PROYECTO 1

201903873 – Joaquin Emmanuel Aldair Coromac Huezo

Resumen

El enunciado demuestra el problema de alojamiento de objetos en una base de datos en sitios distribuidos, por lo cual se espera que el costo total de la transmisión de datos para el procesamiento de todas las aplicaciones sea minimizado, debido a la fuerte demanda de recursos de memoria y tiempo, para esto se propone una metodología de agrupamiento.

Para “nt” tuplas y “ns” sitios, el método consiste en tener la matriz de frecuencia de acceso en los sitios F[nt][ns] de la instancia objetivo, transformarla en una matriz de patrones de acceso y agrupar las tuplas con el mismo patrón. El patrón de acceso para una tupla es el vector binario indicando desde cuál sitio la tupla es accedida.

Para la solución se decidió usar como recurso de memoria Listas Circulares Simplemente Enlazadas, donde se guarda como objeto cada matriz de frecuencia y se procesa para encontrar su matriz reducida y así ahorrar recurso de memoria.

Palabras clave

> Matriz > Objeto
> Frecuencia > Lista > Grafo

Abstract

The statement demonstrates the problem of hosting objects in a database on distributed sites, so it is expected that the total cost of data transmission for the processing of all applications is minimized, due to the heavy demand of memory resources and time, for this a clustering methodology is proposed.

For "nt" tuples and "ns" sites, the method consists of having the access frequency matrix in the sites F[nt][ns] of the target instance, transforming it into a matrix of access patterns and grouping the tuples with the same pattern. The access pattern for a tuple is the binary vector indicating from which site the tuple is accessed.

For the solution it was decided to use as a memory resource Simply Linked Circular Lists, where each frequency matrix is stored as an object and processed to find its reduced matrix and thus save memory resources.

Keywords

> Matrix > Object
> Frequency > List > Network.

Introducción

Teniendo como solución la metodología de agrupamiento, y optando por el paradigma de programación la orientada a objetos (POO), se desarrollo una Lista Circular Simple General donde en cada objeto se tiene los datos de las matrices que nos encontramos en el archivo XML de entrada.

Para así procesar cada matriz y encontrar su respectiva reducida y frecuencia y poder generar el archivo de salida en formato XML, para cada matriz también se tiene la opción de graficar mediante Graphviz que genera un archivo .dot.

Desarrollo del tema

Lógica Programa

➢ **Archivo(main.py)**

- Funciones
    - def main():

  Función principal para ejecutar el programa y llama a la función Menu del archivo proyecto.py

➢ **Archivo(proyecto.py)**

- Clases
    - class proyect:
        - def menú():
- Funciones
    - def ReconstruirLista():
    - def crearNodo():
    - def unirNodo():
    - def matrizBinaria():

➢ **Archivo(lista.py)**

- Clases
    - class Lista:
    - class Nodo:
    - class ListaCircular:

➢ **Archivo(grafo.dot)**

- Archivo generado para cada matriz de entrada que desee graficar

Paradigma de programación en uso

Programación orientada a objetos(POO)

Es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Este muestra un realza mayor al momento de guardar en espacio de memoria cada matriz en un objeto nodo de la lista circular.

Lógica de Lista Circular

Cada nodo tiene un objeto Lista con las variables código, nombre, n, m y g.

En la variable código se guarda otra lista circular en la que cada nodo es una fila de la matriz en forma de String.

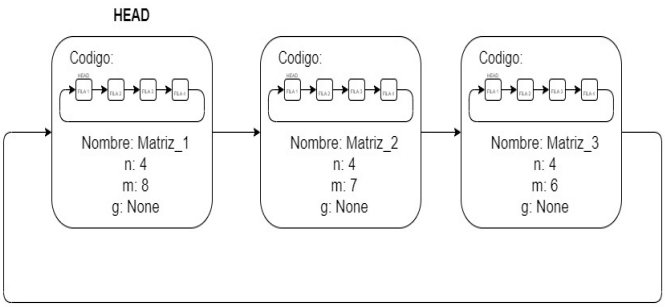

Figura 1. Lógica Lista Circular Simple

Fuente: elaboración propia

Especificaciones

Especificaciones de Editor de Texto, versión Python y Graphviz

Visual Studio Code

python.3.8.1

Graphviz 2.46.0

Especificaciones de computadora

Procesador: Intel(R) Core™ i7-9750H CPU@

2.60 GHz 2.59GHz

RAM instalada: 16.0GB (15.9 GB usable)

Tipo de sistema: Sistema operativo de 64 bits,

Procesador basado en x64

Windows 10.

Control de versiones

Para el proyecto se trabajó con GitHub para controlar las versiones durante su realización.

