PLATFORM V

PG BootCamp Russia 2025 Ekaterinburg

PGBootCamp.ru

Подходы к реализации сжатия данных

Екатеринбург 10.04.2025

Селезнев Павел
Community-manager команды Pangolin
Работа с open source

Не так давно мы реализовали сжатие данных на уровне хранения

Матчасть

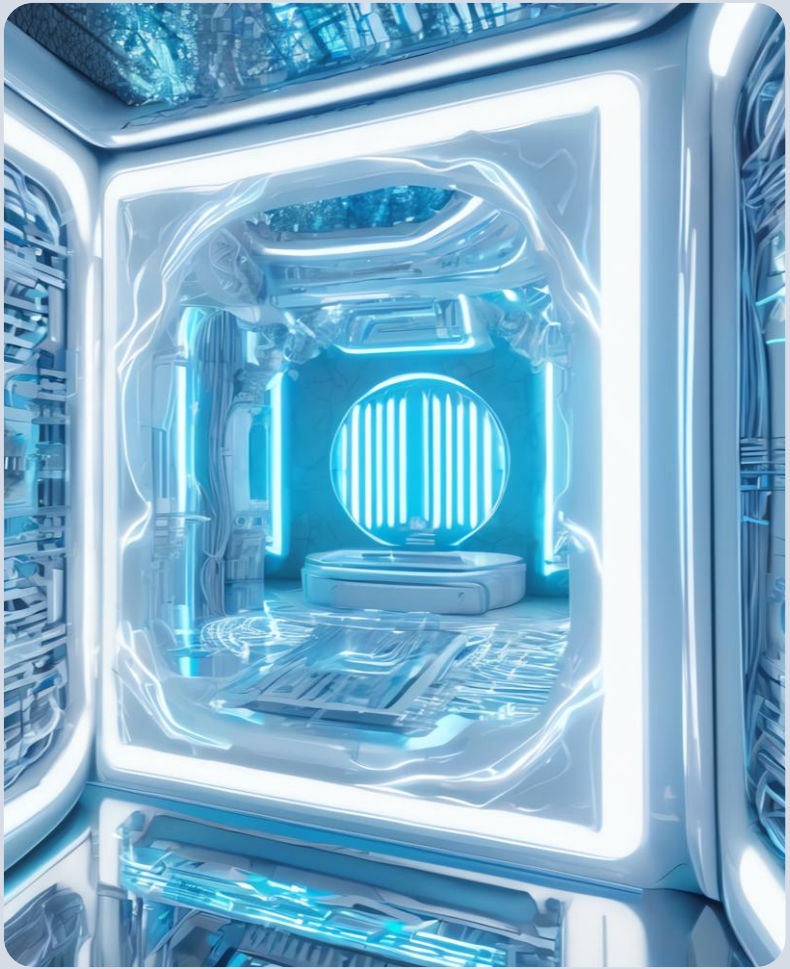
Идеи алгоритмов

Матчасть

Существующие инструменты

Затем

Особенности реализации сжатия



Зачем экономить на объеме информации

01

Каждый год рост объема данных на 20% (300 млн терабайт в день)

02

База данных используется как универсальное хранилище

03

Экономия на оборудовании

PLATFORM V

Алгоритмы сжатия

- Коды Хаффмана
- LZ77
- Арифметическое кодирование
- 860 магическое число
- Как используется CPU

