Computación Distribuida 2023-2
- Practica 4 Consenso -

Profesor: Fernando Michel Tavera
Ayudantes: Yael Antonio Calzada Martín y Mauricio Riva Palacio Orozco

Fecha de entrega: Miércoles 19 de abril 2023

1. Descripción Practica

En esta práctica se tendrá que implementar el algoritmo de consenso (el que no tiene terminación temprana).

2. Desarrollo

Esta práctica estará conformada por tres archivos (con sus respectivas interfaces) y un test:

- NodoConsenso.py
- CnalaRecorridos.py (utilizaremos el mismo canal de la practica anterior.)
- Test.py

Tendrás que implementar la clase del NodoConsenso para la ejecución de este algoritmo.

3. Prueba (Test.py)

Estas pequeñas pruebas tienen la finalidad de validar los resultados que sus implementaciones. **Ojo:** que las pruebas unitarias sean pasadas con éxito no implica que su calificación sea la máxima Así que tómenlas como un apoyo para el diseño de sus algoritmos

3.1. Prerrequisitos

Para ejecutar las pruebas basta con ejecutar el siguiente comando en la terminal:

```
pytest -q test.py
```

4. Observaciones

- Respeta los constructores proporcionados.
- Tendremos como parámetro el número de fallos f en la función consenso.
- Por convención llamaremos $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_5, v_6, v_7\}$ y se tiene $f = 4$, entonces los vértice que fallaran serán $\{v_0, v_1, v_2, v_3\}$
- Simularemos los fallos haciendo que los nodos ya no envíen mensajes ni procesen nada del algoritmo. Pueden hacerlos fallar en cualquier ronda que ustedes quieran, pero recuerden que si un proceso falla en la ronda r en todas las demás rondas siguientes fallará.
- Noten que este algoritmo es sensible a la ronda actual. Recuerden que el número de ronda en simpy podemos obtenerlo con env.now. Si no logran hacerlo con simpy, son libres de simular este comportamiento usando solo Python. (El comportamiento que tienen que simular es que los nodos solo ejecuten el algoritmo un número fijo de veces).
- Atributos importantes de un nodo:
  - *V:* corresponde al arreglo de **id's** que tiene el nodo. El pseudocódigo original se nos dice que se trata de la referencia al nodo en cuestión, sin embargo para la practicas solo usaremos los *id's* por facilidad.
  - *New:* el conjunto de mensajes de cada nodo. De nuevo, en el pseudocódigo original el consenso se hace sobre las referencias a los nodos en sí, por lo que basta con que en este conjunto se tengan de igual manera sólo los *id's* de cada uno de los nodos y no las referencias de estos.
  - **lider:** valor del *id*
  - *fallare:* booleano que será verdadero solo si el nodo fallará en algún punto del algoritmo.

5. Lineamientos de entrega

Dado que el esqueleto de la practica esta hecho y se espera que solamente se rellene, se debe entregar cada clase bien documentada, además dentro de la práctica se deberá agregar un ReadMe con el número de la practica, en este caso Practica 4, nombre de los integrantes con su número de cuenta y una explicación de su implementación para cada algoritmo.
Solo un integrante debe subir la practica, el otro integrante con que la marque como entregada antes de la fecha límite es suficiente, además, se deberá subir la carpeta comprimida como un .zip con el nombre "Practica4"

Cualquier duda escríbanme a la brevedad.