

## СИСТЕМНЫЙ АНАЛИЗ И ОЦЕНКА МЕТОДОВ СЖАТИЯ ДАННЫХ ДЛЯ БАЗ ДАННЫХ ЛАЗЕРНЫХ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСОВ

*В статье приведены результаты исследования характеристик основных методов и алгоритмов сжатия данных, проведена их сравнительная оценка по критериям для определения наиболее эффективных алгоритмов сжатия данных баз данных для лазерных технологических комплексов.*

*Ключевые слова: сжатие данных, методы сжатия табличных данных, упаковка битов, кодирование длин серий, статистическое кодирование, дифференциальное кодирование, словарные методы, методы контекстного моделирования, лазерный технологический комплекс.*

O.V. NECHYPORENKO, S.A. MITSSENKO

Cherkassy State Technological University

### SYSTEMS ANALYSIS AND EVALUATION OF METHODS OF DATA COMPRESSION FOR DATABASES LASER COMPLEXES

*Abstract - The purpose of the work is the realization of analysis systems and estimation of properties and principles of operations and the characteristics of data compressions methods. Determination of effective algorithms for reduction of volume of database.*

*In the article the system analysis and algorithms of non-destructive compression of tabular data's is given. It's description is given and considered basic advantages and disadvantages. The basic criteria's for comparison of descriptions of methods and algorithms of data compression are educe. A comparative analysis of the basic non-destructive methods and algorithms is conducted.*

*The results of conducted analyses are testifies in a benefits of algorithms of differential encryption and dynamic algorithms of arithmetical encryption's.*

*Keywords: data reduction, tabular data compression techniques, packing bits, coding lengths series, entropy coding, differential encoding, dictionary methods, methods context modeling, laser processing facility.*

#### Введение

В настоящее время стремительное развитие баз данных (БД) порождает необходимость хранения и обработки все больших объемов информации, что требует огромных вычислительных и временных затрат. Увеличению объемов данных в БД также способствует введение информационной избыточности, которая используется для повышения надежности функционирования вычислительных систем. К отрицательным последствиям введения избыточности относят повышение интенсивности потока и увеличение вероятности возникновения ошибок, а также уменьшение вероятности их исправления [1].

Для сокращения объемов БД и ускорения выдачи необходимой информации по запросу, целесообразно применять методы сжатия информации. Вопрос экономного кодирования информации в системах управления базами данных (СУБД) был поставлен в первой половине 1970-х годов, но не потерял актуальности до сих пор. Многие современные исследователи отмечают недостаточную теоретическую проработку проблемы и неэффективность поддержки сжатия данных в промышленных СУБД [2].

Интерес к задаче сжатия данных в СУБД обусловлен стремлением уменьшить физический объем БД. При надлежащем интегрировании в СУБД методов неразрушающего сжатия данных достигается значительная экономия. Небольшое увеличение числа используемых тактов процессора для кодирования-декодирования более чем компенсируется снижением затрат на подсистему ввода-вывода. В настоящее время интерес представляет также увеличение производительности СУБД за счет использования сжатия.

Поэтому актуальным является проведение системного анализа и оценки основных методов сжатия. Для определения алгоритмов, наиболее эффективных при обработке различных типов данных БД, целесообразно провести их упорядочение и выбор критериев сопоставления их характеристик. Это позволит провести оценку методов сжатия, которая базируется на анализе свойств, принципов работы и основных характеристик и определить наиболее эффективные алгоритмы для сжатия различных типов данных БД лазерных технологических комплексов (ЛТК).

**Цель работы** – проведение системного анализа и оценки свойств, принципов работы и основных характеристик методов сжатия данных, определение эффективных алгоритмов для уменьшения объема данных БД ЛТК.

#### Результаты исследований

##### 1. Основные понятия и определения

**Определение. Сжатие данных** (data compression) — алгоритмическое преобразование данных, производимое с целью уменьшения занимаемого ими объёма. Применяется для более рационального использования устройств хранения и передачи данных. Синонимы – упаковка данных, компрессия, сжимающее кодирование, кодирование источника [3].

