

Porównanie wydajności platform integracyjnych

Bartłomiej Karol Flis*, Łukasz Kołyga, Maria Skublewska-Paszkowska

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule zostały porównane dwie nowoczesne platformy integracyjne stosowane w środowiskach produkcyjnych, jak i testowych. Jest to rozwiązanie darmowe -szyna danych WSO2 oraz szyna OSB, czyli Oracle Service Bus, która jest rozwiązaniem korporacyjnym. Zostały omówione testy, które były przeprowadzone na wyżej wymienionych szynach oraz uzyskane wyniki. Dodatkowo zostało dołączone porównanie wyników wraz z wnioskami, które zostały wyciągnięte po przeprowadzonych testach.

Słowa kluczowe: OSB; WSO2; Integracja; Enterprise

*Autor do korespondencji. Bartlomiej.Flis1@gmail.com

Comparing the performance of integration platforms

Bartłomiej Karol Flis*, Łukasz Kołyga, Maria Skublewska-Paszkowska

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

Abstract. The article compares two modern integration platforms used in production and test environments. These are, a free integrity solution WSO2 data bus and OSB bus, called Oracle Service Bus, which is a commercial software. The tests that were carried out on the above-mentioned buses and results were discussed. In addition, a comparison of results with conclusions that were drawn after the tests was attached.

Keywords: OSB; WSO2; Integrity; Enterprise

*Corresponding author. Bartlomiej.Flis1@gmail.com

1.

Wstęp

Wraz z informatyzacją kolejnych dziedzin życia, powstała potrzeba wymiany informacji pomiędzy kolejnymi firmami, które zdecydowały się na takie kroki. Potrzeba ciągłej integracji skłoniła większość producentów do stworzenia produktów i standardów, które będą umożliwiały wydajną implementację tych zagadnień.

Pojęcie integracji systemów informatycznych można dostrzec na wielu płaszczyznach życia. Dzięki użyciu platform integracyjnych klienci banków mogą wykonywać dowolną liczbę przelewów w bardzo krótkim czasie. Dzieje się tak dlatego, że instytucje finansowe komunikują się ze sobą i przekazują informacje o wykonanych operacjach. Wraz z rozbudową funkcjonalności, liczba wymienianych komunikatów pomiędzy systemami dziedzinowymi rośnie. Dlatego pojęcie wydajności w zagadnieniu integracji jest tak ważne. Natomiast, zagadnienie integracji może być postrzegane jako komunikacja pomiędzy aplikacjami na poziomie rozumienia przesyłanych bitów danych.

Przez długi czas firma Oracle posiadała jedno z bardziej konkurencyjnych rozwiązań integracyjnych, a jej produkt Oracle Service Bus był stosowany w wielu korporacjach z różnych branż. Rok 2005 był bardzo przełomowy, został wypuszczony produkt o nazwie WSO2[6]. W dość szybkim tempie objął znaczną część rynku prezentując rozwiązanie całkowicie darmowe.

Celem artykułu jest przeprowadzenie analizy wydajnościowej omawianych platform. Wydajność jest rozumiana jako szybkość przepływu informacji w stosunku do pracy, jaką musi wykonać platforma. Badania zostały przeprowadzone w środowisku przygotowanym w oparciu o maszyny wirtualne. Celem pośrednim było zaimplementowanie platformy do testowania wydajności tych dwóch platform.

2.

Przygo-

towanie do przeprowadzenia testów wydajnościowych

W celu przeprowadzenia testów wydajnościowych mających za zadanie zbadać i porównać wydajność platform integracyjnych zostało przygotowane środowisko testowe w oparciu o maszyny wirtualne. W tym przypadku jako hipernadzorca był wykorzystany komputer klasy PC. Na maszynie zostało zainstalowane oprogramowanie VMware ESXi 5.5. Środowisko posiadało następujące zasoby:

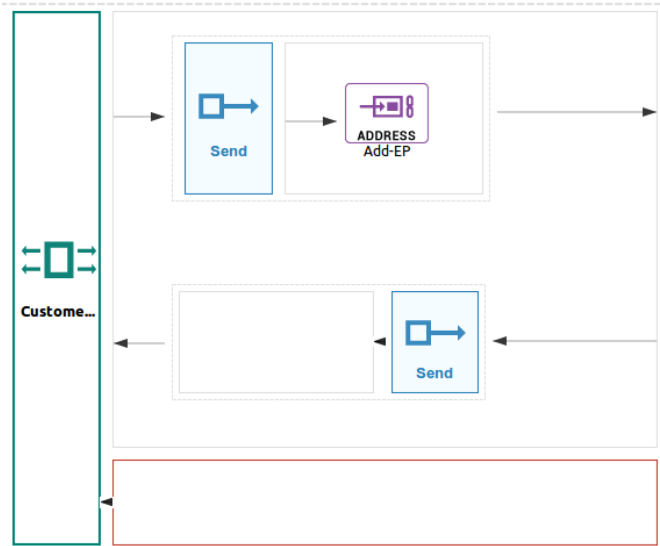
- CPU: Intel(R) Core(TM) i7-4790K CPU @ 4.00 GHz;
- RAM: 2x8GB DDR3 1600 MHz;
- Płyta główna: Asus Z97X Gaming 3;
- HDD: WD Green 1TB 7200 RPM;
- SDD: Samsung EVO 540 120 GB;
- Network: Gigabit Ethernet.

W środowisku testowym zostały utworzone dwie maszyny wirtualne o takich samych parametrach, dla każdej z platform. Trzecia maszyna natomiast została utworzona w celu przeprowadzenia testu obciążeniowego za pomocą oprogramowania SoapUI.

2.1. Scenariusze testowe

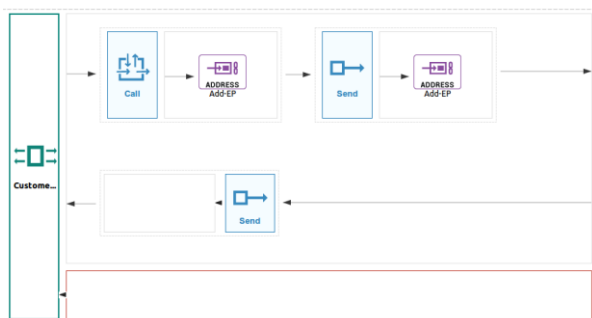
W celu przeprowadzenia miarodajnych testów na każdej z platform zostały zaimplementowane trzy usługi. Każda z usług charakteryzowała się innym przebiegiem wiadomości by sprawdzić zachowanie szyn w różnych warunkach[1]:

- scenariusz usługi prostej – usługa jest odpowiedzialna za komunikację z systemem dziedzicznym, ciało komunikatu podczas wymiany nie ulega transformacji. Tego typu scenariusz pozwoli określić, jak wydajne są transporty obu rozwiązań. Implementacja pokazana jest na Rys. 1.



Rys. 1. Usługa prosta

- scenariusz usługi wzbogacania wiadomości – celem usługi jest wzbogacenie wiadomości, która ma trafić do systemu dziedzicznego[2]. Jako przykład przypadku użycia można podać dodawanie punktów lojalnościowych przy wykonywaniu transakcji przez klienta. Schemat usługi można zobaczyć na Rys. 2. Na scenariusz składa się:
 - odbiór komunikatu SOAP;
 - nawiązanie połączenia oraz wysłanie wiadomości do serwisu, który jest odpytywany o dodatkowe informacje;
 - wysłanie komunikatu do systemu dziedzicznego;
 - odbiór komunikatu od systemu dziedzicznego i dostarczenie go do klienta usługi.

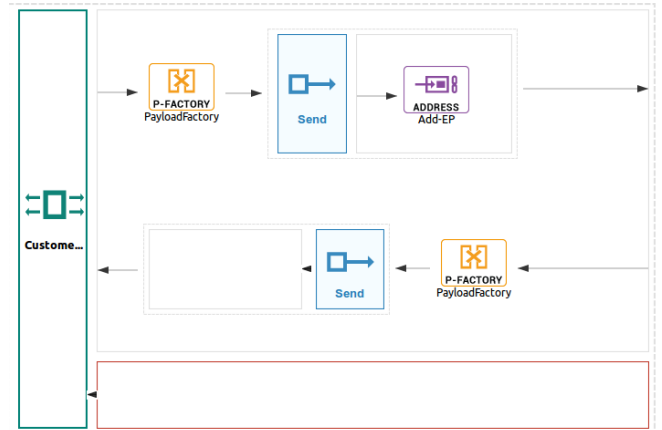


Rys. 2. Usługa wzbogacania wiadomości

- scenariusz usługi transformacji (XQuery) – usługa za pomocą której można zmienić strukturę wiadomości XML i dostosować ją

do wymagań usługi zewnętrznej. Usługa została przedstawiona na Rys. 3. Struktura scenariusza jest następująca:

- odebranie komunikatu od klienta SOAP;
- wywołanie transformacji XQuery komunikatu oraz nawiązanie połączenia z systemem zewnętrznym;
- wysyłanie i odebranie komunikatów z systemu zewnętrznego;
- wywołanie odpowiedzi oraz przekazanie wiadomości do klienta SOAP.



Rys. 2. Usługa transformacji

2.2. Struktura wiadomości

Każda z usług posiada taką samą definicję wiadomości, czyli tak zwany znacznik message (ang. wiadomość), który definiuje z jakich składowych owe wiadomości się składają przy komunikacji SOAP. Na listingu 1 można zauważyć, że oprócz elementów wiadomości definiuje się również ich nazwy do późniejszego korzystania. Można zauważyć również, że definiuje się porttype, który nazywa zestaw komunikatów.

Przykład 1. Zastosowana wiadomość Message

```
<wsdl:message name="PackageRegistrationRequest">
  <wsdl:part name="parameters" element="tns:PackageRegistration"/>
</wsdl:message><wsdl:message name="PackageRegistrationResponse">
  <wsdl:part name="parameters"
    element="tns:PackageRegistrationResponse"/>
</wsdl:message><wsdl:portType
  name="CustomerServiceSimplePortType"><wsdl:operation
  name="PackageRegistration"><wsdl:input
  message="tns:PackageRegistrationRequest"
  wsaw:Action="http://www.example.org/PackageService/PackageRegistrati
  on"/><wsdl:output message="tns:PackageRegistrationResponse"
  wsaw:Action="http://www.example.org/PackageService/PackageService/Pa
  ckageRegistrationResponse"/></wsdl:operation></wsdl:portType></pre>
```

3. Metody ka przeprowadzenia testów

Przeprowadzenie testów wydajnościowych obejmuje przetestowanie scenariuszy zawartych w punkcie 2.1. Testowanie przebiegło w kolejności sekwencyjnej, czyli jeden po drugim, używając tego samego środowiska. Podczas testów na maszynie nie były prowadzone inne akcje oraz nie były wysyłane wiadomości na inne usługi. Każda z wymienionych usług posiada unikatowy adres, pod którym jest osiągalna.

Testy oraz ich parametry zostały uwarunkowane w sposób następujący:

- czas trwania testu wynosił 5 minut;
- wielkość komunikatu wynosiła 15 KB, 500 KB;
- liczba wątków wysyłających kolejno 3, 15;
- kolejność wysyłania zachowana sekwencyjnie.

Testy wydajnościowe zostały wykonane w projekcie programu utworzonego w oprogramowaniu SoapUI. Wewnątrz projektu zostały umieszczone treści komunikatów, które były wysyłane do usług. Dodatkowo, oprogramowanie umożliwiało wygodne i szybkie sterowanie wartościami testu, takimi jak: czas trwania testu, liczba wątków, czy czas trwania wykonania testu. Do sterowania rozmiarem komunikatu zastosowano wartość pola „City” w dokumencie XML [3]. Zmniejszając i zwiększając długość tekstu, który był w wartości pola, regulowany był rozmiar wysyłanego zapytania.

