Publish-Subscribe System

Author: Chaitali Kamble (csk3565@rit.edu)
Date: 03/12/2017

What is Publish-Subscribe System?

Publish-Subscribe is a flexible communication system that performs asynchronous communication message passing. It is a distributed system. Here, we the system components do not need to maintain connection during the message exchange. It is not necessary for a publisher and a subscriber to be present at the same time while communicating with each other. Publishers are unaware of the subscribers and vice a versa.

How does Publish-Subscribe system work?

Publish-subscribe system is a loosely coupled distributed system in which Publisher, Subscriber and Event manager play key roles. The message passing happens in an asynchronous manner. Subscriber can subscribe for a specific topic and whenever an event occurs related to that specific topic, subscriber gets notified. Also, whenever someone advertises a new topic it gets notified to everyone in the system. Publisher can publish an event and whenever publisher publishes an event for a specific topic, he/ she automatically gets subscribed to that topic.
The important aspect of this system is that it is not necessary for publisher and subscriber to be present at the same time. Even if a subscriber is not present, whenever he comes online, he will get notifications for the same.

Different components of Publish-Subscribe System:

1. **Subscribe**: A user can subscribe to a topic of interest and whenever new event occurs related to that topic, this user will get notified.
2. **Unsubscribed**: A user can either unsubscribe from a specific topic or he can unsubscribe from a system as whole. If he does so, he will not get notifications regarding the same topic.
3. **Publish**: User can publish an event on any topic and he/she will get subscribed to it automatically.
4. **Advertise:** User can advertise about new topic in the system and everyone will receive notification for the same.
5. **Notify:** User gets notifications about new topics in the system or events for which he/she subscribed for.

System built

The Publish-Subscribe system built is user friendly. As soon as event manager is up, publisher and subscriber can come and connect to the system. They can disconnect anytime whenever they want and it wont affect event manager at all.

Below is the User interface for client when it connects to Publish-Subscribe System:

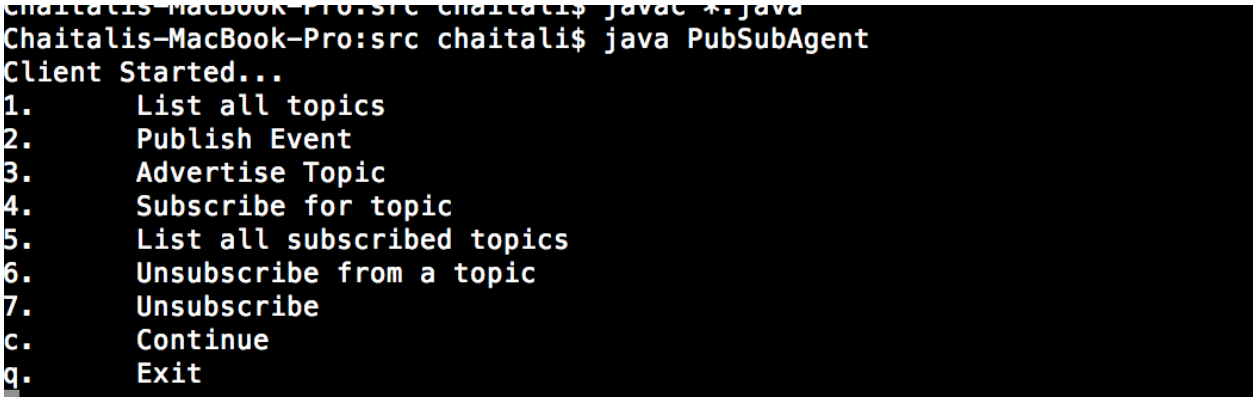

Figure: User Interface for a client when he connects to system

1. **List all topics**: This option will list all the topics which are present currently in the system.
2. **Publish Event**: This option will ask various inputs from user and create an event and sends it to event manager.
3. **Advertise a topic**: This option will advertise a new topic and its related keywords.
4. **Subscribe for topic**: This option will ask user a specified topic and subscribe him for that topic.
5. **List all subscribed topics**: This option will list all the subscribed topics till now for a user.
6. **Unsubscribe from a topic**: This option will ask user a specified topic and unsubscribe him from that topic.
7. **Unsubscribe**: This option will unsubscribe a user permanently from the system.
8. **Continue**: This option will keep user using the system.
9. **Exit**: This option will disconnect user from the system.

All the above options are on the user's screen and whenever user enters any option, necessary action will be taken and request will be sent to event manager. Event manager manages all the data structures and notifies the client regarding the same.

There are certain errors that are handled:

1. Suppose a user is not subscribed to any topic and he/ she enters option 5, then an error message will get popped up as you are not subscribed to any topic.
2. Suppose a publishes an event for a topic which is not present in the system then the user will get a message accordingly.

3. Suppose a user wants to unsubscribe from a topic, then he has to give a topic which is already present in the system and he has to be subscribed for that topic else he will get a response message.
4. If user enters any other option than 1-7 and c and q, then he/she will get all the options again.
5. If any exception occurs on user side, then he/she will get automatically disconnected.
6. If any exception occurs on Event manager side, then he/she will get disconnected.

User and event manager communicates via TCP/IP protocol. Event manager opens port 4000 and allow all the clients to communicate through same port. Here, a client comes in and event manager assigns it to a thread to process all the clients in parallel. Dynamic threading allows event manager to process all the clients parallel and avoid congestion.

Pros of Publish-Subscribe System

- **Asynchronous communication** within client and server.
- **Loosely coupled distributed system** where publisher and subscriber are unaware of the functionality and hence can disconnect from the system anytime when they want without affecting the system.
- **Fast information retrieval** as hash map data structure is used to store data, hence information retrieval happens in O (1) time for best case and worst case time complexity can reach to O(n).
- The system is **scalable** enough to operate on thousands of clients at a time.
- **Synchronization –** Publishers are not blocked while publishing events and subscribers can receive notifications.
- If a user goes offline, still he can receive his notifications as and when he comes online because messages are queued.

Cons of Publish-Subscribe System

- If an event manager goes down then the system will not handle clients. It does not have a backup server.
- Data structures hold just specified information but in future they can be extended to hold more information about event and topics.
- Each client is identified by its IP address uniquely but if two clients log in into the system through same machine then the clients cannot be distinguished using their corresponding IPs. Hence, in future if each client logs in with a specified unique ID then they can be distinguished.

Application of Publish-Subscribe System:

This system has its varied range of applications that we encounter in day-to-day life.

Stock Market:
Working on a Stock market which is a real time application of Publish-Subscribe system.

The idea is similar to Publish-Subscribe system. In fact, communication between server and multiple clients is exactly similar to that of Pub Sub system. In stock market system, all the options displayed on user's screen are different. Also, data structures will hold different information than that of Pub Sub system. Here, some mathematics should be performed such as if a user wants to sell some shares and another user buys that shares than the the money should be increased for first user and simultaneously no of shares should be decreased. A seller publishes a sell command for a share, the offering will be notified to all subscribers who subscribed to receive events of this stock. If a subscriber wants to buy the share, it can publish a buy command to get the share. The available share will sell to the first buyer. Once this buy command of a share is received, the share is removed from the market.

Application of Publish-Subscribe System:

1. I learned about TCP/IP, RPC and message passing functionalities.
2. I learned several aspects parallel processing.
3. Learned about connection of different components over distributed network.
4. Learned to avoid congestion and bottle neck issues.
5. Learned about message delays and notifications.

REFERENCE:

- Patrick Th. Eugster et. al. The many faces of publish/subscribe, In Proceedings of ACM Computing Surveys, 2003