Сжатие основано на устранении избыточности, содержащейся в исходных данных. В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности, которая может быть статической, неизменной для всего сжимаемого сообщения, либо строиться на этапе сжатия. Методы, позволяющие на основе входных данных изменять модель избыточности информации, называются

адаптивними. Неадаптивними являються обычно узкоспеціалізовані алгоритми, застосовувані для роботи з даними, що мають добре визначені та незмінні характеристики. Подавляюча частина універсальних алгоритмів є в той чи інший міру адаптивними [3].

Всі існуючі методи стиснення інформації умовно розділяють на дві основні групи. Перша з них об'єднує методи стиснення інформації без втрат, а друга група – стиснення з втратами [4].

Якщо при стисненні даних відбувається зміна їх вмісту, метод стиснення є необратимим і при відновленні даних з стисненого файлу не відбувається повного відновлення вихідної послідовності. Такі методи називають також методами стиснення з втратами (lossy compression). Вони застосовні тільки для певних типів даних, для яких формальна втрата частини вмісту не призводить до значущого зниження споживчих властивостей. Методи стиснення з втратою інформації зазвичай забезпечують значно вищу ступінь стиснення, ніж зворотні методи, але їх не можна застосовувати до текстових документів, БД і, тим більше, до програмного коду. Якщо при стисненні даних відбувається тільки зміна їх структури, то метод стиснення є зворотним. З отриманого коду можна відновити вихідний масив шляхом застосування зворотного методу. Зворотні методи або методи стиснення без втрати інформації (lossless compression) застосовують для стиснення будь-яких типів даних.

## 2. Аналіз основних методів стиснення даних в СУБД

Використання стиснення даних в СУБД пов'язано з рішенням множини завдань:

- вибір методу стиснення і рівня стискаемого блоку в ієрархії фізичної структури БД;
- вибір між апаратним і програмним стисненням;
- вибір місцеположення модуля стиснення в архітектурі обчислювальної системи;
- забезпечення оновлення даних і виконання запитів до стиснених даних [2].

Основна маса досліджень і розробок, що стосуються питання підтримки економічного кодування в СУБД, явним чином або неявно орієнтовані на реляційні системи (РСУБД). Подавляючий сегмент ринку СУБД займає РСУБД. Дослідження показали, що реляційна модель є найбільш оптимальною для БД ЛТК. Результати проведеного аналізу свідчать також на користь застосування багатовимірної моделі [5].

Багато методів стиснення і супутні прийоми обробки запитів застосовні до БД різного типу, наприклад, багатовимірної і об'єктної. Виходячи з вищезазначеного, даний аналіз стосується питань ефективного підтримки стиснення реляційних БД (РБД) методами стиснення без втрат.

Багатоманітність інформаційної надлишковості обумовило багатоманітність підходів до стиснення даних в РБД і класифікації цих підходів. Для табличних даних характерні наступні типи надлишковості, які зазвичай проявляються одночасно [2]:

- різниця в частоті зустрічальності окремих символів (елементів);
- наявність послідовностей однакових символів;
- часте повторення певних послідовностей символів;
- часте виникнення символів або послідовностей символів в певних місцях.

Для табличних даних характерні як вертикальні кореляційні зв'язи (по стовпцю), так і горизонтальні (по рядку). Найкраще стиснення буде забезпечуватися тими методами, які ефективно експлуатують всі типи надлишковості.

Найбільш поширеними типами даних БД є рядкові і числові, методи стиснення яких мають специфіку, тому вони розглянуті окремо. В табл. 1 наведено основні методи і алгоритми стиснення без втрат для РБД, дано їх описання, розглянуті основні переваги і недоліки.

## 3. Оцінка методів і алгоритмів стиснення табличних даних БД ЛТК

Для визначення методів і алгоритмів, найбільш ефективних при обробці різних типів даних цілорозумно виробити їх упорядкування і вибір критеріїв порівняння їх характеристик.

Визначення ефективних алгоритмів базується на аналізі властивостей, принципів роботи і основних характеристик. Основними критеріями для порівняння методів і алгоритмів стиснення в контексті результативності їх роботи є:

- ступінь стиснення - відношення об'ємів вихідного і отриманого представлення даних;
- швидкість роботи (кодування і декодування) – час, витрачений на стиснення і відновлення даних;
- об'єм вимованої пам'яті;
- зворотність алгоритма – можливість відновлення даних (з втратами або без втрат);
- послідовність обробки даних – визначається, однопрохідним або багаторазовим методом;
- можливість зростання надлишковості при застосуванні алгоритму стиснення;
- сфера застосування.

