

Raport de evaluare - Pub/Sub

Miruna, Curduman, Daria Melinte, Tudor Aungurenci

June 17, 2025

Abstract

Această lucrare evaluează performanța unui sistem distribuit de tip publish/subscribe, bazat pe filtrare după conținut și capabil să proceseze ferestre de publicații. Evaluarea a fost realizată prin simularea unui flux de 10.000 de subscripții simple, transmise printr-o rețea de brokeri care filtrează și direcționează publicațiile către subscrieri.

Contents

1	Introducere	2
2	Arhitectura sistemului	2
2.1	Componentele principale	2
2.2	Caracteristici importante	3
3	Procedura experimentală	3
3.1	Inițializarea sistemului și generarea subscripțiilor	3
3.2	Pornirea rețelei de brokeri	4
3.3	Pornirea nodurilor subscriber și înregistrarea subscripțiilor	4
3.4	Pornirea nodului publisher și rularea experimentului	4
3.5	Oprirea sistemului și colectarea datelor	4
3.6	Prelucrarea și analiza rezultatelor	4
4	Rezultate și interpretare	5
5	Discuții și concluzii	7

1 Introducere

Acest experiment urmărește evaluarea performanței unui sistem publish-subscribe bazat pe conținut, capabil să proceseze atât mesaje individuale, cât și secvențe de mesaje grupate în ferestre temporale. Sistemul este alcătuit dintr-un nod publisher, care generează în mod continuu un flux de mesaje cu date generate aleatoriu, o rețea de brokeri ce se ocupă cu filtrarea și distribuirea mesajelor către nodurile subscriber, care primesc și procesează doar mesajele relevante pentru subscripțiile lor.

Pentru transmiterea eficientă a datelor între noduri, mesajele sunt serializate folosind un mecanism binar (Google Protocol Buffers), ceea ce reduce timpul și spațiul necesar comunicării. Rețeaua de brokeri suportă atât subscripții simple, bazate pe condiții directe asupra câmpurilor din mesaje, cât și subscripții complexe care implică agregări statistice pe ferestre de mesaje (de exemplu, medii sau valori maxime pe un interval de timp).

Experimentul se concentrează pe măsurarea următorilor parametri esențiali pentru evaluarea sistemului:

- **Rata de livrare a mesajelor** — câte mesaje sunt livrate cu succes către subscriberii corespunzători într-un interval continuu de 3 minute, în condiții de încărcare cu 10.000 de subscripții simple.
- **Latența medie de livrare** — timpul mediu de la generarea unui mesaj în publisher până la primirea lui de către subscriber, măsurat pentru mesajele transmise în același interval.
- **Rata de potrivire (matching)** — procentajul de mesaje care corespund criteriilor de filtrare în două scenarii diferite: unul în care toate subscripțiile folosesc operatorul de egalitate pe un câmp specific și altul în care operatorul de egalitate este utilizat în aproximativ 25% din subscripții.

2 Arhitectura sistemului

Sistemul implementat folosește un model **publish-subscribe** distribuit, în care un nod numit *publisher* generează și trimite mesaje, iar mai mulți *subscriberi* primesc doar mesajele care corespund criteriilor lor de interes. Între publisher și subscriberi se află o rețea de brokeri care gestionează trimiterea și filtrarea mesajelor.

2.1 Componentele principale

- **Publisher:** Generează continuu mesaje cu date aleatorii folosind mai multe thread-uri pentru performanță. Mesajele sunt serializate cu Protocol Buffers pentru a reduce dimensiunea și timpul de transmisie.

- **Brokeri:** O rețea formată din 2-3 brokeri care distribuie sarcina între ei folosind o metodă simplă de echilibrare a încărcării (round-robin). Fiecare broker procesează subscripțiile și face potrivirea mesajelor cu condițiile definite de subscriberi. Subscripțiile pot fi simple (filtrare directă pe câmpuri) sau bazate pe ferestre de date (agregări statistice, cum ar fi media sau valoarea maximă pe un set de mesaje).
- **Subscribers:** Nodurile care primesc mesajele care se potrivesc cu subscripțiile lor. Pot înregistra subscripții noi în timpul rulării sistemului și măsoară latența de livrare a mesajelor.
- **Generator de date:** Creează publicații și subscripții aleatorii pe baza unor modele configurabile, folosind generare multi-threaded pentru a testa sistemul la scară.

