

Proyecto Fase III

Almacenamiento distribuido

1. Descripción

Se desea implementar la abstracción de archivos con un sistema distribuido donde los objetos almacenados son particionados en fragmentos (posiblemente replicados) y almacenados en múltiples nodos diferentes participantes en la abstracción, como se ve en la figura 1.

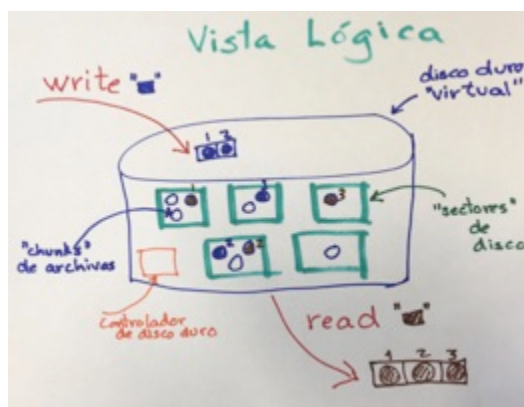


Figura 1. Visión lógica del almacenamiento distribuido a implementar.

2. Detalles técnicos

Establecimiento del almacenamiento distribuido

El establecimiento del almacenamiento se conforma de dos partes: el establecimiento del controlador de topología y la instanciación del grafo de almacenamiento en el que se van a guardar los chunks de archivo. En la explicación que sigue, los colores se refieren a la figura 2, que muestra un ejemplo.

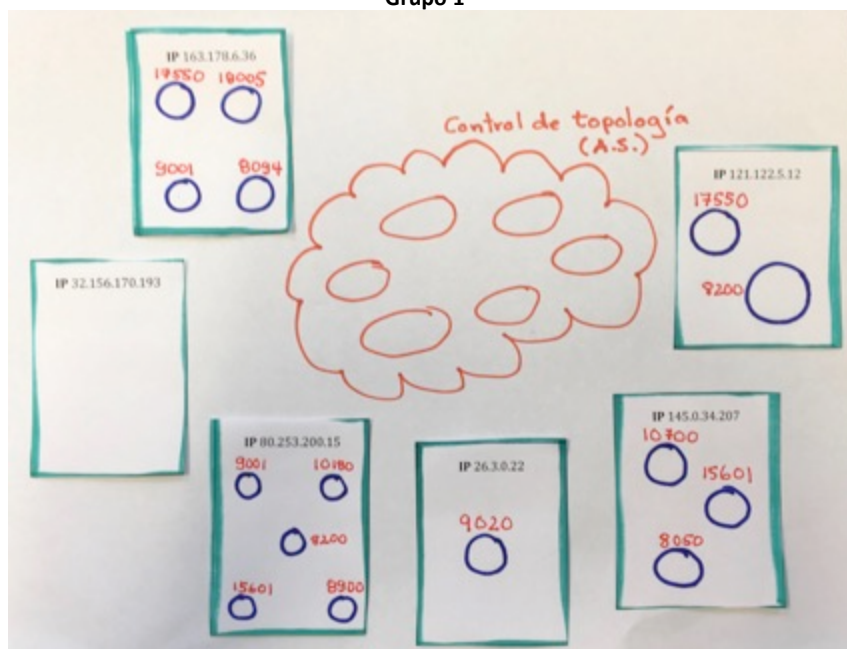


Figura 2. Ejemplo de visión de red del almacenamiento distribuido.

Parte 1. Establecimiento del controlador de topología.

Los controladores de topología (nube anaranjada) forman una red separada con reglas propias. Cada grupo (Sistema Autónomo) tiene al menos un servidor de topología (círculo anaranjado) que provee de servicios a los procesos que almacenan "chunks" de datos (círculos azules).

A cada servidor se le provee un archivo en formato CSV que describe el grafo. Cada línea tiene el siguiente formato:

nombre_nodo, nombre_vecino_1, nombre_vecino_2,..., nombre_vecino_n

Los nombres de los nodos son enteros positivos. Por ejemplo, el grafo de la figura 3 se representa como se indica en la tabla 1.

