**PROIECT**

**-** **Raport de analiză și evaluare comparativă -**

Kafka ,RabitMQ și ActiveMQ

**Proiect realizat de studentele**: Băldean Adela-Ștefania

Cusco Ana-Maria

**Profesor coordonator**: Ivan Cosmina

**An:** 4

**Grupa**: 30241

Cuprins

[1. Scop 3](#_Toc61599599)

2. Introducere 10

2.1 Kafka ......................................................................................................

2.2 RabbitMQ ...............................................................................................

2.3 ActiveMQ ...............................................................................................

3. Aspecte funcționale și de arhitectură 10

4. Aspecte de calitatea serviciului oferit ............................................................ . 12

5. Referinte ............................................................ . 12

# 1. Scop

Raport de analizăsi evaluare comparativăa tehnologiilor de mesagerie Kafka ,RabitMQ și ActiveMQ din perspectiva următoarelor cerințe: a) aspecte funcționale și de arhitectură: logica de rutare,garantarea livrării,ordonarea mesajelor,transmitere multicast, modul de stocare , suport tranzacțional,gradul de securizarec) aspecte decalitatea serviciuluioferit:disponibilitate, performanțăși scalabilitatedinamică. Deosebit de important pentru aplicații sunt latența și debitul, proprietăți ce diferențiază tehnologiilepentru diverse use-case-uri, si pentruacestea se va realiza o analiză comparativă.Deasemenea, se vor menționa și caracteristicile distinctiveale fiecărei tehnologii analizate

# 2. Introducere

2.1 Kafka

Apache Kafka este o platformă de streaming și de mesagerie distribuită, proiectată pentru a gestiona fluxuri masive de date în timp real. Kafka utilizează un model de publicare/abonare, în care producătorii trimit mesaje la topic-uri și consumatorii se abonează la aceste topic-uri pentru a primi mesajele relevante. Cu un accent puternic pe performanță și scalabilitate, Kafka poate gestiona volume mari de date și oferă o toleranță ridicată la întârzieri (latență redusă). De asemenea, Kafka garantează livrarea mesajelor "cel puțin o dată" și oferă capacitatea de a stoca mesajele pe termen lung într-un mod persistent. Este folosit în mod frecvent în aplicații care necesită procesare în timp real, analiză de date în flux sau sisteme de tranzacționare.

2.2 RabbitMQ

RabbitMQ este o platformă de mesagerie open-source, bazată pe protocoalele de mesagerie standard AMQP (Advanced Message Queuing Protocol). RabbitMQ utilizează un model de cozi de mesaje, în care producătorii trimit mesaje către cozi, iar consumatorii le preiau din cozi. Rutarea mesajelor se realizează prin intermediul schimburilor (exchanges), care direcționează mesajele în cozi în funcție de reguli de rutare definite. RabbitMQ oferă o flexibilitate mare în ceea ce privește rutarea și gestionarea mesajelor, permițând dezvoltatorilor să implementeze scenarii complexe de mesagerie. De asemenea, RabbitMQ oferă suport pentru confirmările de livrare și tranzacții, ceea ce facilitează asigurarea consistenței mesajelor în aplicații. RabbitMQ este utilizat în principal în aplicații de tip enterprise, în care este necesară o livrare sigură a mesajelor și o manipulare eficientă a volumelor mari de date.

2.3 ActiveMQ

Apache ActiveMQ este o platformă de mesagerie distribuită și open-source, bazată pe standardul JMS (Java Message Service). ActiveMQ utilizează, similar cu RabbitMQ, un model de cozi de mesaje și rutare prin intermediul schimburilor. Platforma oferă suport extins pentru tranzacții JMS, permițând tranzacționarea mesajelor și integrarea cu sistemele tranzacționale. ActiveMQ oferă diferite niveluri de garanție a livrării, inclusiv "cel puțin o dată" și "exact o dată", ceea ce îi permite să se adapteze la diferite cerințe de fiabilitate. De asemenea, ActiveMQ oferă suport nativ pentru retenția mesajelor pe termen lung și capacitatea de a persista mesajele în baze de date externe. ActiveMQ este utilizat în principal în aplicații Java și în mediul enterprise, în care interoperabilitatea și suportul tranzacțional sunt esențiale pentru asigurarea unei mesagerii fiabile și gestionarea complexă a fluxului de mesaje.

# 3. Aspecte funcționale și de arhitectură

**Logica de rutare**

**RabbitMQ**

**ActiveMQ**

**Kafka**

-utilizează, de asemenea, un model de cozi de mesaje similar cu RabbitMQ.

