

Способы сжатия данных в PostgreSQL

Жилин Михаил

PGMeetup.NN

PosgresPro

PGMeetup.NN

О себе:

Telegram: [@mizhka](https://www.instagram.com/mizhka)
E-mail: mizhka@gmail.com



PostgresPro

- Физтех
- 14 лет работы с системной производительностью
- 2 года в компании Postgres Professional
- Контрибьютор в open source проекты
- Счастливый муж и отец
- Волейбол, дайвинг, путешествия

PGMeetup.NN

План разговора

- Полезно ли сжатие для баз данных?
- Сжатие бывает разным
- Встроенные механизмы и не только
- Примеры и рекомендации

Полезно ли сжатие для баз данных?

PGMeetup.NN

Полезно ли сжатие для баз данных?

Достоинства

- Экономия дискового пространства
- Экономия оперативной памяти
- Меньше нагрузка на диски

PGMeetup.NN

Полезно ли сжатие для баз данных?

Достоинства

- Увеличение пропускной способности системы (больше tps)
- Улучшается время ответа (меньше ms)

Полезно ли сжатие для баз данных?

Недостатки

Overhead	Shared Object	Symbol
63.82%	postgres	[.] pglz_decompress
16.04%	libc-2.31.so	[.] __memcpy_avx_unaligned_erms
1.99%	postgres	[.] LWLockAttemptLock
0.89%	postgres	[.] hash_search_with_hash_value
0.89%	postgres	[.] LWLockRelease
0.88%	postgres	[.] _bt_compare
Overhead	Shared Object	Symbol
46.70%	liblz4.so.1.9.3	[.] LZ4_decompress_safe
5.81%	postgres	[.] LWLockAttemptLock
2.59%	postgres	[.] LWLockRelease
2.19%	postgres	[.] _bt_compare
2.14%	postgres	[.] hash_search_with_hash_value

PGLZ (~80% или x4)

LZ4 (~45% или x2)

PGMeetup.NN

PostgresPro

Сжатие бывает разным

PGMeetup.NN

Сжатие бывает разным

Закономерности

- 1, 2, 3, 4, 5, 6, 7, 8, 9...
- 1, 2, 3, 5, 7, 11, 13, 17, 19, 23...
- 53363, 40550, 33191, 69437...

PGMeetup.NN

Сжатие бывает разным

Закономерности

- 1, 2, 3, 4, 5, 6, 7, 8, 9...
 - 1, 2, 3, 5, 7, 11, 13, 17, 19, 23...
 - 53363, 40550, 33191, 69437...
- +1
 - поиск простого числа
 - ???

Сжатие бывает разным

Теория информации

- Алгоритмическая теория информации
 - Андрей Николаевич Колмогоров (1960-ые)
 - <https://nautil.us/kolmogorov-complexity-and-our-search-for-meaning-237158/>
- Колмогоровская сложность
 - Минимальная длина программы, которая выводит необходимый набор символов (данных)
- Парадокс неизмеримости: **всегда может существовать алгоритм с меньшей длиной программы**

Сжатие бывает разным

Энтропийное сжатие

- Шеннон-Фано (1948-1949)
 - Вероятность символов
 - Чем чаще символ, тем меньше бит на символ
 - Двойной проход, операции на каждой символ
- Хаффман (1954)
 - Бинарное дерево частот и префиксы
 - Оптимальный набор
- Арифметическое сжатие

PGMeetup.NN

Сжатие бывает разным

Сжатие по словарю

- Lempel-Ziv-Welch (1978-1984)
 - рекурсивная копия, повторяемость последовательностей символов
 - LZ1 (LZ77) / LZ2 (LZ78) / LZW
 - Операция на целую последовательность символов
 - gif
- Deflate (1991) с добавлением энтропийного сжатия
 - png, zip, gzip and others

