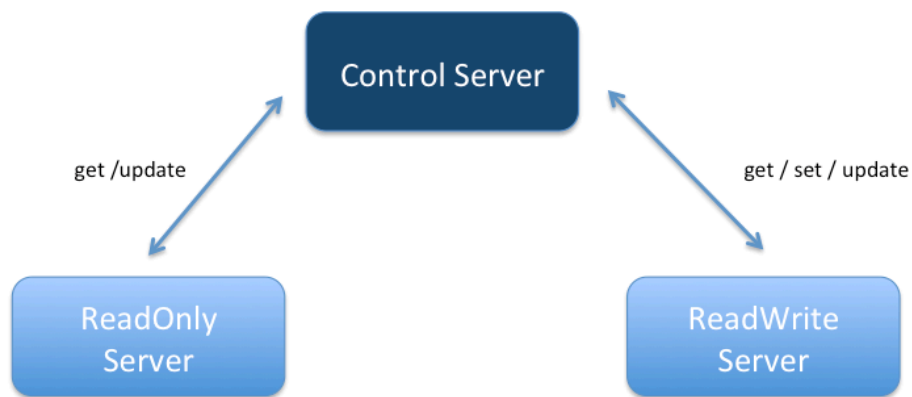


Qüestionari de l'exercici 0:

1-. Explica el funcionament del teu algorisme.

En l'algorisme s'han definit tres tipus de servidors, tal i com es mostra en la següent imatge.



- ControlServer → és el centralitzador de les peticions dels diferents servidors. Aquest mateix serà doncs, l'encarregat de rebre, executar i avisar als altres servidors dels canvis que es realitzen sobre el valor de la variable entera compartida entre ells, per tal de que tots tinguin un valor correcte de la variable abans d'utilitzar-la.
- ReadWriteServer → servidor que llegeix i modifica el valor de la variable entera compartida.
- ReadOnlyServer → servidor que només llegeix el valor de la variable entera compartida.

Les trames que s'han definit per tal de que els diferents servidors puguin comunicar-se entre ells són les següents:

- *id:get:value* → petició de lectura del valor de la variable compartida.
- *id:get:value* → lectura del valor actual de la variable compartida.
- *id:set:value* → petició d'escriptura del valor de la variable compartida.
- *id:update:valor* → nou valor de la variable compartida.

Tots tres tipus de servidors hereten de la classe Server, el qual té com a propietats un Id i estableix el valor a 0.

En inicialitzar el seu funcionament, en funció del nombre de servidors que es defineix (per defecte el número és 5), es configuren els diferents tipus segons el valor de Id que se li assignarà a cada servidor:

- 0 → per defecte sempre serà el ControlServer
- Imparells → definits com a ReadOnlyServer
- Parells → definits com a ReadWriteServer

Així doncs, una vegada configurats tots els servidors, s'espera a que l'usuari inicialitzi el seu funcionament de manera manual, introduint una lletra 'e' per teclat.

Una vegada introduït, cadascun comença la seva rutina de funcionament, que dependrà de la seva configuració:

- Control Server → inicialitza el server socket que permetrà controlar la connexió dels altres nodes i roman escoltant noves peticions per a nous nodes.

Per cada node connectat, inicialitza un nou thread, ControlServerThread, el qual s'encarrega de gestionar les diferents peticions del servidor.

En cas de que la petició sigui de lectura, respondrà al mateix servidor amb el valor de la variable. Per altra banda, si la petició és d'escriptura, modificarà el valor local que té de la variable i enviarà una trama de modificació amb el nou valor de la variable a tots els servidors que té connectats.

Aquesta variable local està protegida amb un ReentrantLock, per tal de que només un fil d'execució la pugui utilitzar a la vegada.

- ReadOnlyServer/ReadWriteServer → primer de tot demanen el valor actual al ControlServer i, a partir de llavors inicialitzen el seu comportament predefinit. A més a més, també inicialitza un altra thread, UpdateThread, que s'encarregarà de rebre les modificacions del valor dels altres servidors del sistema i actualitzar el valor local de la variable. Aquesta variable local, a cada servidor, està protegida amb un ReentrantLock, per tal de que només un fil d'execució la pugui utilitzar a la vegada.

Amb el disseny implementat, tot i que tot el tràfic de peticions l'hagi de gestionar el ControlServer, no cal que cada servidor sigui conscient de tots els altres servidors que estan connectats al sistema.

2-. Quines limitacions li veus al teu algorisme?

El problema principal de l'algorisme és que tot el tràfic d'intercanvi de dades passa pel node central (Control Server), el qual només n'hi ha un. D'aquesta manera, en cas de que el node caigués, deixaria de funcionar completament el servei.

A més a més, no s'està realitzant una gestió de peticions de lectura paral·leles, fet que augmentaria el rendiment de l'aplicació en cas de que la majoria de servidors estiguessin configurats com a ReadOnlyServers.

3-. Com reacciona el rendiment del teu algorisme quan s'incrementa arbitràriament el número de processos?

Amb un augment del número de processos el rendiment de l'algorisme baixaria de manera considerable, ja que s'augmentaria el nombre de paquets que el ControlServer ha de gestionar i reenviar, en cas de que la petició sigui d'escriptura. Si el nombre de processos és molt gran, podria ser que un sol node no fos capaç de gestionar totes les peticions i que es produís un coll d'empolla en la connexió.

4-. Què caldria canviar, o com de complicat seria permetre que nous servidors s'afegissin al sistema a mitja execució?

No seria complicat, ja que el ControlServer està acceptant en tot moment noves peticions, i els nodes connectats els primer que fan es demanar el valor actual de la variable compartida al ControlServer, per començar la seva rutina de funcionament. A partir d'aquest moment, tot node connectat a mitja execució rebrà també les actualitzacions que es produeixin en els altres servidors connectats al sistema, ja que el valor actualitzat que rep és absolut i no relatiu a les modificacions anteriors.