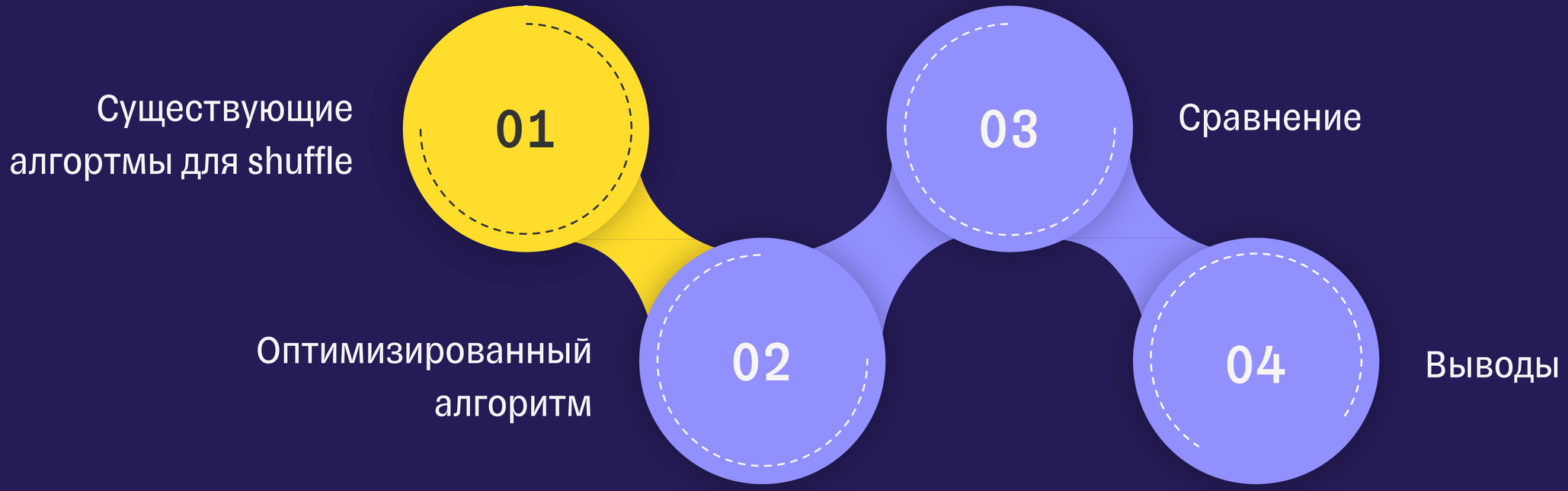
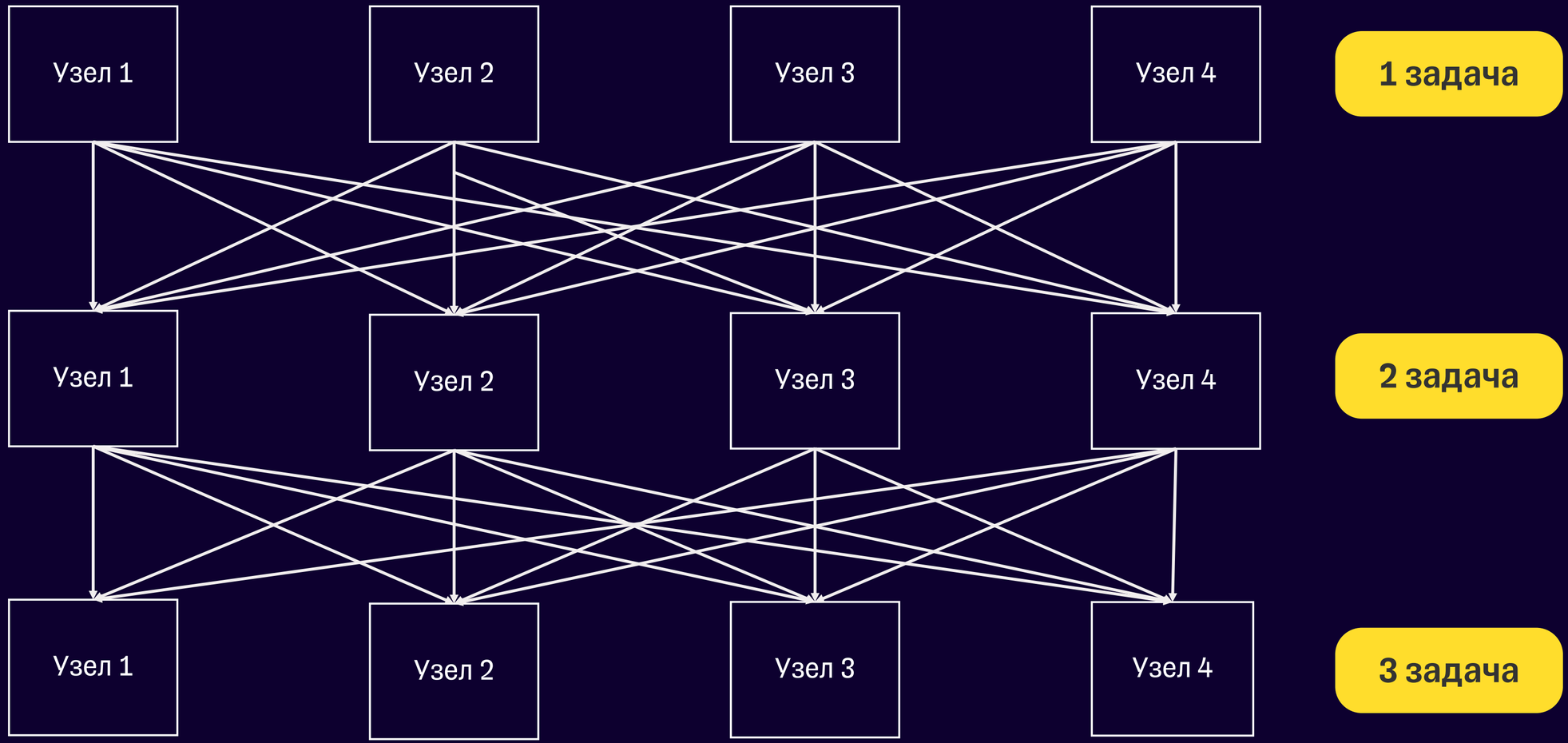
Оптимизация распределения партиций в последовательности задач распределенной обработки данных