2.2 Caracteristici importante

- Sistemul distribuie încărcătura între brokeri pentru a evita blocajele.
- Folosește execuția paralelă (thread-uri) pentru o procesare rapidă.
- Permite subscripții dinamice și condiții complexe, inclusiv pe ferestre de date.
- Colectează statistici și măsoară performanța în timp real.
- Utilizează un format eficient de serializare pentru transmiterea datelor.

3 Procedura experimentală

Experimentul a fost conceput pentru a evalua performanța unui sistem publish-subscribe bazat pe conținut, care să poată procesa atât mesaje individuale, cât și secvențe de mesaje grupate în ferestre temporale. Pentru aceasta, s-au urmat pașii detaliați mai jos, folosind un set de componente software implementate în Python.

3.1 Inițializarea sistemului și generarea subscripțiilor

Mai întâi, s-a încărcat o configurație specifică dintr-un fișier JSON care conține parametrii pentru generarea de subscripții. Această configurare controlează tipurile de filtre folosite, frecvența operatorului de egalitate pe anumite câmpuri și numărul total de subscripții. Folosind clasa `GeneratorPubSub`, au fost generate un număr mare de subscripții (peste 10.000) cu condiții variate, atât simple (filtrare directă pe câmpuri), cât și complexe (care includ condiții pe ferestre de mesaje).

S-a calculat și raportat procentul subscripțiilor care conțin operatorul de egalitate pe un câmp specific (de exemplu, „rain”), pentru a diferenția scenariile de testare (25% versus 100%).

3.2 Pornirea rețelei de brokeri

S-a inițiat o rețea compusă din trei brokeri, care au rolul de a primi mesajele publicate, a face filtrarea în funcție de subscripțiile înregistrate și de a trimite mai departe mesajele către nodurile subscriber relevante. Rețeaua folosește un mecanism de distribuție a încărcăturii și suportă procesarea pe ferestre temporale, cu o dimensiune prestabilită.

3.3 Pornirea nodurilor subscriber și înregistrarea subscripțiilor

Trei noduri subscriber s-au conectat la rețeaua de brokeri. Aceste noduri au înregistrat subscripțiile generate anterior, astfel încât să primească doar mesajele care corespund condițiilor definite. Subscripțiile au fost distribuite uniform între subscriberi pentru a simula un scenariu realist.

Fiecare nod subscriber a fost programat să înregistreze mesajele primite și să măsoare latența de la emiterea mesajului în publisher până la primirea acestuia.

3.4 Pornirea nodului publisher și rularea experimentului

Publisherul a fost pornit pentru a genera în mod continuu mesaje care conțin date serializate folosind Google Protocol Buffers. Mesajele sunt plasate în cozi și apoi trimise către brokeri.

Experimentul a fost desfășurat pe o durată fixă de 3 minute, timp în care publisherul a generat un flux constant de publicații, iar brokerii au procesat și transmis aceste mesaje către subscriberii interesați. În acest interval s-au colectat date privind numărul total de mesaje livrate cu succes, timpul mediu de livrare (latența) și statistici privind potrivirea mesajelor cu subscripțiile.

3.5 Oprirea sistemului și colectarea datelor

La finalul perioadei de testare, toate componentele (publisher, brokeri și subscriberii) au fost oprite în mod ordonat pentru a evita pierderi de date.

Fiecare nod subscriber a salvat toate mesajele primite într-un fișier JSON pentru analiză ulterioară. Brokerii au generat rapoarte în format CSV care conțin statistici detaliate precum numărul de mesaje primite, numărul de mesaje trimise către subscriber, numărul de încercări de potrivire și numărul efectiv de potriviri găsite, împreună cu latențele medii.

3.6 Prelucrarea și analiza rezultatelor

Rezultatele obținute pentru fiecare configurație de subscripții au fost centralizate într-un fișier CSV sumar care conține datele esențiale: numărul total de mesaje livrate, latența medie în milisecunde și rata de potrivire exprimată în procente.

