
Proyecto 1

201801105 – Daniel Eduardo Izas Marroquín

Resumen

Se realizó la creación de un programa de consola, para la distribución consistente de alojamiento de datos, para disminución de costos y la comunicación entre sí son mínimas, que nos permitan alojar objetos en un sitio de red o base de datos.

El problema consiste en obtener un nuevo esquema replicado de alojamiento que se adapte a un nuevo patrón de uso de la base de datos y minimice los costos de transmisión. El problema de diseño de distribución consiste en determinar el alojamiento de datos de forma que los costos de acceso y comunicación son minimizados. Para «nt» tuplas y «ns» sitios, el método consiste en tener la matriz de frecuencia de acceso en los sitios F de la instancia objetivo, transformarla en una matriz de patrones de acceso y agrupar las tuplas con el mismo patrón.

El programa consiste en el manejo de matrices que acepten, n matrices de frecuencia de acceso.

Palabras clave

- POO.
- TDA's.
- XML.
- Tuplas.
- Listas.

Abstract

The creation of a console program was carried out, for the consistent distribution of data hosting, to reduce costs and the communication between them is minimal, which allows us to host objects in a network site or database.

The problem is to obtain a new replicated hosting scheme that adapts to a new database usage pattern and minimizes transmission costs. The distribution design problem consists of determining the data accommodation in such a way that access and communication costs are minimized. For “nt” tuples and “ns” sites, the method consists of taking the access frequency matrix in the F sites of the target instance, transforming it into an access pattern matrix and grouping the tuples with the same pattern.

The program consists of managing matrices that accept n access frequency matrices.

Keywords

- POO.
- TDA's.
- XML.
- Tuples.
- Lists.

Introducción

Usando un IDE compatible con el lenguaje de programación Python, creamos un menú de consola donde tenga las opciones tales como cargar archivo, procesar archivo y escribir archivo.

Cada una de estas opciones realizara su debida función, con una matriz de frecuencia de acceso inicial, se crea su matriz correspondiente de patrones de acceso, donde se verificará por filas si tienen el mismo patrón y si se da el caso se reducirán las filas, sumándose entre ellas los datos que correspondan en x, y.

Esto nos hará que la matriz se reduzca de frecuencia de acceso obtenidos de su suma, con el cual se mandara un archivo de salida con la matriz reducida.

con las matrices originales se creará un árbol o grafo por medio de Graphviz, el cual se obtendrá como una imagen (.png), la cual mostrará el nombre de las matrices, las nfilas, las mcolumnas y el contenido de la matriz.

Desarrollo del tema

Con la ayuda de listas enlazadas simples y lista circular creamos un nodo donde iremos guardando la información proporcionada por el archivo XML.

Lista: Es una estructura de datos formada por una secuencia ordenada de objetos. Los elementos de una **lista** pueden accederse mediante su índice, siendo 0 el índice del primer elemento.

Nodo: es un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar.

Condicionales: La estructura de control if ... else ... permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple esa condición.

XML: se utiliza para el intercambio de datos estructurados. Más que un formato de **archivos** rígido, **XML** es un lenguaje que define los formatos aceptados que pueden utilizar los grupos para intercambiar información.

Los nodos tendrán almacenados los datos tales como, los nombres, las filas, las columnas y los datos de la matriz. Se utiliza una lista simple dentro de una lista circular para poder almacenar mejor los datos obtenidos

Tuplas: una **tupla** es un conjunto ordenado e inmutable de elementos de este o diferente tipo.

Las **tuplas** se representan escribiendo los elementos entre paréntesis y separados por comas. Una **tupla** puede no contener ningún elemento, es decir, ser una **tupla** vacía.

POO (Programación Orientada a Objetos):

Es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

Se utilizo el paradigma POO, para que el código fuera reutilizable y organizado, lo que evitara estar creando duplicados de código, y crear un programa eficiente.

Al tener las matrices dentro de las listas se podrán manipular para generar un árbol o grafo usando la herramienta Graphviz, para obtener una imagen png o del formato deseado.

Graphviz: (Graph Visualization) es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje descriptivo DOT.

Diccionario de palabras

While: Permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor True).

For: se utiliza para repetir una o más instrucciones un determinado número de veces, se suele utilizar cuando sabemos seguro el número de veces que queremos que se ejecute.

Matriz: Es un arreglo bidimensional de números.

Se utilizan para múltiples aplicaciones y sirven para representar transformaciones lineales dada una base.

Grafo: Un **grafo** se representa gráficamente como un conjunto de puntos (vértices o nodos) unidos por líneas (aristas).

Desde un punto de vista práctico, los **grafos** permiten estudiar las interrelaciones entre unidades que interactúan unas con otras.

Versionamiento: Es el proceso de asignación de un nombre, código o número único, a un software para indicar su nivel de desarrollo.

Frecuencia: esta etiqueta representa la frecuencia de los grupos de registros utilizados para crear la matriz reducida.

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

Conclusiones

- Se determino una solución integral que implemente **TDA's**, combinados con la **POO**, para poder almacenar objetos de base de datos en sitios distribuidos.
- Se determine que, mediante la **POO**, se facilitan los programas como conjunto de objetos que se ayudan entre ellos, para realizar las acciones.
- Se llego a utilizar estructuras de programación secuenciales tales como, el for, el while, que permitían recorrer los archivos de entrada, también se implantaron las condicionales como if, else, para el menú, para darle lógica a la lista circular, y así poder manipularla.
- Se determino que por medio de la librería **xml.etree.ElementTree**, se facilitó el manejo de los datos tipo etiqueta, para su lectura, y poder extraer la información del archivo.

Referencias bibliográficas

1. Problem Solving with Algorithms and Data Structures using Python by Bradley N. Miller, David L. Ranum
<https://runestone.academy/runestone/static/pythoned/index.html>
2. <https://wiki.python.org/moin/XmlBooks>
3. <https://graphviz.org/documentation/>
4. <https://diveintopython3.net/xml.html>

Apéndices

Apéndice A.

Requerimientos del Sistema mínimos

- Memoria RAM de 2 GB
- Windows 7
- Procesador dual Core

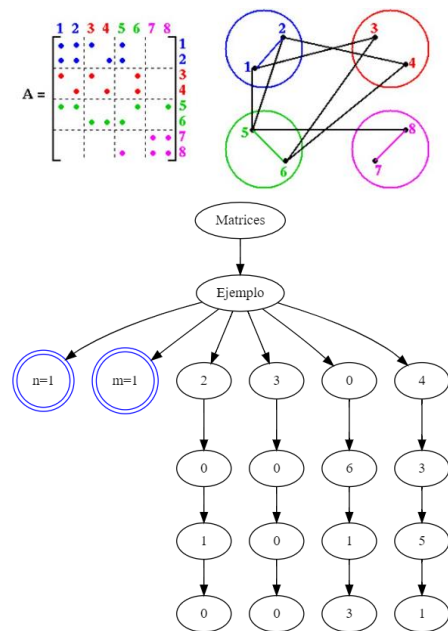
Apéndice B.

Programas necesarios en el sistema

- Python 2.0 en adelante
- Graphviz
- Git hub o Git hub Desktop

Apéndice C.

Problema de Particionamiento de grafos



Apéndice D.

Listas