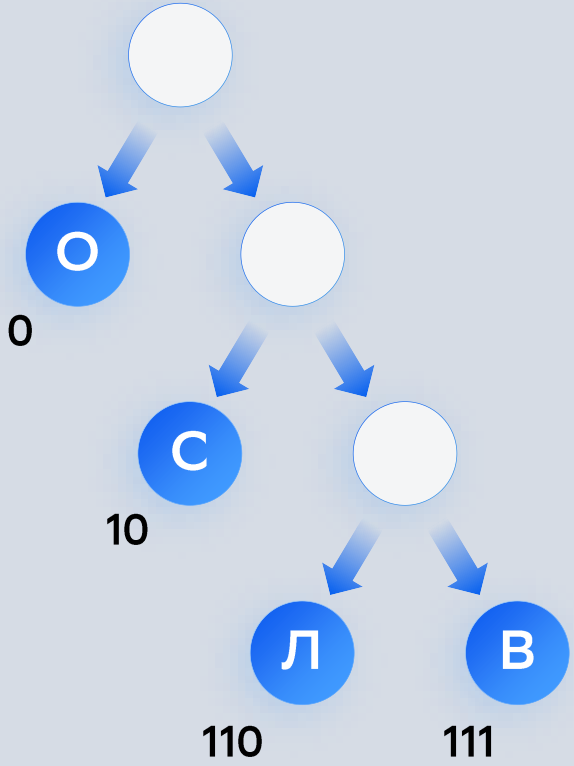
Алгоритм сжатия: коды Хаффмана

Разработан в 1952

Для каждого символа префиксный код уникален

Чем вероятнее символ, тем меньше префикс

Оптимален, когда частоты появления символов пропорциональны $1/2^n$



Входящее сообщение
СЛОВО

Вероятность символа
О: 0.4
С: 0.2
Л: 0.2
В: 0.2

Сжатое сообщение
1011 0011 10

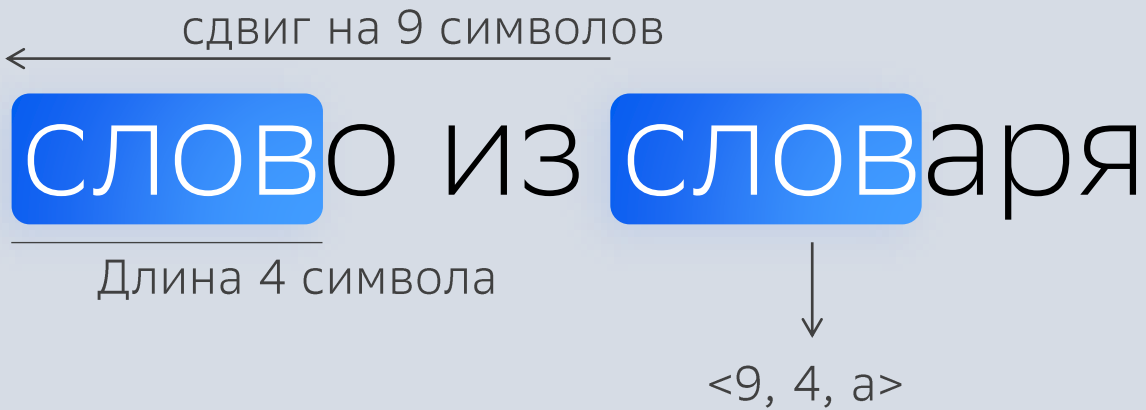
Алгоритм сжатия: LZ77

Семейство LZ

Разработан в 1977

Кодирование ссылки блоками из трёх элементов — ⟨**сдвиг, длина, следующий символ**⟩

