

# **MPP системы и колоночное хранение**

# Массивно-параллельные системы обработки

При использовании массивно-параллельной архитектуры данные разделяются на фрагменты, обрабатываемые независимыми центральными процессорами (CPU) и хранящиеся на разных носителях.

Это похоже на загрузку разных фрагментов данных на несколько объединенных в сеть персональных компьютеров.



# Аппаратная архитектура на примере VERTICA

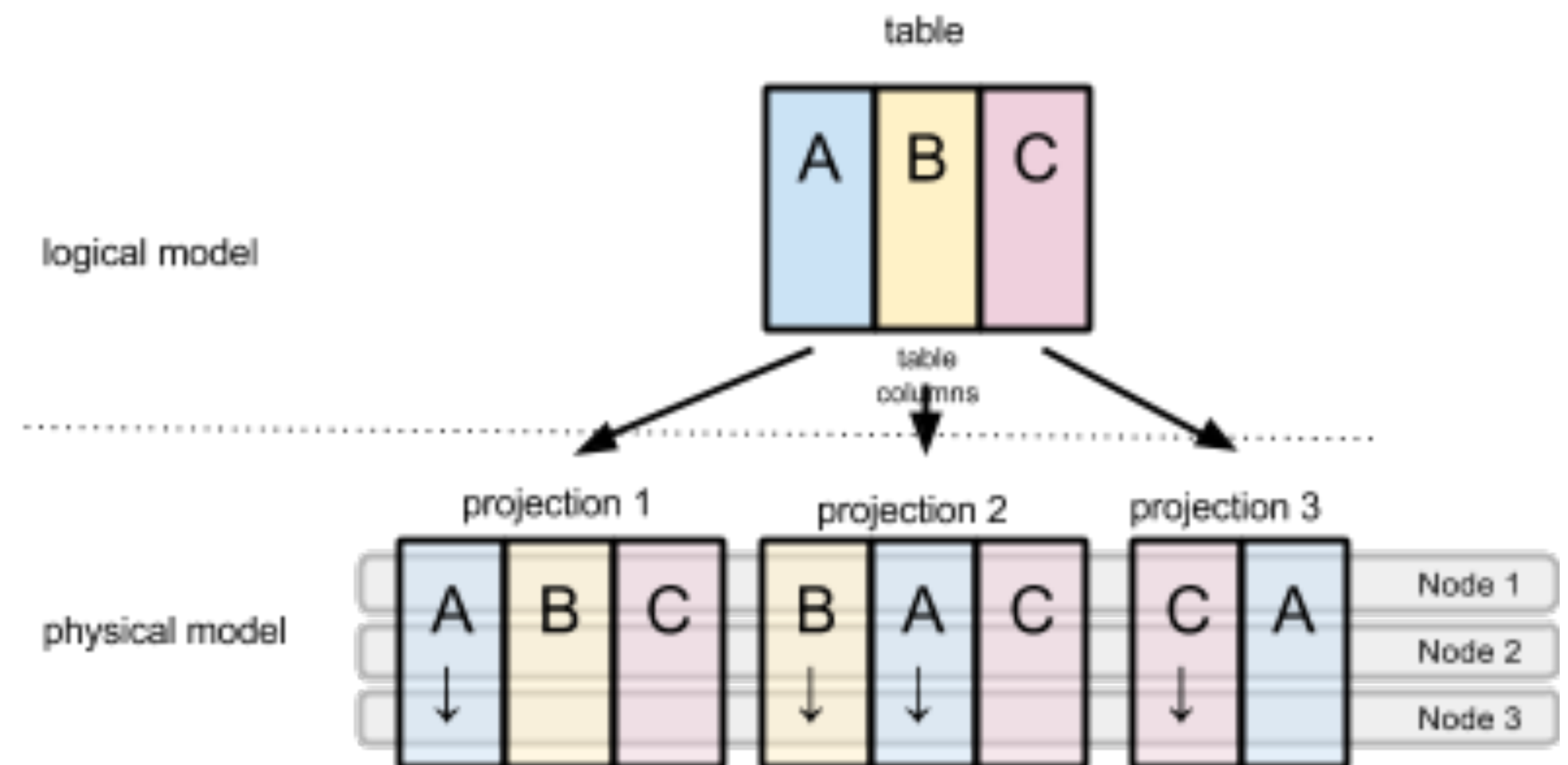
Основные свойства:

- отсутствует единая точка отказа,
- каждый узел независим и самостоятелен,
- отсутствует единая для всей системы точка подключения,
- узлы инфраструктуры дублируются,
- данные на узлах кластера автоматически копируются.

