

Sistemas Distribuidos

Tarea # 1

Raúl Monge Anwandter
rmonge@inf.utfsm.cl

Roberto Catricura Silva
roberto.catricura@alumnos.usm.cl

17 de Agosto de 2016

1. Introducción

En el modelo de sistemas distribuidos los componentes ubicados en una red de computadores se comunican y coordinan sus acciones a través del paso de mensajes, encontrándose en una gran cantidad de servicios, por ejemplo: world wide web, juegos en línea, email, redes sociales, etc. A manera de comunicar procesos o computadores se utilizan transmisiones de tipo Unicast, en la cual se envían mensajes a una dirección única (UDP/TCP) o Multicast donde un solo mensaje es capaz de llegar a varios receptores (one-to-many o many-to-many) a través de la subscripción a un segmento IP específico. Ambos tipos de comunicación hacen uso de Sockets, una abstracción la cual permite la comunicación entre procesos utilizando su dirección IP y un puerto.

En esta actividad los alumnos podrán familiarizarse con el uso de sockets para comunicar procesos, utilizar transmisiones de tipo Unicast y Multicast con diferentes propósitos y solucionar problemas relacionados a la sincronización entre clientes y servidores.

2. Objetivos

- Experimentar con programación con socket y uso de transmisión, tanto Unicast como Multicast.
- Reconocer algunos problemas propios de la programación distribuida y las dificultades en el diseño de servicios distribuidos que requieren la presencia de varios servidores.

3. Actividades

3.1. Motivación

Como se mencionó anteriormente, existe una gran cantidad de aplicaciones que utilizan modelos distribuidos para entregar diferentes tipos de servicios, entre estos se encuentran los juegos en línea: los cuales requieren mantener una gran cantidad de clientes de manera concurrente e intercambiando una gran cantidad de datos para lograr la propagación de eventos y mantener una visión del mundo virtual consistente y compartida.

La actividad consistirá en desarrollar una arquitectura que sea capaz de mantener esta clase de servicio, específicamente el videojuego **Distribumon Go**, el cual consiste en la captura de criaturas a través de múltiples regiones en el mundo. Para esto se propone la utilización de diferentes servidores, los cuales tendrán que administrar una región específica operando de manera individual; de esta manera, se podrá distribuir la carga a través de varios servidores, sin la necesidad de interactuar entre ellos o un servidor principal.

3.2. Arquitectura

Una de las características principales es que Distribumon Go mantiene instancias aisladas para cada una de las regiones que componen su mundo. Por lo tanto, es posible distribuir estas regiones sin necesidad de intercambiar mensajes entre ellas, la interacción se realiza solamente entre servidor y cliente (y viceversa), lo que simplifica

bastante su implementación. Aún así, es necesario mantener un servidor principal el cual administra de manera transparente para el usuario las direcciones de estos servidores.

3.2.1. Requerimientos

- **Servidor Central:** servidor único quien coordinará con el cliente la comunicación entre éste y las diferentes zonas del juego, debiendo encargarse de entregar al cliente la información necesaria para interactuar con el servidor de zona requerido. En específico debe proveer:
 - Nombre de Zona: nombre por el cual el cliente identificará la zona (ejemplo: Casa Central, San Joaquín).
 - Dirección IP del grupo Multicast: direcciones válidas del segmento Multicast (ejemplo: 224.0.2.3)
 - Dirección IP de recepción: dirección IP donde correrá el proceso encargado de la recepción de peticiones para la captura.
 - Puerto de recepción: puerto que escuchará el socket para la recepción de peticiones.
- **Servidores de Zona:** existirá un servidor de este tipo por cada zona, el cual administra para una zona individual, la difusión de las criaturas se realizará a través de Multicast. De esta manera, múltiples clientes podrán ser informados sobre eventos en la región utilizando una mínima cantidad de mensajes. Este servicio deberá escuchar peticiones de clientes en un puerto específico a través de socket (UNicast UDP) para realizar la captura de criaturas. Éste debe contar con las siguientes opciones:
 - Inicializar el servidor utilizando una dirección Multicast y un puerto requerido por pantalla.
 - La capacidad de instanciar Distribumones en la zona, éstos al menos deben contar con un id local, nombre y nivel.
 - Enviar Periódicamente el estado de Distribumones en la zona.
 - Responder a peticiones de captura de Distribumones en la zona. Para esto debe aceptar la primera petición que reciba, descartando otras que hagan referencia al mismo id e informando a través de Multicast la captura; se debe remover a la criatura capturada de la zona, actualizando su estado por Multicast.
- **Clientes:** los cuales podrán conectarse a los Servidores de Zona a través de un Servidor Central. Para esto el cliente conoce con anterioridad la dirección IP del Servidor Central y que puerto escucha. Éste debe contar con la capacidad de:
 - Cambiar de zona: a través de el servidor central debe dejar el grupo multicast actual y subscribirse a uno nuevo.
 - Sincronizar el estado de Distribumones utilizando los mensajes periodicos enviados por el Servidor de Zona
 - Listar Distribumones en la zona: una lista simple mostrando los datos de cada Distribumon presente en la zona.
 - Capturar Distribumon: utilizando el id de los distribumones presentes en la zona debe enviar una petición al servidor de zona en la IP y puertos definidos, de ser exitoso debe pasar a la criatura capturada a una lista local.
 - Listar Distribumones capturados: una lista simple mostrando los datos de cada Distribumon capturado (sin importar la zona)

