

Tarea 1

Cruz Perez Ramon -315008148
Marco Antonio Orduña Avila 315019928

Octubre 2020

1. ¿Cuál es la diferencia entre el cómputo paralelo y el cómputo distribuido?

El cómputo paralelo divide un proceso en tareas que son ejecutadas al mismo tiempo para hacer una tarea pesada. (En el mismo hardware)
El cómputo distribuido divide un proceso en tareas que son ejecutadas en diferentes ubicaciones. (En diferentes hardwares)

2. ¿Por qué no hay un único modelo de cómputo distribuido?
Pues como el cómputo distribuido esta siendo ejecutado por varias computadoras el modelo debe puede ser eficiente para n computadoras y diferentes tipos de problemas especificos a resolver, entonces si se cambian en numero de computadoras drásticamente o el tipo de problema, el modelo ya no sera de utilidad como antes.
3. Explica con tus propias palabras qué es un evento.
Un evento es una accion que ejecuta un proceso por ejemplo al enviar o recibir un mensaje en ciertas unidades de tiempo.
4. Diseña un algoritmo distribuido que calcule la distancia entre la raíz de una gráfica y el nodo que se está visitando.

En el algorirmo posterior no se envian mensajes ,pero en el momento de que el contador sube, es como si cada vertice supiera que ya esta contado, es facil ver que es como si se enviaran mensajes de que ya han sido contados y por lo tanto ya se sabe su distancia, se puede hacer otro algoritmo, primero para cada vertice aprenda la distancia de la raíz hacia ese mismo vertice y que se lo vaya pasando a sus vecinos por lo que para cada iteracion del mensaje a los vecinos ya visitados seria $(v_i, \text{dame_tu_distancia} + 1)$, asi que en escencia el algoritmo seria muy parecido.

Algorithm 1 Algoritmo distribuido para calcular la distancia de la raiz a cualquier nodo de una grafica simple G

Require: Lista de vecinos de G, un nodo r distinguido llamado raiz

Ensure: Lista con la distancia de la raiz a cualquier nodo, regresa una lista de duplas (v_i, d) donde v_i son los vertices de la grafica y d es la distancia hacia la raiz

```

1:  $id_i = i$ ;
2: FinalList = {lista final con las distancias}
3:  $neighbors_i = \{ \text{lista de vecinos para cada } v_i \}$ 
4:  $visitados_i = \{r\}$ 
5:  $ListAux_i = \{ \text{una lista auxiliar} \}$ 
6: main()
7: for each  $i \in neighbors_i$  of  $visitados_i$  do
8:   contador = 1;
9:    $listAux_i = listAux_i \cup neighbors_k$  of  $visitados_i$ ;
10:  elimina-repetidos(listAux);
11:   $listAux_i = listAux_i - visitados_i$ ;
12:  FinalList = FinalList  $\cup \{listAux_i \cup contador\}$ 
13:  contador+1;
14:   $visitados_i = visitados_i \cup listAux_i$ 
15: return FinalList;
```

5. Para el algoritmo propuesto en el ejercicio anterior, argumenta cuántos mensajes es necesario enviar para diseminar el mensaje y cuál es el tamaño de cada mensaje enviado. En el peor de los casos se tiene una grafica donde solo sea una linea recta, es decir que el grado max de G sea 2, y que el nodo distinguido o raiz sea uno de los dos extremos, por lo que se necesitaran a lo mas n-1 mensajes para obtener la distancia. el mensaje puede ser de tamaño a lo mas n-vertices, por lo que se puede codificar en log n bits.