PGMeetup.NN

Сжатие бывает разным

PGLZ internals

00001810	c1	40	00	00	00	00	00	00	40	1f	00	00	b8	18	00	00	.@.....@.....	toast_compress_header→tc_i
00001820	00	42	75	6d	70	73	20	5b	65	00	76	65	6e	74	73	6f	.Bumps [e.ventso	
00001830	75	72	00	63	65	5d	28	68	74	74	70	00	73	3a	2f	2f	ur.ce](http.s://	
00001840	67	69	74	68	00	75	62	2e	63	6f	6d	2f	45	15	01	20	gith.ub.com/E..	ctl:15=10101, len=4, off=32
00001850	53	02	20	2f	08	2c	29	20	66	00	72	6f	6d	20	31	2e	S. /.,) f.rom 1.	
00001860	30	2e	80	37	20	74	6f	20	31	2e	01	02	00	0a	3c	64	0..7 to 1.....<d	
00001870	65	74	61	69	6c	00	73	3e	0a	3c	73	75	6d	6d	00	61	etail.s>.<summ.a	
00001880	72	79	3e	43	68	61	6e	80	67	65	6c	6f	67	3c	2f	05	ry>Chan.ge log</.	
00001890	13	00	0a	3c	70	3e	3c	65	6d	3e	05	03	57	64	03	4b	...<p>..Wd.K	
000018a0	3c	61	20	68	72	10	65	66	3d	22	0f	85	18	2f	62	6c	<a hr.ef=".../bL	
000018b0	00	6f	62	2f	6d	61	73	74	65	00	72	2f	48	49	53	54	.ob/maste.r/HIST	

PGMeetup.NN

Сжатие бывает разным

PGLZ internals

00001810	c1	40	00	00	00	00	00	00	40	1f	00	00	b8	18	00	00	.@.....@.....	toast_compress_header→tc_i
00001820	00	42	75	6d	70	73	20	5b	65	00	76	65	6e	74	73	6f	.Bumps [e.ventso	
00001830	75	72	00	63	65	5d	28	68	74	74	70	00	73	3a	2f	2f	ur.ce](http.s://	
00001840	67	69	74	68	00	75	62	2e	63	6f	6d	2f	45	15	01	20	gith.ub.com/E..	ctl: 15=10101, len=4, off=32
00001850	53	02	20	2f	08	2c	29	20	66	00	72	6f	6d	20	31	2e	S. /.,) f.rom 1.	
00001860	30	2e	80	37	20	74	6f	20	31	2e	01	02	00	0a	3c	64	0..7 to 1.....<d	
00001870	65	74	61	69	6c	00	73	3e	0a	3c	73	75	6d	6d	00	61	etail.s>.<summ.a	
00001880	72	79	3e	43	68	61	6e	80	67	65	6c	6f	67	3c	2f	05	ry>Chan.ge log</.	
00001890	13	00	0a	3c	70	3e	3c	65	6d	3e	05	03	57	64	03	4b	...<p>..Wd.K	
000018a0	3c	61	20	68	72	10	65	66	3d	22	0f	85	18	2f	62	6c	<a hr.ef=".../bL	
000018b0	00	6f	62	2f	6d	61	73	74	65	00	72	2f	48	49	53	54	.ob/maste.r/HIST	

→ Bumps [eventsource](https://github.com/E

PGMeetup.NN

Сжатие бывает разным

PGLZ internals

00001810	c1	40	00	00	00	00	00	00	40	1f	00	00	b8	18	00	00	. @.....@.....	toast_compress_header→tc_i
00001820	00	42	75	6d	70	73	20	5b	65	00	76	65	6e	74	73	6f	.Bumps [e.ventso	
00001830	75	72	00	63	65	5d	28	68	74	74	70	00	73	3a	2f	2f	ur.ce](http.s://	
00001840	67	69	74	68	00	75	62	2e	63	6f	6d	2f	45	15	01	20	gith.ub.com/E..	ctl: 15=10101, len=4, off=32
00001850	53	02	20	2f	08	2c	29	20	66	00	72	6f	6d	20	31	2e	S. /.,) f.rom 1.	
00001860	30	2e	80	37	20	74	6f	20	31	2e	01	02	00	0a	3c	64	0..7 to 1.....<d	
00001870	65	74	61	69	6c	00	73	3e	0a	3c	73	75	6d	6d	00	61	etail.s>.<summ.a	
00001880	72	79	3e	43	68	61	6e	80	67	65	6c	6f	67	3c	2f	05	ry>Chan.ge log</.	
00001890	13	00	0a	3c	70	3e	3c	65	6d	3e	05	03	57	64	03	4b	...<p>..Wd.K	
000018a0	3c	61	20	68	72	10	65	66	3d	22	0f	85	18	2f	62	6c	<a hr.ef=".../bL	
000018b0	00	6f	62	2f	6d	61	73	74	65	00	72	2f	48	49	53	54	.ob/maste.r/HIST	

