

ОСОБИНЕ БРОКЕРА ПОРУКА И ЊИХОВА ПРИМЕНА У ДИСТРИБУИРАНИМ СИСТЕМИМА

CHARACTERISTICS OF MESSAGE BROKERS AND THEIR APPLICATION IN DISTRIBUTED SYSTEMS

Драгана Јовић, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – Рад истражује и анализира изабране брокере порука, са фокусом на њихове кључне карактеристике као што су постојаност и очување редоследа порука. У раду се разматра како се ове особине примењују и имплементирају у дистрибуираним системима. Такође, рад се бави евалуацијом перформанси брокера порука.

Кључне речи: Брокер порука, Перформансе, Кашњење, Пропусност, Постојаност, Редослед порука, Евалуација

Abstract – The paper explores and analyzes selected message brokers, focusing on their key features such as durability and message ordering. The paper examines how these characteristics are applied and implemented in distributed systems. Additionally, the paper evaluates the performance of the message brokers.

Keywords: Message broker, Performance, Latency, Throughput, Durability, Messaging order, Evaluation

1. УВОД

Савремене софтверске апликације често зависе од удаљених услуга и података, што чини интеграцију кључном у дистрибуираним архитектурама. Асинхрона комуникација, реализована путем брокера порука, омогућава да задаци који захтевају више времена буду обрађени у позадини без блокирања захтева. Овакав приступ помаже у решавању изазова дистрибуираних система као што су непоуздане мреже и чврста повезаност [1]. У последње време, нови брокери порука фокусирају се на ниско кашњење и високе перформансе, што значајно утиче на ефикасност и поузданост система.

2. ОПИС РЕШАВАНОГ ПРОБЛЕМА

Рад решава проблем избора адекватног брокера порука за дистрибуирани финансијски систем, у коме су постојаност порука, очување редоследа и перформансе од суштинске важности. У ери када дистрибуирани системи играју кључну улогу у обради података у реалном времену, избор одговарајућег брокера порука постаје критичан фактор који утиче на укупне перформансе, скалабилност и поузданост система.

НАПОМЕНА:

Овај рад произтекао је из мастер рада чији ментор је био проф. др Александар Ердџан

Брокери порука представљају кључни комуникациони слој у дистрибуираним системима, омогућавајући безбедан пренос, постојаност података и отпорност на грешке. У системима са великим количинама података и критичним апликацијама, избор брокера директно утиче на ефикасност и способност система да одговори на различита оптерећења. У зависности од потреба, неки брокери су бољи за високу пропусност, док су други погоднији за обраду у реалном времену, где је кашњење кључно. Кашњење је посебно важно за апликације које захтевају брзу реакцију, попут финансијских или SCADA система.

Постојаност порука обезбеђује поуздано чување података и при кваровима, што је критично за системе где је губитак података неприхватљив. Редослед обраде је такође важан када је потребано очувати интегритет података.

Циљ рада је анализа постојаности и очувања редоследа порука код различитих брокера, како би се утврдило који најбоље одговара захтевима дистрибуираног финансијског система. Истраживање обухвата *Apache ActiveMQ*, *Artemis* и *Kafka*-у, кроз анализу њихових карактеристика, перформанси под оптерећењем и понашања у реалним сценаријима.

Студија случаја заснована је на *AdInsure* систему, који представља платформу за животна и неживотна осигурања, развијеном у C# и .NET 8. Постојећа инфраструктура за размену порука омогућава laku интеграцију са брокерима и поуздано тестирање. На основу резултата, рад предлаже брокер који најбоље испуњава критеријуме поузданости, постојаности и перформанси у систему са високим захтевима за скалабилност и ефикасност.

