Proyecto Final de Curso

Aplicación Distribuida y Paralela

## Propuesta básica:

El sistema simulará un pool de minería de Bitcoins.

En este, los clientes se conectarán con un nodo central, el cual tendrá un bloque de tareas. Dicho nodo central enviará una tarea a cada cliente conectado y éste utilizará su GPU para realizar la tarea recibida del nodo central.

Una vez resuelto el problema, el cliente le enviará su respuesta al nodo central y este se encargará de guardar la información de dicha operación y por quien fue resuelta. A continuación el nodo central procederá a enviar una o varias nuevas tareas al cliente.

Cuando todo el bloque de tareas sea completado, el nodo central dará un premio a cada cliente en base a la cantidad de tareas resueltas, y proseguirá a repartir tareas de otro bloque.

## Funciones que realizará el sistema:

#### Nodo central:

* Debe ser capaz de leer (BD, archivo) o crear bloques de tareas.
* Debe ser capaz de aceptar varios clientes en simultáneo, y repartirles tareas de manera distribuida.
* Debe ser capaz de registrar nuevos usuarios y permitir su logueo.
* Debe ser capaz de recibir el resultado de una tarea realizada por el cliente, y guardar la información pertinente de dicha operación.
* Debe ser capaz de validar la respuesta recibida por el cliente.
* Debe medir el tiempo de respuesta de un cliente, para luego determinar si le envía una tarea nueva o un conjunto de ellas.
* Debe ser capaz de verificar si todo el bloque de tareas fue resuelto y crear o leer otro.
* Debe ser capaz de verificar que el cliente está trabajando en la tarea y no se ha desconectado. Si este se desconecto, deberá asignarle la tarea a otro cliente.
* Contará con un servidor de backup el cual estará a disposición de este ante cualquier fallo.

#### Cliente:

* Debe ser capaz de conectarse al nodo central y pedirle tareas
* Debe loguearse.
* Debe ser capaz de recibir una o más tareas, y ejecutarlas.
* Debe ser capaz de devolverle tareas sin resolver al nodo central.
* Debe ser capaz de utilizar su GPU para resolver la tarea de forma paralela.

## Diagrama1.png

## Aspecto distribuido:

El aspecto***distribuido*** se verá reflejado en la distribución de tareas del nodo central a los clientes.

El nodo central será capaz de aceptar varias conexiones de clientes y distribuir tareas entre ellos.

## Aspecto paralelo:

El aspecto ***paralelo*** consistirá en realizar la tarea en la GPU del cliente, de esta forma en una misma unidad de tiempo se incrementa considerablemente la posibilidad de encontrar la respuesta a la tarea.

Se presentan diferentes casos para la utilización del paralelismo:

1- Que cada cliente reciba una única tarea y esta sea ejecutada en cada core de la GPU, es decir que se ejecutará la misma instrucción en todos los cores del GPU sobre un dato, permitiendo así que todos trabajen sobre la misma tarea generando mayor efectividad a la hora de obtener el resultado buscado.

2- Que cada bloque de tareas sea para un cliente, y este divida las tareas del bloque entre los core de su GPU.

3- Que un bloque de tareas sea para varios clientes, y estos dividan las tareas entre los core de su GPU.

Sin embargo en este trabajo se realizará la alternativa número 1.

Los tiempos y cantidad de intentos para terminar el bloque se comparan con otras dos pruebas:

1- En lugar de usar la GPU se utilizará la CPU de cada cliente, de esta forma se compara el rendimiento entre GPU y CPU no paralelizada.

2 - En vez de usar la GPU se utilizará la CPU, pero en este caso, el código se paralelizará usando OpenMP, de tal forma que se aprovechen todos los núcleos de la CPU. Esto servirá para comparar la velocidad entre CPU y GPU paralelizadas.

En cuanto al servidor de backup, este se estará actualizando en base al nodo central, y estará constantemente chequeando que dicho nodo central esté levantado. En caso de que el nodo central se haya caído, surgen diferentes posibilidades:

*1ra opción:* Que los clientes se conecten directamente con el servidor backup.

*2da opción:* Que el servidor backup lo reemplace, asignandose la IP que corresponda.

En ambos casos, cuando el servidor caido se recupere, actuara como servidor Backup

## Descripción de la tarea a realizar

Cada tarea estará representada por un array de bytes, cada cliente recibirá una tarea.

Cada core de la GPU de dicho cliente concatenara un nuevo carácter al array de bytes recibido.

Cada vez que se concatena un nuevo carácter se probará que esa combinación sea la correcta.

Una vez que se resuelva la tarea se le enviará el resultado al nodo central y este se encargará de evaluar el resultado.

Tarea original

Caracter

concatenacion

Hash

Hash < valor

no

Nuevo caracter

Fin

Se chequeará que la salida respete la solución buscada, por ejemplo que el valor obtenido sea menor que un valor prefijado.

## Literatura asociada al tema elegido:

<https://www.bitcoinmining.com/>