Величина сдвига ограничена плавающим окном

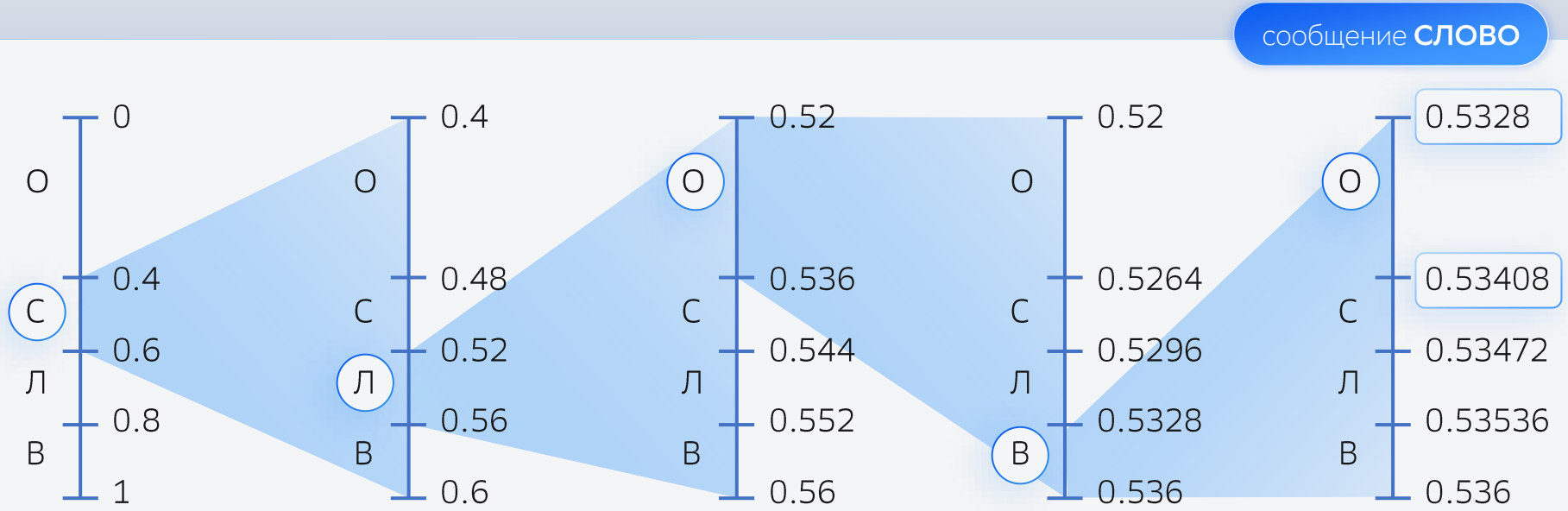


PLATFORM V

Есть ли что-то новое после 1977 года?

Арифметическое кодирование

Zstandard использует наработки арифметического кодирования и сжимает эффективнее за счет более плотной упаковки битов



Магическое число

860 байт

Длина, свыше которой сжатие имеет смысл

Меньше

и так влезет в сетевой пакет

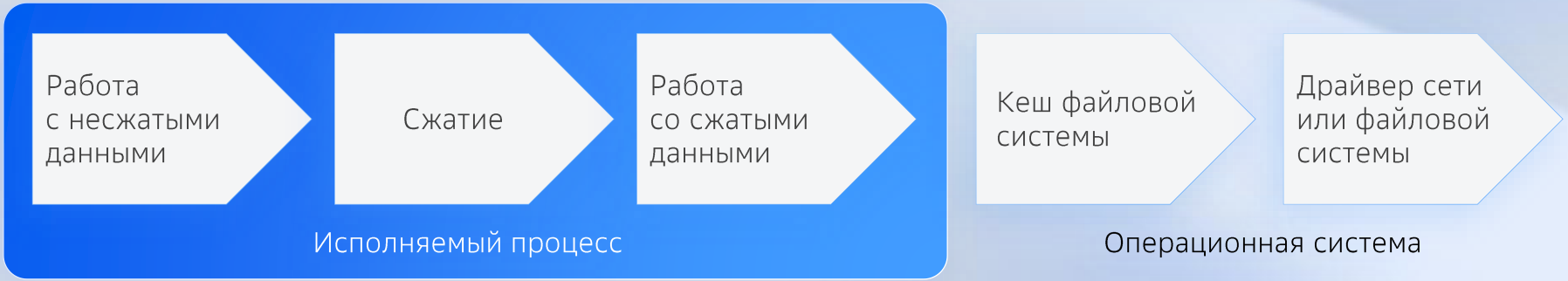
Меньше

затраты CPU
на кодирование будут большими

Нагрузка на CPU при сжатии

Дополнительное потребление CPU

Низкое потребление CPU/IO операций для работы с данными



PLATFORM V

Алгоритмы в PostgreSQL

Существующие алгоритмы

Поддерживаемые алгоритмы

Базовые алгоритмы:

- pglz (по умолчанию)
- lz4 (сжимает быстрее, чем pglz, начиная с PG14)
- zstandard (сжимает лучше, чем pglz, начиная с PG15)

- deflate для pg_dump

- Семейство LZ77
- BSD лицензия

Расширения

С помощью расширений вы можете добавить любой существующий алгоритм для сжатия и распаковки данных

pgsql-gzip

```
SELECT gzip('this is my text'), также есть поддержка BYTEA[]
```

Pgbrotli

```
SELECT pgbrotli_compress('this is my text');
```

postgres-protobuf

```
SELECT protobuf_from_json_text('pgpb.test.ExampleMessage',
'{"scalars":{"int32Field":123}}')
```

pgbrotli хорош для заимствования кода

Как выбирать алгоритм

База знаний

Сжимаете/разжимаете/передаете?