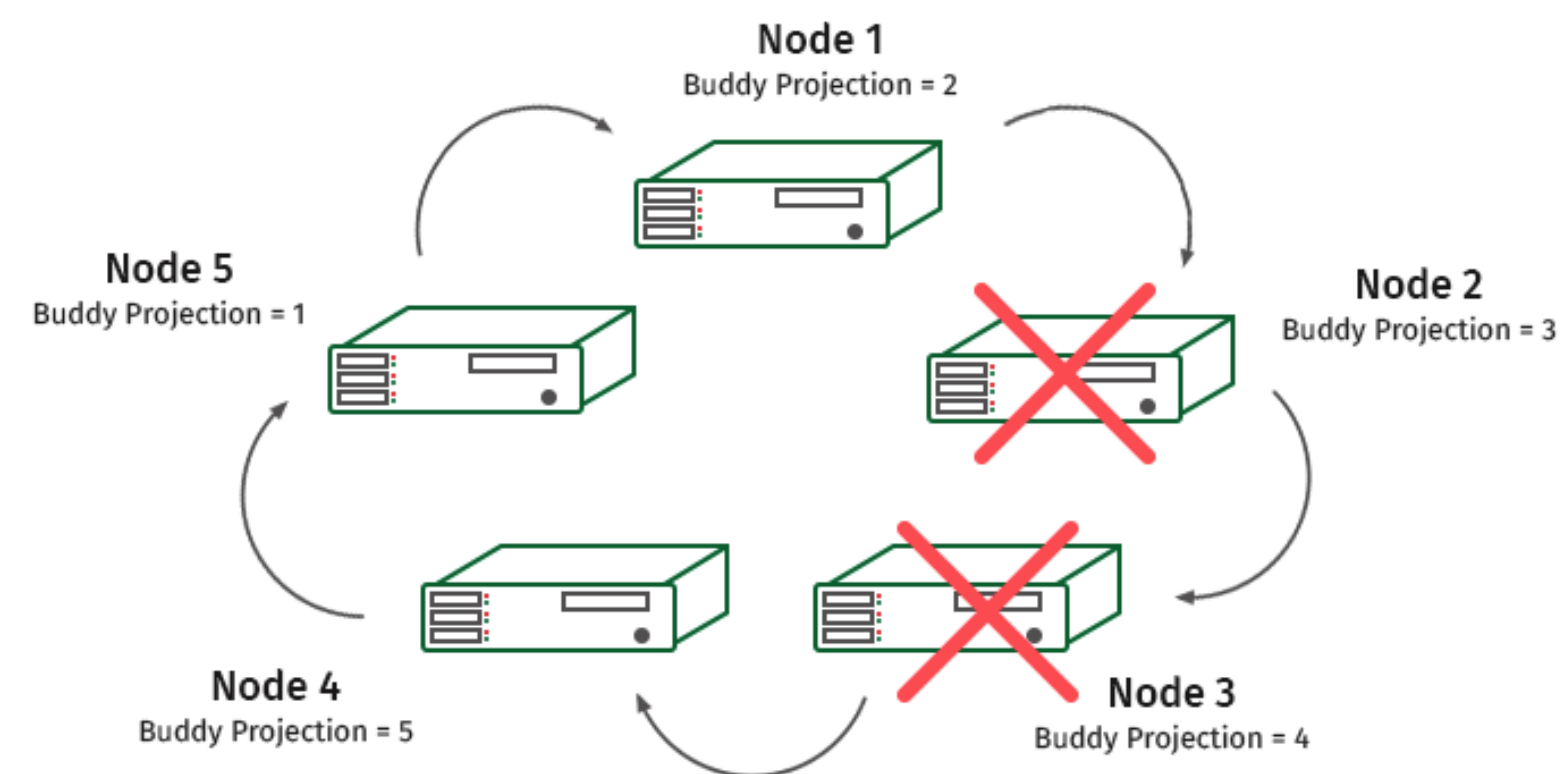
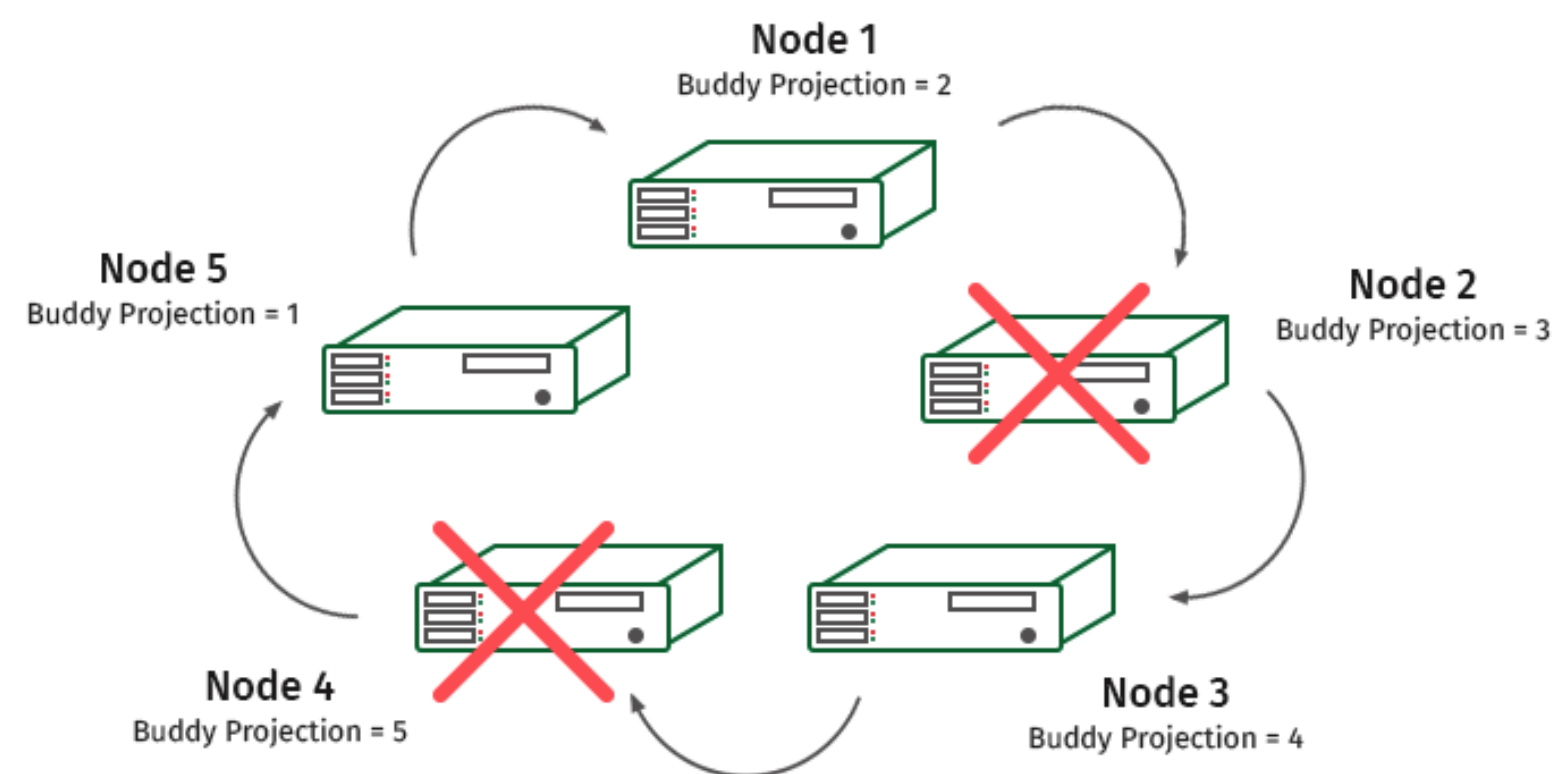