Булкина Милена
Data Engineer Т-Банк

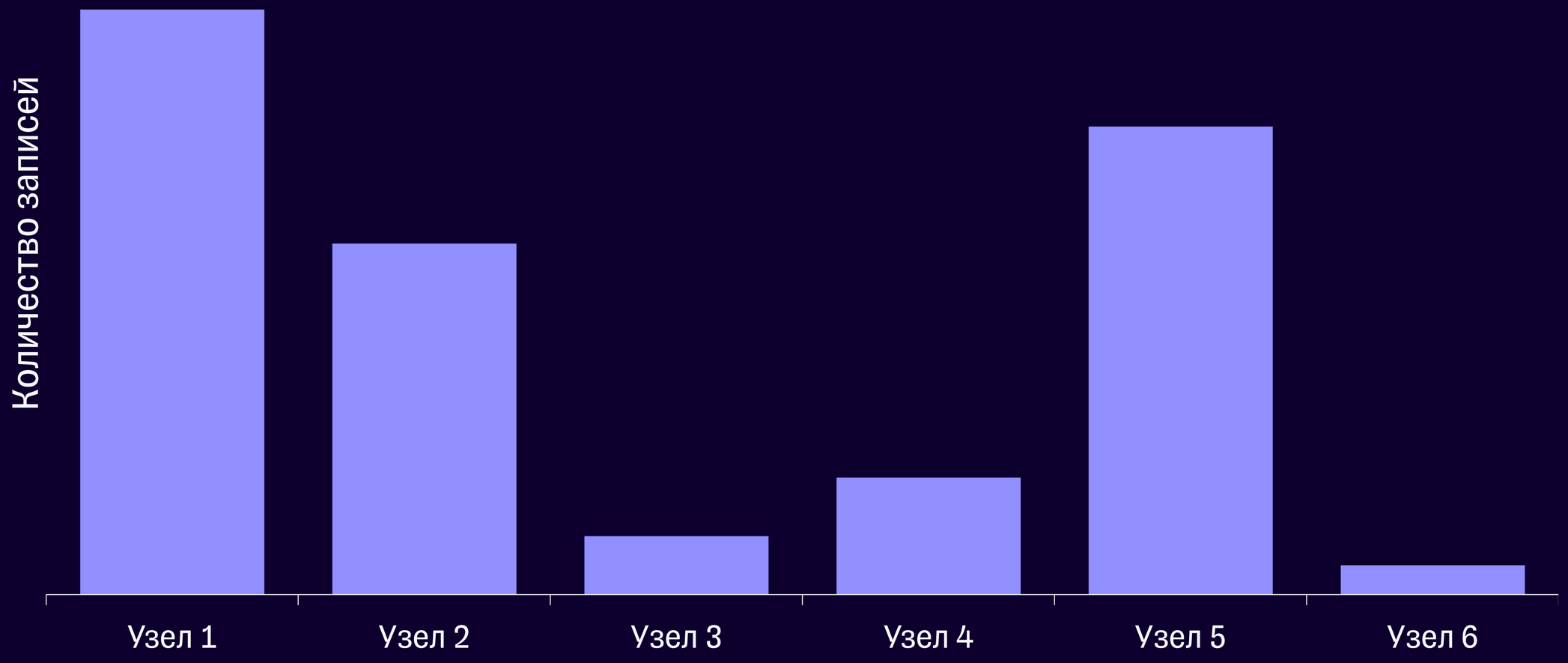
План доклада



Shuffle


Узел 1
Узел 2
Узел 3
Узел 4
1 задача
Узел 1
Узел 2
Узел 3
Узел 4
2 задача
Узел 1
Узел 2
Узел 3
Узел 4
3 задача

Data skew



Мотивация

- Большое количество перераспределяемых данных приводит к существенному замедлению времени выполнения пайплайна

- Необходимо минимизировать объем shuffle-данных при минимальном data skew

Hash Shuffle

- Не возникает перекос на узлах, если равномерное распределение возможно
- Отсутствует оптимизация количества перемещаемых данных
- Задачи не учитываются как последовательность

Hash Shuffle

- Не возникает перекос на узлах, если равномерное распределение возможно
- Отсутствует оптимизация количества перемещаемых данных
- Задачи не учитываются как последовательность

Holistic Shuffle [1]

- Часто возникает перекос после нескольких заданий
- Присутствует оптимизация shuffle-данных, но данные часто находятся на малом количестве узлов
- Задачи не учитываются как последовательность