Grupo 1

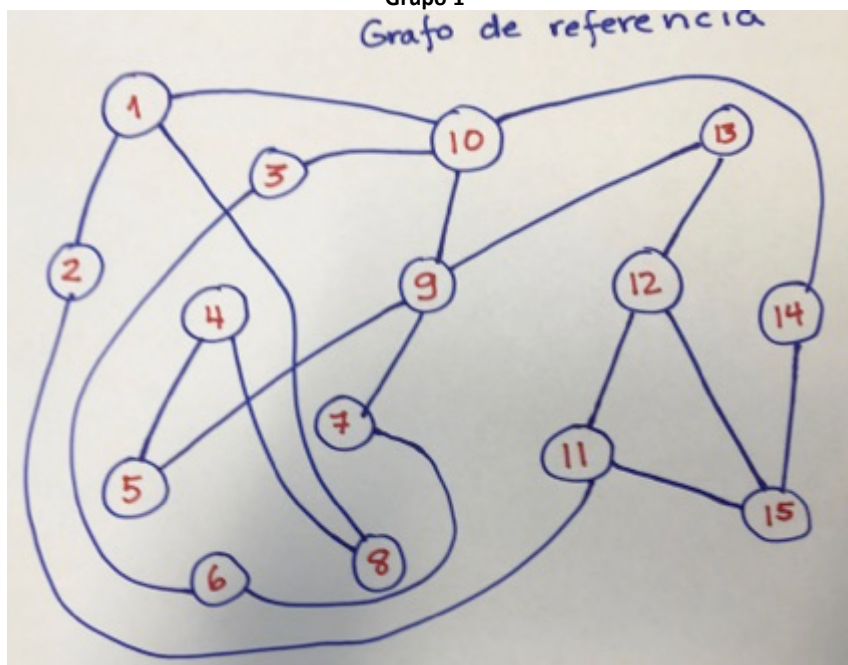


Figura 3. Ejemplo de grafo de referencia

Tabla 1. Representación CSV del grafo ejemplo (Figura 3).

Nodo	Vecinos	Nodo	Vecinos	Nodo	Vecinos
1,	2, 8,10	6,	3,7	11,	2, 12, 15
2,	1, 11	7,	6,9	12,	11, 13, 15
3,	6,10	8,	4, 1	13,	9, 12
4,	5,8	9,	5,7, 10, 13	14,	10,15
5,	4,9	10,	1, 3, 9, 14	15,	11, 12, 14

Los controladores **no** instancian nodos de almacenamiento, lo que hacen es responder solicitudes de incorporación al grafo. Para ello, necesitan encontrar un nombre de nodo que aún no esté siendo ocupado, y devolverle al nodo solicitante la línea correspondiente, esto es, el nombre abstracto del nodo y sus vecinos. A su vez, el nodo solicitante debe informarle al servidor de topología su dirección IP y el puerto en el que va a atender solicitudes. El servidor debe mapear nombres abstractos a direcciones IP y puertos.

Como los servidores consultan sus copias de grafo abstracto de forma asincrónica, una vez que comienzan a asignar nodos, deben asegurarse que únicamente uno de ellos asigne cada nodo abstracto. Para ello, deben correr alguna clase de protocolo que les permita ponerse de acuerdo en quien puede usar un nombre dado, y poder compartir la información de mapeo de nombres asignados.

Parte 2. Instanciación del grafo de almacenamiento.

Los nodos en los que van a estar almacenados los “chunks” de archivo (representados por círculos azules) nacen en alguna máquina (o sea que tienen un IP asociado), y se apropian de algún puerto libre. Sin embargo, para unirse al grafo de almacenamiento distribuido deben contactar a su servidor de control de topología (servidores representados por círculos anaranjados) para que les asignen un nombre y vecinos en el grafo de referencia que manejan los servidores de topología (ver figura 3). A su vez, el nodo azul le

Grupo 1

informa al servidor naranja cuál es su IP y puerto, que se constituirán en sus verdaderos identificadores a partir de este punto. Una vez que se haya integrado completamente al grafo, podrá almacenar chunks de archivos y responder consultas relacionadas con los archivos.

En las figuras 4 y 5 se muestra un ejemplo de como queda la visión de red y el grafo de referencia una vez que todos los nodos azules han sido instanciados.