Enlace Repositorio.

https://github.com/jeach27/IPC2_Proyecto1_201903873.git

Forma de Uso Programa

Menu principal

Escoja una opción entre 1-6 para realizar la función requerida.

```
----------------Menú principal--------------------

> Elija una opcion

1.Cargar Archivo
2.Procesar Archivo
3.Escribir Archivo Salida
4.Mostrar Datos Estudiantes
5.Generar Gráfica
6.Salir

> Ingrese valor
```

Imagen 1. Menu Principal

Fuente: Programa, elaboración propia

Opcion1 Cargar Archivo

Ingresa ruta del archivo XML y guarda cada matriz en memoria de la Lista Circular.

```
> Ingrese valor
1
----------------------Cargar Archivo-----------------------
Ingrese ruta de archivo
D:\Quincho\Vsemestre\IPC2\Laboratorio\entrada1.xml

--> El archivo fue cargado correctamente
```

Imagen 2. Cargar Archivo

Fuente: Programa, elaboración propia

Opcion2 Procesar Archivo

Procesa cada matriz y va identificando cada acción realizada.

```
> Ingrese valor
2
-------------------Procesar Archivo------------------

<Calculando Matriz Binaria>
<Calculando Matriz Binaria>
<Calculando Matriz Binaria>
<Calculando posiciones>
<Calculando posiciones>
<Calculando posiciones>
<Realizando Suma de Tuplas>
<Realizando Suma de Tuplas>
<Realizando Suma de Tuplas>
```

Imagen 3. Procesar Archivo

Fuente: Programa, elaboración propia

Opcion4 Datos Estudiantiles

Muestra los datos del estudiante desarrollador.

```
> Ingrese valor
4
---------------Datos Estudiantiles--------------------

Carné: 201903873
Nombre: Joaquin Emmanuel Aldair Coromac Huezo
Curso: Introducción a la Programación y Computación 2
Carrera: Ingenieria en Ciencias y Sistemas
5to Semestre
```

Imagen 4. Datos Estudiantiles
Fuente: Programa, elaboración propia

Opcion5 Generar Grafica

Muestra las matrices guardadas en memoria y pide escoger la que desee graficar.

```
> Ingrese valor
5
---------------Generar Gráfica-------------------------
-------------Matrices---------------
1. matriz_n_1
2. matriz_n_2
3. matriz_n_3
Ingrese valor de matriz a graficar
```

Imagen 5. Generar Grafica
Fuente: Programa, elaboración propia

Modelo Grafica

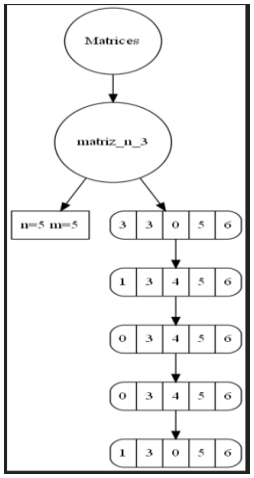

Imagen 6. Modelo Grafica
Fuente: Programa, elaboración propia

Opcion6 Salir

Pulse una tecla para salir de la ejecución del programa.

```
> Ingrese valor
6
----------------------Salir----------------------------

----> Pulsa una tecla para salir


D:\Quincho\Vsemestre\IPC2\Laboratorio\IPC2_Proyecto1_201903873>
```

Imagen 7. Salir
Fuente: Programa, elaboración propia

Conclusiones

El uso de Paradigma de programación orientado a objetos nos permite trabajar de forma más sencilla cada objeto de la Lista de matrices.

Los archivos XML es una forma sencilla y segura de guardar datos de este tipo.

La herramienta Graphviz para realizar los grafos es bastante fácil al momento de generar sus archivos .dot.

Referencias bibliográficas

9.3. Programación orientada a objetos —

Materiales del entrenamiento de

programación en Python - Nivel básico. (s.

f.). Materiales del entrenamiento de

programación en Python - Nivel básico.

Recuperado 24 de febrero de 2021, de

https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/poo.html

GeeksforGeeks. (s. f.). *Parse XML using Minidom in Python*. Recuperado 1 de marzo de 2021, de https://www.geeksforgeeks.org/parse-xml-using-minidom-in-python/

graphviz. (2020, 24 diciembre). PyPI. https://pypi.org/project/graphviz/

Salcedo, L. (s. f.). *Linked List: Listas enlazadas - Implementación en Python*. Mi Diario Python. Recuperado 29 de febrero de 2021, de https://pythondiario.com/2018/07/linked-list-listas-enlazadas.html

uniwebsidad. (s. f.). *Capítulo 16. Listas enlazadas (Algoritmos de programación con Python)*. Recuperado 27 de febrero de 2021, de https://uniwebsidad.com/libros/algoritmos-python/capitulo-16