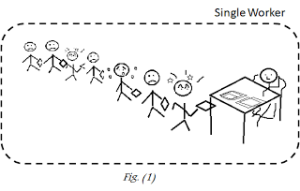
Tìm hiểu về Apache ActiveMQ

ActiveMQ là gì? Tại sao ta lại cần đến nó?

Trước khi bàn đến ActiveMQ tôi xin đề cập đến Producer–Consumer pattern để trả lời câu hỏi thứ 2.

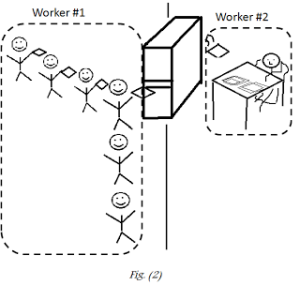
Để dễ hiểu ta có thể hình dung một tình huống như sau:

Việc xếp hàng để thực hiện một hành vi nào đấy là rất phổ biến, ví dụ như xếp hàng mua vé tàu, người mua vé phải xếp hàng dài đợi chờ rất lâu để có thể mua vé tàu, vì việc này phải xử lý tuần tự, người này mua xong mới tới người khác. Trong kỹ thuật người ta gọi đây là xử lý **Synchronized**tức là xử lý đồng bộ.



Với một vài trường hợp như các tác vụ là độc lập không cần chờ nhau, thì việc này sẽ gây ra việc tắc nghẽn hệ thống vì task vụ trước phải sử lý xong với đến tác vụ sau.

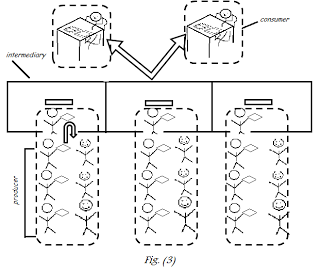
Để giải quyết tình huống này ta có thể xây dựng một hệ thống tiếp nhận yêu cầu liên tục mà không cần xử lý ngay, rồi dữ liệu sẽ được rút ra rồi xử lý dần dần như cái cabinet trong hình sau.



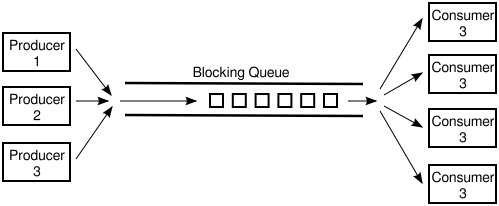
Trong kỹ thuật người ta gọi đây là xử lý bất đồng bộ **as-Synchronized**

Việc này sẽ giúp cho việc hệ thống có thể nhận yêu cầu liên tục từ “Customer” mà không cần chờ đợi, tác vụ sẽ được “Worker” xử lý dần cho đến khi các tác vụ được xử lý hết.

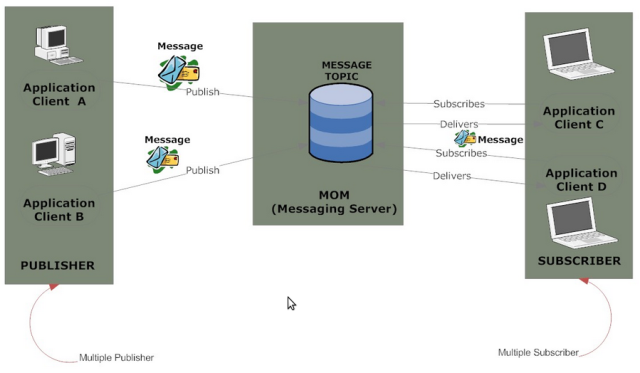
Nhưng sẽ có tình huống quá nhiều yêu cầu mà Worker không xử lý kịp, dẫn đến cabinet bị đầy và việc tiếp nhận sẽ bị chậm đi, để giải quyết việc này ta có thể bổ xung thêm nhiều Worker hoặc chia ra thành nhiều cabinet mỗi cabinet có một Worker.



Vần đề nêu trên trong lập trình người ta gọi là Producer–Consumer Problem. Producer ở đây là khách hàng, người gửi yêu cầu, còn Consumer ở đây được hiểu là Worker người xử lý yêu cầu của khách hàng, còn cái cabinet chưa yêu cầu được gọi là Queue.



Bây giờ về lại với chủ đề chính là ActiveMQ. ActiveMQ là một Message-oriented middleware (MOM) giúp cho việc chuyển nhận message theo cơ chế queue as-Synchronized dùng để giải quyết Producer-Consumer Problem. Điểm mạnh của ActiveMQ ta có thể dùng để gửi nhận message với nhiều hệ thông với các nền tảng khác nhau, đúng như tên gọi “Hệ thống trung gian chuyển tải gói tin”.



Các thành phần chính trong ActiveMQ

1. Producer/Publisher: Thành phần tạo và gửi tin (ActiveMQ-Client).
2. Broker trung gian hay Message Oriented Middleware (MOM) (ActiveMQ-Broker).
3. Consumer/Subcriber: Thành phần nhận tin từ Producer thông qua MOM (ActiveMQ-Client).

Cách cài đặt: (Trên \*nix system)

Bước 1: Down package từ trang chủ

# wget <http://activemq.apache.org/path/tofile/apache-activemq-x.x.x-bin.tar.gz>

Bước 2: giải nén

cd [activemq\_install\_dir]

tar zxvf activemq-x.x.x-bin.tar.gz

Bước 3: Chạy dịch vụ thông qua thư viện

# cd [activemq\_install\_dir]/bin

# ./activemq console #foregroud process

# ./activemq start #daemon process

Bước 4: kiểm tra dịch vụ

# netstat -nl | grep 61616

Giao diện giám sát ActiveMQ

- Đường dẫn

<http://localhost:8161/admin>

- User: admin|admin. Có thể cấu hình trong “conf/jetty-real.properties”

Để dừng activeMQ

# cd [activemq\_install\_dir]/bin

# ./activemq stop

Cấu hình activeMQ

Cấu hình cơ bản

- activeMQ sử dụng file activemq.xml cho việc cầu hình cơ bản.

- Phần dưới đây ta sẽ sử dụng file cấu hình có sẵn cung cấp bởi puppet Mcollective để cấu hình phục vụ cho kết nối:

Bước 1: Truy cập vào nơi chứa file activemq

cd [activemq\_install\_dir]/config

Bước 2: copy nội dung file có sẵn vào file cấu hình:

- Nội dung trong file: activemq.xml (trong thư mục chứa docs)

- Truy cập vào link để lấy file: <https://raw.githubusercontent.com/puppetlabs/marionette-collective/master/ext/activemq/examples/single-broker/activemq.xml>

vim activemq.xml

### Ghi đè nội dung

…..

Bước 3: Restart service activemq

# cd [activemq\_install\_dir]/bin

# ./activemq stop

# ./activemq start

e