Tìm hiểu Apache Tribes

- Apache Tribes là group or peer-to-peer framework kết nối cho phép bạn dễ dàng kết nối các remote object tới các thành phần khác.

Bao gồm:

Import: org.apache.catalina.tribes.Channel

Import: org.apache.catalina.tribes.Member

Import: org.apache.catalina.tribes.MembershipListener

Import: org.apache.catalina.tribes.ChannelListener

Import: org.apache.catalina.tribes.group.GroupChannel

Create a class that implements: org.apache.catalina.tribes.ChannelListener

Create a class that implements: org.apache.catalina.tribes.MembershipListener

Simple class to demonstrate how to send a message:

Vd:

//create a channel

Channel myChannel = new GroupChannel();

//create my listeners

ChannelListener msgListener = new MyMessageListener();

MembershipListener mbrListener = new MyMemberListener();

//attach the listeners to the channel

myChannel.addMembershipListener(mbrListener);

myChannel.addChannelListener(msgListener);

//start the channel

myChannel.start(Channel.DEFAULT);

//create a message to be sent, message must implement java.io.Serializable

//for performance reasons you probably want them to implement java.io.Externalizable

Serializable myMsg = new MyMessage();

//retrieve my current members

Member[] group = myChannel.getMembers();

//send the message

myChannel.send(group,myMsg,Channel.SEND\_OPTIONS\_DEFAULT);

- Có rất nhiều ví dụ về Tribes mà ta có thể shown, xem thêm tại docs.

Note:

Tribes được xây dựng trên JDK1.5

What is Tribes

- Tribes là messaging framework với khả năng giao tiếp giữa các nhóm. Tribes cho phép bạn gửi và nhận message thông qua network, nó cũng cho phép tìm kiếm các node khác trong mạng.

- Tribes module bắt đầu năm 2006, là 1 phần nhỏ của code base tới từ clustering module tồn tại 2003 – 2004.

- Tribes nắm giữ phần phức tạp của messaging của replicatuon module và trở thành 1 thành phần độc lập, linh động trong group communication module.

- Trong Tomcat cữ “molules/cluster” trở thành “modules/groupcom”(Tribes) và “modules/ha” (replication). Nó sẽ cho phép các phát triển tập trung và các dev có thể tập trung vào các vấn đề mà đang làm việc mà không cần quan tâm đến các phần bên dưới.

- Tribes sẽ cho phép bảo điểm messaging và có thể tùy chỉnh theo nhiều các. Đây là 1 thành phần quan trọng vì người phát triển muông biết tin nhắn họ gửi đến được đích mong muốn, và hơn thế nữa, nếu message không tới được đích, app trên top Tribes sẽ thông bao message không được gửi và node nào lỗi.

Feature Overview

- Một số tính năng quan trọng trong Tribes

Pluggable modules

- Tribes được xây dựng sử dụng interfaces. Bất kỳ modules hoặc thành phần thuộc Tribes có thể tách ra customize thành phần đó.

Guaranteed Messaging

- Mặc định thực thi Tribes sử dụng TCP hoặc UDP cho tin nhắn. TCP đã có khả năng bảo đảm sự vận chuyển message và flow control built in. Và tin rằng hiệu năng của Java TCP sẽ hơn hẳn so với Java/UDP/flow-control/message. UDP messaging đã được thêm vào cho việc gửi message thông qua UDP thay vì TCP khi thiết kế. Có kịch bản bảo đảm được mô tả cho UDP nhưng vấn đề xảy ra kaf khi message mất đi, nó không được xem xét là lỗi.

- Tribes hỗ trợ cả non-blocking và blocking IP operations. Khuyến khích setting sử dụng non blocking vì nó sẽ hỗ trợ tính song song tốt hơn khi gửi là nhận messages. Blocking thực thi có sẵn trong NIO nhưng vẫn tồn tại lỗi.

Different Guarantee Levels

- Có 3 mức bảo đảm khác nhau khi gửi message.

1. IO based send guarantee – nhanh nhất, ít tin cậy.

- Có nghĩa tribes sẽ cần nhắc truyền message thành công nếu message đã gửi tới socket buffer và accepted.

- Trên blocking IO, nó sẽ là “socket.getOutputStream().write(msg)”

- Trên non-blocking IP, nó sẽ là “socketChannel.write()”, buffer byte buffer sẽ được làm mới cùng với “socketChannel.read()” để chắc chắn channel vẫn mở.

2. ACK based – Khuyến khác, đảm bảo sự vận chuyển

- Khi message đã được nhận trên remote node, ACK sẽ gửi lại sender, cho biết rằng message đã được nhận.