Aceste date permit compararea impactului diferitelor configurații asupra performanței sistemului, în special a influenței procentului de subscripții care utilizează operatorul de egalitate în filtrare.

Prin această procedură experimentală detaliată, s-a asigurat colectarea unor date relevante și precise pentru a evalua eficiența și scalabilitatea arhitecturii publish-subscribe dezvoltate, în condiții apropiate de aplicații reale de procesare a fluxurilor de date în timp real.

4 Rezultate și interpretare

În acest experiment am evaluat performanța sistemului publish-subscribe folosind două configurații diferite pentru subscripții, denumite „25%” și „100%”, în funcție de procentul subscripțiilor care utilizează operatorul de egalitate pe un câmp.

Pentru fiecare configurație, au fost măsurate următoarele aspecte:

1. Numărul de publicații livrate cu succes în intervalul de 3 minute

Sistemul a generat aproximativ același număr de mesaje în ambele configurații: **36.675 mesaje** pentru configurația **25%** și **35.675 mesaje** pentru configurația **100%**. Dintr-acestea, numărul de mesaje livrate efectiv către subscriber a fost foarte apropiat: **35.590 pentru 25%** și **36.675 pentru 100%**. Astfel, procentul mesajelor livrate față de cele generate a fost de aproximativ 97% pentru ambele cazuri.

Table 1: Rezultate detaliate ale experimentului

Configurație	Publicații generate	Publicații livrate	% Livrate	Latență medie (ms)	Rată potrivire (%)
25%	36.675	35.590	97,07	7,85	27,73
100%	36.675	35.675	97,27	7,91	9,89

2. Latența medie de livrare a unei publicații

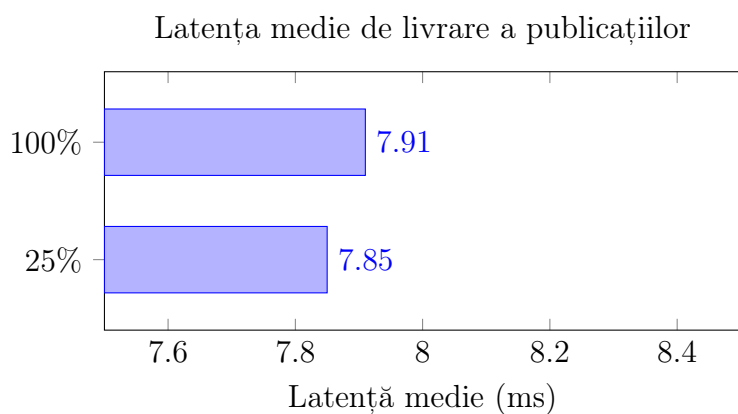


Figure 1: Comparația latenței medii între configurațiile 25% și 100%.

Timpul mediu de la emiterea unei publicații de către publisher până la recepția acesteia de către subscriber a fost foarte mic și comparabil pentru ambele configurații: **7,85 ms pentru 25%** și **7,91 ms pentru 100%**.

3. Rata de potrivire a mesajelor cu subscripțiile

Rata de potrivire reprezintă procentul mesajelor pentru care brokerii au găsit o corespondență în subscripțiile înregistrate, calculată ca raportul dintre numărul de potriviri găsite și numărul total de încercări de potrivire.

$$\text{Rata de potrivire} = \frac{\text{Numar potriviri gasite}}{\text{Numr incercari de potrivire}} \times 100\%$$