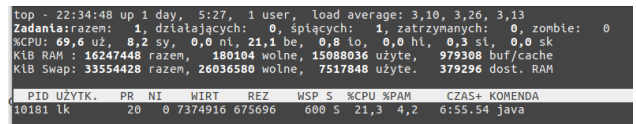
Oracle Service Bus jak i WSO2EI to oprogramowanie oparte o technologie Java, uruchamiane są więc na maszynie wirtualnej Java (JVM – Java Virtual Machine). W związku ze specyfiką maszyny wirtualnej Java[4], przed rozpoczęciem testu, usługi były obciążane przez kilku minut. Zapobiegało to występowaniu anomalii w wynikach testu. Szyny integracyjne to bardzo skomplikowane oprogramowanie i stosuje się w nich wiele technik optymalizacji. Jednym z takich zabiegów może być mechanizm ”Codecache” [5] na maszynie JVM. W tym przypadku pewna czynność za pierwszym razem może trwać dłużej, lecz za każdym kolejnym wykonaniem czas wykonania tej czynności powinien być krótszy.

4. Wyniki przeprowadzonych testów wydajnościowych

Po zrealizowaniu testów uzyskane wyniki zostały przedstawione w postaci tabel dla poszczególnych pomiarów i danej szyny. Każda z usług miała cztery pomiary:

- pierwszy pomiar:
 - liczba wątków wysyłających: 3;
 - wielkość komunikatu: 5kb.
- drugi pomiar:
 - liczba wątków wysyłających: 15;
 - wielkość komunikatu: 5kb.
- trzeci pomiar:
 - liczba wątków wysyłających: 3;
 - wielkość komunikatu: 500kb.
- czwarty pomiar:
 - liczba wątków wysyłających: 15;
 - wielkość komunikatu: 500kb.

Podczas wykonywania testów sprawdzany był stan pamięci RAM oraz użycia procesora. W przypadku WSO2 Enterprise Service Bus poziom zużycia procesora był na poziomie 8% - 14% oraz RAM był na poziomie od 500 mb do 950 mb, co można zobaczyć na Rys 5.



Rys. 5. Pomiar użycia CPU oraz RAM

Parametry systemu operacyjnego wahały się w podanych granicach. Dla Oracle Service Bus miały kolejno 10% - 16% użycia CPU, zaś RAM był na poziomie 1300 mb – 1500 mb.

Wyniki testów dla szyny WSO2 zaprezentowana w Tabeli 1.

Tabela 1. Wyniki testów wydajnościowych przeprowadzone na szynie WSO2

Usługa prosta				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości [s]	Suma przesłanych wiadomości
3	5kb	654,61	3,98	196383
15	5kb	969,60	14,89	290880
3	500kb	49,68	59,69	14904
15	500kb	75,59	195,56	22677

Tabela 2. Wyniki testów wydajnościowych dla usług wzbogacania na szynie WSO2

Usługa wzbogacania wiadomości				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości [s]	Suma przesłanych wiadomości
3	5kb	360,03	5,33s	108010
15	5kb	620,54	20,39s	186163
3	500kb	34,77	84,16s	10432
15	500kb	49,13	283,56s	14740

Tabela 3. Wyniki testów wydajnościowych dla usługi transformacji na szynie WSO2

Usługa transformacji				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości [s]	Suma przesłanych wiadomości
3	5kb	510,59	5,13s	153178
15	5kb	688,41	18,31s	206524
3	500kb	38,75	68,64s	11625
15	500kb	60,47	228,80s	18141

Dla porównania zostały przedstawione tabele (4-6) z wynikami dla Oracle Service Bus, na której zostały wykonane takie same testy wydajnościowe.

Tabela 4. Wyniki testów wydajnościowych przeprowadzone na szynie Oracle Service Bus

Usługa prosta				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości [s]	Suma przesłanych wiadomości
3	5kb	534,78	4,51s	160434
15	5kb	819,86	17,98s	245958
3	500kb	58,68	39,12s	17604
15	500kb	94,59	183,61s	28377

Tabela 5. Wyniki testów wydajnościowych dla usług wzbogacania na szynie Oracle service Bus

