma, N. Gupta, Apache hadoop, nosql and newsql solutions of big data, International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE) 1(6) (2014) 28–36, https://doi.org/10.13140/2.1.3454.9444.

[7] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, Proceedings of the 2010 IEEE 26th symposium on mass storage systems and technologies (MSST), Ieee, Incline Village, NV, USA (2010) 1–10, https://doi.org/10.1109/MSST.2010.5496972

[8] J. Dittrich, J. A. Quiané-Ruiz, Efficient big data processing in Hadoop MapReduce, Proceedings of the VLDB Endowment 5(12) (2012) 2014-2015, https://doi.org/10.14778/2367502.2367562.

[9] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, E. Baldeschwieler, Apache hadoop yarn: Yet another resource negotiator., Proceedings of the 4th annual Symposium on Cloud Computing, Santa Clara California (2013) 1–16, https://doi.org/10.1145/2523616.2523633