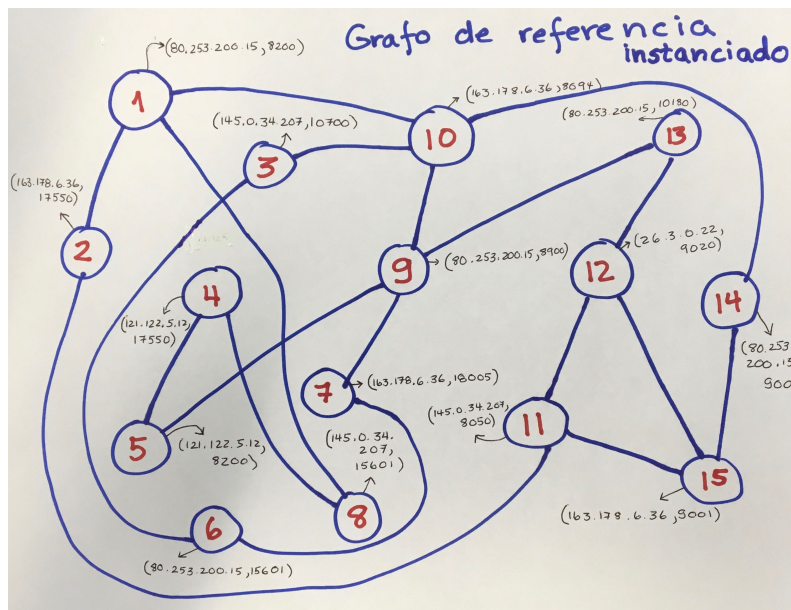


Figura 4. Grafo de referencia ejemplo ya instanciado.

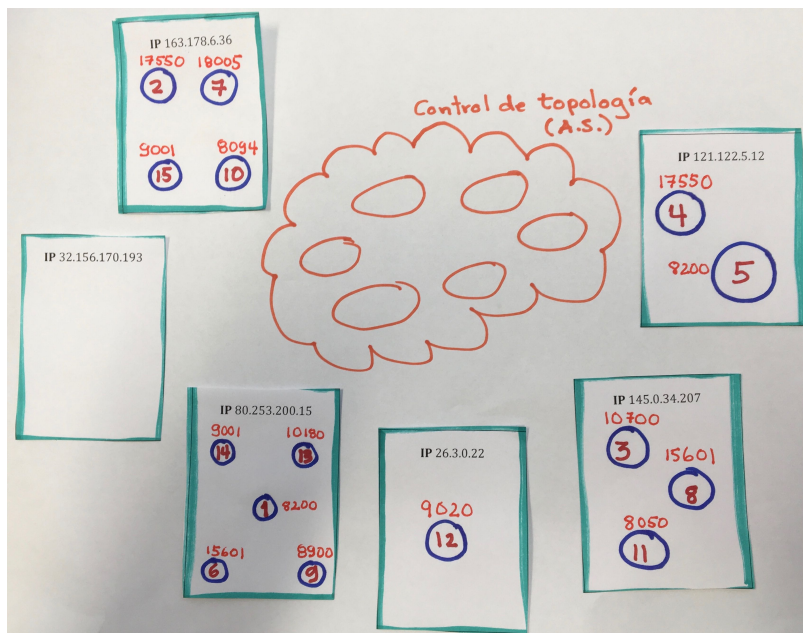


Figura 5. Visión de red ejemplo instanciada.



3. Clientes del sistema de almacenamiento

Existe un tercer tipo de entidad en el diseño, que son los clientes que realizan las acciones de interfaz que permiten el funcionamiento del sistema de almacenamiento. Para simplificar el diseño, estos son clientes “gordos”, en el sentido de que no únicamente proveen una interfaz con el usuario final que le permite especificar los objetos sobre los que se realizarán las operaciones (depositar, encontrar, devolver), sino que se encargan de toda la lógica de particionamiento del objeto en chunks, selección de nodos de inserción, verificación de que la operación se completa, e integración de un objeto a partir de múltiples chunks. Estos clientes son entidades efímeras: esto es, el usuario los invoca para realizar una operación, y terminan luego de que la acción se realiza exitosamente.

4. Acciones que se deben poder solicitar a los clientes

Depositar(objeto)

“Depositar” es la acción que deposita un objeto en el repositorio. Esta acción se le solicita a un cliente, con la referencia a un objeto que se encuentra en el sistema de archivos del computador donde se solicita.