- **Одно имя**
  zstandard
- **Распространённость алгоритма**
  deflate медленный, но есть везде
- **Медленный процессор на сжатии/распаковке?**
  lz4
- **Архитектура Arm64?**
  zstandard + нагрузочное тестирование
- **Важна скорость передачи по сети**
  zstandard + уровень компрессии

PLATFORM V

Компоненты PostgreSQL, поддерживающие сжатие

- Репликация и создание резервных копий
- TOAST-сжатие

Существующие алгоритмы

Утилиты Pg_dump/pg_basebackup/ pg_receivewal поддерживают сжатие

Через ключ compress задается алгоритм и уровень сжатия

```
pg_dump --
compress=gzip|zstd|lz4:1..20
```

Поддержка алгоритмов зависит от версии ядра

Сжатие при работе с WAL-файлами (postgres.conf)

```
wal_compression=pglz|lz4|zstd
```

Сценарий нагрузки с WAL-файлами

Сценарий тестирования:

```
pgbench --client=10 --jobs=2 --transactions=100000 test_db
```

Запрос для расчет размера WAL-файлов:

```
SELECT 'WAL_AFTER'::pg_lsn - 'WAL_BEFORE'::pg_lsn;
```

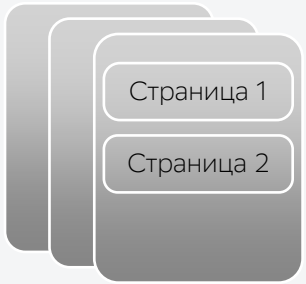
Оценка загрузки CPU показывает прирост 5-10%, но практически без влияния на общее время (при достаточных ресурсах).

	Сжатие отключено	pglz	lz4	zstandard
Размер WAL-файлов, байт	6 501 306 296	893 750 464	935 227 352	**757 001 640**
Время выполнения скрипта, сек	350.0	351.8	348.0	**345.6**

TOAST-сжатие

- **CREATE TABLE** `<tablename>(<fieldname> TEXT COMPRESSION pglz|lz4)`
- **Срабатывает при превышении 2 КБ**, из расчета, что 4 кортежа (tuple) должны поместиться на страницу. Параметр toast_tuple_target позволяет уменьшить предел до 128 байт.
- **Стратегия EXTENDED/MAIN**
- **Можно установить для всей базы данных**
  `SET default_toast_compression = pglz|lz4;`

Отношения хранятся в файлах (сегментах) по 1 ГБ



TOAST-файл для кортежей, которые не помещаются на страницу

- Один файл на всю таблицу, разные колонки в один файл
- Максимальная длина одного объекта — 2^30 (1 gb)
- Максимальное количество объектов в таблице 2^32 (4 млрд записей, включая удалённые)
- 32 ТВ — ограничение на размер файла

TOAST-сжатие и производительность

Размер TOAST-таблицы, байт	Без сжатия, стратегия EXTERNAL	pglz, стратегия EXTENDED,	lz4, стратегия EXTENDED
Исполняемые файлы	53 198 848	**25 696 608**	27 484 160
Исходный код и графика	22 470 656	**8 249 344**	8 060 928
Каталог звезд	7 577 600	7 577 600	**7 102 464**

Время выполнения контрольной точки, мс	Без сжатия, стратегия EXTERNAL	pglz, стратегия EXTENDED	lz4, стратегия EXTENDED
Исполняемые файлы	85	53	53
Исходный код и графика	48	26	30
Каталог звезд	30	31	26

PLATFORM V

Подходы к сжатию данных на диске

Вариант 1
экономичное сжатие данных

Вариант 2
сжатие данных с помощью фрагментов

Есть ли резерв для сжатия?