3.2.2. Restricciones

- El cliente solo conoce la IP y el puerto del servidor central, debe obligatoriamente preguntar a éste cada vez y conectarse automáticamente a los servidores de zona solamente utilizando el nombre de la región.
- El servidor de zona debe mantener la consistencia entre todos los clientes que están suscritos al grupo Multicast, inclusive si éstos se suscribieron después de la instanciación de la criatura.
- El output tanto de servidores como clientes debe seguir el formato de la sección Ejemplo de Ejecución.

3.3. Ejemplo de Ejecución

Servidor central registra zona

```
[Servidor Central]: Agrega Servidor de Zona
[Servidor Central]: Nombre:
>Casa Central
[Servidor Central]: IP Multicast
>224.0.2.3
[Servidor Central]: IP Peticiones
>192.168.0.102
[Servidor Central]: Puerto Peticiones
>8889
```

Comunicación entre cliente y servidor central

```
[CLIENTE] Ingresar IP Servidor Central
>192.168.0.102

[CLIENTE] Introducir Nombre de Zona a explorar, Ej: Casa Central, San Joaquin
>Casa Central
```

Respuesta servidor central

```
[Servidor Central]: Respuesta a /192.168.0.102 por Casa Central
[Servidor Central]: Nombre: Casa Central, IP Multicast: 224.0.2.3, IP Peticiones: 192.168.0.102,
Puerto Peticiones: 8889
```

Opciones luego de completar la conexión a una zona

```
[CLIENTE] Consola
[CLIENTE] (1) Listar Distribumones en Zona
[CLIENTE] (2) Cambiar Zona
[CLIENTE] (3) Capturar Distribumon
[CLIENTE] (4) Listar Distribumones Capturados
```

Inicialización servidor de zona

```
[SERVIDOR ZONA]Nombre Servidor
>Casa Central
[SERVIDOR ZONA:Casa Central]IP Multicast:
>224.0.2.3
[SERVIDOR ZONA:Casa Central] Puerto Multicast
>9999
[SERVIDOR ZONA:Casa Central] IP Peticiones
>192.168.0.102
[SERVIDOR ZONA:Casa Central] Puerto Peticiones
>8889
```

Publicación de distribumon RMIchu en zona Casa Central

```
[SERVIDOR ZONA:Casa Central] Publicar Distribumon
[SERVIDOR ZONA:Casa Central] Introducir nombre
>RMIchu
[SERVIDOR ZONA:Casa Central] Introducir nivel
>22
[SERVIDOR ZONA:Casa Central] Se ha publicado al Distribumon: RMIchu
*****
id: 0
nombre: RMIchu
nivel: 22
```

[CLIENTE] Aparece nuevo Distribumon!: RMIchu
--

4. Consideraciones

- Se realizará una ayudantía con el fin de presentar y resolver dudas de la tarea, tanto para Casa Central como San Joaquín, fecha y horario a coordinar por moodle
- Consultas sobre la tarea se deben realizar en moodle o enviar un correo a **roberto.catricura@alumnos.usm.cl**

5. Reglas de la entrega

- La tarea se realiza en grupos de 2 personas.
- La fecha de entrega es el día **23 de septiembre**.
- Se entregará acceso a máquinas virtuales para el desarrollo de la tarea, su evaluación será en base al estado de la máquina el día de la entrega.