Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2024

# ALOJAR OBJETOS DE BASES DE DATOS EN SITIOS DISTRIBUIDOS

**202200389 – Juan José Rodas Mansilla**

## Resumen

Este problema consiste en alojar objetos de bases de datos en sitios distribuidos, de manera que el costo total de la transmisión de datos para el procesamiento de todas las aplicaciones sea minimizado.

Un objeto de base de datos es una entidad de una base de datos, esta entidad puede ser un atributo, un set de tuplas, una relación o un archivo.

El problema de diseño de distribución consiste en determinar el alojamiento de datos de forma que los costos de acceso y comunicación son minimizados.

## Palabras clave

**Bases de Datos:** Sistema que organiza, almacena y gestiona grandes volúmenes de información.

**Tuplas:** tupla es una secuencia ordenada de elementos.

**NP-Hard:** Estos problemas no tienen una solución eficiente conocida y resolver uno eficientemente permitiría resolver todos los problemas NP de manera eficiente.

**Matriz:** Estructura de datos en forma de tabla bidimensional o multidimensional.

## *Abstract*

*This problem involves hosting database objects in distributed sites so that the total cost of data transmission for processing all applications is minimized.*

*A database object is an entity within a database. This entity could be an attribute, a set of tuples, a relation, or a file.*

*The distribution design problem consists of determining the data placement in a way that access and communication costs are minimized.*

## *Keywords*

***Databases**: A system that organizes, stores, and manages large volumes of information.*

***Tuples**: A tuple is an ordered sequence of elements.*

***NP-Hard**: These problems have no known efficient solution, and solving one efficiently would allow all NP problems to be solved efficiently.*

***Matrix**: A data structure in the form of a two-dimensional or multidimensional table.*

## Introducción

El problema de diseño de distribución consiste en determinar el alojamiento de datos de forma que los costos de acceso y comunicación son minimizados. Como muchos otros problemas reales, es un problema combinatorio NP-Hard. Algunas de las situaciones comunes que hemos observado cuando se resuelven instancias muy grandes de un problema NP-Hard son:

Fuerte requerimiento de tiempo y fuerte demanda de recursos de memoria.

Un método propuesto para resolver este tipo de problemas consiste en aplicar una metodología de agrupamiento.

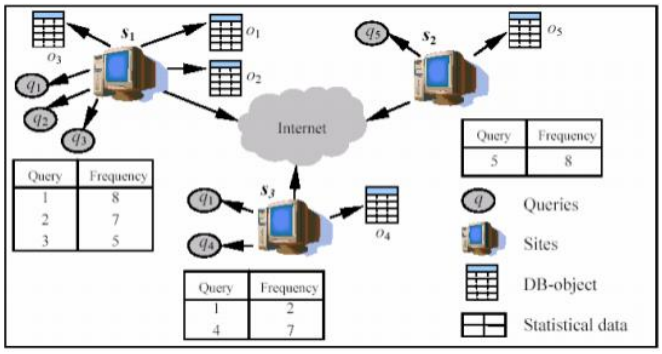


Figura 1. Problema de diseño de distribución en una base de datos

## Desarrollo del tema

El patrón de acceso para una tupla es el vector binario indicando desde cuál sitio la tupla es accedida. Por ejemplo:

Para la siguiente matriz de frecuencia de acceso:

| | | | |
|---|---|---|---|
| 2 | 3 | 0 | 4 |
| 0 | 0 | 6 | 3 |
| 3 | 4 | 0 | 2 |
| 1 | 0 | 1 | 5 |
| 0 | 0 | 3 | 1 |

Figura 2. Ejemplo de formato

Entonces, su correspondiente matriz de patrones de acceso es

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |

Figura 3. Ejemplo de matriz convertida a binario

Se puede observar que las filas 1 y 3 tienen los mismos patrones de acceso, así como también las filas 2 y 5. Entonces, habrá tres grupos considerando el tercer grupo formado simplemente por la fila 4. La cardinalidad de un grupo es definida por el número de tuplas incluidas en él. La matriz reducida de frecuencia de accesos obtenida de la suma de las tuplas en los grupos será:

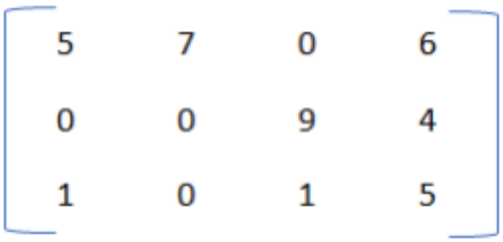


Figura 4. Ejemplo de matriz resultante

Se debe diseñar un programa que acepte “n” matrices de frecuencia de accesos y por cada una obtener los grupos formados por las tuplas con el mismo patrón de acceso. Finalmente, se debe obtener la matriz de frecuencia de acceso reducida.