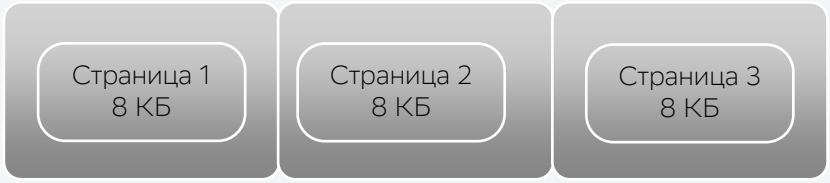
```
CREATE TABLE transaction(
    id BIGINT,
    transaction_at TIMESTAMP,
    bank_in INTEGER,
    bank_out INTEGER,
    amount INTEGER,
    account_in VARCHAR(40),
    account_out VARCHAR(40)
);

SELECT
get_raw_page('transaction', 0);
```

```
 040007000208180004000000000000dfc32b2f
 23d50200a566a702ac66a702409c00002b333031
 303138313034303030303030303232322b3330
 313031383130343030303030303030232390000
 060000000000000000000000000003000700
 020818000300000000000009bf8a22a23d50200
 a966a702a966a702307500002b33303130313831
 303430303030303030303232332b333031303138
 313034303030303030303030232380000040000000
 000000000000000000000000200070002081800
 020000000000000024a5142423d50200a966a702
 a966a702204e00002b333031303138313034303030
 303030303030303232342b333031303138313034303030
 303030303030303232370000030000000000000000
 000000000000000010007000208180001000000
 000000001a52592323d50200a966a702a966a702
 102700002b333031303138313034303030303030
 30303232352b333031303138313034303030303030
 303030323236000ad13000000000000000000000
 000000001010000000000000
(410 rows)
```

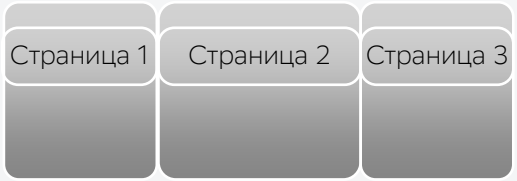
Экономичное сжатие данных

Размер

Сегмент без сжатия, до 1 ГБ



Сегмент со сжатыми страницами



Сегмент со сжатыми страницами после обновления записей

Требуется дефрагментация

Сжатие с помощью фрагментов

```
CREATE TABLE test(
  id BIGINT, name TEXT
)
WITH (compresstype=pglz|lz4|zstd, compresslevel=1..22,
compress_chunk_size=512..4096, compress_prealloc_chunks=1..16);
```

Задаём параметры:

- алгоритм
- уровень сжатия
- размер фрагмента
- количество выделенных фрагментов при создании страницы

Хранение сжатой страницы

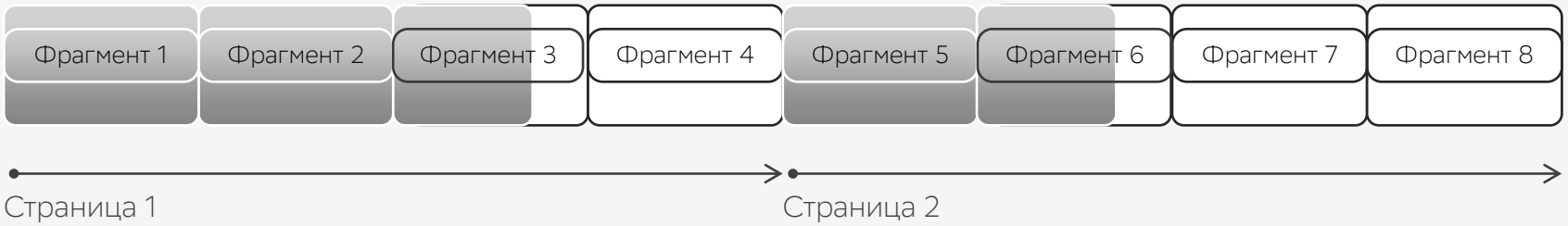
Оценка необходимого количества фрагментов

```
SELECT relation_estimate_compression_ratio(<имя таблицы>,
<алгоритм>, <уровень сжатия>, <размер фрагмента>, TRUE, <начальная страница>, <конечная страница)
---
2.4080865966252786;
```

Количество фрагментов = Размер страницы / 2.408 ...

