PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2523 Sistemas Distribuidos (II/2017)

Actividad 3

Raimundo Herrera (rjherrera@uc.cl)

1 Discusión IPFS

1. A continuación se presentan ciertas desventajas que presenta el protocolo `HTTP` frente a `IPFS`, y la finalmente sobre necesidad de una renovación al protocolo.

   - El protocolo `HTTP` es un protocolo totalmente centralizado, es decir existe un ente que posee cierta información y si se desea acceder a ella, de todas maneras se debe pasar por él. Por ejemplo, para descargar un archivo de un sitio web, se debe descargar de su servidor, o de las copias que él explícitamente disponga, es decir, un nodo central es quien controla la disponibilidad. Por esa misma razón, ese protocolo es poco tolerante a fallas o poco resiliente, ya que un nodo es el que posee la información requerida y entonces la existencia de dicha información depende en su totalidad de la disponibilidad del nodo. Por su parte, en `IPFS`, al distribuir el almacenamiento de la información, sucede que la disponibilidad ya no depende de un nodo, sino que es posible realizarla desde cualquier nodo que haya descargado la información.
   - Por otro lado, cuando se desea servir alguna información, digamos un archivo, a través del protocolo `HTTP`, se requiere contratar un servidor donde alojarla para que quede disponible. Este servidor, dependiendo de las necesidades, debe ser capaz de responder a la demanda de dicho archivo. Esto implica un costo que no es despreciable, y que puede aumentar mucho a medida que la demanda crece. Sin embargo, en `IPFS`, como se distribuye la labor de servir, la disponibilidad del archivo va creciendo a medida que más nodos descargan y sirven dicho elemento, de este modo, el sistema va escalando sin necesidad de añadir un costo adicional asumido por un único nodo centralizado.
   - Del mismo modo, al existir costos permanentes de alojamiento al servir las cosas bajo el protocolo `HTTP`, es posible que exista una "fecha de caducidad" en los elementos que uno disponibiliza, esto hace que, cuando alguien decide dejar de compartir un archivo, o deja de contratar los servicios de un servidor, este archivo deje de estar disponible para toda la red, y de cierto modo, hay una unilateralidad de las decisiones, ya que se depende de un ente central. En cambio, en `IPFS` para que se indisponibilice un archivo, se requeriría que todos quienes tengan una copia del elemento dejen de servirla/la borren, lo que, asumiendo una gran cantidad de nodos, se vuelve virtualmente imposible.
   - Muy ligado a lo anterior, esto permite que la censura sea muy sencilla de realizar ya que en `HTTP` basta con retirar el servicio del nodo central, para que ya no sea accesible en la red, en cambio en `IPFS`, como se dijo, se requiere una acción muy impracticable.
   - Desde un punto de vista de la eficiencia, en `HTTP` las tasas de descarga dependen absolutamente (y están limitadas) por el ancho de banda que ofrece el servidor, en cambio en `IPFS` depende de las de los nodos, y además es posible descargar trozos de los archivos de distintos nodos, diversificando la carga en distintos miembros de la red, y no cargando únicamente a uno. También, desde ese mismo punto de vista, en `IPFS` se toma en cuenta la duplicación, por lo que se evita la misma

para ahorrar espacio, así en el macro, hay un ahorro potencialmente muy grande, lo que no ocurre en `HTTP`, ya que si hay copias, no se verifica (ni se puede por construcción) si esta ya existe en la red.

- Por último, cuando un sitio bajo el protocolo `HTTP` está alojado en cierta zona geográfica, esto es de cierto modo una limitante a la hora de hablar de tiempos de respuesta. La latencia que existe entre servidor y cliente está dada por eso y es de cierto modo inmutable si se consideran ubicaciones fijas. Sin embargo, en `IPFS`, se puede aprovechar la potencial cercanía geográfica de nodos. Si un nodo está cerca de otro en términos físicos, se insta a que la descarga de archivos se de entre ellos, en vez de ir a un nodo más lejano que los posea, por lo que se pueden ir reduciendo significativamente esos tiempos dados por la distancia al haber una mayor cantidad de nodos.

Las razones anteriores llevan a dejar explícita una necesidad de renovar el protocolo `HTTP` ya que no se están aprovechando las características de un mundo interconectado. Asumir la vulnerabilidad que tiene confiar en el protocolo `HTTP` pudiendo distribuir riesgos y minimizarlos se ve como poco sensato. A día de hoy, es posible sacar partido de la arquitectura de la red de mucho mejor manera que con el tradicional método cliente(s) - servidor.

2. En cuanto a la seguridad, existen potenciales preocupaciones respecto a cualquier implementación de un protocolo donde se vea involucrado el intercambio de archivos. Uno de los primeros aspectos que se pueden observar en cualquier protocolo es el de ver como manejan la suplantación de datos, es decir, el hacer pasar un dato por otro. Sin embargo, en `IPFS`, esto está considerado, y no es posible servir un archivo bajo el falso identificador de otro para lograr distribuir información fraudulenta, virus, etc. Esto porque se implementa *hashing* de los archivos a nivel de bloques del archivo, de modo que si el contenido es distinto, el *hash* será distinto, mitigando el riesgo de *swap* de archivos.