4 байта со сдвигом на 32 байта

→ Bumps [eventsource](https://github.com/E

PGMeetup.NN

Сжатие бывает разным

PGLZ internals

00001810	c1	40	00	00	00	00	00	00	40	1f	00	00	b8	18	00	00	.@.....@.....	toast_compress_header→tc_i
00001820	00	42	75	6d	70	73	20	5b	65	00	76	65	6e	74	73	6f	.Bumps [e.ventso	
00001830	75	72	00	63	65	5d	28	68	74	74	70	00	73	3a	2f	2f	ur.ce](http.s://	
00001840	67	69	74	68	00	75	62	2e	63	6f	6d	2f	45	15	01	20	gith.ub.com/E..	ctl:15=10101, len=4, off=32
00001850	53	02	20	2f	08	2c	29	20	66	00	72	6f	6d	20	31	2e	S. /.,) f.rom 1.	
00001860	30	2e	80	37	20	74	6f	20	31	2e	01	02	00	0a	3c	64	0..7 to 1.....<d	
00001870	65	74	61	69	6c	00	73	3e	0a	3c	73	75	6d	6d	00	61	etail.s>.<summ.a	
00001880	72	79	3e	43	68	61	6e	80	67	65	6c	6f	67	3c	2f	05	ry>Chan.getLog</.	
00001890	13	00	0a	3c	70	3e	3c	65	6d	3e	05	03	57	64	03	4b	...<p>..Wd.K	
000018a0	3c	61	20	68	72	10	65	66	3d	22	0f	85	18	2f	62	6c	<a hr.ef=".../bL	
000018b0	00	6f	62	2f	6d	61	73	74	65	00	72	2f	48	49	53	54	.ob/maste.r/HIST	

5 байт со сдвигом на 32 байта

→ Bumps [events**source**](https://github.com/Event**Source**)

PGMeetup.NN

Сжатие бывает разным

Соревнование на скорость

- LZO (1996) - Markus F.X.J. Oberhumer
 - быстрая распаковка: read-only filesystems
 - <https://github.com/markus-oberhumer>
- LZ4 (2011) + Zstandard (2016) - Yann Collet
 - tradeoff и оптимизации под 32-/64-бит архитектуры
 - <https://github.com/Cyan4973>

PGMeetup.NN

Сжатие бывает разным

Современные реалии

- Смесь из энтропийного сжатия и сжатия по словарю
- Степень сжатия: от 1 до N (размер плавающего окна)
- Неэффективность на маленьких объемах
- LZ - словарь находится в сжатом поток

PGMeetup.NN

Сжатие бывает разным

LZ4 vs PGLZ

- Нет побитовых операций
- Копирование по несколько байт вместо по одному

PGMeetup.NN

Сжатие бывает разным

LZ4 vs PGLZ

PGLZ

LZ4

Overhead	Shared Object	Symbol
63.82%	postgres	[.] pglz_decompress
16.04%	libc-2.31.so	[.] __memcpy_avx_unaligned_erms
1.99%	postgres	[.] LWLockAttemptLock
0.89%	postgres	[.] hash_search_with_hash_value
0.89%	postgres	[.] LWLockRelease
0.88%	postgres	[.] _bt_compare
Overhead	Shared Object	Symbol
46.70%	liblz4.so.1.9.3	[.] LZ4_decompress_safe
5.81%	postgres	[.] LWLockAttemptLock
2.59%	postgres	[.] LWLockRelease
2.19%	postgres	[.] _bt_compare
2.14%	postgres	[.] hash_search_with_hash_value

PGMeetup.NN

Сжатие бывает разным

zstd vs LZ4

- Добавлено энтропийное сжатие (Huffman + FSE)
- Многопоточный
- Отрицательные уровни сжатия (-7 - самый быстрый)
- Колоссальные размеры окон (до 128MiB)