*Creación de Clases*

| NOMBRE | CATEGORÍA |
|---|---|
| Nodo | Creación de nodos |
| ListasC | Control de nodos |
| Matrices | Control de objetos y funciones claves |

Fuente: elaboración propia, o citar al autor, año y página.

**Nodo:** Un nodo es un componente fundamental en diversas estructuras de datos, redes o sistemas distribuidos. Dependiendo del contexto, puede realizar diferentes funciones:

**Lista Circular:** Es un tipo de estructura de datos abstracta (TDA) basada en listas enlazadas, donde el último nodo está conectado de nuevo al primer nodo, formando un ciclo. A diferencia de las listas enlazadas tradicionales, no tiene un final "nulo" explícito.

**Matriz:** Es una estructura de datos bidimensional que organiza elementos en filas y columnas, permitiendo acceder a ellos mediante índices. Es comúnmente usada en matemáticas, ciencia de datos, gráficos computacionales y programación.
donde:

**Método de lectura de datos:**

Los archivos de entrada y salida consistirán en archivos con extensión y estructura xml en el cual se limitará a utilizar únicamente las etiquetas:

- matrices: este será necesario para la lectura inicial del archivo, ya que será la etiqueta padre de todo.
- matriz: esta etiqueta será la que indica que una nueva matriz de frecuencia de accesos será creada para su respectivo análisis y únicamente puede estar dentro de la etiqueta matrices y puede tener los atributos:
    - nombre: este contendrá el identificador de la matriz leída (se deberá validar la existencia de matrices con el mismo nombre, para mantener la consistencia de los datos).
    - n: representa el número de filas que tendrá la matriz, si los datos dentro de ella son mayores a este atributo o menor a 1 será error.
    - m: representa el número de columnas que tendrá la matriz, si los datos dentro de ella son mayores a este atributo o menor a 1 será error.
- dato: esta etiqueta únicamente podrá estar dentro de la etiqueta matriz y contendrán los valores respectivos a cada celda de la matriz, esta etiqueta puede tener los siguientes atributos.
    - x: será la fila de la matriz y no puede ser mayor al atributo n de la matriz ni menor a 1

    - y: será la columna de la matriz y no puede ser mayor al atributo m de la matriz ni menor a 1
- frecuencia: esta etiqueta representa la frecuencia de los grupos de registros utilizados para crear la matriz reducida.

**Conclusiones:**

**Minimización de costos:** El problema busca optimizar la distribución de datos en sistemas distribuidos, minimizando los costos de transmisión de datos, que es esencial para mejorar la eficiencia del sistema.

**Complejidad NP-Hard:** Debido a que este es un problema NP-Hard, encontrar una solución óptima es computacionalmente complejo, y se requieren enfoques heurísticos, como el agrupamiento de patrones de acceso, para obtener soluciones viables.

**Agrupamiento como solución práctica:** El método de agrupamiento basado en patrones de acceso es una estrategia efectiva para reducir la complejidad del problema al agrupar tuplas con comportamientos similares, lo que permite crear una matriz reducida más manejable y eficiente.

**Uso de XML y Graphviz:** Se utilizarán XML para el manejo estructurado de las matrices de acceso, y Graphviz para una visualización clara de los datos procesados, lo que facilita la interpretación y análisis de la solución.

**Ideas Principales:**

**Método de resolución propuesto**:

- Se propone aplicar un método de agrupamiento para resolver el problema. Esto implica crear una matriz de frecuencias de acceso, transformarla en una matriz de patrones de acceso, agrupar las tuplas que comparten los mismos patrones y, finalmente, generar una matriz reducida de frecuencias de acceso.

**Visualización gráfica**:

- Se utilizará Graphviz para crear un gráfico que represente la estructura del archivo de entrada procesado, mostrando la matriz, sus dimensiones y los valores que contiene.

**Formato de entrada y salida**:

- El formato de entrada y salida debe ser en XML, con etiquetas específicas para organizar la información de las matrices, sus valores y las frecuencias.

**Referencias bibliográficas**
Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems.* Addison-Wesley Publishing Company, Inc.

Cerrada Somolinos, José y Collado Machuca, Manuel (2015). Fundamentos De Programación. Madrid: Editorial Universitaria Ramón Areces.

Quetglás, Gregorio; Toledo Lobo, Francisco; Cerverón Lleó, Vicente (1995). Fundamentos de informática y programación. Valencia: Editorial V.J.