3. SYNC\_ACK based. – đảm bản vận chuyển, bảo đảm xử lý, chậm nhất.

- Khi message đã được nhận trên remote node, node sẽ xử lý message và nếu message được xử lý thành công, ACK sẽ gửi lại tới sender cho biết message đã được nhận và xử lý thành công. Nếu message được nhận nhưng xử lý lỗi, ACK\_FAIL sẽ gửi lại tới sender. Đây là 1 tính năng riêng hỗ trợ cho việc phát triển ứng dụng. Hầu như frameworks sẽ cho bạn biết message đã được gửi và application developer sẽ xây dựng logic cho message đã thực sự xử lý trên app tại remote node. Nếu config, Tribes sẽ gửi exception khi nhận được ACK\_FAIL cùng với thông điệp member không xử lý message.

Per Message Delivery Attributes (Thuộc tính phân phối trên mỗi message)

- Có thể tính năng kiến Tribes tách biệt khởi group commnication frameworks. Tribes cho phép bạn gửi và quyết định nhưng semantisc message truyền nên có trên mối message. Có nghĩa là, message sẽ không truyền dựa trên 1 số static config giữa nguyên khi message framework được bắt đầu.

VD: Gửi 10 message

Message\_1 - asynchronous and fast, no guarantee required, fire and forget

Message\_2 - all-or-nothing, either all receivers get it, or none.

Message\_3 - encrypted and SYNC\_ACK based

Message\_4 - asynchronous, SYNC\_ACK and call back when the message is processed on the remote nodes

Message\_5 - totally ordered, this message should be received in the same order on all nodes that have been

send totally ordered

Message\_6 - asynchronous and totally ordered

Message\_7 - RPC message, send a message, wait for all remote nodes to reply before returning

Message\_8 - RPC message, wait for the first reply

Message\_9 - RPC message, asynchronous, don't wait for a reply, collect them via a callback

Message\_10- sent to a member that is not part of this group

Note:

As you can imagine by now, these are just examples. The number of different semantics you can apply on a per-message-basis is almost limitless. Tribes allows you to set up to 28 different on a message and then configure Tribes to what flag results in what action on the message.

Imagine a shared transactional cache, probably >90% are reads, and the dirty reads should be completely unordered and delivered as fast as possible. But transactional writes on the other hand, have to be ordered so that no cache gets corrupted. With tribes you would send the write messages totally ordered, while the read messages you simple fire to achieve highest throughput.

There are probably better examples on how this powerful feature can be used, so use your imagination and your experience to think of how this could benefit you in your application.

Interceptor based message processing (Xử lý dựa trên tin nhắn nhận được)

- Tribes sử dụng interceptor stack được tùy chỉnh để xử lý các tin nhắn gửi và nhận.

- Tribes interceptor có thể phản ứng lại các message dựa trên mỗi “message-attributes” gửi runtime. Có nghĩa là, nếu ta thêm bào encryption interceptor mà sẽ encrypt toàn bộ tin nhắn hoặc nhưng message đc chỉ định.

- Số lượng các interceptor có sẵn đang tiếp tục tăng và có thể đóng góp.

Threadless Interceptor stack (Giảm bớt thread Interceptor stack)

- Interceptor không yêu cầu bất kỳ thread riêng biệt để thực hiện thao tác thông báo chung.

- Message được gửi sẽ piggy back trên thread mà đã gửi chúng tới tất cả way thông qua transmission. Exception là “MessageDispatchInterceptor” sẽ xếp trên tin nhắn và gửi nó thông qua 1 thread riêng biệt đổi với các bất đồng bộ message được truyền. Message nhận được quản lý bởi thread pool trong “receiver” component. Kênh object có thể gửi “heartbeat()” thông qua interceptor stack để cho phep timeout, cleanup và 1 số event khác.

Note:

The MessageDispatchInterceptor is the only interceptor that is configured by default.

Parallel Delivery

- Tribes hỗ trợ truyền giao song song cho message. Có nghĩa là node\_A gửi 3 message tới node\_B song song. Tính năng này trở nên hữu ích khi gửi message khác delivery semantics. Bên cạnh đó, nếu Message\_1 được gửi là ordered, Message\_2 sẽ chờ cho đến khi message đó hoàn thành.

- Thông qua NIO (Network Input Output), Tribes cũng cho phép gửi message đến 1 số receiver cùng 1 thời điểm trên cùng 1 thread.

Silent Member Messaging

- Với Tribes, ta có thể gửi message tới member không nằm trong group. Mặc định, ta có thể gửi message thông qua WAN, mặc dù dynamic discover module (module phát hiện động) này được giới hạn trong mạng LAN bằng các sử dụng multicast để tìm các node động. Tính năng gửi mạng WAN sẽ được phát triển thêm trong tương lai. Nhưng nó sẽ rất hữu ích khi ta muốn dấu member khỏi phần còn lại của nhóm.

e