В області стиснення даних діє закон важеля: алгоритми, що використовують більше ресурсів (часу і пам'яті), зазвичай досягають кращої ступеня стиснення, і навпаки, менш ресурсомісткі алгоритми за якістю стиснення, як правило, поступають більш ресурсомістким. Таким чином, побудова оптимального з практичної точки зору алгоритму стиснення даних є складною задачею, так як необхідно отримати достатньо високу якість стиснення при невеликому об'ємі використовуваних ресурсів [6].

## Анализ основных методов сжатия табличных данных

Метод	Описание	Преимущества	Недостатки
1	2	3	4
<b>Кодирование числовых данных</b>			
<b>Упаковка битов</b> - классический алгоритм (bit packing) - алгоритм сжатия кадра ссылок (frame of reference compression)	Заключается в кодировании значения атрибута битовыми последовательностями фиксированной длины	- Простой способ сжатия на уровне значений атрибутов. - Высокая скорость обработки и сохранение упорядоченности.	- Обработка появления новых значений атрибута. - Не обеспечивает значительной степени сжатия.
<b>Кодирование длин серий и устранение констант</b> - классический алгоритм - алгоритм заголовочного сжатия - метод «VAR»	Выявление повторяющихся последовательностей данных и замена их простой структурой, состоящей из кода и коэффициента повтора. Устранение константы реализуется с помощью битовой карты, заголовочного вектора или трех векторов одновременно.	- Эффективность достигается за счет экономного кодирования часто повторяющихся значений. - Независимость от объема данных - Отличается простотой и высокой скоростью работы - Эффективен для БД с фиксированной длиной полей	- Актуально только для сильно разреженных БД - Медленное декодирование - В среднем обеспечивает недостаточное сжатие
<b>Статистическое кодирование</b> - метод Хаффмана - арифметическое сжатие - интервальное кодирование	Явным образом опираются на теорему Шеннона и включают в себя два этапа: оценку вероятности кодируемых элементов (моделирование) и собственно кодирование	- Может применяться к данным произвольного типа - Высокая скорость обработки данных	- Достаточно затратное кодирование - Эффективно только при большой статистической избыточности - Классический алгоритм требует двух проходов по сообщению
<b>Дифференциальное кодирование</b> - классический алгоритм - алгоритм «сжатие окна» (window-based compression)	Значение $T[i,j]$ $i$ -ой строки $j$ -го столбца заменяется на разность $T[i,j] - T[i-1,j]$ . Если имеется тенденция в изменении данных, то результирующие разности экономно представляются простыми методами	- Один из основных способов сжатия данных в СУБД и часто используется в качестве одного из шагов алгоритма сжатия - Высокая скорость работы - Высокая степень сжатия данных	- Сложность обеспечения случайного доступа к информации в БД
<b>Кодирование строковых данных и данных произвольного типа</b>			
<b>Метод Хаффмана</b> - классический (статический) алгоритм - неадаптивный алгоритм с расширением алфавита символов - полуадаптивное кодирование - адаптивное кодирование	Определяет процедуру построения кода переменной длины, средняя избыточность для которого минимальна для всех неблочных кодов. Сопоставляет символам входного потока, которые встречаются большее число раз, цепочку бит меньшей длины.	- Один из основных и наиболее популярных способов экономного кодирования - Простота и высокая скорость декодирования - Простая аппаратная реализация - Уникальный префикс, позволяющий однозначно декодировать коды - Используется как один из этапов компрессии в более сложных схемах	- Малоэффективен для небольших файлов - Сложное декодирование - В классическом алгоритме длина сжатого сообщения увеличивается на длину таблицы частот - Зависимость степени сжатия от близости вероятностей символов к 2 в отрицательной степени