PGMeetup.NN

Сжатие бывает разным

Benchmarks

Алгоритм	Описание	Сжатие	Распаковка	Степень
deflate	добрый старый	5-100MiBps	10-200MiBps	2.8
lzo	быстрая распаковка	8MiBps	850MiBps	2.8
lz4	самый быстрый	780MiBps	4500MiBps	2.1
zstd	лучший баланс	480MiBps	1200MiBps	2.8

PGMeetup.NN

PostgresPro

Встроенные механизмы сжатия данных в PostgreSQL

PGMeetup.NN

Встроенное сжатие: inline и TOAST

Для строк/значений
больше чем 2килобайта

- короткие в таблице
- длинные в TOAST
- TOAST: btree + heap (2KiB per chunk)
- PGLZ (по умолчанию)
- LZ4 (доступно с 14ой версии)
- Параметр COMPRESSION (create/alter table)
- pg_attribute.attcompression ('p','l')
- postgresql.conf:
default_toast_compression=lz4

PGMeetup.NN

Встроенное сжатие: Index Key Dedup

Группировка TID-ов в Posting List

- PostgreSQL 12+
- PostgresPro 10+

- rebuild всех неуникальных индексов после pg_upgrade

PGMeetup.NN

Встроенное сжатие: WAL Full Page Image

Сжатие блоков в WAL файлах

- PGLZ
- LZ4 (с 15ой версии)
- выключен по умолчанию
- postgresql.conf:
[wal_compression=on](#)

PGMeetup.NN

Может быть завтра...

- Быстрый и компактный TOAST-а
- Дедупликации значений в таблицах
- Сжатие ключей в индексах

PGMeetup.NN

Дополнительно...

Колоночное хранилище

- GreenPlum & ZedStore (fork)
 - Citus Columnar & cstore_fdw (extension)
 - Append-only optimizations
 - lz4, zstd, zlib, rle
- Нет индексного сжатия
 - Ряд ограничений

PGMeetup.NN

PostgresPro

Встроенные механизмы сжатия данных поблочно



PGMeetup.NN

Блочное сжатие: PostgresPro CFS

- Покрывает все типы данных (индексы, таблицы)
- Прозрачно для пользователей
- Простота конфигурации
 - задаётся на уровне tablespaces-ов
- lz4, zstd, zlib, pglz, levels...
- Входит в PgPro Enterprise 9.6+

Блочное сжатие: PostgresPro CFS

- `CREATE TABLESPACE cfs LOCATION '/var/data/cfs' WITH (compression=true);`
- `postgresql.conf`: [cfs_level](#)

PGMeetup.NN

Блочное сжатие: на уровне ядра OS

Сжатие на уровне OS

- OpenZFS
- Почти прозрачно для базы данных
- lz4, zstd
- snapshot-ы

- Пределы масштабирования
- Требуется опыт в настройках

PGMeetup.NN

PostgresPro

Типовые сценарии использования сжатия

PGMeetup.NN

Типовые сценарии использования сжатия

Современные базы
небольшого размера

- < 500GiB
- < 500tps
- Достаточное количество RAM для кэширования
- SSD/NVMe

Встроенное сжатие

PGMeetup.NN

Типовые сценарии использования сжатия

Современные базы
большого размера

- > 1TiB

CFS или OpenZFS

PGMeetup.NN

Типовые сценарии использования сжатия

Медленные диски и/или мало
RAM

- HDD / NAS
- Всего 2-4GiB RAM

CFS или OpenZFS

PGMeetup.NN

Типовые сценарии использования сжатия

Файлохранилка (PDF, photos)

- Хранится в TOAST-е
- Медленно, но надёжно
- По копии на каждой реплике и в бекапах

Вынос на внешнее хранилище (S3)

PGMeetup.NN

Типовые сценарии использования сжатия

Аналитические базы
данных, витрины

- Колонки без индексов
- Низкоэнтропийные данные

Колоночное хранилище

Типовые сценарии использования сжатия

Зашифрованные данные

- Энтропия
- Нет встроенного шифрования
- Не включайте сжатие TOAST для зашифрованных данных

Сначала сжимать, потом шифровать

PGMeetup.NN

Итого:

Use compression, Luke!

- Вопросы?

PGMeetup.NN

СПАСИБО!

postgrespro.ru