

DATA COMPRESSION CONCEPTS SEC (SEMESTER-1)

BY

SANTOSH KUMAR LAL

DEPT. OF COMMERCE

SARIYA COLLEGE, SURIYA

INTRODUCTION TO DATA COMPRESSION

DATA COMPRESSION IS THE PROCESS OF REDUCING THE SIZE OF DATA SO THAT IT OCCUPIES LESS STORAGE SPACE AND REQUIRES LESS TIME TO TRANSMIT OVER NETWORKS.

COMPRESSION REMOVES **REDUNDANT OR UNNECESSARY DATA** FROM FILES WHILE KEEPING THE ORIGINAL INFORMATION INTACT.

EXAMPLE

- A **100 MB VIDEO** COMPRESSED TO **25 MB**
- A **TEXT FILE** COMPRESSED INTO A **ZIP ARCHIVE**

IMPORTANCE

- **SAVES STORAGE SPACE**
- **REDUCES DATA TRANSMISSION TIME**
- **IMPROVES NETWORK BANDWIDTH USAGE**
- **HELPS IN EFFICIENT DATA BACKUP**

COMMON COMPRESSED FORMATS

- ZIP
- RAR
- MP3
- JPEG
- MPEG

डेटा कम्प्रेसन

डेटा कम्प्रेसन (DATA COMPRESSION) वह प्रक्रिया है जिसमें किसी डेटा के आकार को कम किया जाता है ताकि वह कम स्टोरेज स्पेस ले और नेटवर्क के माध्यम से जल्दी भेजा जा सके।

इस प्रक्रिया में **अनावश्यक या दोहराए गए डेटा (REDUNDANT DATA)** को हटाया जाता है।

उदाहरण

- 100 MB की वीडियो को 25 MB में बदलना
- टेक्स्ट फाइल को ZIP फाइल में कम्प्रेस करना

महत्व

- स्टोरेज स्पेस की बचत
- डेटा ट्रांसमिशन तेज होता है
- नेटवर्क बैंडविड्थ की बचत
- बैकअप और आर्काइविंग आसान

NEED FOR DATA COMPRESSION

THE NEED FOR DATA COMPRESSION ARISES BECAUSE MODERN SYSTEMS HANDLE **LARGE VOLUMES OF DATA** SUCH AS IMAGES, VIDEOS, AND MULTIMEDIA FILES.

REASONS

- **LIMITED STORAGE CAPACITY**
- **FASTER DATA TRANSMISSION**
- **REDUCED BANDWIDTH CONSUMPTION**
- **COST EFFICIENCY**
- **EFFICIENT MULTIMEDIA STREAMING**

EXAMPLE

- STREAMING SERVICES LIKE **YOUTUBE** USE COMPRESSION ALGORITHMS TO DELIVER VIDEOS SMOOTHLY.

आज के समय में बड़ी मात्रा में डेटा जैसे **वीडियो, इमेज और मल्टीमीडिया फाइलें** उपयोग होती हैं, इसलिए डेटा कम्प्रेसन आवश्यक है।

कारण

- सीमित स्टोरेज क्षमता
- डेटा भेजने की गति बढ़ाना
- बैंडविड्थ की बचत
- लागत में कमी
- ऑनलाइन वीडियो स्ट्रीमिंग को आसान बनाना

TYPES OF DATA COMPRESSION

DATA COMPRESSION IS MAINLY OF **TWO TYPES**:

- **LOSSLESS COMPRESSION**
- **LOSSY COMPRESSION**

LOSSLESS COMPRESSION

LOSSLESS COMPRESSION REDUCES FILE SIZE **WITHOUT LOSING ANY ORIGINAL DATA**. AFTER DECOMPRESSION, THE FILE BECOMES **EXACTLY THE SAME AS THE ORIGINAL**.

CHARACTERISTICS

- NO DATA LOSS
- PERFECT RECONSTRUCTION
- USED FOR IMPORTANT FILES

EXAMPLES

- TEXT FILES
- SOFTWARE PROGRAMS
- DATABASE FILES

FILE FORMATS

- PNG
- ZIP
- GIF

LOSSLESS COMPRESSION में डेटा को कम्प्रेस किया जाता है लेकिन कोई भी जानकारी खोती नहीं है।
डिकम्प्रेसन के बाद फाइल बिल्कुल पहले जैसी बन जाती है।

विशेषताएँ

- कोई डेटा लॉस नहीं
- मूल डेटा पूरी तरह वापस मिलता है
- महत्वपूर्ण डेटा के लिए उपयोग

उदाहरण

- टेक्स्ट फाइल
- प्रोग्राम फाइल
- डेटाबेस

LOSSY COMPRESSION

LOSSY COMPRESSION REMOVES **SOME UNNECESSARY DATA PERMANENTLY** TO ACHIEVE A HIGHER COMPRESSION RATIO.

AFTER DECOMPRESSION, THE FILE IS **NOT IDENTICAL** TO THE ORIGINAL BUT REMAINS VISUALLY OR AUDIBLY ACCEPTABLE.

USED IN

- IMAGES
- VIDEOS
- AUDIO FILES

FILE FORMATS

- JPEG
- MP3
- MPEG

LOSSY COMPRESSION में फाइल का आकार कम करने के लिए कुछ डेटा को स्थायी रूप से हटा दिया जाता है।

डिकम्प्रेस करने के बाद फाइल मूल फाइल से थोड़ी अलग हो सकती है।

उपयोग

- इमेज
- वीडियो
- ऑडियो

उदाहरण

- JPEG
- MP3
- MPEG

COMPRESSION RATIO

COMPRESSION RATIO MEASURES HOW MUCH A FILE SIZE IS REDUCED DURING COMPRESSION.