Продовження табл. 1

1	2	3	4
<b>Арифметическое сжатие</b> - классический алгоритм - адаптивное кодирование - полуадаптивное кодирование - интервальное кодирование - неадаптивное кодирование атрибута по колонкам («COLA»)	Блок элементов представляется дробью, определяемой произведением оценок вероятности всех элементов блока. Сжатие достигается за счет экономного кодирования окончательных пробелов, принудительно увеличивающих длину значения до заданной ширины столбца.	- Высокая степень сжатия, достигается теоретическая граница степени сжатия - Сохраняется порядок сортировки элементов в соответствии с их накопленной вероятностью - Не требует явной перестройки кода при изменении оценок вероятности	- Сложное декодирование - Работает медленнее, чем алгоритм Хаффмана - Арифметический кодер является двухпроходным и требует передачи вместе с закодированным текстом таблицы частот символов.
<b>Методы Зива-Лемпела</b> - LZ77, LZSS - LZ78, LZW - LZW с сохранением упорядоченности - LZMW (метод Миллера-Уэгнама)	Заключается в замене последовательностей элементов исходных данных на идентификаторы фраз некоторого словаря, совпадающие с замещаемой последовательностью.	- Используются как эталонные при сравнении схем сжатия - Позволяют выполнять операции сравнения без декодирования данных - Эффективен для массивов большого объема	- Непрактичны для сжатия на уровне столбца - Сложность обеспечения произвольного доступа - Невысокая скорость работы классических алгоритмов
<b>Дифференциальное кодирование</b> - классический алгоритм - векторное квантование (VQ)	Кодирование разности между предыдущим и текущим символами. Метод VQ разбивает множества объектов на группы, в каждой из которых выбирается образцовый объект, к нему приравниваются все остальные объекты	- Высокая скорость работы - Высокая степень сжатия данных - Эффективно используется как средство простого учета вертикальных взаимосвязей в данных	- Усложняется организация эффективного произвольного доступа к информации
<b>Методы контекстного моделирования</b> - статическое моделирование - полуадаптивное моделирование - адаптивное моделирование (динамическое) - блочно-адаптивное моделирование	Методы контекстно-зависимого моделирования ограниченного порядка, позволяющие оценить вероятность символа в зависимости от предыдущих символов	- Универсальный метод - Хорошая степень сжатия при высокой скорости - Возможность получения хорошей статистической модели обработанной последовательности данных	- Медленное декодирование - Несовместимость кодера и декодера в случае изменения алгоритма оценки - Неэффективны для малоизбыточных данных - Большие запросы памяти
<b>Сортировка, группировка и преобразование столбцов</b> - сортировка параллельных блоков (PBS) - преобразование Барроуза-Уилера (BWT) - метод перемещения стопки книг (MTF) - алгоритм нахождения дерева зависимости групп колонок - алгоритм нахождения оптимальных группировок	Сортировка - упорядочение записей по значениям одного из полей. Группировка - образование однородных групп по ряду признаков. PBS сортирует элементы входного блока и раскладывает их в несколько выходных. BWT меняет порядок символов во входной строке, при этом повторяющиеся подстроки образуют на выходе идущие подряд последовательности одинаковых символов.	- Как предварительная операция позволяет применить к полученным данным простые методы сжатия - Перестановка столбцов может существенно улучшить сжатие при построчной обработке - Сортировка позволяет значительно повысить степень сжатия, в особенности на текстовых данных. - Декомпрессия в 3-4 раза быстрее сжатия.	- Степень сжатия сильно зависит от типа данных. - Требуется достаточно много памяти - Для однозначного восстановления сортировка должна быть устойчива, с сохранением порядка одинаковых элементов - Алгоритмы группировки работают сравнительно медленно - Скорость сжатия – относительно невысокая.

Критерии оценки методов сжатия с практической точки зрения сильно зависят от предполагаемой области применения. Например, при использовании сжатия в системах реального времени необходимо обеспечить высокую скорость кодирования и декодирования; для встроенных систем – объем требуемой

памяти; для систем долговременного хранения данных – степень сжатия и (или) скорость сжатия.

Проведем сравнительный анализ и оценим основные неразрушающие методы и алгоритмы сжатия данных, используя вышеперечисленные критерии (табл. 2).