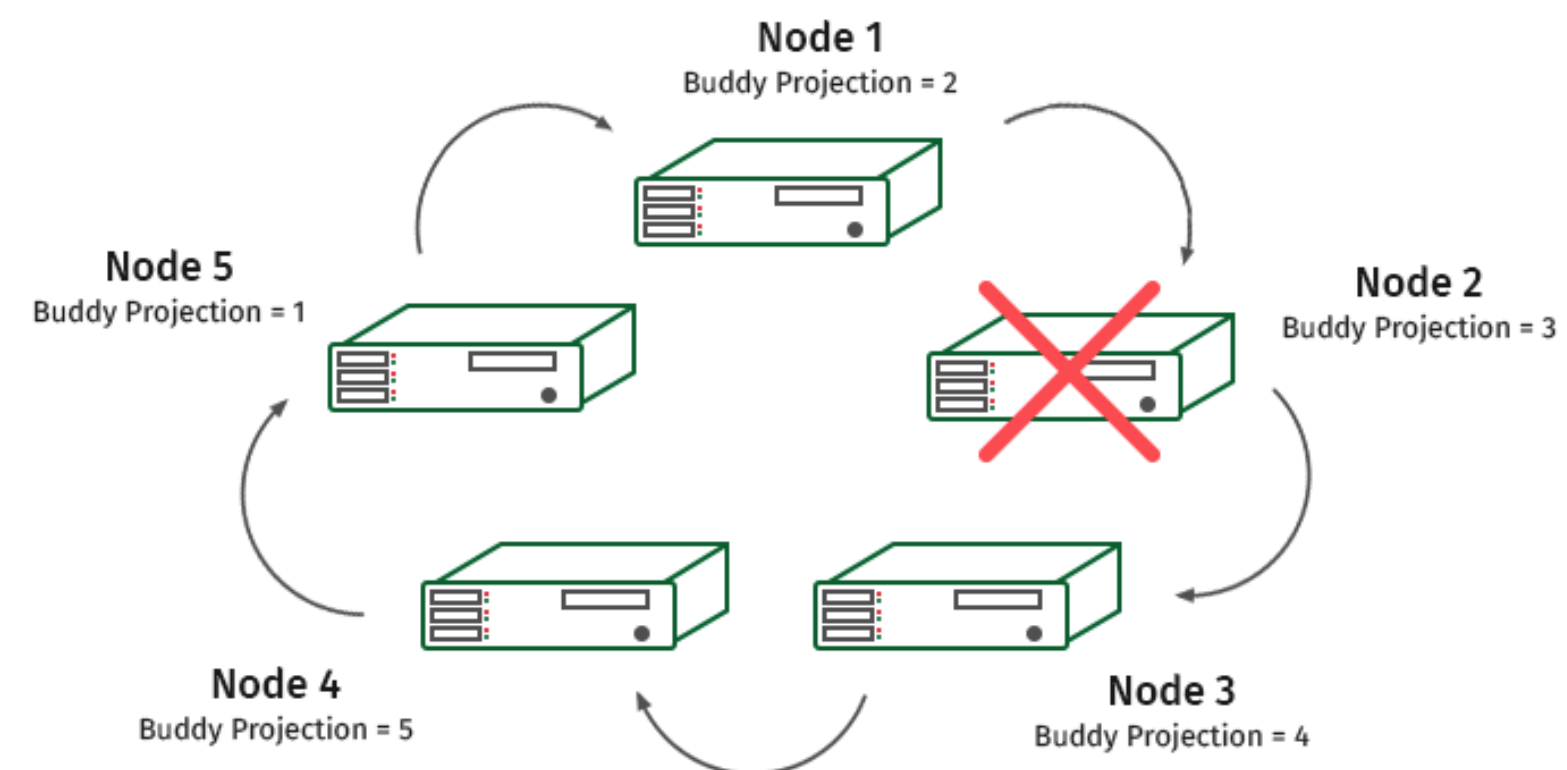
