

## Proyecto Tracker BitTorrent [65h - 45% calificación]

El objetivo final de este proyecto es el desarrollo de un sistema distribuido que implemente la funcionalidad de un clúster de trackers de bitTorrent. El proyecto se organizará en varias entregas en las que pondrás en práctica diferentes aspectos teóricos de la asignatura.

Los principales requisitos funcionales del sistema distribuido serán 3:

- **Funcionalidad de Tracker bitTorrent:** El tracker debe implementar la especificación del protocolo bitTorrent UDP<sup>1</sup> y la extensión correspondiente a la lista compacta de peers<sup>2</sup>.
- **Almacenamiento de información persistente:** Cuando esté activo, el tracker debe almacenar los datos correspondientes a los ficheros .torrent y peers asociados a cada uno de ellos en un repositorio de información persistente. Se recomienda utilizar una BBDD sqlite o una estructura de ficheros XML (para simplificar la gestión del gestor de BBDD), pero podría usarse cualquier otra tecnología.
- **Gestión redundada de trackers:** El tracker, y toda la información que gestiona debe estar redundado. Dicha redundancia tiene que ser totalmente transparente para los peers.
  - Las diferentes instancias formarán parte de un grupo multicast. De esta forma, los peers se comunican con la dirección IP grupal y las peticiones de los peers llegan a todos los trackers. A pesar de la redundancia de trackers, a todos los efectos, los peers deben creer que sólo hay uno. Por lo tanto, deben recibir la respuesta de un único tracker.
  - El esquema de redundancia contemplará una instancia principal (Master) y el resto serán copias idénticas (Slaves). En cada instante sólo existe un Master y la única labor de los Slaves será garantizar la integridad de la información de los peers y estar pendientes ante un fallo en el Master. El proceso de elección del Master se realizará utilizando un algoritmo de elección (por ejemplo el algoritmo bully<sup>3</sup>). Dicho proceso se realizará al iniciar el sistema y ante un fallo en el Master. Cuando sólo haya una instancia de tracker activa, ésta será el Master. Cada tracker posee un ID numérico que se utiliza como referencia durante el proceso de elección del Master. Cada vez que se inicia una instancia de tracker debe tener un ID distinto a cualquier otro activo.
  - Todas las instancias implementarán un sistema de tolerancia a fallos basado en un mensaje de tipo keepalive<sup>4</sup> que enviarán en base a un intervalo prefijado (por ejemplo, 1 seg.). Si dicho mensaje no se recibe en 2 intervalos del tiempo prefijado, se supone que la instancia no está operativa. Cuando la instancia que deja de estar operativa es el Master, habrá que realizar un proceso de elección para escoger al nuevo Master. Durante el proceso de elección dejarán de atenderse los mensajes de los clientes para garantizar la integridad tanto del sistema como de la información almacenada.
  - Cada tracker dispondrá de un repositorio de información en el que almacenará la información de los enjambres activos o swarms mientras el tracker está activo. Todas las instancias deben tener sincronizado su repositorio de información. En este sentido, se

---

<sup>1</sup> UDP Tracker Protocol for BitTorrent. URL: [http://www.bittorrent.org/beps/bep\\_0015.html](http://www.bittorrent.org/beps/bep_0015.html) [Fecha de consulta: 07-2016]

<sup>2</sup> Tracker Returns Compact Peer Lists. URL: [http://www.bittorrent.org/beps/bep\\_0023.html](http://www.bittorrent.org/beps/bep_0023.html) [Fecha de consulta: 07-2016]

<sup>3</sup> Bully Algorithm. URL: [https://en.wikipedia.org/wiki/Bully\\_algorithm](https://en.wikipedia.org/wiki/Bully_algorithm) [Fecha de consulta: 07-2016]

<sup>4</sup> Keepalive. URL: <http://en.wikipedia.org/wiki/Keepalive> [Fecha de consulta: 07-2016]

implementará un sistema de consenso para realizar la actualización de los datos. Cada vez que se recibe un mensaje de un cliente que suponga una actualización del repositorio los datos no se harán persistentes hasta que el Master lo indique. Este sistema de consenso debe funcionar de acuerdo al siguiente esquema:

1. Cuando el Master detecta que es necesario actualizar el repositorio de información envía un mensaje a todos los trackers preguntando si están preparados para almacenar la información.
2. Cada tracker responde OK u ERROR dependiendo de si está preparado para almacenar o no. Si la respuesta mayoritaria es OK, el Máster envía un nuevo mensaje para que todos actualicen el repositorio. Si la respuesta mayoritaria es ERROR, se descarta la información y se gestiona de forma adecuada la respuesta al peer.
3. Después de recibir las respuestas de los trackers activos, el Master envía un mensaje para almacenar la información de manera persistente o para abortar el proceso.

Además, cada vez que se inicia una nueva instancia de tracker es responsabilidad del Master la transmisión de la información actual del repositorio de información para garantizar en todo momento que todos los trackers tienen la misma información.

Para poder implementar la funcionalidad descrita, será necesario disponer de **dos canales de comunicación diferenciados**:

- Un canal de comunicación a través del que se recibirán las peticiones de los peers y se enviarán las respuestas de acuerdo al protocolo definido por la especificación de bitTorrent. En este caso se utilizarán comunicaciones UDP tal y como se definen en el protocolo tracker-peer.
- Un canal de comunicación a través del que se intercambiarán los mensajes relacionados con la gestión de la redundancia de trackers y la persistencia de información. Este canal de comunicaciones estará basado en JMS.

Además de los dos canales de comunicación será necesaria la **definición de varios hilos de ejecución** para que el tracker pueda realizar varios procesos en paralelo.

Por último, cada tracker debe tener una interfaz gráfica con al menos la siguiente información, distribuida en pestañas:

- Información de configuración para inicializar el tracker: IP y puertos de comunicaciones, ID; así como un botón para iniciar/parar, e incluso provocar un error por pérdida de conexión (especialmente interesante para simular la caída del Máster).
- Información de las diferentes instancias de tracker que están activas, con su correspondiente ID, el detalle de quién es el Master y cuándo se recibió el último mensaje de keepalive. Esta lista se refrescará de forma automática. La información de los trackers activos no se hará persistente.
- Información de todos los swarms activos con: nombre del contenido, tamaño, número total de seeders y leechers, y listado de peers con sus IDs, ip+puerto e información del estado de la descarga (cantidad de bytes descargado/pendiente) para cada uno de ellos. Esta información también se refrescará de forma automática. La información de los swarms debe hacerse persistente únicamente mientras el tracker esté activo de tal forma que al cerrarse el tracker se vaciará el repositorio de información por completo.