Simulador de memoria compartida distribuida

Universidad Nacional de Costa Rica
Escuela de informática y computación
Sistemas Distribuidos

1. Objetivos del proyecto

El objetivo del proyecto consiste en familiarizar a los estudiantes con ambientes distribuidos donde la carga de consistencia de la memoria debe de prevalecer para poder completar las diferentes tareas que se pueden ejecutar con apoyo de una red de computadoras de bajo costo.

2. Descripción del problema

Una de las aplicaciones más comunes en sistemas distribuidos consiste en implementar un DSM (Sistema de Memoria Distribuida por sus siglas en inglés), el cual permite que múltiples páginas sean modificadas simultáneamente por varios sitios, con el fin de agilizar un proceso paralelizable que requiere de mucha memoria y que puede verse beneficiado del trabajo conjunto de un sistema distribuido. Un ejemplo sencillo donde este modelo se puede aplicar sería en los sistemas operativos que digitalizan películas animadas por computadora. Entretanto, podemos contar con descripciones como esta: Wikipedia y un algoritmo puede verse descrito en la clase de esta página

3. Descripciones del simulador

Su misión consiste en tener computadoras que simularán estar corriendo un programa que accesa a diferentes páginas en forma aleatoria y uniforme. Cada vez que una página es solicitada por un sitio, se verificará si la operación que quiere hacer es de escritura o de lectura y si tiene la página en su lista de páginas. Esto puede llevar a 6 posibles estados:

1. Quiere leer, es dueño de la página y tiene la página

   lee

2. Quiere leer, no es dueño de la página y tiene la página

   lee

3. Quiere leer, no es dueño de la página y no tiene la página

   solicita al servidor principal quién tiene la página (dueño) y le pide al dueño una copia

4. Quiere escribir, es dueño de la página y tiene la página

   al ser dueño sabe quién tiene copia de las páginas, pide que las borren y la actualiza

5. Quiere escribir, no es dueño de la página y tiene la página

   Solicita la propiedad al servidor, espera a que el dueño se la pase, el dueño anterior borra las copias (y la suya propia) y le pasa la propiedad a la página.

6. Quiere escribir, no es dueño de la página y no tiene la página

   Los pasos son los mismos que para el punto 3 y el 5.

El siimulador debe de correr en varias computadoras que fingirán ser los nodos, al inicio todas las páginas estarán en una ip dada y en su versión (0) cero, la cual será la del servidor principal. Las demás computadoras van a tener un archivo de configuración en donde se especificará:

- La IP del servidor
- El número de páginas en el sistema
- Media exponencial con que se solicitarán las páginas
- Probabilidad de lectura (la probabilidad de escritura será el complemento con 1)

El proceso que ejecutarán los nodos será sencillo, debe saber cuáles páginas tiene, la versión en que se encuentra cada página (una página cambia de versión cada vez que alguien escribe en ella, sumando 1 a la versión en que se encontraba anteriormente).

Cada nodo debe de generar una página aleatoria, generar el tipo de operación que va a tener (escritura o lectura), si no la tiene, preguntar al servidor la ip del dueño de la páginas y seguir el procedimiento descrito anteriormente. Siempre se guardarà la versión actual de la página, cada vez que una página sea solicitada como copia, se enviará entre nodos el número de versión en que se encuentra la página.

Para verificar el estado de las páginas, cada nodo y el servidor, deberá imprimir en pantalla las páginas que tienen y la versión. En en caso del servidor, además deberá mostrar las IP's de los nodos dueños de una página.

El programa deberá correr indefinidamente hasta que el usuario digite en el servidor e en cuyo caso, deberá informar a todos los nodos que deben cesar de trabajar y luego él mostrará la versión de cada una de las páginas y finalizará su accionar.

4. Evaluación

Su programa debe de cumplir con las siguientes características:

1. Documentación 20 % (se restará un punto por cada falta ortográfica)

   Explicación del problema de las funciones que trabajan correctamente y una justificación de aquellas funciones faltantes.

2. Conectividad entre nodos y protocolo de comunicación 20 %

3. Manejo de las páginas y demostración de su estado en todo momento 60 %

Toda programación debe realizarse en C y correr sobre Linux y en particular en los laboratorios provistos por la universidad.