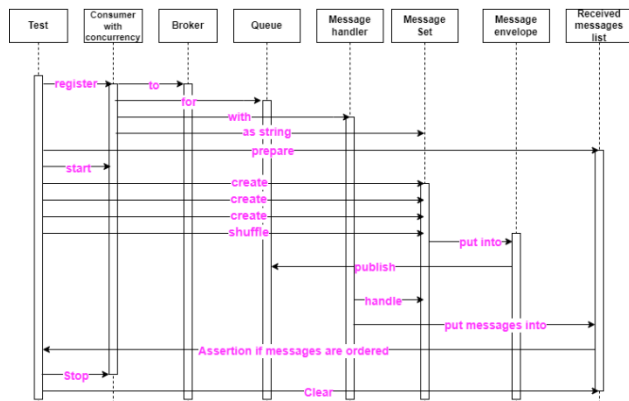
3. ЕВАЛУАТОР БРОКЕРА ПОРУКА

MessageBrokerEvaluator је .NET 8 конзолна апликација за тестирање постојаности и редоследа порука код различитих брокера. Користи инфраструктуру за размену порука *AdInsure* платформе и конфигурацију из *messagingSettings.json* датотеке, која дефинише магистралу, канале, редове и теме. Циљ је утврдити који брокер најбоље одговара потребама финансијских система попут *AdInsure*-а.

Тестирање се спроводи кроз интеграцијске тестове написане у C#-у и NUnit-у. Прослеђивање порука од произвођача до потрошача није посебно приказано, јер је то већ обезбеђено самим брокером.

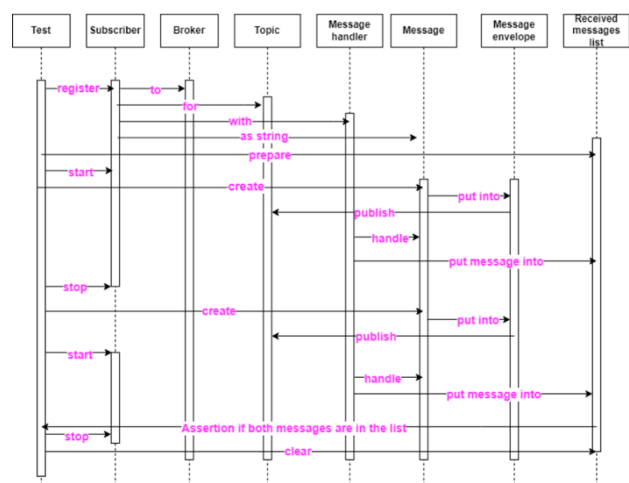
Тестови редоследа порука: Ови тестови проверавају да ли брокер може гарантовати очување редоследа

порука које се шаљу у ред или тему. Више порука се шаље у кратким временским интервалима, а тест осигурава да се поруке конзумирају редоследом којим су послате. Посебно је важно да се провери да ли је *FIFO* (енгл. *First-In-First-Out*) редослед очуван чак и у случају конкуренције међу конзументима. На слици 3.2 је приказан ток тестирања очувања редоследа порука у реду у облику секвенцијалног дијаграма.



Слика 3.2. Провера очувања редоследа порука у реду

Тестови постојаности: Ови тестови су осмишљени да провере да ли брокер подржава постојаност претплатника. Тестови симулирају сценарије где претплатник на тему губи везу или је привремено недоступан, па се након неког времена поново повезује. Проверава се да ли претплатник прима поруке које су послате док је био недоступан, што је кључно за финансијске системе где ниједна порука не сме бити изгубљена. На слици 1 је приказан ток тестирања постојаности претплатника на теми у облику секвенцијалног дијаграма.



Слика 1. Провера издржљивости претплатника на теми

3.1 Резултати евалуације *ActiveMQ* брокера

У евалуацији *ActiveMQ* брокера, инстанца је покренута као Docker контејнер коришћењем званичне слике *rmohr/activemq* са *Docker Hub*-а. За транспортни протокол коришћена је библиотека *Apache.NMS.ActiveMQ*.

ActiveMQ подржава очување редоследа порука, посебно у сценарију са једним конзументом по реду, као и кроз групе порука, где се очувава редослед унутар сваке групе путем идентификатора.

Што се тиче постојаности претплатника, у случају постојаних тема сваком претплатнику се чува његова копија поруке. То се постиже конфигурисањем *clientID* и имена претплатника у *JMS* конекцији, што омогућава да претплатник добије све поруке након поновног повезивања [2].

3.2 Резултати евалуације *Artemis* брокера

При евалуацији *Artemis* брокера, инстанца је покренута као Docker контејнер користећи званичну слику *apache/activemq-artemis*. Због подршке за *OpenWire* протокол, коришћена је иста библиотека *Apache.NMS.ActiveMQ*, без потребе за додатним изменама у коду.

