Task 2: Strong Consistency through 2PC + Lamport Clocks
Distributed Systems 2015-2016

Alumnes: Carlos Villar Robles, Roger Llaveria Gómez
Assignatura: Sistemes Distribuïts

1.- Decisions de disseny preses

Per a la realització d'aquesta segona pràctica hem utilitzat la solució statefull de la primera pràctica.

En aquesta pràctica hem afegit databases.
Aquestes databases estaran conectades amb els servidors i utilitzaran un Two face commit per a indicar si el canvi d'estat es pot realitzar o no en funció de la resposta de les databases.

El two face commit té el següent funcionament:

Fase 1:

Al haver de realitzar un canvi d'estat el servidor coordinador(escollit segons el round robin) indicarà que totes les databases han de realitzar una votació, aquestes databases indicaran si estan d'acord o no de fer el canvi d'estat en funció del lamport actual que reben i el que tenen guardat en les seves variables lamport (seguint la formula: si lamport actual- lamport_database>1 commit, sino abort).
Aquestes votacions seràn guardades en un fitxer individual per a cada databases o s'indicarà l'últim lamport guardat per a que en cas de caiguda es pugui recuperar el valor del lamport clock.

Fase 2:

Segons les votacions realitzades per totes les databases el servidor decidirà realitzar el commit actualitzant el valor lamport guardat a les databases i realitzant el canvi d'estat si totes les databases hi estan d'acord o un abort en cas que alguna database no hi estigui d'acord.

Per a que el funcionament sigui correcte les funcions de les databases han de ser síncrones i el metode push_data del servidor també ha passat a ser-ho ja que sinó els servidor estarien enviant les dades al mateix temps,es creuarien les dades i el funcionament no seria correcte.

Per a la realització del primer apartat nomes utilitzem 1 servidor i 3 databases. En aquest apartat no hi ha problemes d'enviament ja que totes les Streetlight utilitzem el mateix servidor i les tasques es realitzen seqüencialment.

En el segon apartat afegim un altre servidor. Com tenim diversos servidors hi pot haver el problema que un sigui més ràpid que un altre i segurament els lamport clock no siguin correctes. Per això s'utilitza el two face commit perquè en cas que el lamport no sigui el consecutiu es torni a enviar a la cua i sigui comprovat més tard. Anant reenviant a la cua els lamports farà que els lamport clock acabin sent correctes i es realitzi correctament la tasca ja que al realitzar un round robin es possible que la dada entri al servidor més ràpid i sigui tractada.

En el tercer apartat simplement hem hagut d'afegir un retard d'un segon a un servidor qualsevol. Com podem veure el servidor sense el retard anirà més ràpid que l'altre i acabarà fent més tasca que aquest però gràcies al two face commit el funcionament serà el correcte.

En el cas del apartat extra el server podria abans de fer la votació realitzar una consulta per veure si la database dona una resposta, en cas contrari no la tindria en compte per a la votació.
Hem simulat la caiguda de la database posant la variable que indica si esta «viva» a none ja que suposem que és el valor que rebria el servidor si no tingues resposta de la database en la database 2 al arribat al lamport 6

2.- Estructura de la solució

Tenim el mateix fitxer .py de la primera pràctica afegint una nova classes anomenada Database.

Com a fitxer d'entrada tenim el mateix fitxer de text de la primera pràctica «arxiu.txt» amb la successió de valors dels Streetlight i com a sortida un fitxer de text anomenat «ResultatsBX.txt» (on X es el numero de solució (1,2,3,4)) on s'indica els resultat de la compilació .

En aquest fitxer de text es veu l'identificador d'Streetlight i les dades rebudes, en el moment de fer un canvi d'estat s'escriu al fitxer de text.
Per exemple per als 7 primers valors tenim d'entrada:

1 1 1 1 1 0 0
1 1 0 0 0 0 1
1 1 1 1 1 1 1

i de sortida obtenim:

'Streetlight1:', 'ON', '1', '1', '1', '1', '1', '0', '0'

'Streetlight2:', 'ON', '1', '1', '0', '0', '0', '0', 'OFF', '1', 'ON'

'Streetlight3:', 'ON', '1', '1', '1', '1', '1', '1','1'

Les Streetlight aniran guardant en un vector propi la informació que vagi rebent i un canvi d'estat si s'ha de realitzar. Al final del programa aquests vector passaran a una classe que els escriurà en el fitxer de text ResultatsBX.txt

Per pantalla del terminal podem veure quins valors de lamport clock han fet un commit i quins un abort.

Al final de l'execució també es veurà el valor de cada vector de Streetlight al terminal