

# Exclusión Mutua Centralizada y Distribuida

Felipe Samur Sansur

201673536-4

jorge.samur@sansano.usm.cl

Jorge Díaz Matte

201673559-3

jorge.diazma@sansano.usm.cl

3 de diciembre de 2020

## ¿Qué se hizo?, ¿Cómo se hizo?

Los objetivos principales de este laboratorio se centraban en el estudio de algoritmos de coordinación en sistemas distribuidos, considerando para ello los enfoques basados en Exclusión mutua centralizada y Exclusión Mutua Distribuida, de ahora en adelante EMC y EMD respectivamente.

Para lograr dichos objetivos se propuso administrar el manejo de existencias digitales de una red de bibliotecas. Considerando que mas de un nodo en la red no puede acceder a un mismo registro a la vez, se debió trabajar los algoritmos de coordinación y se consideró el de EMC y EMD.

En terminos tecnicos, para el desarrollo de esta labor se trabajo con el lenguaje de programación Go y para establecer las comunicaciones se usó gRPC.

En la red figuran tres tipos de nodos; Cliente, DataNode y NameNode. El nodo Cliente se encargará de abastecer a la red con archivos; esto lo hará enviando el archivo por partes (*chunks*) de 250 KB a un DataNode en particular. Asimismo, este nodo también tiene la tarea de recuperar estos chunks tras la distribución de estos en la red, además de unirlos finalmente en un solo archivo, idéntico al original. El nodo DataNode tiene como cometido principal almacenar aquellas porciones de archivo enviadas por el nodo Cliente. Por otra parte, debe entregar una propuesta de distribución de estos chunks, en pos de cargar la red de chunks que deben ser recuperados por un cliente. Dependiendo de la coordinacion escogida, la propuesta debe ser aprobada por los otros nodos DataNode o por el nodo NameNode. Finalmente, el nodo NameNode es el delegado de almacenar un `log` de las distribuciones aceptadas, para que eventualmente el nodo Cliente pueda saber donde buscar cada parte del archivo que requiere.

Para lograr una coordinación con EMC, un nodo Cliente envía chunks a un nodo DataNode. Dicho nodo propone una distribución al nodo NameNode, el cual debe decidir sobre esta propuesta. Una vez dicha propuesta es aceptada, el NameNode la almacena en su `log` y el DataNode distribuye las partes. Finalmente el Cliente accede al nodo NameNode para conocer la distribución que debe seguir para obtener los chunks del archivo buscado, para finalmente rearmarlo. Por otra parte, el caso de EMD funciona de manera similar, excepto que en este caso las propuestas de distribución se deciden entre los nodos DataNode, y el nodo NameNode solo se ocupa de dejar registro de aquella

## Resultados

Una vez ejecutados ambos algoritmos se llegó a los siguientes resultados:

### Algoritmo de Coordinación con Exclusión Mutua Centralizada

Peso Archivo [MB]	Número de Mensajes Enviados	Tiempo de Ejecución

Cuadro 1: Datos de Ejecución de un algoritmo de Coordinación con EMC

### Algoritmo de Coordinación con Exclusión Mutua Distribuida

Peso Archivo [MB]	Número de Mensajes Enviados	Tiempo de Ejecución

Cuadro 2: Datos de Ejecución de un algoritmo de Coordinación con EMC

## Análisis

## Discusión

Previo a trabajar con los resultados, es importante saber lo que la teoría nos dice de los algoritmos de coordinación. Por un lado la EMC trae consigo una fácil implementación e involucra tan solo 3 mensajes por cada acceso. Sin embargo, tiene un punto unico de falla, en el caso del laboratorio seria el nodo DataNode que almacene en primera instancia todos los chunks. Por otra parte la EMD usando el algoritmo de Ricart y Agrawala tiene como ventaja que su retraso de sincronización es solo del tiempo de transmisión del mensaje, no tiene un delay de ida y vuelta.

## Conclusión

Basandonos en lo visto en clases y lo expresado en la Discusión, de haber obtenido resultados, lo esperado habria sido que el algoritmo de Ricart y Agrawala hubiese sido mas eficiente que solo EMC.