За разлику од *ActiveMQ*, *Artemis* користи **Fully Qualified Queue Name (FQQN)** за управљање трајним претплатама и виртуелним темама. Уместо класичних назива, користи се структура као што је *VirtualTopic.{Тема}::Consumer.{Претплатник}.VirtualTopic.{Тема}*, што омогућава директан приступ редовима претплатника. Ово понашање се конфигурише параметром *virtualTopicConsumerWildcards* [2].

3.3 Резултати евалуације *Kafka* брокера

Kafka брокер је покренут као Docker контејнер користећи званичну *apache/kafka* слику, а за транспорт је коришћена библиотека *Confluent.Kafka*.

Редослед порука је загарантован само унутар једне партиције, што ограничава паралелизам. Ради очувања редоследа на вишем нивоу користе се приступи као што су **алгоритам агрегатора и сортирања, коришћење јединственог потрошача унутар групе, и примена протокола групног потврђивања и емитовања**. [3].

Постојаност потрошача у *Kafka*-и се остварује кроз **чување и комитовање позиције**, као и коришћењем **потрошачких група**. У *AdInsure* систему, потрошачи деле исти *group ID*, а поставка *AutoOffsetReset = Latest* омогућава наставак читања од последње обрађене поруке. Ручно комитовање позиције након обраде повећава поузданост у односу на аутоматско.

Резултати евалуације показују да сви брокери подржавају концепте очувања редоследа порука и постојаност претплатника. *Artemis* се истиче као једноставна замена за *ActiveMQ*, са идентичним протоколима и библиотекама, осим код виртуелних дестинација које захтевају мање измене конфигурације. У поређењу са *Kafka*-ом, *Artemis* и *ActiveMQ* нуде једноставније очување редоследа, док је у *Kafka*-и редослед ограничен на ниво партиција, што ограничава паралелизам. У финансијском дистрибуираном систему, где су редослед, постојаност и перформансе кључни, ови аспекти су важни за избор брокера. Не постоји универзални избор – прави избор зависи од специфичних приоритета система и захтева пословања.

4. ПЕРФОРМАНСЕ БРОКЕРА

Метрике перформанси играју важну улогу у откривању ефикасности и поузданости брокера порука, пружајући драгоцен увид у њихово оперативно понашање и потенцијал за скалабилност. **Пропусност** и **кашњење** су две основне метрике перформанси за брокере порука [4].

Пропусност указује на ефикасност система у обради порука, одражавајући његову способност да ефикасно управља великим количинама података. С друге стране, кашњење се односи на кашњење које једна порука доживљава док пролази кроз систем, наглашавајући брзину и одзив у испоруци порука.

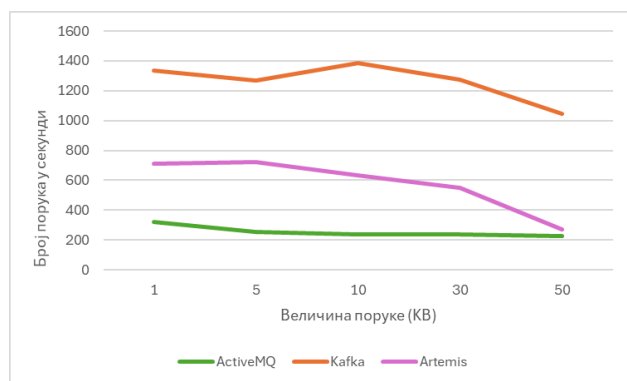
За тестирање перформанси брокера порука *ActiveMQ*, *Artemis* и *Kafka* коришћено је локално окружење, које омогућава потпуну контролу над хардверским ресурсима и елиминисање мрежних варијација. Спецификација хардвера на којем су тестиране перформансе су: *Intel i7* процесор, *32 GB RAM*, *500 GB SSD*.

4.1 Пропусност

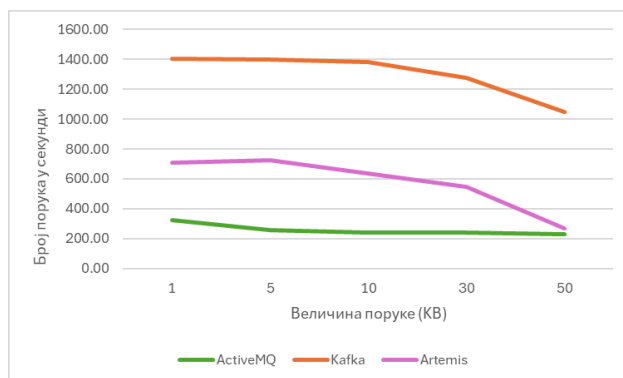
Пропусност порука означава број порука које се могу обрадити у просеку по секунди за N број порука. Овај показатељ мери колико порука може бити обрађено у одређеном временском оквиру. Израчунава се као:

$$\text{пропусност} = \frac{\text{укупан број обрађених порука}}{\text{укупно време потребно за обраду порука}}$$