1 – Holistic Shuffler for the Parallel Processing of SQL Window Functions

Fábio Coelho, José Pereira, Ricardo Vilaça, Rui Oliveira

DOI: 10.1007/978-3-319-39577-7_6

Основная идея Holistic Shuffle

Снижение объёма shuffle-данных
с помощью использования
статистики по ключам

A	NODE
1	1
1	2
1	3
1	3
2	1
2	1
2	3
3	4

Определение узла Holistic Shuffle

A	NODE
1	1
1	2
1	3
1	3

Партицирование по A

A	NODE
1	3
1	3
1	3
1	3

Определение узла Holistic Shuffle

A	NODE
1	1
1	2
1	3
1	3
2	1
2	1
2	3
3	4

Партицирование по A

A	NODE
1	3
1	3
1	3
1	3
2	1
2	1
2	1
3	4

Holistic Shuffle – data skew

A	B	C	NODE
1	…	…	1
1	…	…	2
1	…	…	3
1	…	…	3
2	…	…	1
2	…	…	1
2	…	…	3
3	…	…	4

Партицирование
по другим ключам

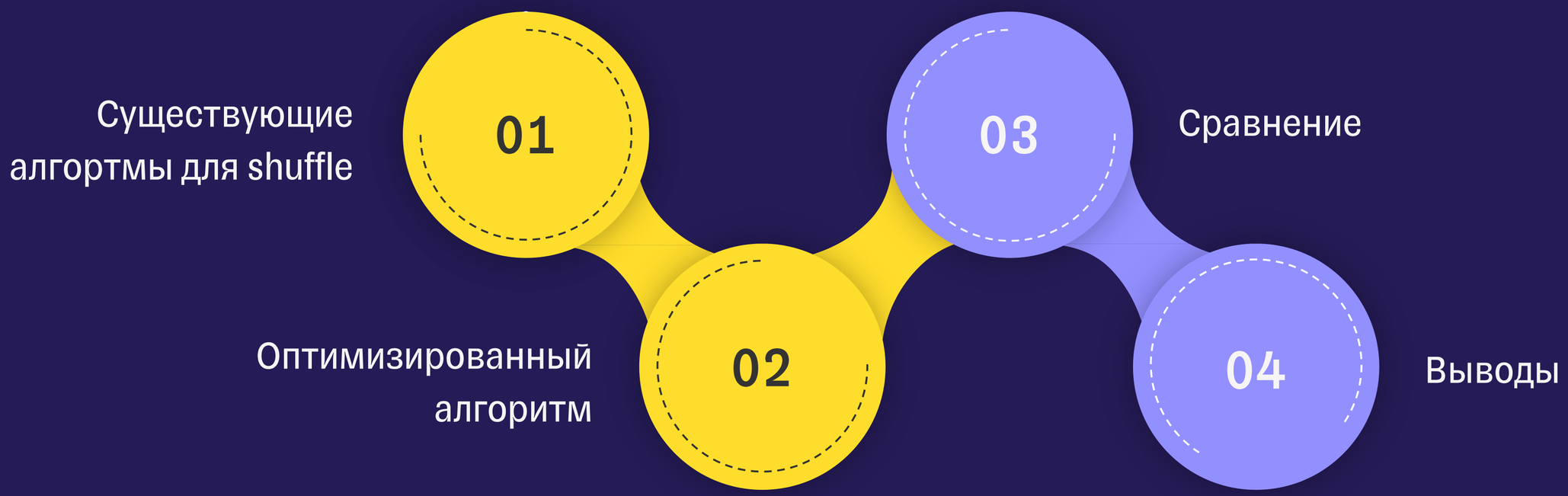
…

A	B	C	NODE
1	…	…	1
1	…	…	1
1	…	…	2
1	…	…	3
2	…	…	1
2	…	…	1
2	…	…	3
3	…	…	1

Партицирование по A

A	B	C	NODE
1	…	…	1
1	…	…	1
1	…	…	1
1	…	…	1
2	…	…	1
2	…	…	1
2	…	…	1
3	…	…	1

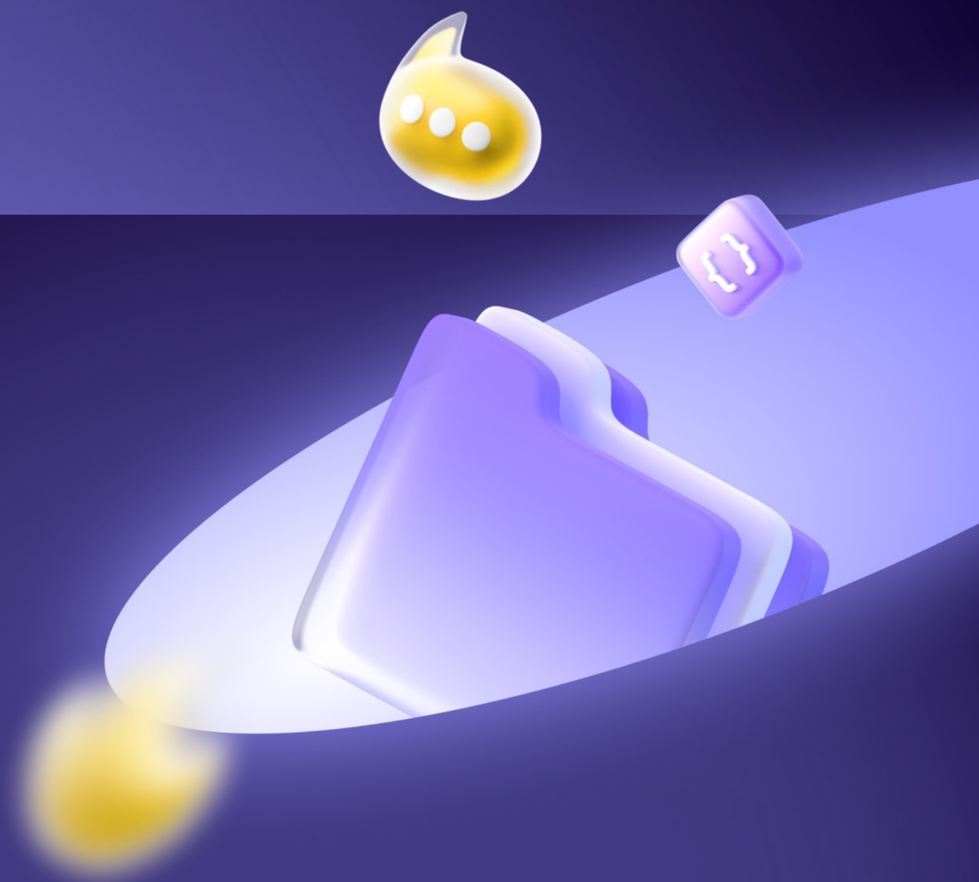
Спустя несколько задач большое количество записей
могут оказаться на одном узле


Существующие алгортмы для shuffle
01
Оптимизированный алгоритм
02
Сравнение
03
Выводы
04

Optimized Shuffle

- В основе идея Holistic Shuffle — использование статистики для сокращения количества shuffle данных
- Ограничивается data skew
- Добавлен поиск общих ключей партицирования у задач, которые идут подряд (оптимизация ключей партицирования в пайплайне)