Obtener(objeto)

“Obtener” es la acción de solicitarle a un cliente que nos entregue un objeto que está en el repositorio. El cliente recupera los chunks del grafo, los reensambla en el sistema de archivos del computador en el que está corriendo, y le devuelve al usuario algún tipo de referencia al mismo una vez que está reensamblado.

Existe(objeto)

La función “existe” determina si el objeto está en el grafo. Devuelve verdadero o falso.

Completo(objeto)

Otra función binaria, “completo” revisa los chunks presentes en el grafo y determina si el objeto es reensamblable o no.

Localizar(objeto)

La función “localizar” devuelve la lista (CSV) de nodos azules que contienen chunks del objeto. Extra: Desplegar el “mapa” de azules que contienen el objeto de forma “bonita”, esto es, más allá del CSV.

Eliminar(objeto)

Como su nombre lo indica, esta función elimina todos los chunks del objeto del grafo.

Redistribuya(objeto) [EXTRA]

Al recibir la orden “redistribuya” el grafo azul debe ingeniársela para que los chunks queden en nodos azules diferentes de los que están en el momento en que se solicita.



Matar(nodo)

La función “matar” cierra un nodo de forma ordenada. Deben asegurarse que el grafo de topología se entere de que el nodo ha sido eliminado. Asimismo, esto puede causar que el grafo se desconecte, y que varios objetos se vean afectados.

5. Sobre el manejo de buffers de entrada y salida

En cada nodo azul (nodo de almacenamiento) existen un buffer de entrada por cada vecino y $m=2$ buffers de salida (el número es configurable) que se pueden asignar a los puertos de salida. Una correcta administración de los buffers es parte de lo que deben lograr. Los buffers son más pequeños que los chunks. Cada nodo azul maneja una cantidad restringida de chunks, $n=5$, configurable.

6. Restricciones

Lenguaje de programación: Ninguna restricción. Incluso pueden combinar varios lenguajes para distintas partes de la solución.

Sistema operativo: Únicamente Linux.

Protocolo de capa de transporte a utilizar: Exclusivamente UDP.

Archivo CSV del grafo: No necesariamente va a estar ordenado por nombre de nodo. El grafo siempre va a ser conexo. No contiene errores (p.ej. nodos repetidos, o vecinos inconsistentes). Provisto por los dueños del producto (los profesores).

Almacenamiento de chunks: En el contexto de un nodo azul, los chunks que le corresponde guardar se deben almacenar en el sistema de archivos (**no** en memoria) del computador en el que estén corriendo.

Redundancia: Los chunks deben estar replicados con algún grado de redundancia. Ningún objeto debe estar representado por únicamente una copia de cada chunk.

Unicidad de nombres: Aunque los usuarios pueden sugerir nombres no únicos para los objetos, es responsabilidad del sistema asignarles nombres únicos para poder distinguirlos. El usuario debe conocer el nombre único con el que el objeto se almacenó en el grafo si desea recuperarlo posteriormente.

Comunicación: Toda comunicación en el grafo azul debe llevarse a cabo mediante la topología del grafo.

Tamaño de buffers: El tamaño es configurable, pero **fijo**.

Tamaño de los chunks: 10 Kbytes, configurable.

7. Posible protocolos y negociaciones

Cuando menos, los siguiente protocolos deben ser definidos entre todos los grupos:

- Coordinación entre servidores de topología (nube anaranjada).
- HELLO entre vecinos durante la fase de inicialización del grafo, dado que uno o más de los vecinos pueden no existir cuando un nodo obtiene una posición en el grafo
- KEEPALIVE entre vecinos durante el estado estable, que permita determinar si el vecino está activo o ha desaparecido



UNIVERSIDAD DE COSTA RICA
Escuela de Ciencias de la Computación e Informática
CI-0123 Proyecto Integrador de Sistemas Operativos y
Redes de Comunicación de Datos
Grupo 1



8. Reglas de entrega

Primera entrega: viernes 24 de mayo de 2019

Deben entregar lo siguiente:

- Propuesta de diseño de la solución (incluye detalles de los protocolos que se van a requerir)
- Fechas y contenido de al menos dos entregas antes de la entrega final.

Entrega final: viernes 5 de julio de 2019

Los contenidos específicos se detallarán posteriormente.

Revisión de funcionamiento global: martes 9 de julio de 2019