   No obstante, al igual que en `HTTP` existe una cierta necesidad de confianza en quienes distribuyen un contenido. Si alguien comparte un virus en la red, y uno descarga dicho virus, este le va a llegar, íntegro, sin que haya sido intercambiado por otra cosa, pero llegará igualmente, por lo que ese riesgo sigue estando y es un riesgo inherente a internet a mi parecer. La gracia es que no hay posibilidad de un *swap* intermedio, pero si puede llegar un virus si lo que intentó descargar lo era, volviendo a necesitar una confianza en el elemento distribuido.

   Por otro lado, existe un riesgo propio de la estructura de la red. Ya que por esta especie de permanencia que tiene, la cual se discutirá más adelante, el hecho de subir a la red bajo este protocolo un archivo infeccioso, puede ser una acción irreversible, ya que una vez que se empiece a distribuir, este potencialmente quedaría perpetuado en la red.

   Un punto a considerar es que el protocolo `HTTP` en sí no es seguro, es por eso que se creó `HTTPS` paar añadir esta capa de encriptación sobre el mismo. Del mismo modo, en `IPFS` no existe una encriptación de los datos enviados, del tráfico, por lo que, así como está, no es posible confiar en transmitir información sensible, ya que esta podría ser observada a nivel de paquetes por ejemplo por medio de programas como *Wireshark*. Esto en sí es una característica más que una vulnerabilidad, pero debe ser considerado para su implementación.

3. La *web permanente* consiste en la idea de que los sitios web estén disponibles de manera permanente y de cierto modo imposible de borrar. La idea es, si un sitio se sube a la web, que no sea tan simple como eliminar un servidor para que el sitio ya no esté. Lo que sucede hoy, y el gran enemigo de la web permanente, es que el protocolo `HTTP` es muy fragil en ese sentido, lo que produce que la disponibilidad de contenido dependa de un servidor, y sea tan simple como "desenchufarlo" para que ya no esté. `IPFS` tiene la respuesta a eso, si los sitios web son subidos basándose en este protocolo, lo que sucede es que una vez que la gente empieza a acceder a esta información, se va distribuyendo por todos estos nodos, y entonces sucede que se va paulatinamente desechando la dependencia de servidores permantentes y esto hace que, con suficientes nodos sirviendo una web, se haga practicamente imposible la eliminación

de un sitio web. Lo que sucede es que hoy por hoy existen muchos links rotos, y no porque se haya migrado, sino porque se ha dejado de servir, por la razón que sea, esto se puede evitar usando IPFS.

4. Para este caso, encontramos que ambos roles se pueden ver favorecidos desde las distintas perspectivas. Por un lado, cuando uno tiene el rol de traficante, es decir de vendedor de esclavos, puede publicar una venta, por ejemplo un archivo con los datos del esclavo. Este archivo al ser descargado por otros miembros de la red, empieza a estar servido por otros y por ende ahora existe un pseudo-anonimato ya que, si bien es posible averiguar quienes están sirviendo un archivo, no es posible ver quién fue quien lo originó. De este modo, se aseguran los vendedores de que no se les identifique como causantes. Por otro lado, para el comercio de ellos, es positivo al pensar que no es tan sencillo botarles el comercio, ya que al estar distribuido tiene una especie de permanencia mayor, y se vuelve complejo borrar cierta información.

   Para el lado de los miembros del DCC, de modo análogo se vuelve posible identificar a los involucrados, ya que si bien no puedo saber si quien descarga y sirve el archivo es un vendedor, un comprador o un interesado, si puedo saber la identidad de los involucrados y rastreo las IP de los mismos, ya que están *hosteando* el archivo, sin embargo, se vuelve muy difícil eliminar este comercio, ya que el archivo se va perpetuando entre múltiples nodos, y eliminarlos o capturarlos a todos se vuelve cada vez más impracticable.

2 Mala Leche

Para esta sección se utilizó el siguiente comando para obtener los directorios de cartas:

```
ipfs get <directory_hash>
```

Las frases a continuación son una mezcla de 1 carta de las negras y 1 de las blancas para cada combinación, el criterio de su elección fue la que me causó más risa en su primera lectura, el orden no implica mi preferencia.

1. **Fui descubierto por el profesor Cristian Ruz con** *el pedazo de un nodo de Grima en mis manos.*
2. **Mi vida era un agujero horrendo sin esperanza hasta que encontré una** *conspiración para derrocar al decano.*
3. **Toda mi vida he soñado fuertemente con ser un** *hamster radioactivo con distemper.*
4. **Una noche romántica en la Burguesía estaría incompleta sin** *una tarea de autómatas.*
5. **Antes de matarlo señor Bond, permítame mostrarle** *una foto mia disfrazado de Geisha.*

Espero les gusten, sobretodo la **1** y la **4**.