Таблица 2

## Сравнительный анализ основных неразрушающих методов и алгоритмов сжатия

Метод сжатия (алгоритмы)	Степень сжатия	Скорость работы	Потребление памяти	Кол-во проходов	Возрастание избыточности	Применение
Упаковка битов						
Классический алгоритм (bit packing)	низкая	высокая	среднее	1	редко	Кодирование данных числового типа
Кодирование длин серий и устранение констант						
Кодирование длин серий (RLE)	низкая	высокая	большое	1	возможно	Сжатие изображений, таблиц
Кодирование Хаффмана (HAFF)						
статический алгоритм	средняя	достаточно высокая	достаточно большое	2	редко	Универсальные алгоритмы, сжатие текстов и бинарной информации
динамический алгоритм	средняя	выше средней	выше среднего	1	редко	
фиксированный алгоритм	средняя	высокая	большое	1	возможно	
Арифметическое кодирование (ARIC)						
статический алгоритм	высокая	средняя	достаточно большое	2	редко	Универсальный метод, сжатие текстов
динамический алгоритм	высокая	выше средней	выше среднего	1	редко	
Дифференциальное кодирование (DIFC)						
классический алгоритм	достаточно высокая	достаточно высокая	среднее	1	возможно	Кодирование данных в БД, сжатие изображений и звука
Словарные методы (LZ)						
LZ77, LZSS	достаточно высокая	ниже средней	небольшое	1	редко	Универсальные методы, сжатие текстов, сжатие изображений
LZHAFF, LZARJ	высокая	низкая	небольшое	2	редко	
LZ78, LZW	средняя	выше средней	среднее	1	редко	
Методы контекстного моделирования (CM)						
предсказание по частичному совпадению (PPM)	высокая	ниже средней	достаточно большое	1	редко	Универсальный, сжатие текстов, изображений
Сортировка, группировка и преобразование столбцов						
преобразование Барроуза-Уилера (BWT)	выше средней	низкая	среднее	>1	редко	Универсальные методы, сжатие текстов
метод перемещения стопки книг (MTF)	ниже средней	выше средней	достаточно большое	1	возможно	

Для БД ЛТК, как системы долговременного хранения данных необходимо обеспечить высокую степень и достаточно высокую скорость сжатия. Анализ показал, что, по первому критерию (степень сжатия), эффективными являются следующие методы: арифметическое кодирование (статический и динамический алгоритмы), дифференциальное кодирование, словарные методы (алгоритмы LZ77, LZSS), методы контекстного моделирования. Двум критериям одновременно (степень и скорость сжатия) полностью удовлетворяет только дифференциальное кодирование, а также динамический алгоритм арифметического кодирования, который имеет высокую степень сжатия и приемлемую скорость работы, немного меньше, чем у дифференциального кодирования.

Для применения в СУБД требуются специфические методы и приемы экономного кодирования, поскольку обычные методы:

- не обеспечивают требуемой скорости декодирования;

- работают с блоками достаточно большого объема и не поддерживают произвольный доступ;
- не учитывают структуру данных;
- не учитывают статистику по данным для последующей оптимизации запросов.

Очень полезным свойством метода сжатия может быть сохранение упорядоченности, что позволяет оперировать при выполнении запросов непосредственно закодированными данными. Многие обычные методы сжатия не обладают такой особенностью. Результатом применения сжатия для БД является повышение производительности и скорости выполнения запросов. Повышение производительности за счет уменьшения размера хранимых данных обусловлено следующими эффектами [2]:

- уменьшается среднее время поиска и доступа к информации в устройствах внешней памяти;
- повышается фактическая пропускная способность канала ввода-вывода внешней памяти;
- повышается фактическая пропускная способность вычислительной сети;
- увеличивается фактический размер буфера устройства внешней памяти;
- увеличивается фактический размер буферного пула СУБД.

Повышение скорости выполнения запросов вследствие использования сжатия вызвано:

- увеличением пропускной способности каналов связи;
- уменьшением времени ожидания извлечения данных из внешней и оперативной памяти.

В то же время применение сжатия для БД связано с рядом отрицательных эффектов:

- увеличение расходов вычислительных ресурсов и, возможно, оперативной памяти;
- переменная длина записей требует усложнения программного обеспечения;
- нарушение характеристик данных, что требует использования сортирующих методов сжатия;
- уменьшение возможности успешного восстановления после ошибок.

Обеспечение действительно высокой производительности СУБД при использовании сжатия возможно только при комплексном изменении механизма выполнения запросов. Увеличение разницы в скорости процессора и элементов памяти диктует необходимость использования более сложных методов, требующих значительного объема вычислений, но предоставляющих большую степень сжатия, которые могут повысить реальную производительность СУБД.