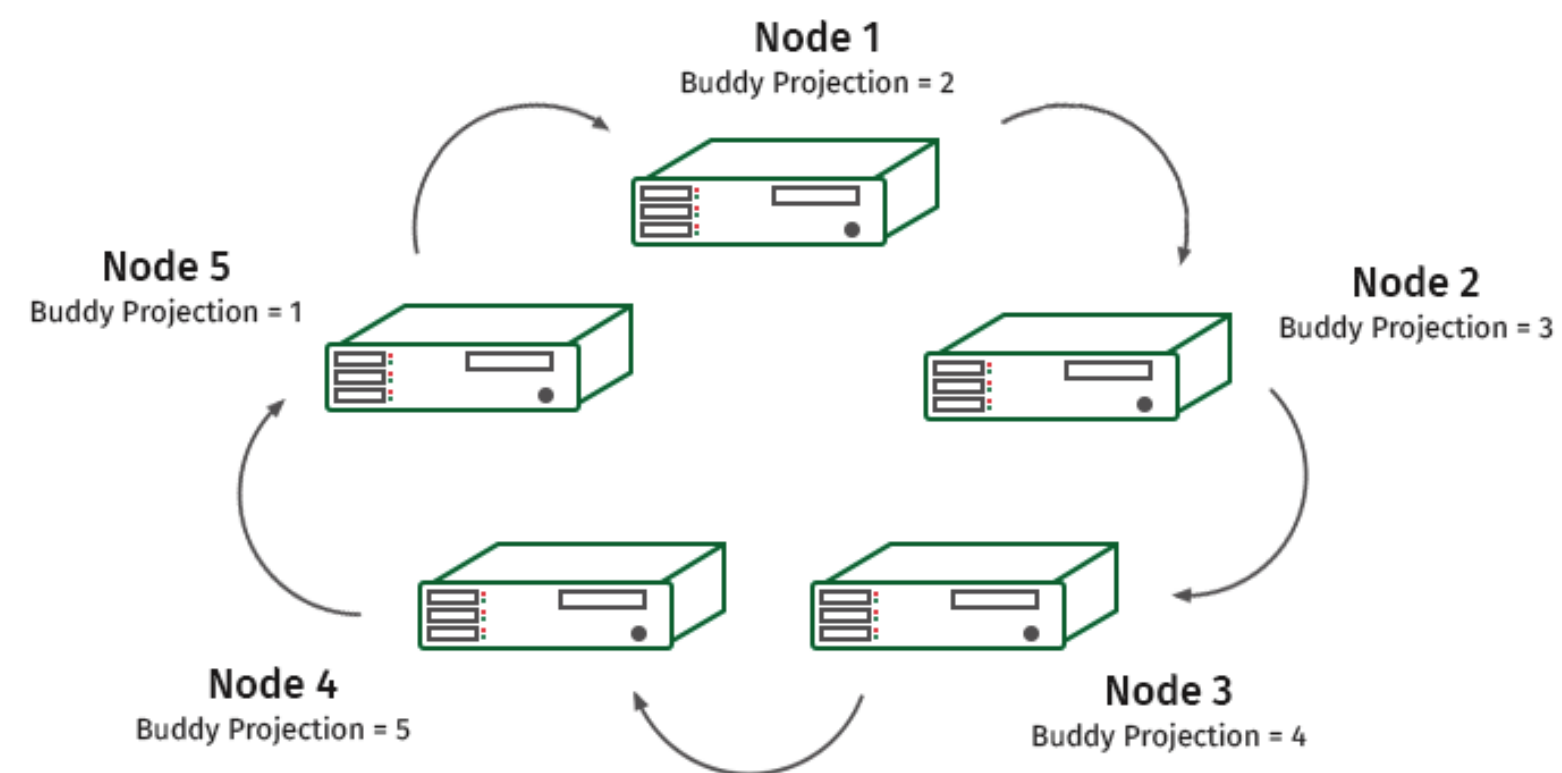
Логические единицы хранения информации — это схемы, таблицы и представления. Физические единицы — это проекции.

