Compresor de datos basado en Huffman

Área Temática: *Fundamentos Informáticos, Análisis y diseño de algoritmos.*
Categoría: *Trabajo de alumnos, Trabajo de Cátedra.*
Cátedra: *Diseño de Lenguajes de Consulta – DLC.*

Abstract

Se presenta la necesidad de modificar la implementación tradicional del algoritmo de Huffman para la compresión de datos con el objetivo de: minimizar el espacio de metadatos en el archivo comprimido, minimizar el uso de memoria principal y optimizar los tiempos de procesamiento.

Metodología

Para la reducción de espacio en el área de metadatos, se reemplazó la estructura de datos del Árbol de Huffman por una matriz definida por alfabeto, longitud y código Huffman.

Para la reducción de memoria principal para la codificación de Huffman, se reemplazó la estructura de datos *ArrayList <Byte>* por registros binarios de 64 bits.

La reducción de tiempos de procesamiento se logra eliminando los ciclos de iteración de listas con desplazamientos binarios en registros de 64bits y reemplazando la estructura del Árbol de Huffman por una tabla Hash<código+longitud, símbolo> en el descompresor.

Resultados

Dado un alfabeto de 256 símbolos:
Para una implementación típica: Un árbol tendrá una longitud de 511 nodos, cada uno de 128 bits + 1 boolean, suponiendo 129 bits, es decir 511 nodos x 129 bits = 65919 bits = 8.3 KBytes.
Empleando la estructura propuesta:
256 * 80 bits = 20480 = 2.5 KBytes.
Esto representa un 70 % de ahorro en metadatos.

Esto también implica que el espacio en memoria principal es de 2.5 KBytes para el alfabeto codificado.

Al reemplazar ciclos de iteración de listas por desplazamientos binarios de registros y del Árbol de Huffman por una tabla Hash en el descompresor, se redujo considerablemente el tiempo de procesamiento.

Discusión final

La implementación propuesta es similar a la solución con codificación Canónica de Huffman. Esta representa un ahorro aproximadamente del 10% de espacio en metadatos, ya que no requiere almacenar el alfabeto, pero requiere que éste sea acordado previamente junto con un procesamiento adicional de los códigos de Huffman. Lo que implica más trabajo para el compresor de datos.

La implementación podría optimizarse implementando un esquema de buffering para disminuir la cantidad de accesos al disco.

Implementación de Árbol de Huffman por Ing. Valerio Frittelli.
Compresor y descompresor por Rodolfo F. Juarez.
rfjuarez@gmail.com