#### Выводы

Проведен системный анализ и оценка основных методов и алгоритмов неразрушающего сжатия табличных данных, дано их описание, рассмотрены основные преимущества и недостатки.

Выявлены основные критерии для сравнения характеристик методов и алгоритмов сжатия данных.

Проведен сравнительный анализ основных неразрушающих методов и алгоритмов сжатия по критериям для БД ЛТК. Результаты проведенного анализа свидетельствуют в пользу применения алгоритмов дифференциального кодирования, а также динамического алгоритма арифметического кодирования.

#### Литература

1. Нечипоренко О.В. Синтез специализированных цифровых устройств по условию надежности на основе систем счисления с постоянным числом единиц : [монография] / Нечипоренко О.В. – Черкассы : ВУЭМ, 2009. – 109 с.
2. Смирнов М.А. Обзор применения методов безущербного сжатия в СУБД [Электронный ресурс] – Режим доступа : [http://compression.ru/download/articles/db/smirnov\\_2003\\_database\\_compression\\_review/index.html](http://compression.ru/download/articles/db/smirnov_2003_database_compression_review/index.html).
3. Сжатие данных [Электронный ресурс]. – Режим доступа : [http://ru.wikipedia.org/wiki/Сжатие\\_данных](http://ru.wikipedia.org/wiki/Сжатие_данных).
4. Дударь З.В. Исследование и оптимизация методов сжатия текстовой информации / З. Дударь, С. Егоров. – Харьков : Вестник ХНТУ, 2012. – № 1 (44).
5. Нечипоренко О.В. Классификационная схема моделей баз данных для лазерных технологических комплексов / О. Нечипоренко, С. Миценко. – Черкаси : Вісник ЧДТУ, 2013. – № 2.
6. Алгоритмы сжатия [Электронный ресурс] – Режим доступа : [http://mf.grsu.by/UchProc/livak/en/po/comprsite/theory\\_contents.html](http://mf.grsu.by/UchProc/livak/en/po/comprsite/theory_contents.html).
7. Лукашенко А.Г. Методологія організації бази даних на основі теорії неповної подібності та розмірностей / Лукашенко А.Г., Лукашенко Д.А., Міценко С.А. – Кіровоград : КНТУ, 2011. – С. 336–339.

#### References

1. Nechyporenko O. V. Syntez spetsyalyzovannykh cyfrovyykh ustrojstv po uslovyju nadezhnosti na osnove system schyslennyja s postojannym chyslom edynyc: monohrafyja. Cherkassy, VUEM, 2009, 109 p.
2. Smyrnov M. A. "Obzor prymenenija metodov bezushherbnogho szhatyja v SUBD" [Elektronnyj resurs] – Rezhym dostupa: [http://compression.ru/download/articles/db/smirnov\\_2003\\_database\\_compression\\_review/index.html](http://compression.ru/download/articles/db/smirnov_2003_database_compression_review/index.html).
3. "Szhatye dannykh. [http://ru.wikipedia.org/wiki/Szhatye\\_dannykh](http://ru.wikipedia.org/wiki/Szhatye_dannykh).
4. Dudary Z.V., Egorov S.V. Yssledovanye y optymyzacyja metodov szhatyja tekstovoj ynformacyy, Kharjkov, Vestnyk KhNTU, 2012, No. 1(44).
5. Nechyporenko O.V., Mycenko S.A. Klassyfykacyonnaja skhema modelej baz dannykh dlja lazernykh tekhnologhycheskykh kompleksov, Cherkasy, Visnyk ChDTU, 2013. No. 2.
6. Alghorytmy szhatyja. [http://mf.grsu.by/UchProc/livak/en/po/comprsite/theory\\_contents.html](http://mf.grsu.by/UchProc/livak/en/po/comprsite/theory_contents.html).
7. Lukashenko A.H., Lukashenko D.A., Mitsenko S.A. Metodolohiia orhanizatsii bazy danykh na osnovi teorii nepovnoi podobnosti ta rozmirmostei. Kirovohrad, KNTU, 2011, P. 336-339.

Рецензія/Peer review : 10.12.2013 р. Надрукована/Printed :6.2.2014 р.

Рецензент: д.т.н., проф., зав. каф. системного програмування ЧДТУ Рудницький В.М.