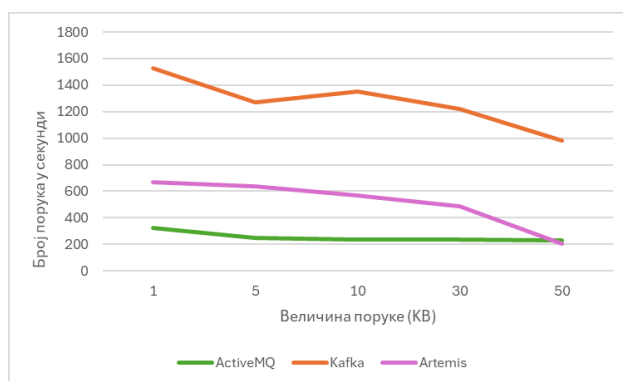
Програм за мерење пропусности развијен је у *C#* као конзолна апликација и симулира паралелно слање порука преко више произвођача. Тестирају се различити сценарији са бројем произвођача (16, 24, 32) и величином порука (1–50 KB), у трајању од 60 секунди. Бележе се укупан број послатих порука и пропусност (поруке/секунди). Апликација користи *AdInsure* инфраструктуру и транспортне протоколе у складу са тестираним брокером. Резултати су приказани на сликама 2, 3 и 4.



Слика 2. Резултати пропусности брокера за 16 паралелних произвођача у зависности од величине порука (од 1 до 50), где је величине порука представљена у килобајтима (kB)



Слика 3. Резултати пропусности брокера за 24 паралелна произвођача у зависности од величине порука (од 1 до 50), где је величине порука представљена у килобајтима (kB)



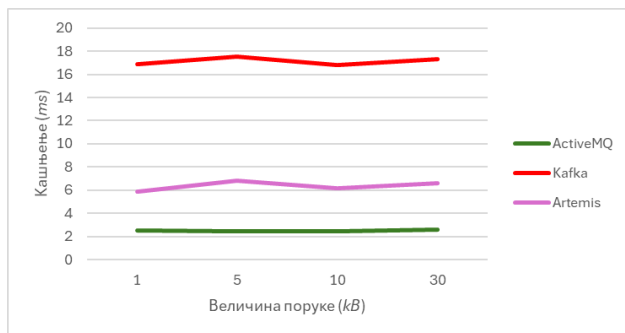
Слика 4. Резултати пропусности брокера за 32 паралелна произвођача у зависности од величине порука (од 1 до 50), где је величине порука представљена у килобајтима (kB)

4.2. Кашњење

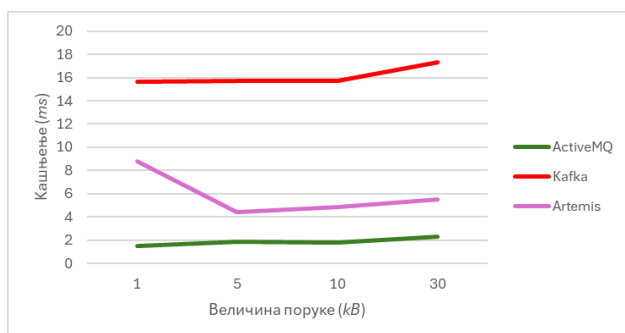
Просечно кашњење представља просечно време кашњења за све поруке у скупу од N порука. Израчунава се као:

$$\text{просечно кашњење} = \frac{1}{N} \sum_{n=1}^N \text{кашњење}(n)$$

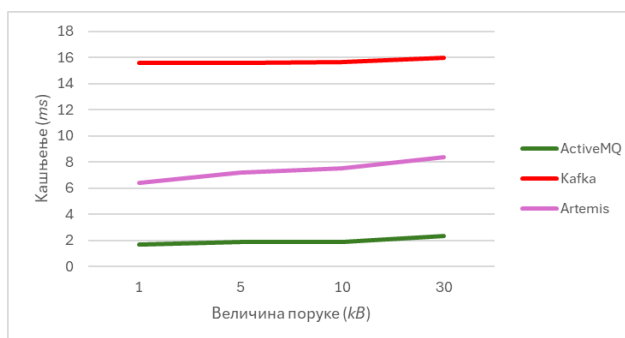
Кашњење се мери у склопу програма развијеног у *C#* конзолној апликацији. Тестирање кашњења подразумева следеће кораке: регистрација потрошача на канал за пријем порука који бележи време пријема, слање порука од стране произвођача, који за сваку поруку бележи време слања и израчунавање разлике између времена слања и пријема порука. На крају, програм израчунава просечно кашњење свих порука и приказује резултате. Тестирање је спроведено за *ActiveMQ*, *Artemis* и *Kafka* брокере, сви са инстанцама на Докеру. Резултати су дати у наставку, на сликама 5, 6, и 7 и графички илуструју кашњење у зависности од броја послатих порука.



Слика 5. Резултати кашњења брокера у милисекундама (ms) за 1000 послатих порука у зависности од величине порука (1-30) у килобајтима (kB)



Слика 6. Резултати кашњења брокера у милисекундама (ms) за 10000 послатих порука у зависности од величине порука (1-30) у килобајтима (kB)



Слика 7. Резултати кашњења брокера у милисекундама (ms) за 50000 послатих порука у зависности од величине порука (1-30) у килобајтима (kB)

На основу спроведених тестова перформанси, утврђено је да сваки од анализираних брокера – *ActiveMQ*, *Artemis* и *Kafka* – има своје специфичне предности и мане у зависности од сценарија употребе. *ActiveMQ* се истиче најнижим кашњењем, што га чини погодним за апликације које захтевају брзу обраду порука, али његова пропусност опада при повећању оптерећења. *Artemis* нуди баланс између пропусности и кашњења, што га чини добрим избором за системе који захтевају компромис између брзине испоруке и обима порука. С друге стране, *Kafka* доминира у

пропусности, нарочито у сценаријима са великим бројем произвођача, али уз жртвовање кашњења, што је чини идеалном за обраду великих количина података где брзина испоруке није критична.

5. ЗАКЉУЧАК

Обезбеђивање постојаности и редоследа у дистрибуираном финансијском систему могуће је код свих разматраних брокера порука, али са различитим нивоима сложености и перформанси. *ActiveMQ* је најбољи избор за системе са ниским кашњењем и умереним оптерећењем, док *Artemis* пружа добар компромис између пропусности и кашњења. Оба брокера омогућавају релативно једноставну имплементацију очувања редоследа порука као и постојаности брокера. *Kafka*, иако најпогоднији за високу пропусност и скалабилност, захтева сложенију имплементацију за очување редоследа и постојаности и има веће кашњење. Избор брокера зависи од специфичних захтева апликације, укључујући критичне факторе као што су кашњење, пропусност и обим обраде података.

6. ЛИТЕРАТУРА

- [1] Magnoni L: "Modern messaging for distributed systems", J. Phys.: Conf. Ser. 608 012038, 2015
- [2] *ActiveMQ*, <https://activemq.apache.org/> - датум последњег приступа новембар 2024. године
- [3] Shashank Kumar, Aryan Jadon, Sachin Sharma: "Global Message Ordering using Distributed Kafka Clusters", 2023 15th International Conference on Innovations in Information Technology (IIT), 2023
- [4] John, Vineet, and Xia Liu. "A survey of distributed message broker queues." *arXiv preprint arXiv:1704.00411* (2017).

Кратка биографија:



Драгана Јовић рођена је у Новом Саду 1991. год. Основну школу „Петар Кочић“ завршила је у Темерину 2006. године. Гимназију „Светозар Милетић“ у Новом Саду завршила је 2010. године. Основне академске студије на Факултету техничких наука у Новом Саду, одсек Електротехника и рачунарство, смер Рачунарство и аутоматика, усмерење Примењене рачунарске науке и информатика, завршила је 2014. године. Мастер академске студије на Факултету техничких наука у Новом Саду, студијски програм Електроенергетски софтверски инжењеринг, уписала је 2014. године.

DOI: <https://doi.org/10.24867/34BE29Jovic>