Data skew

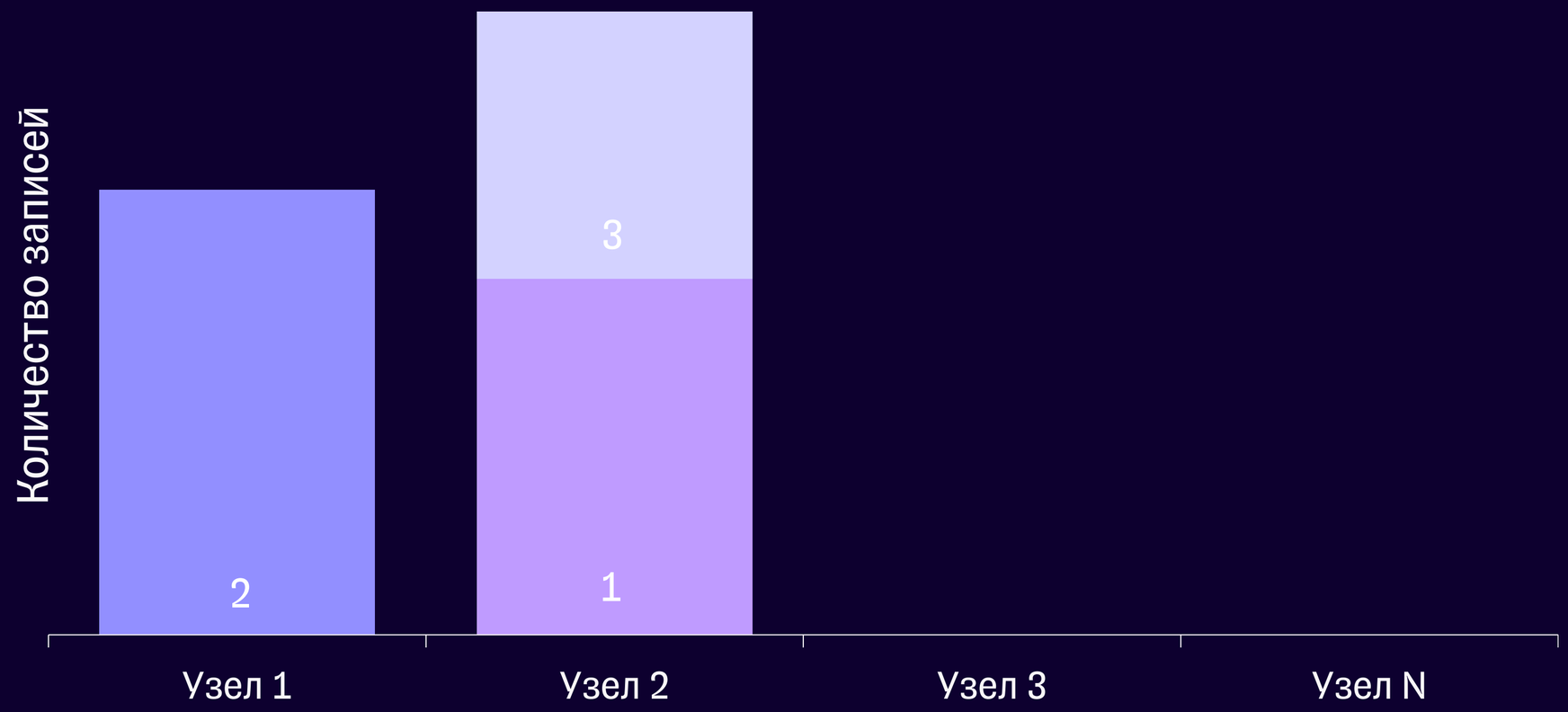


- Входной аргумент – коэффициент, с помощью которого контролируется data skew
- Допускается перекос данных в заданных пределах

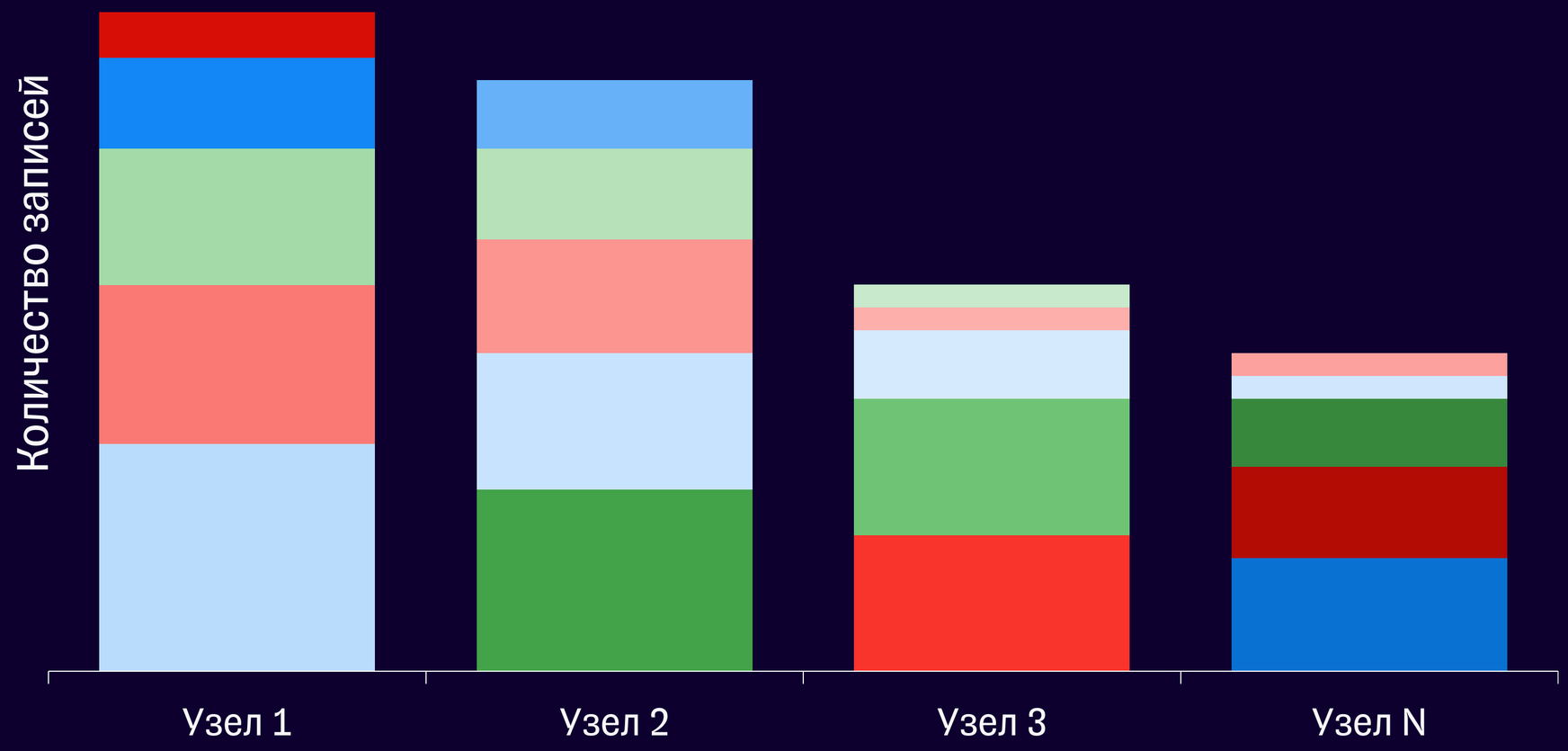
$$\text{rows\_cnt} < k \times \frac{M}{N},$$

