Evaluation Report of the Publication Delivery System

Introduction

This report evaluates the performance of a publication delivery system, specifically measuring its efficiency in handling 10,000 simple subscriptions over a continuous feed interval of 3 minutes. The evaluation covers three key metrics:
1. The number of publications successfully delivered.
2. The average delivery latency.
3. The match rate under different conditions of equality operators in subscriptions.

The tests were conducted three times to ensure the reliability of the results, and the average values were computed.

System architecture and setup

Threading and concurrent execution

- **Publishing**: Publishers operate in separate threads (`publisher1_thread`, `publisher2_thread`), ensuring concurrent publication of content to the broker network.
- **Subscriber Handling**: Subscribers are managed in a list and interact with brokers asynchronously to receive publications matching their subscriptions.

Subscription filters

- **100% Equality**: Subscriptions are filtered using `filter_func1`, ensuring that only publications containing the character 'A' in their content are considered for matching.
- **25% Equality**: Subscriptions are filtered using `filter_func2`, which introduces randomness by including the character 'A' in publications with a 25% probability.

Match Rates

1. Match rate for 100% equality:

For subscriptions using `filter_func1`, which checks if the character 'A' is present in the publication content (`'A' in pub['content']`), the match rate is calculated as follows:

```
match_rate_100 = sum(len(sub.received_publications) for sub in
subscribers[:5000]) / 5000
```

Calculation:

- We iterate through the first 5000 subscribers (`subscribers[:5000]`), which are configured with `filter_func1`.
- For each subscriber, `len(sub.received_publications)` gives the number of publications received that match their subscription criteria.
- We sum these counts and divide by 5000 to get the average number of publications received per subscriber.
- This average represents the match rate for 100% equality, indicating how effectively the system delivers publications that exactly match the subscription criteria.

2. Match rate for 25% equality:

For subscriptions using `filter_func2`, which introduces randomness by including 'A' in publications with a 25% probability (`random.choice([True, False, False, False]) and 'A' in pub['content']`), the match rate is calculated as:

```
match_rate_25 = sum(len(sub.received_publications) for sub in subscribers[5000:]) / 5000
```

Calculation:

- We iterate through the last 5000 subscribers (`subscribers[5000:]`), which are configured with `filter_func2`.
- Similar to the 100% equality calculation, `len(sub.received_publications)` gives the count of publications received per subscriber.
- Sum these counts and divide by 5000 to obtain the average number of publications received per subscriber.
- This average represents the match rate for 25% equality, indicating the effectiveness of delivering publications that probabilistically match the subscription criteria.

Test Results

Run number	Total Delivered Publications	Average Latency	Match Rate for 100% Equality	Match Rate for 25% Equality
1	855,757	0.0149 seconds	137.0000	34.1514
2	1,012,631	0.0106 seconds	162.0000	40.5262

3	1,043,326	0.0099 seconds	167.0000	41.6652
AVERAGE	970,571	0.0118 seconds	155.3333	38.7809

Analysis

Successful Deliveries
The average number of publications delivered successfully across the three runs was 970,571. This demonstrates the system's robust capacity for handling high throughput, with some variation between runs likely due to network or system load fluctuations.

Latency
The average latency for delivering a publication was 0.0118 seconds. This low latency indicates a highly efficient system capable of rapid publication delivery, though slight variations in latency were observed across the runs.

Match Rate
The match rate under different conditions shows:
- 100% Equality: An average match rate of 155.3333, indicating the system performs well with straightforward, equality-based subscriptions.
- 25% Equality: An average match rate of 38.7809, which is lower, reflecting the increased complexity and reduced likelihood of matches under less frequent equality conditions.

Implications

- High Match Rate with 100% Equality: The system performs exceptionally well when subscriptions are strictly defined with equality operators, ensuring high match rates and efficient delivery.
- Lower Match Rate with 25% Equality: The reduced match rate under less frequent equality operator usage indicates the need for potential optimizations in handling more complex subscription conditions.

Conclusion

The evaluated publication delivery system exhibits high efficiency and reliability, with significant throughput and low latency. The system performs best when subscriptions are straightforward and equality-based. However, there is room for improvement in handling more complex subscription scenarios with less frequent equality operators. Further optimizations could enhance the match rate and overall performance under varied subscription conditions.

This evaluation underscores the system's capability to support high-volume, real-time data delivery, making it suitable for applications where timely and accurate data dissemination is critical.