DATA COMPRESSION

Techniques and Applications, Hardware and Software Considerations

GILBERT HELD

SECOND EDITION

DATA COMPRESSION

Techniques and Applications
Hardware and Software Considerations

Second Edition

GILBERT HELD
4-Degree Consulting,
Macon, Georgia, USA

and

THOMAS R. MARSHALL
(software author)

John Wiley & Sons
Chichester · New York · Brisbane · Toronto · Singapore

***Library of Congress Cataloging-in-Publication Data*:**
Held, Gilbert, 1943–
Data compression.
Bibliography: p.
Includes index.
1. Data compression (Computer science) I. Marshall, Thomas (Thomas R.) II. Title.
QA76.9.D33H44 1987 005.74′6 86-18942
ISBN 0 471 91280 8

***British Library Cataloguing in Publication Data*:**
Held, Gilbert
Data compression: techniques and applications hardware and software considerations. —
2nd ed.
1. Data compression (Computer science)
I. Title II. Marshall, Thomas
005.74′6 QA76.9.D33
ISBN 0 471 91280 8

Typeset by Photo·Graphics, Honiton, Devon
Printed in Great Britain

CHAPTER ONE

Rationale and Utilization

In the chronology of computer development, large-scale information transfer by remote computing and the development of massive information storage and retrieval systems have witnessed a tremendous growth. Concurrent with this growth, several problem areas have developed which can result in major, but unnecessary, economic expenditures.

One problem is the so-called 'run-away database'. Here the size of the database used by an organization for its information storage and retrieval programs becomes larger and larger, requiring additional disc drives for online systems and reels of magnetic tapes for those systems that can be processed in a batch environment.

Accompanying the growth in the size of databases has been a large increase in the number of users and duration of usage by personnel at remote locations. These factors result in tremendous amounts of data being transferred between computers and remote terminals. To provide transmission facilities for the required data transfers, communications lines and auxiliary devices, such as modems and multiplexers, have been continuously upgraded by many organizations to permit higher data-transfer capability.

Although the obvious solutions to these problems of data storage and information transfer are to install additional storage devices and expand existing communications facilities, to do so requires an additional increase in an organization's equipment and operating costs. One method that can be employed to alleviate a portion of data storage and information transfer problems is through the representation of data by more efficient codes. If one examines an organization's database or monitors a transmission line, there is an excellent chance that the individual characters that both make up the database and the transmission sequence could be encoded more effectively. Two techniques that can result in a more effective encoded data representation are logical and physical data compression.

1.1 LOGICAL COMPRESSION

When a database is designed, one of the first steps of the analyst is to obtain as much data reduction as possible. This data reduction results from the elimination of redundant fields of information while representing the data elements in the remaining fields with as few logical indicators as is feasible.

Although logical compression is data dependent and the method employed can vary based upon the analyst's foresight, the following two examples will illustrate the ease of implementation and benefits of this compression technique.

One simple example of logical data compression is the occupational field on a personnel database. Suppose 30 alphanumeric positions are allocated to this field. If the field is fixed, occupations such as the 10-character occupational description 'DISHWASHER' have 20 blanks inserted into the remainder of the field. Then, 30 million characters of storage would be required for the occupational field of 1 million workers. Suppose at most there were 32 768 distinct occupations. Instead of indicating the occupational title, one could encode the equivalent 5-digit data code, eliminating 25 character positions per field. The size of the field could be reduced further by allocating the binary value of 1 or more characters to the occupational code. As an example, an 8-bit character could represent $2^8 - 1$ or 255 distinct values or occupational codes. Linking two 8-bit characters through appropriate software would provide $2^{16}-1$ or 65 535 distinct codes. This would reduce the field size from 30 to 2 characters, saving 28 million characters of storage. If our counting begins at zero instead of a conventional starting place of 1, an 8-bit character could represent 256 codes while a 16-bit character could be employed to represent 65 536 distinct values.

A second example of logical compression is a date field. This type of field frequently occurs in databases. Normally, the numeric equivalents of the subfields representing day, month and year are used in place of longhand notation. Thus, 01 04 81 would represent 1 April 1981. While this logical compression results in 6 numeric characters of storage, additional data reduction can result from storing the date as a binary value. Since the day will never exceed 31, 5 bits would suffice to represent the date field. Similarly, 4 bits could be used to represent the month value while 7 bits could represent 127 years, permitting a relative year ranging from 1900 to 2027.

Logical compression using numerical and binary representation is illustrated in Figure 1.1 for the preceding date field example. It is interesting to note that employing binary representation reduces the date field to 16 binary digits or two 8-bit concatenated characters of storage. As discussed, many logical compression methods can be considered by an analyst during the database design process. Each method may result in a distinct degree of data storage reduction. Correspondingly, when logically compressed databases or portions of such databases are transmitted between locations, transmission time is reduced since fewer data characters are transmitted.

While logical compression can be an effective tool in minimizing the size of a database, it only reduces transmission time when logically compressed data is transmitted. Thus, the transmission of inquiry and response data

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.