-Mesajele sunt publicate în cozi și consumatorii le preiau din cozi

-Rutarea mesajelor se realizează prin intermediul schimburilor și regulilor de rutare definite.

-utilizează un model de cozi de mesaje, în care producătorii trimit mesaje la cozi și consumatorii le preiau din cozi.

-Rutarea mesajelor se realizează prin intermediul schimburilor (exchanges), care direcționează mesajele în cozi în funcție de reguli de rutare definite.

-utilizează un model de publicare / abonare, în care producătorii trimit mesaje către unul sau mai mulți consumatori, numiți consumatori de tip abonat

-Rutarea mesajelor se bazează pe conceptul de topic-uri, unde producătorii trimit mesaje la anumite topic-uri, iar consumatorii se abonează la aceste topic-uri pentru a primi mesajele relevante.

**Garantarea livrării**

**RabbitMQ**

**ActiveMQ**

**Kafka**

-oferă diferite niveluri de garanție a livrării, inclusiv "cel puțin o dată" (at least once) și "exact o dată" (exactly once), utilizând tranzacții JMS sau alte mecanisme specifice.

-asigură o garanție "cel puțin o dată" (at least once) pentru livrarea mesajelor prin intermediul confirmărilor de livrare (delivery acknowledgments)

-Mesajele pot fi livrate către consumatori de cel puțin o dată, dar pot exista și cazuri în care un mesaj să fie livrat de mai multe ori.

-asigură o garanție "cel puțin o dată" (at least once) pentru livrarea mesajelor

-Acest lucru înseamnă că un mesaj publicat în Kafka va fi livrat către consumatorii abonați cel puțin o dată, asigurând astfel o semantica a mesajelor fără pierderi..

**Ordonarea mesajelor**

**Kafka**

**RabbitMQ**

**ActiveMQ**

-nu oferă implicit garantia ordonării mesajelor.

-garantează ordonarea mesajelor în cadrul unei singure partiții. Aceasta înseamnă că mesajele publicate într-o anumită partiție vor fi livrate consumatorilor în aceeași ordine în care au fost publicate.

-nu oferă implicit garantia ordonării mesajelor.

**Transmitere multicast**

**Kafka**

**RabbitMQ**

**ActiveMQ**

-nu oferă suport nativ pentru transmiterea multicast a mesajelor. Cu toate acestea, este posibilă implementarea unui mecanism personalizat pentru a realiza transmiterea multicast.

-suportă transmiterea multicast a mesajelor prin intermediul consumatorilor de tip consumator de grup

-permite trimiterea mesajelor către mai mulți consumatori simultan prin intermediul consumatorilor de tip consumator de grup

**Modul de stocare**

**Kafka**

**RabbitMQ**

**ActiveMQ**

-utilizează un model de stocare pe disc persistent, unde mesajele sunt păstrate pentru o anumită perioadă de timp sau în funcție de dimensiunea logurilor de mesaje.

-Aceasta permite retenția mesajelor și reluarea consumului de la un punct anterior în timp.

-stochează mesajele pe disc și oferă suport nativ pentru retenția mesajelor pe termen lung.

-mesajele pot fi păstrate în cozi pentru o perioadă definită sau pot fi persistate în baze de date externe.

-stochează mesajele în memorie și, implicit, pe disc, dar nu oferă suport nativ pentru retenția mesajelor pe termen lung.

-poate fi configurat pentru a utiliza plugin-uri suplimentare pentru a obține retenția mesajelor.

**Suport tranzacțional**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-oferă suport extins pentru tranzacții JMS (Java Message Service) și alte mecanisme de tranzacționare, permițând tranzacționarea mesajelor și integrarea cu sistemele tranzacționale.

-oferă suport pentru tranzacții în cadrul producătorului și consumatorului, permițând astfel să se asigure consistența mesajelor între producție și consum.

-oferă suport pentru tranzacții în cadrul producătorului și consumatorului, permițând astfel să se asigure consistența mesajelor între producție și consum.

**Gradul de securitate**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-asigură securitatea prin autentificare și autorizare bazate pe roluri, criptarea mesajelor în tranzit și suport pentru protocoale de securitate precum TLS/SSL.

-De asemenea, oferă integrare cu servicii externe de autentificare și autorizare.

-oferă opțiuni puternice de securitate, inclusiv autentificare, autorizare și criptare.

-Suportă autentificarea utilizatorilor și autorizarea pe baza rolurilor, iar mesajele pot fi criptate în tranzit.