где M — общее количество записей, N —количество узлов

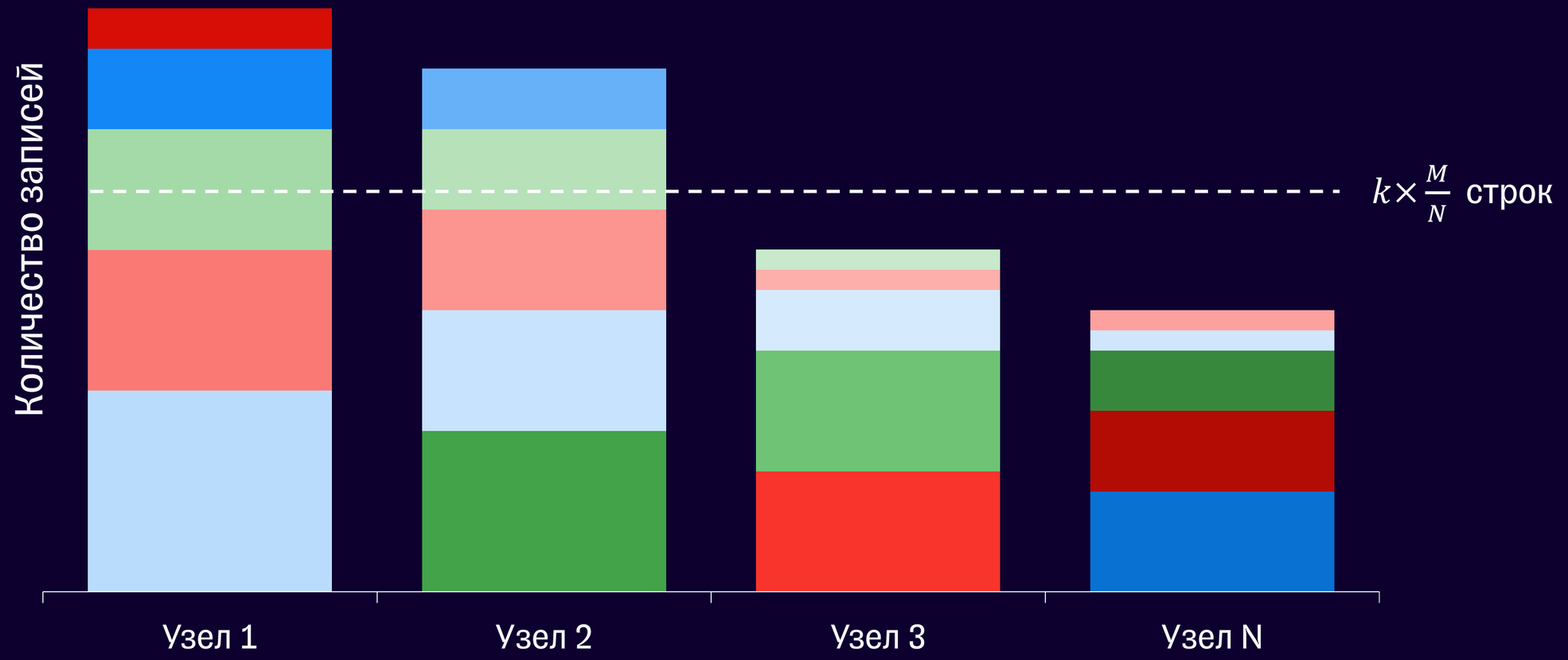
Выбор нового узла



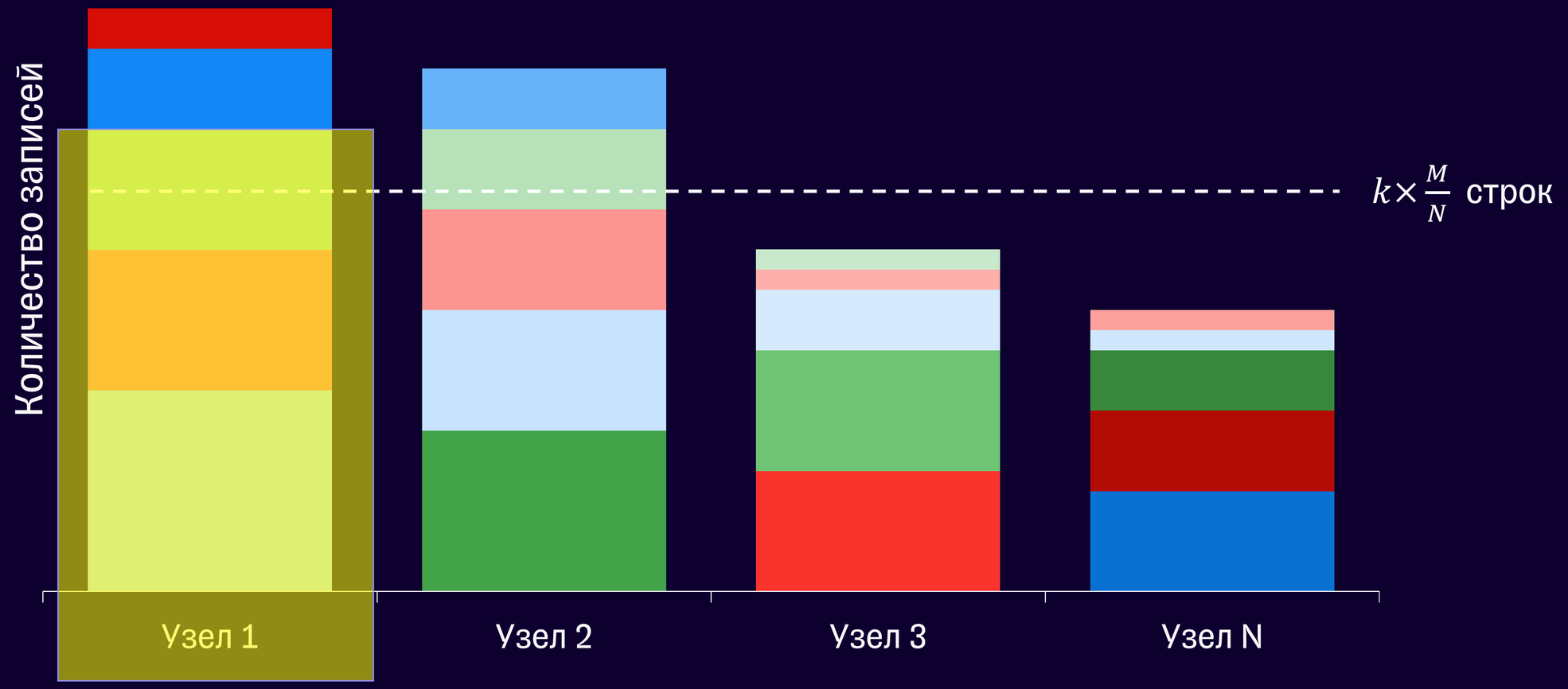
Выбор нового узла



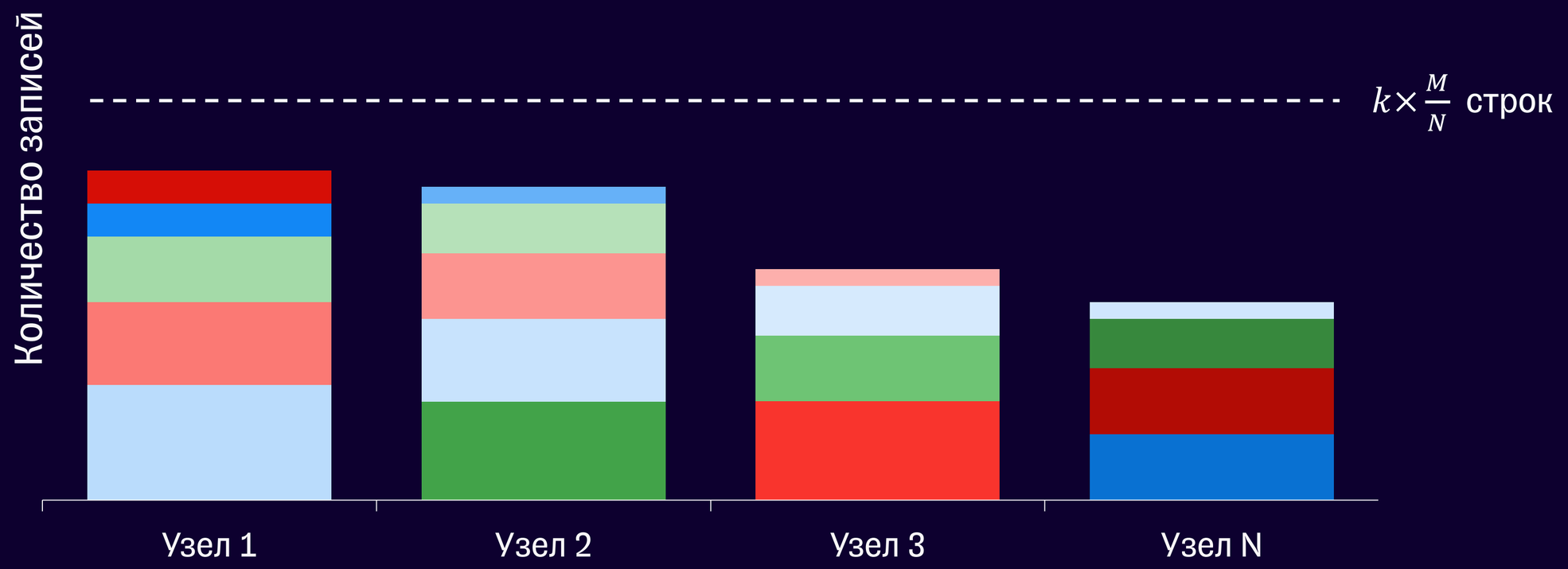
Недопустимый перекос



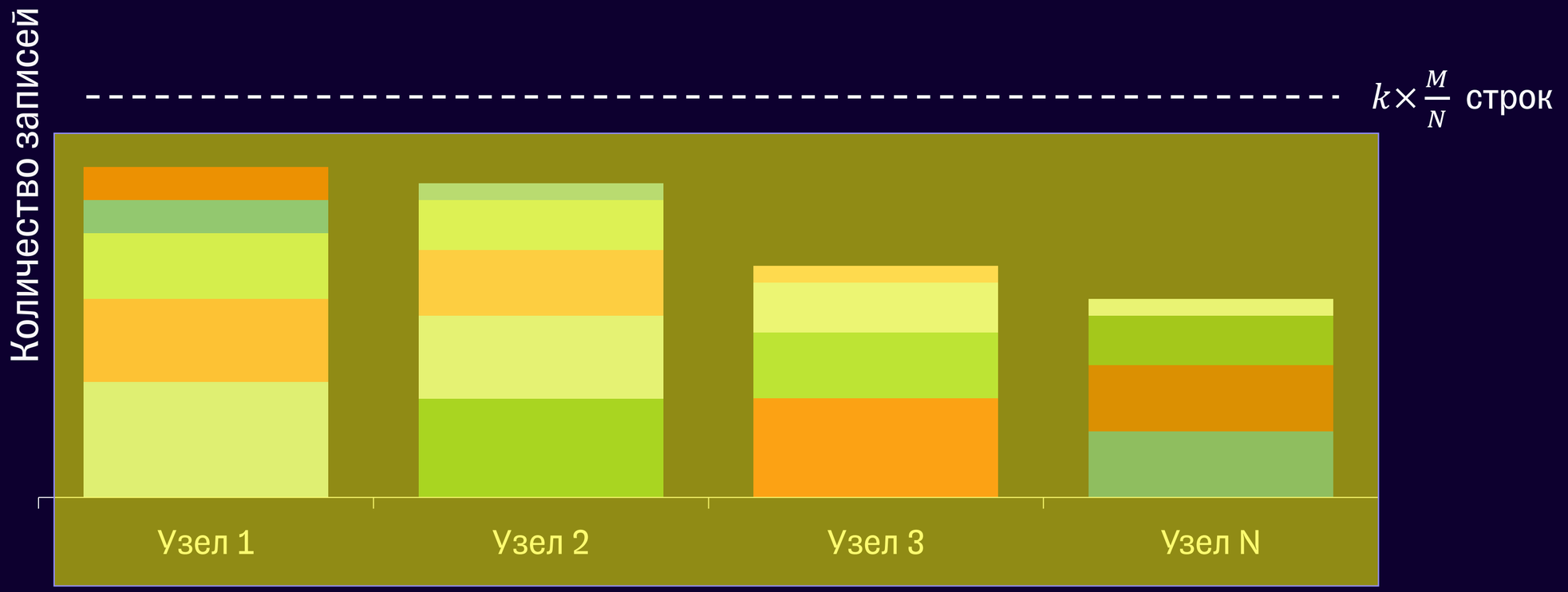
Новый узел


Количество записей
$k \times \frac{M}{N}$ строк
Узел 1
Узел 2
Узел 3
Узел N

Новый узел


Количество записей
$k \times \frac{M}{N}$ строк
Узел 1
Узел 2
Узел 3
Узел N

Новый узел


Количество записей
$k \times \frac{M}{N}$ строк
Узел 1
Узел 2
Узел 3
Узел N

Сокращение количества ключей

- partition by a, b
- partition by b, c
- partition by b, d, e
- partition by f

При достаточной кардинальности

- partition by b
- partition by f

Для задач, которые идут подряд и имеют общее подмножество ключей, можно сократить набор ключей при достаточной кардинальности подмножества ключей