FORMULA

COMPRESSION RATIO = ORIGINAL SIZE / COMPRESSED SIZE

EXAMPLE

- ORIGINAL FILE = 100 MB
COMPRESSED FILE = 20 MB

COMPRESSION RATIO = $100 / 20 = 5:1$

MEANING THE FILE IS **5 TIMES SMALLER**.

COMPRESSION RATIO यह बताता है कि कम्प्रेसन के बाद फाइल का आकार कितना कम हुआ।

सूत्र

COMPRESSION RATIO = ORIGINAL SIZE / COMPRESSED SIZE

उदाहरण

- ORIGINAL = 100 MB
COMPRESSED = 20 MB

COMPRESSION RATIO = 5:1

COMPRESSION TECHNIQUES

SEVERAL TECHNIQUES ARE USED IN DATA COMPRESSION.

MAIN TECHNIQUES

- RUN LENGTH ENCODING (RLE)
- HUFFMAN CODING
- LEMPEL-ZIV-WELCH (LZW)
- ARITHMETIC CODING

RUN LENGTH ENCODING (RLE)

RLE COMPRESSES DATA BY REPLACING **REPEATED CHARACTERS WITH A COUNT AND VALUE.**

EXAMPLE

AAAAAA → A6

THIS REDUCES REDUNDANT DATA.

RLE तकनीक में बार-बार आने वाले अक्षरों को एक अक्षर और उसकी संख्या से दिखाया जाता है।

उदाहरण

AAAAAA → A6

HUFFMAN CODING

HUFFMAN CODING IS A **LOSSLESS COMPRESSION ALGORITHM** THAT ASSIGNS **SHORTER BINARY CODES TO FREQUENTLY OCCURRING CHARACTERS** AND LONGER CODES TO LESS FREQUENT CHARACTERS.

BENEFITS

- EFFICIENT STORAGE
- REDUCES AVERAGE CODE LENGTH

HUFFMAN CODING एक **LOSSLESS COMPRESSION ALGORITHM** है जिसमें अधिक बार आने वाले अक्षरों को **छोटा कोड** दिया जाता है।

LZW COMPRESSION

LZW (LEMPERL-ZIV-WELCH) IS A **DICTIONARY-BASED COMPRESSION ALGORITHM** WIDELY USED IN FILE COMPRESSION UTILITIES.

APPLICATIONS

- GIF IMAGES
- ZIP COMPRESSION TOOLS

LZW एक **DICTIONARY** आधारित कम्प्रेसन एल्गोरिदम है जो फाइल कम्प्रेसन में उपयोग होता है।

उपयोग

- GIF इमेज
- ZIP फाइलें

APPLICATIONS OF DATA COMPRESSION

DATA COMPRESSION IS WIDELY USED IN MANY FIELDS.

APPLICATIONS

- IMAGE STORAGE
- VIDEO STREAMING
- AUDIO COMPRESSION
- FILE ARCHIVING
- DATABASE STORAGE
- WEB APPLICATIONS

EXAMPLES INCLUDE PLATFORMS LIKE **NETFLIX** AND **YOUTUBE**, WHICH COMPRESS LARGE VIDEO FILES FOR FASTER STREAMING.

डेटा कम्प्रेसन कई क्षेत्रों में उपयोग होता है।

उपयोग

- इमेज स्टोरेज
- वीडियो स्ट्रीमिंग
- ऑडियो फाइल
- फाइल आर्काइव
- डेटाबेस
- वेब एप्लिकेशन

ADVANTAGES OF DATA COMPRESSION

- SAVES STORAGE SPACE
- FASTER FILE TRANSFER
- EFFICIENT BANDWIDTH USAGE
- REDUCES BACKUP TIME
- SUPPORTS MULTIMEDIA COMMUNICATION
- स्टोरेज स्पेस की बचत
- तेज फाइल ट्रांसफर
- बैंडविड्थ का बेहतर उपयोग
- बैकअप समय कम
- मल्टीमीडिया डेटा ट्रांसफर आसान

DISADVANTAGES OF DATA COMPRESSION

- COMPRESSION REQUIRES PROCESSING TIME
- LOSSY COMPRESSION REDUCES QUALITY
- SOME ALGORITHMS ARE COMPLEX
- DECOMPRESSION IS REQUIRED BEFORE USE
- कम्प्रेसन में प्रोसेसिंग समय लगता है
- LOSSY कम्प्रेसन में गुणवत्ता कम हो सकती है
- कुछ एल्गोरिदम जटिल होते हैं
- उपयोग से पहले डिकम्प्रेसन करना पड़ता है

CONCLUSION

DATA COMPRESSION PLAYS A CRUCIAL ROLE IN MODERN COMPUTING BY REDUCING STORAGE REQUIREMENTS AND ENABLING EFFICIENT DATA TRANSMISSION.

डेटा कम्प्रेसन आधुनिक कंप्यूटिंग में बहुत महत्वपूर्ण है क्योंकि यह स्टोरेज बचाता है और डेटा ट्रांसमिशन को तेज बनाता है।