Проекции бывают нескольких типов:

- суперпроекции (Superprojection),
- запрос-ориентированные проекции (Query-Specific Projections),
- агрегированные проекции (Aggregate Projections).



# Отказоустойчивость



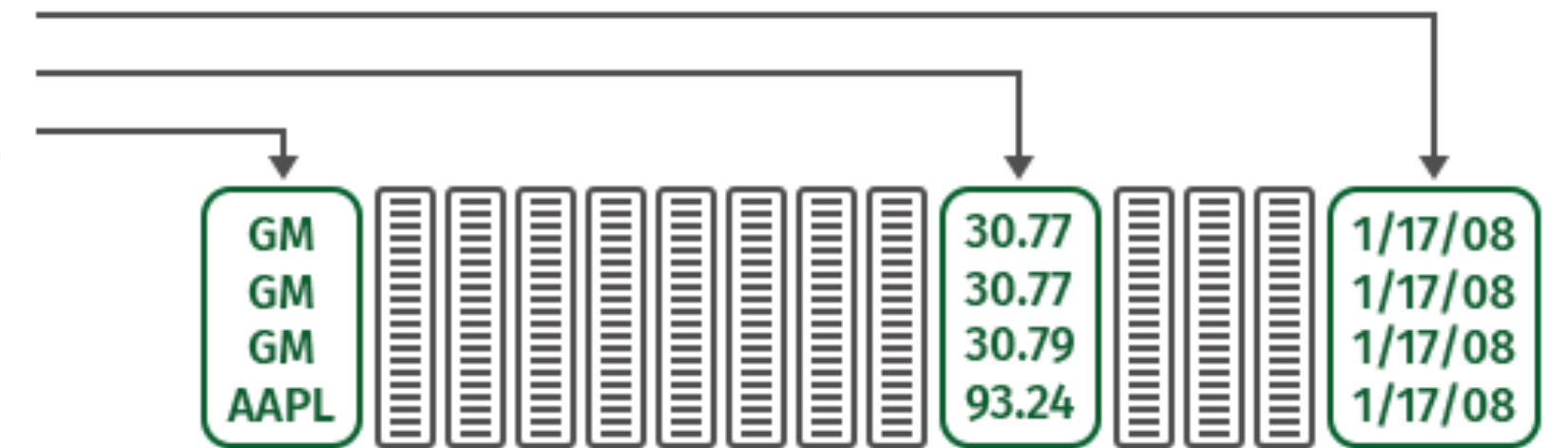
# В чем выгода колоночного хранения

Если мы читаем строки, то, например, для выполнения команды

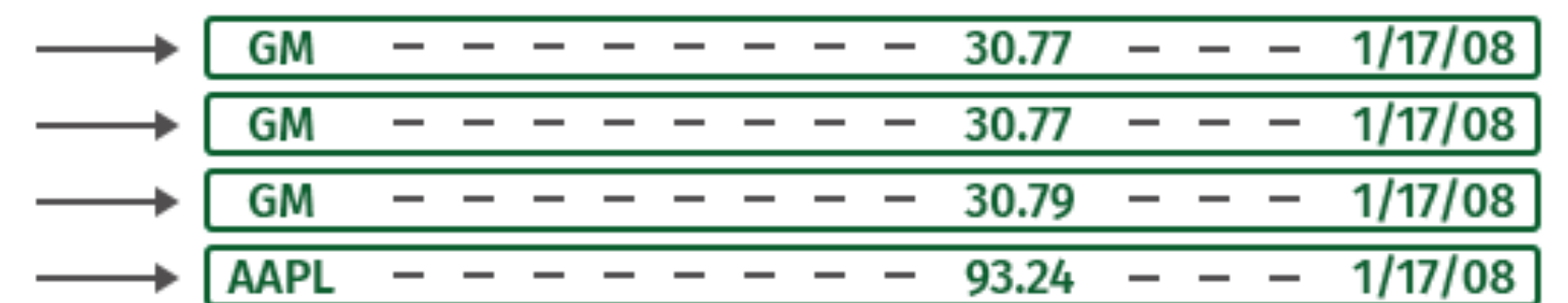
```
SELECT 1,11,15 FROM table1
```

нам придется читать всю таблицу. Это огромный объем информации. В данном случае колоночный подход выгоднее. Он позволяет считать только три нужных нам столбца, экономя память и время