Применимость алгоритма

Ограничения

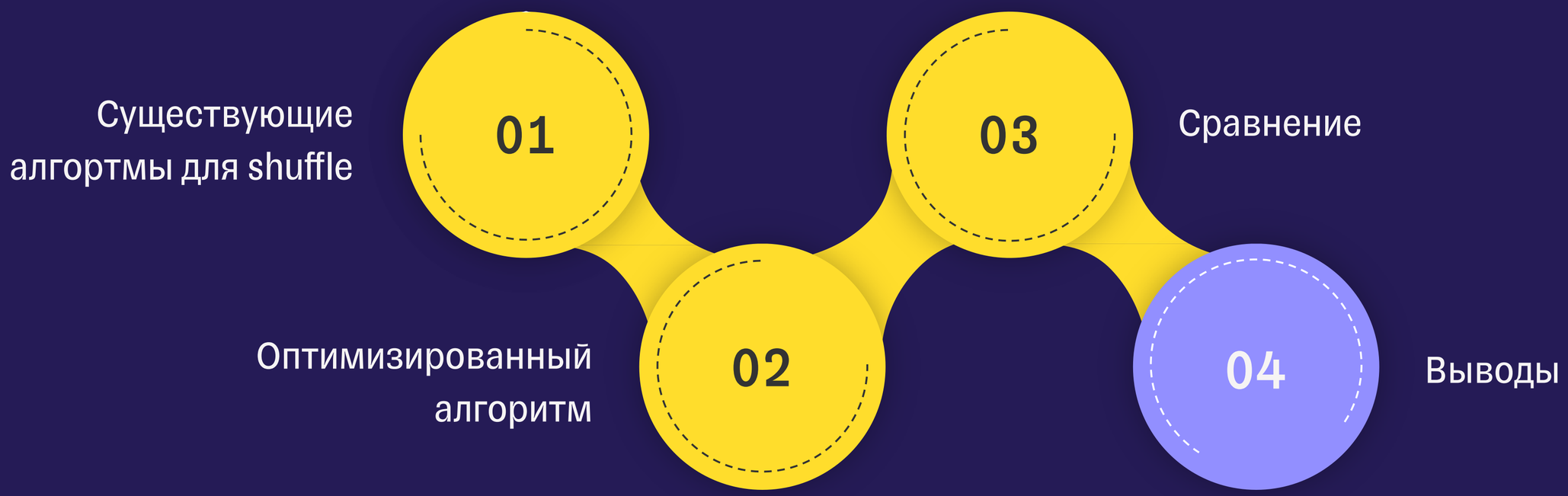
- Объём ключей, используемых для репартицирования, занимает незначительную часть всего датасета
- Только для колоночного формата данных

Пример данных, для которых может быть применен алгоритм

- Данные сейсморазведки
- Посты в блогах
- Широкие таблицы

Применимость алгоритма

Не используются в качестве
ключей репартицирования


Существующие алгортмы для shuffle
01
Оптимизированный алгоритм
02
Сравнение
03
Выводы
04

Тестовые данные

- Тест проводился на данных сейсморазведки
- Особенностью данных является большое количество атрибутов, малый процент из которых используется в качестве ключей партицирования
- 30 млн записей
- Сравнение проводилось на трех пайплайнах