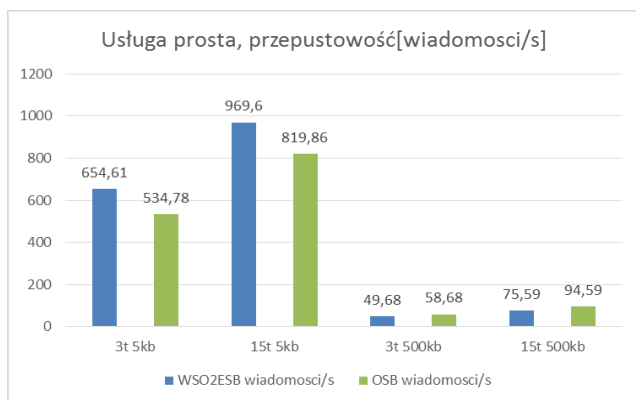
Usługa wzbogacania wiadomości				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości [s]	Suma przesłanych wiadomości
3	5kb	374,34	6,44s	112303
15	5kb	508,31	24,45s	152493
3	500kb	39,31	52,02s	11794

Tabela 6. Wyniki testów wydajnościowych dla usługi transformacji na szynie Oracle Service Bus

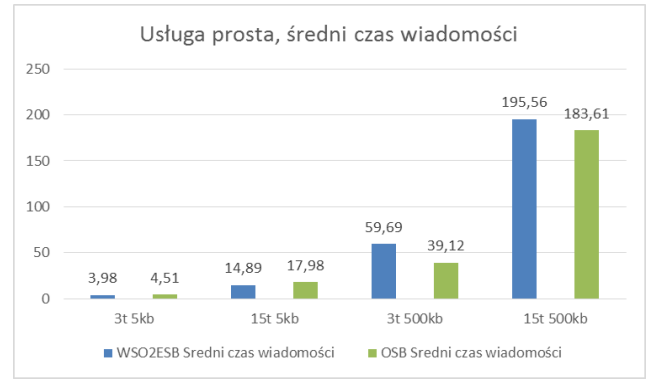
Usługa transformacji				
Liczba wątków wysyłających	Rozmiar	Przepustowość [wiadomości/s]	Średni czas wiadomości	Suma przesłanych wiadomości
3	5kb	390,38	5,41s	117116
15	5kb	590,29	21,93s	177089
3	500kb	48,70	44,98s	14611
15	500kb	69,99	216,65s	20998

5. Analiza otrzymanych wyników

Na podstawie wyników z testów wydajnościowych zostały stworzone wykresy. Na Rys. 6 przedstawiona jest liczba wysłanych wiadomości na sekundę, dla usługi prostej dla poszczególnej liczby wątków oraz rozmiaru komunikatu. Na tym wykresie można dostrzec, że przepustowość szyny WSO2 dla mniejszych komunikatów jest większa niż dla szyny OSB. Natomiast w przypadku komunikatów o większym rozmiarze, szyna WSO2 radziła sobie gorzej. Na kolejnym Rys. 7 został przedstawiony średni czas przetwarzania komunikatów. Na tym wykresie również widać, że szyna ESB znacząco lepiej radzi sobie z komunikatami o mniejszym rozmiarze, lecz przy obsłudze komunikatów, których rozmiar sięga 500kb czas obsługi komunikatu spada. Dla większych komunikatów oprogramowanie z firmy Oracle wypada lepiej.

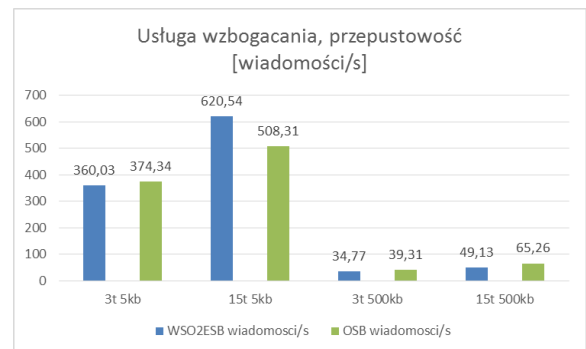


Rys. 6. Liczba przesłanych wiadomości na sekundę dla usługi prostej

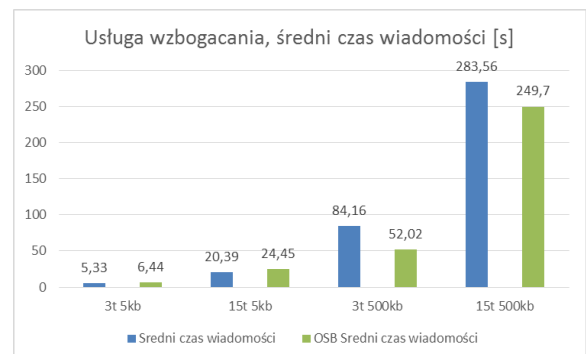


Rys. 7. Średni czas przetwarzania komunikatów w sekundach.