Хранение сжатой страницы

4 заранее выделенных фрагмента на страницу



Устройство адресного файла

Если данные не сжимаются, то помещаем их так, как есть и ставим флаг в заголовке

Размер PCA файла:

16 фрагментов
размер фрагмента —512 байт
адресная таблица занимает 9 МБ

2 фрагмента
размер фрагмента — 4096 байт
адресная таблица занимает 2 МБ

Файл с данными (PCD)

Адресный файл (PCA), пример для размера 2048 байт

Номер страницы	N 1 фрагмента	N 2 фрагмента	N3 фрагмента	N4 фрагмента	N5 фрагмента
1	1				
2	2	4			
3	3	5	6		

Как может выглядеть план тестирования

01. Действительно ли данные сжимаются

02. Проверка задания опций сжатия

```
WITH (compresstype=pglz, compresslevel=1,
compress_chunk_size=2048,
compress_prealloc_chunks=0);
```

03. Тестируем различные виды объектов таблицы, разные типы индексов

```
CREATE TABLE test(
  id BIGINT,
  col_btree TEXT,
  col_gin TEXT,
  col_gin_tsv TSVECTOR,
  col_hash TEXT,
  col_gist TEXT,
  col_gist_tsv TSVECTOR,
  col_spgist TEXT )
WITH (compresstype=pglz …
```

04. Низкоуровневая работа — page inspect, page surgery

05. Нагрузочное тестирование

06. Уронить базу (kill -9) в различных вариациях

- при репликации
- при работе с нежурнализированными таблицами
- N раз подряд
- restore сжатых данных (pg_dump/pg_restore)

Влияние на производительность

1000

Количество выполняемых транзакций в секунду. Сценарий нагрузки TPCC

20

Количество активных клиентов, выполняющих запросы

30 минут

Время выполнения теста

Показатель	Без сжатия	Compresslevel =default, chunk_size = 512	Compresslevel =19, chunk_size = 512	Compresslevel =1, zstd 1.5.7, chunk_size = 512	Compresslevel =default, chunk_size = 512, pgbouncer	Compresslevel =1, chunk_size = 4096
Размер, ГБ	126	61	**59**	60	61	94
CPU%	54	51	59	54	57	**51**
RAM, ГБ	1.4	1.4	1.5	**1.35**	1.49	1.38

Сравнение алгоритмов

Дамп базы 1С (с дополнительными тестовыми данными, всевозможные типы объектов), который используется для тестирования обновлений версий ядра или изменения системного каталога

Размер tablespace, МБ	Без сжатия	pglz	lz4	zstandard
Размер фрагмента 4096 байт	29 087	14 696	14 697	14 697
Размер фрагмента, 2048 байт	29 087	8 580	9 016	**7 167**

Оценка производительности 1С Appdex

	Без сжатия	chunk_size = 1024, zstd 1.4.4, compresslevel = default	
APPDEX	0,924	0,925	
Размер базы, байт	239 109 273 100	172 461 966 348	Экономия ~30%

Выводы

Сжатие
не приводит к деградации производительности, если сжатые данные используются дальше

Осознанно подходите к выбору алгоритма сжатия и его параметров — идеально через нагрузочное тестирование

Спасибо за внимание!

PG BootCamp Russia 2025 Ekaterinburg

PGBootCamp.ru

Pangolin

http://pangolin.sbertech.ru

PG BootCamp

https://pgbootcamp.ru/

Список литературы

- **Сравнение алгоритмов** https://habr.com/ru/articles/570694/
- **Сжатие при создании резервной копии** https://habr.com/ru/articles/843264/
- **Понимание математики алгоритмов** https://kadm.kmath.ru/files/lect5_cmdc.pdf
- **Арифметическое кодирование** https://habr.com/ru/articles/142492/
- **Лекции по сжатию** http://msiit.ru/x/ti/__.html
- **Алгоритм Хаффмана** https://habr.com/ru/companies/samsung/articles/771572/