**Muty: a distributed mutual-exclusion lock**

**Pau Alcázar**

**Open questions:**

* **What is the behavior of the lock when you increase the risk of a conflict?**

Quan el risc de conflicte per a un bloqueig augmenta, normalment a causa d'un major nombre de processos que lluiten per la mateixa secció crítica, el comportament del bloqueig pot canviar de les maneres següents:

1. Augment del temps d'espera: els processos poden haver d'esperar més temps per adquirir el bloqueig perquè és més probable que el bloqueig sigui mantingut per un altre procés.
2. Estimulació de recursos: En alguns casos, certs processos poden estar "passats de fam", és a dir, poden haver d'esperar una quantitat inordinada de temps per adquirir el bloqueig, especialment si no hi ha una política d'equitat en vigor.
3. Reducció del rendiment: El rendiment general del sistema pot disminuir perquè es dedica més temps a gestionar la contenció del bloqueig i es dedica menys temps a fer el treball real.
4. Deadlocks: Si el mecanisme de bloqueig no està ben dissenyat, l'augment de la contenció podria conduir a bloqueigs, on dos o més processos estan esperant indefinidament per alliberar panys.

* **Justify how your code guarantees that only one process is in the critical section at any time. ii) What is the main drawback of lock2 implementation?**

El mòdul muty.erl (que sembla canviar el nom a lock1.erl) implementa una forma bàsica d'un bloqueig d'exclusió mútua utilitzant el pas de missatges. Un procés sol·licita el bloqueig enviant un missatge {take, self(), Ref} i espera una resposta {taken, Ref} abans d'introduir la secció crítica. Aquest mecanisme, si està correctament seqüenciat, assegura que només un procés pot entrar a la secció crítica alhora perquè el procés de bloqueig (lock1) no envia un missatge {taken, Ref} a cap altre procés fins que rep i processa un missatge de llançament del blocador actual.

ii) Principal Retrat de la implementació de lock2:

1. Manca d'equitat: el lock2 no implementa un mecanisme de cua o no considera l'ordre de les sol·licituds de bloqueig, podria conduir a la injustícia, on alguns processos accedeixen a la secció crítica més sovint que altres.
2. Problemes de rendiment: La sobrecàrrega associada a la gestió de l'estat de bloqueig i la comunicació entre processos poden provocar colls d'ampolla de rendiment, especialment en escenaris d'alta contenció.
3. Complexitat en la gestió d'errors: gestionar les fallades de node, les particions de xarxa i assegurar que els bloquejos es distribueixin correctament pot afegir complexitat significativa a la implementació.

* **Note that the workers are not involved in the Lam-port clock. According to this, would it be possible that a worker is given access to a critical section prior to another worker that issued a request to its lock instance before (assuming real-time order)?**

En sistemes distribuïts, el rellotge Lamport és un mecanisme utilitzat per determinar l'ordre d'esdeveniments en un sistema que no es basa en rellotges sincronitzats en temps real. És un rellotge lògic que s'incrementa pel procés que el posseeix, cada vegada que aquest procés realitza un esdeveniment. Quan els processos es comuniquen, intercanvien aquests valors de rellotge i actualitzen els seus propis per assegurar l'ordenació.

En absència de rellotges Lamport o una altra forma de cronòmetre lògic o físic, el sistema ha de confiar en l'ordre de pas del missatge i la planificació del procés per determinar quin workers té accés a la secció crítica. Si els workers no estan utilitzant les marques horàries de Lamport o un altre mecanisme de sincronització per ordenar les seves sol·licituds, hi ha diverses possibilitats:

Sense rellotges sincronitzats ni marques de temps lògiques, el sistema pot ser per defecte d'una estratègia first-come-first-served (FCFS) basada en l'ordre en què el servidor de bloqueig rep les sol·licituds. No obstant això, la latència de la xarxa i el mallot poden afectar l'ordre en què arriben les sol·licituds al servidor de bloqueig, el que significa que una sol·licitud enviada abans per un worker pot arribar després d'una sol·licitud enviada més tard per un altre worker.

Si es reben dues sol·licituds gairebé al mateix temps, la planificació i l'ordre de processament del servidor de bloqueig determinaran quina sol·licitud es concedeix primer. Això podria permetre a un worker entrar a la secció crítica abans que un altre worker la sol·licitud del qual es va emetre primer però que va arribar més tard a causa de variacions en la latència de la xarxa o retards en el processament.

Per tant, en absència d'un mecanisme com els rellotges Lamport, és possible que un worker tingui accés a una secció crítica abans que un altre worker que va emetre una sol·licitud abans en temps real. Això es deu al fet que sense un rellotge lògic, no hi ha un acord a tot el sistema sobre l'ordre dels esdeveniments, i "ordre en temps real" no és un indicador fiable a causa de la naturalesa impredictible dels retards del sistema distribuït.