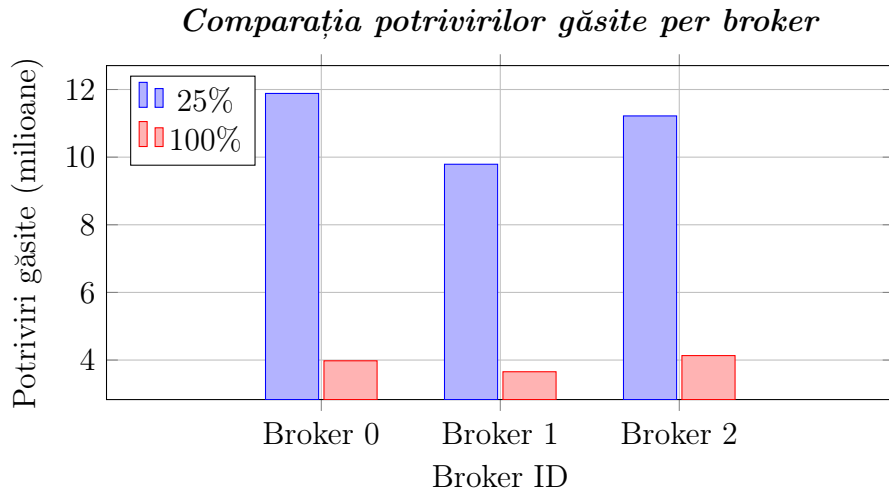


Figure 2: Numărul de potriviri găsite de fiecare broker în cele două configurații.

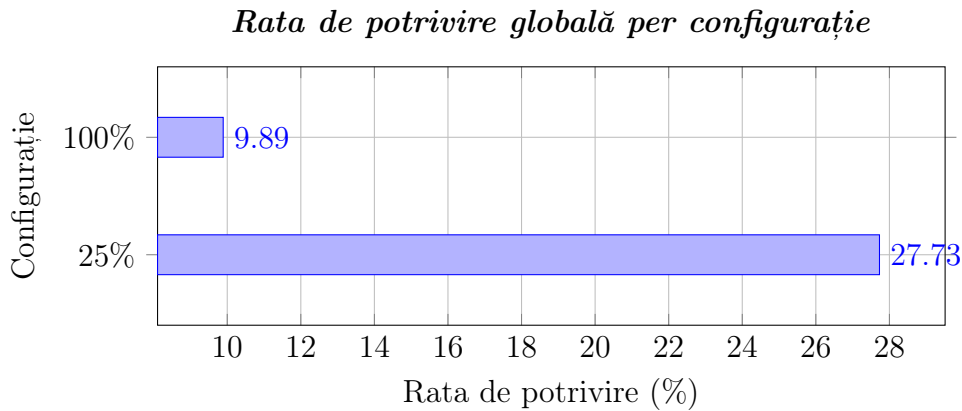


Figure 3: Comparația ratelor de potrivire globale între cele două configurații.

Pentru configurația cu **25%** operator egal, rata de potrivire a fost mult mai mare, **27,73%**, comparativ cu doar **9,89%** pentru configurația **100%**. Această diferență este explicată prin faptul că subscripțiile cu operator egal sunt mai restrictive, deci se potrivesc unui număr mai mic de mesaje, în timp ce subscripțiile din configurația cu **25%** permit o filtrare mai largă.

Table 2: Statistici brokeri - configurație 25%

Broker ID	Publicații primite	Trimise subscriberilor	Încercări potrivire	Potriviri găsite
broker_0	11,863	11,863	39,551,242	11,884,493
broker_1	11,864	11,864	39,542,712	9,790,136
broker_2	11,863	11,863	39,539,379	11,219,146
Total	35,590	35,590	118,633,333	32,893,775

Table 3: Statistici brokeri - configurație 100%

Broker ID	Publicații primite	Trimise subscriberilor	Încercări potrivire	Potriviri găsite
broker_0	11,891	11,891	39,644,594	3,977,794
broker_1	11,892	11,892	39,636,036	3,653,540
broker_2	11,892	11,892	39,636,036	4,131,937
Total	35,675	35,675	118,916,666	11,763,271

5 Discuții și concluzii

Rezultatele obținute în acest experiment sunt concludente și relevante pentru evaluarea performanței sistemului *publish-subscribe* dezvoltat. Sistemul demonstrează următoarele caracteristici de performanță:

- Latență foarte scăzută (sub 8 ms)
- Rată de livrare ridicată (peste 97%)

Diferența semnificativă în ratele de potrivire între cele două configurații confirmă impactul direct al tipului de operatori folosiți în subscripții asupra eficienței filtrării:

- 27,73% rată de potrivire pentru configurația cu 25% operatori de egalitate
- 9,89% rată de potrivire pentru configurația cu 100% operatori de egalitate
- Subscripțiile cu operator de egalitate sunt mai restrictive, ceea ce duce la mai puține potriviri
- Poate fi avantajos în scenarii unde este necesară o filtrare foarte precisă