Kolejnym elementem, który był porównywany, była praca usługi wzbogacania. Wyniki zostały przedstawione na Rys. 7 i Rys. 8. W tym przypadku w porównaniu z prostą usługą, była znacznie wolniejsza. Spowodowane jest to specyfiką obsługi takiego wywołania. Na wywołanie takiej usługi składa się wykonanie dwóch zapytań do systemów dziedzicznych. Podczas pierwszego wywołania systemu dziedzicznego wątek, który obsługuje całą usługę jest zablokowany i oczekuje na jego zakończenie. W tym przypadku również szyna Oracle Service Bus lepiej radziła sobie z obsługą większych komunikatów zarówno w przypadku 3 i 15 wątków wysyłających.



Rys. 8. Liczba wiadomości na sekundę dla usługi wzbogacania



Rys. 9. Średni czas przetwarzania komunikatu dla usługi wzbogacania

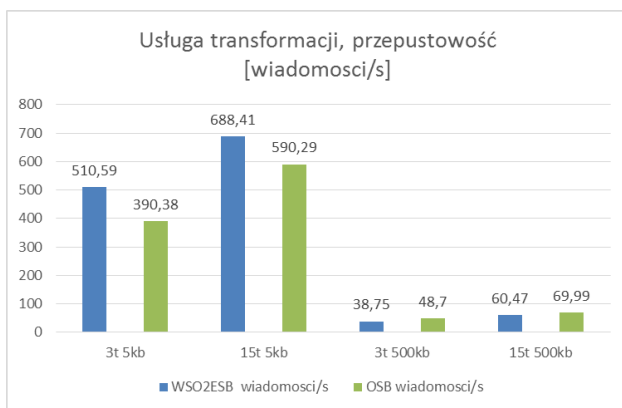
Ostatnią testowaną usługą była ta, której wyniki pokazano na Rys. 10 i Rys. 11. Usługa ta była szybsza od usługi wzbogacania lecz wolniejsza od usługi prostej. Realizacja tej usługi polega na wywołaniu silnika XQuery wewnątrz szyny. Za wywołanie transformacji odpowiada

osobny wątek. W tym przypadku również widać różnice w prędkościach przetwarzanych komunikatów. W tym przypadku, jak w dwóch wcześniejszych, szyna WSO2 radzi sobie lepiej z komunikatami o mniejszym rozmiarze, natomiast szyna Oracle Service Bus przetwarza z większą wydajnością wiadomości o dłuższej treści.

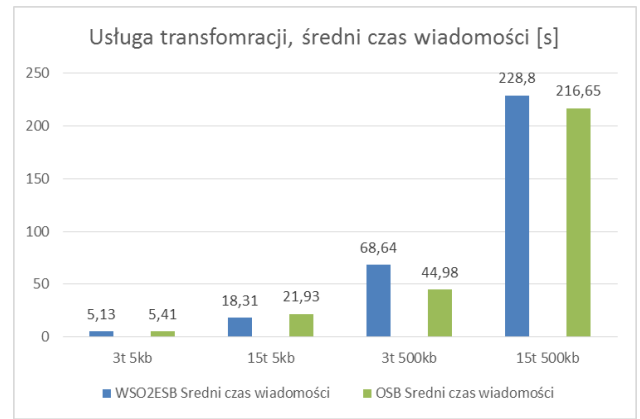
6. Wniosek i

Na podstawie implementacji usług, przeprowadzonych testów wydajnościowych i analizy wyników można wyciągnąć następujące wnioski:

- podejście oraz budowa usług na szynie WSO2 Enterprise Integrator oraz Oracle Service Bus są bardzo podobne. Posiadając podstawową wiedzę na temat usług w intuicyjny sposób można poruszać się po obu środowiskach IDE;
- zapotrzebowanie Oracle Service Bus na pamięć RAM jest większa niż w przypadku szyny WSO2 Enterprise Integrator;
- dla małych komunikatów rozwiązanie oparte o WSO2 Enterprise Integrator jest wydajniejsze niż Oracle Service Bus;
- zwiększenie liczby wątków wysyłających zwiększa liczbę przesłanych komunikatów w ciągu jednostki czasu, lecz wydłuża czas obsługi pojedynczego zapytania;
- w przypadku transformacji komunikatu, wydajność spada w porównaniu usługi prostej, lecz nie jest to znacząca strata wydajności w porównaniu z zyskiem funkcjonalności, którą daje transformacja komunikatu;
- najwolniejszym typem usługi jest usługa wzbogacająca, gdyż wiąże to się z dodatkową alokacją wątku na czas obsługi komunikatu;
- transformacje XQuery jest dostępna zarówno w przypadku WSO2 Enterprise Integrator jak i Oracle Service Bus.



Rys. 10. Liczba wiadomości na sekundę dla usługi transformacji



Rys. 11. Średni czas przetwarzania komunikatu dla usługi transformacji

7. Podsumowanie

Cel pośredni, czyli budowa platformy do testowania oraz cel bezpośredni porównanie wydajności tych dwóch platform zostały zrealizowane. Wyniki testów wydajnościowych dla wszystkich typów wykonywanych testów jednoznacznie wskazują na rozbieżności wydajności dla różnych wielkości komunikatów. Dla większych komunikatów wydajniejszym rozwiązaniem jest szyna Oracle Service Bus, w przypadku komunikatów o mniejszym rozmiarze większą przepustowością charakteryzowała się szyna WSO2 Enterprise Integrator.

Nie można jednoznacznie określić, która platforma integracyjna jest wydajniejsza, wybór rozwiązania zależy od kontekstu wykorzystania produktu oraz ekosystemu oprogramowania, jakie zastosowane jest w przedsiębiorstwie. Jeżeli w przedsiębiorstwie używane jest inne oprogramowanie Oracle np. Oracle Database lub Oracle Message Broker, w takim przypadku lepszym wyborem może okazać się OSB. Powodem tego jest bardzo dobra i wspierana integracja pomiędzy Oracle Service Bus, a brokerem lub bazą danych. Przykładem takiej zalety może być integracja z bazą danych, która umożliwi używanie kolejek bazodanowych na szynie. Ponadto posiadając wiele produktów Oracle można liczyć na obniżenie sumarycznej ceny za licencje. Szyna WSO2 Enterprise Integrator w porównaniu z Oracle Service Bus jest młodszym produktem. Jest dystrybuowany na licencji Open Source, dzięki czemu można jej używać bez ponoszenia dodatkowych kosztów licencyjnych. W przypadku, kiedy przedsiębiorstwo chciałoby korzystać ze wsparcia podczas implementacji lub utrzymania ma możliwość wykupienia oferowanego płatnego wsparcia firmy WSO2.

Oba rozwiązania świetnie sobie radzą z przetwarzaniem równoległym wielu wiadomości jednocześnie. Dzięki temu oba rozwiązania są w stanie obsługiwać jednocześnie bardzo wiele klientów korzystających z szyny. Ponadto dla zwiększenia wydajności jest możliwość zbudowania klastra, który zwiększy potencjał oraz możliwości platformy.

Dodatkowym plusem budowy klastra jest wysoka dostępność produkty. W przypadku, kiedy jedna z instancji zostanie uszkodzona, reszta klastra przejmuje jego rolę.

Literatura

- [1] J. Duckett, Beginning HTML, XHTML, CSS and JavaScript, Wiley Publishing Inc., New York (2010), 507 - 510
- [2] M. Havey, SOA Cookbook, Packt Publishing, New York(2008) 89-90,
- [3] XML, <https://en.wikipedia.org/wiki/XML>, [01.06.2018]
- [4] J. Duckett, Beginning HTML, XHTML, CSS and JavaScript, Wiley Publishing Inc., New York (2010), 507 - 510
- [5] Oracle Java SE Embedded: Developer's Guide, <https://docs.oracle.com/javase/8/embedded/develop-apps-platforms/codecache.htm>,: [01.06.2018r]
- [6] D. Chappell, Enterprise Service Bus: Theory in Practise, O'Reilly Media, New York(2010) 112 - 114