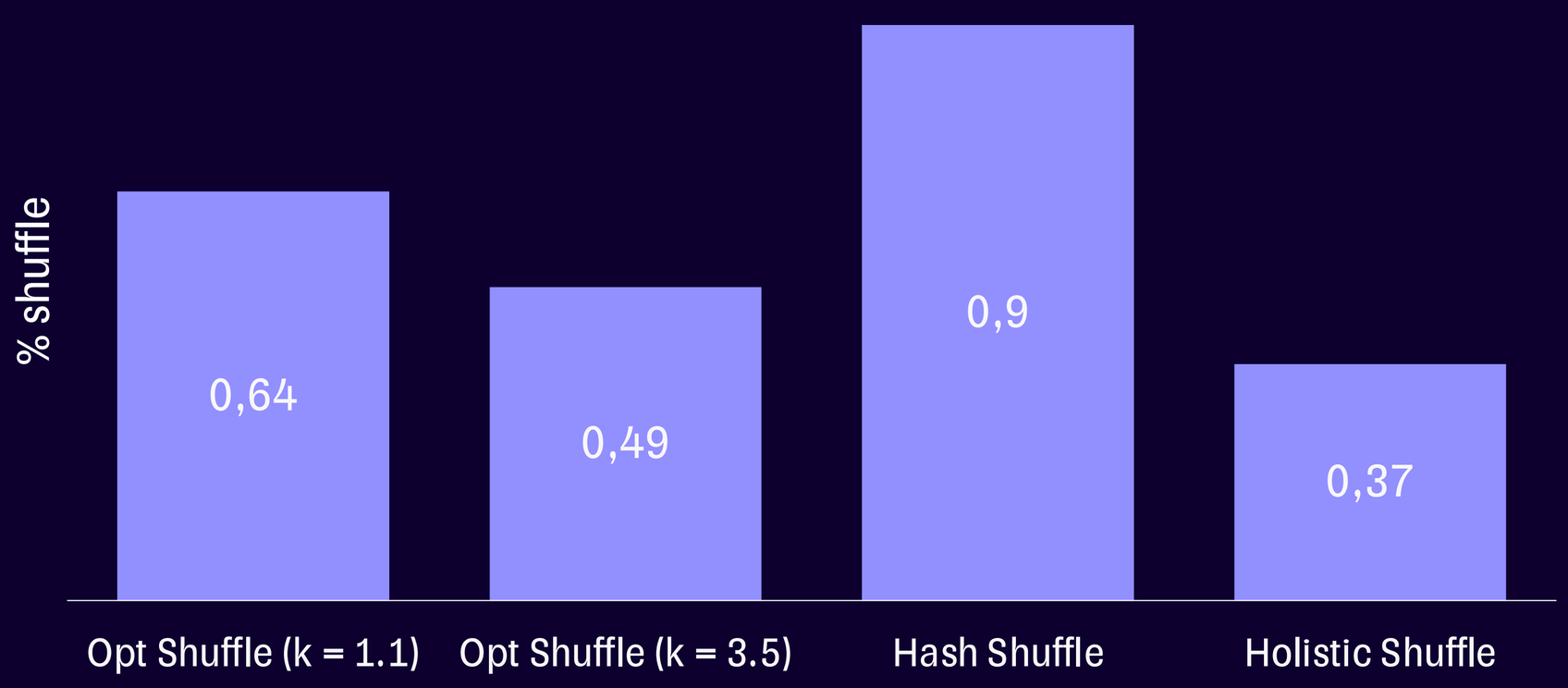
Метрики

Метрики используемые для сравнения

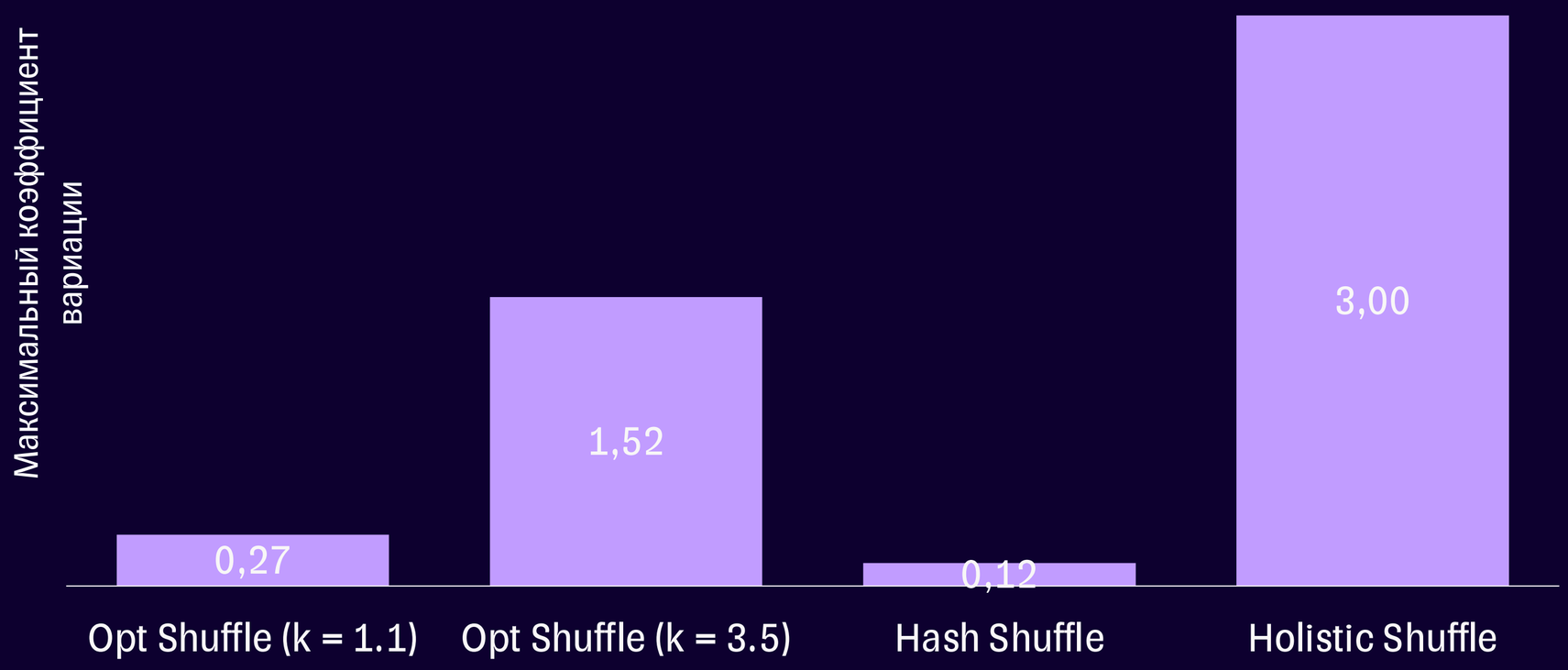
- Доля передаваемых данных
- Коэффициент вариации

$$c_v = \frac{\sigma}{\mu}, \text{где } \sigma \text{ — стандартное отклонение, } \mu \text{ — среднее}$$

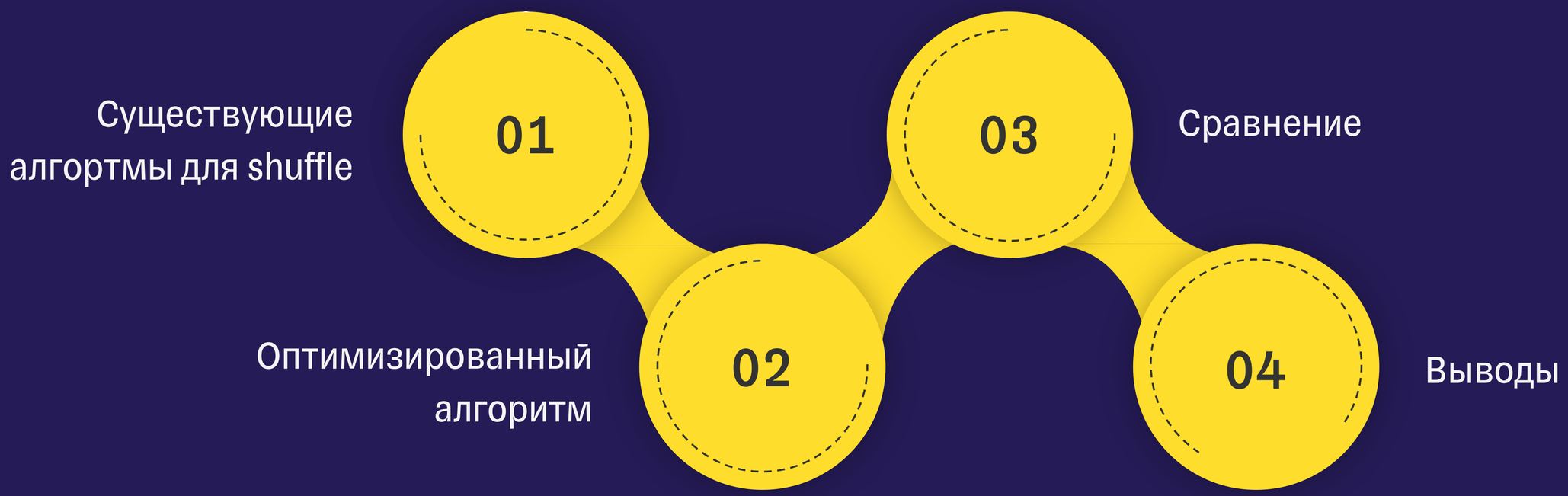
Сравнение – доля shuffle-данных



Сравнение – коэффициент вариации



Сравнение – доля shuffle-данных


Существующие алгортмы для shuffle
01
Оптимизированный алгоритм
02
Сравнение
03
Выводы
04

Выводы

Для тестовых данных и k = 1.1

- Процент shuffle-данных — 64% (Holistic — 45%, Hash — 90%)
- Коэффициент вариации не превышает 0.28 (Holistic — 3.0, Hash — 0.12)

Внедрение

Идея алгоритма была внедрена у индустриального партнера университета ИТМО, который занимается анализом данных сейсморазведки

Future Work

Поиск оптимального коэффициента k с помощью моделирования конфигурации системы (задача оптимизации, оптимизируем время выполнения пайплайна)

Масштабируемость

Определение максимального значения доли атрибутов, используемых в качестве ключей для репартицирования при котором выгодно использовать алгоритм

БАНК

Спасибо!