-oferă opțiuni de securitate, inclusiv autentificare, autorizare și criptare a mesajelor în tranzit.

-suportă integrarea cu diverse mecanisme de securitate și oferă opțiuni de configurare pentru autentificare externă.

**4. Aspecte de calitatea serviciului oferit**

**Disponibiliate**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-este conceput pentru a fi scalabil și disponibil, având suport pentru replicare și clustering.

-În cazul în care un nod eșuează, ActiveMQ poate redirecționa cererile către nodurile disponibile, asigurând astfel o disponibilitate continuă a serviciului.

-este proiectat pentru a fi foarte rezistent la eșecuri și are o arhitectură distribuită, care permite scalabilitate orizontală și toleranță la eșecuri.

-Prin replicarea datelor și distribuirea mesajelor în partiții, Kafka poate asigura disponibilitatea ridicată în cadrul unui cluster.

-are suport pentru arhitectură distribuită și replicare, ceea ce îi permite să ofere o disponibilitate crescută.

-În plus, RabbitMQ poate fi configurat în mod redundat pentru a evita punctele de eșec.

**Performanță**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-este recunoscut pentru performanța sa de înaltă viteză.

-are un model de stocare pe disc eficient și o arhitectură distribuită, care îi permite să manipuleze un volum mare de mesaje cu o lată de bandă ridicată.

-Kafka este proiectat pentru a oferi o performanță excelentă în procesarea în timp real și în transmiterea mesajelor la viteze mari.

-oferă o performanță decentă, dar poate fi mai puțin performant în comparație cu Kafka sau RabbitMQ în ceea ce privește manipularea unui volum mare de mesaje sau cerințele de procesare în timp real de mare viteză.

-oferă performanță bună în ceea ce privește mesageria tradițională.

-cu toate acestea, performanța sa poate fi mai scăzută în comparație cu Kafka în scenarii care implică un volum mare de mesaje sau cerințe de procesare în timp real de mare viteză.

**Scalabilitate dinamică**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-oferă și el suport pentru scalabilitate prin configurarea de cluster-uri.

-cu toate acestea, scalabilitatea sa poate fi mai limitată în comparație cu Kafka, în special în ceea ce privește gestionarea unui număr mare de producători și consumatori sau cerințele de procesare în timp real de mare viteză.

-este cunoscut pentru scalabilitatea sa orizontală excelentă.

-poate fi ușor extins prin adăugarea de noi noduri la cluster și poate gestiona cu ușurință creșterea cererii.

-Kafka distribuie partițiile și replica mesajelor pe noduri multiple, asigurând o scalabilitate eficientă și o distribuire echilibrată a sarcinilor.

-poate fi configurat într-un cluster pentru a oferi scalabilitate și redundanță.

-cu toate acestea, scalabilitatea sa poate fi mai puțin eficientă și mai puțin flexibilă în comparație cu Kafka sau RabbitMQ.

**Latența și debit**

**ActiveMQ**

**RabbitMQ**

**Kafka**

-oferă o latență decentă și un debit moderat. Este potrivit pentru aplicații cu volum mediu de mesaje și cerințe de procesare.

-oferă, în general, latență scăzută și un debit bun, dar poate întâmpina anumite limitări în ceea ce privește cer deosebit de mari de mesaje sau cerințe de procesare în timp real de mare viteză.

-oferă latență scăzută și un debit ridicat, ceea ce îl face potrivit pentru aplicații cu cerințe stricte de timp real și procesare în flux continuu.

* Kafka: Modelul de publicare/abonare, garantarea livrării "cel puțin o dată", scalabilitatea excelentă, suport tranzacțional și stocare persistentă.
* RabbitMQ: Modelul de cozi de mesaje, flexibilitatea rutării mesajelorși suportul extins pentru protocoale de mesagerie, cum ar fi AMQP și MQTT.
* ActiveMQ: Suport extins pentru standardul JMS (Java Message Service), inclusiv tranzacții JMS și persistența mesajelor pe termen lung. ActiveMQ este puternic integrat în ecosistemul Java și oferă o gamă largă de caracteristici și extensibilitate.

**5. Referinte**

- <https://www.conduktor.io/blog/comparing-apache-kafka-activemq-and-rabbitmq/>

- <https://medium.com/double-pointer/kafka-vs-activemq-vs-rabbitmq-vs-amazon-sns-vs-amazon-sqs-vs-google-pub-sub-4b57976438db>

- <https://www.designgurus.io/blog/RabbitMQ-Kafka-ActiveMQ-System-Design>

Code repository:

<https://github.com/anamaria-cusco/kafka-chat.git>