Хранение  
по столбцам  
Чтение трёх столбцов

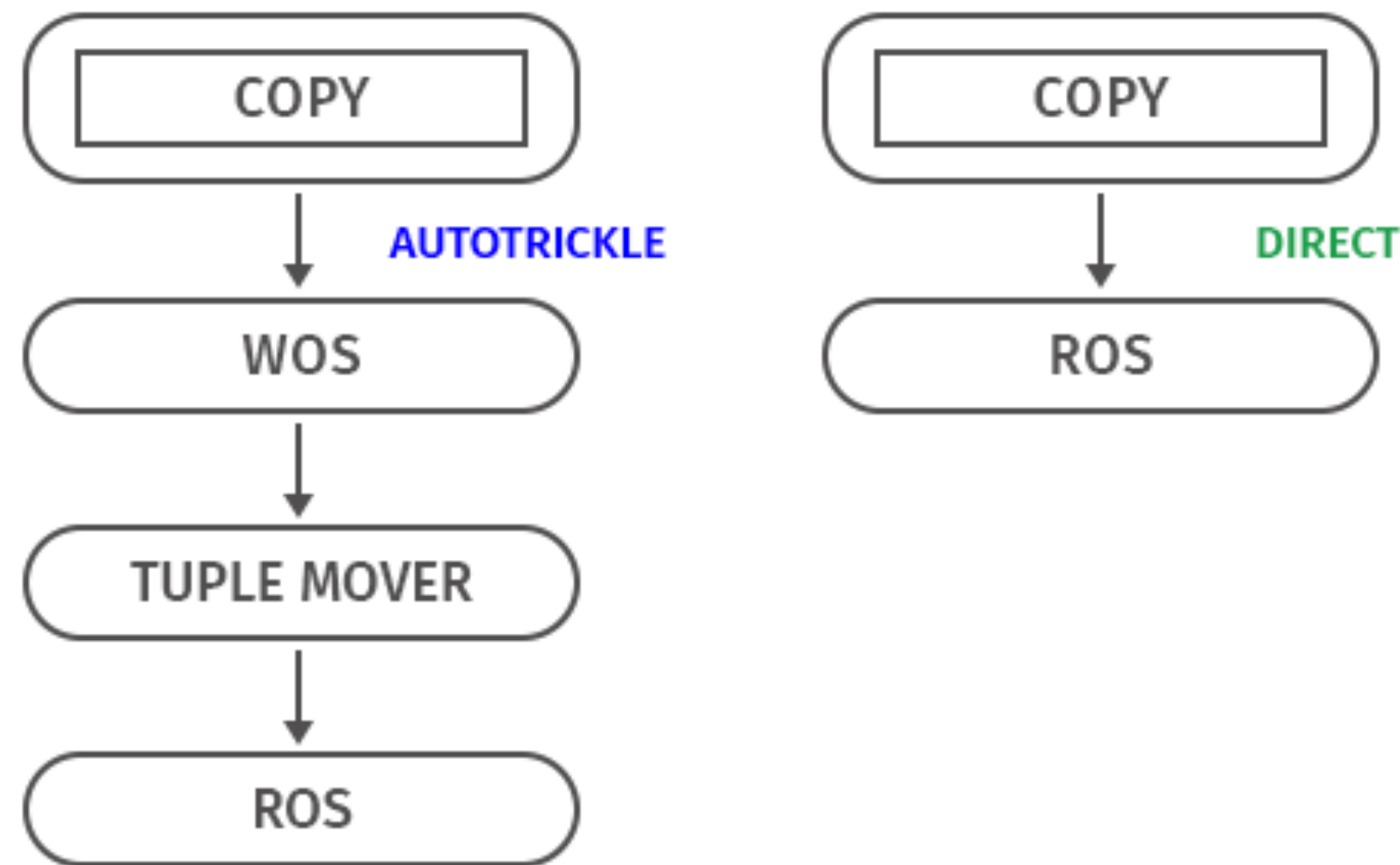


Хранение  
по строкам  
Чтение всех столбцов





# Логическое хранение данных

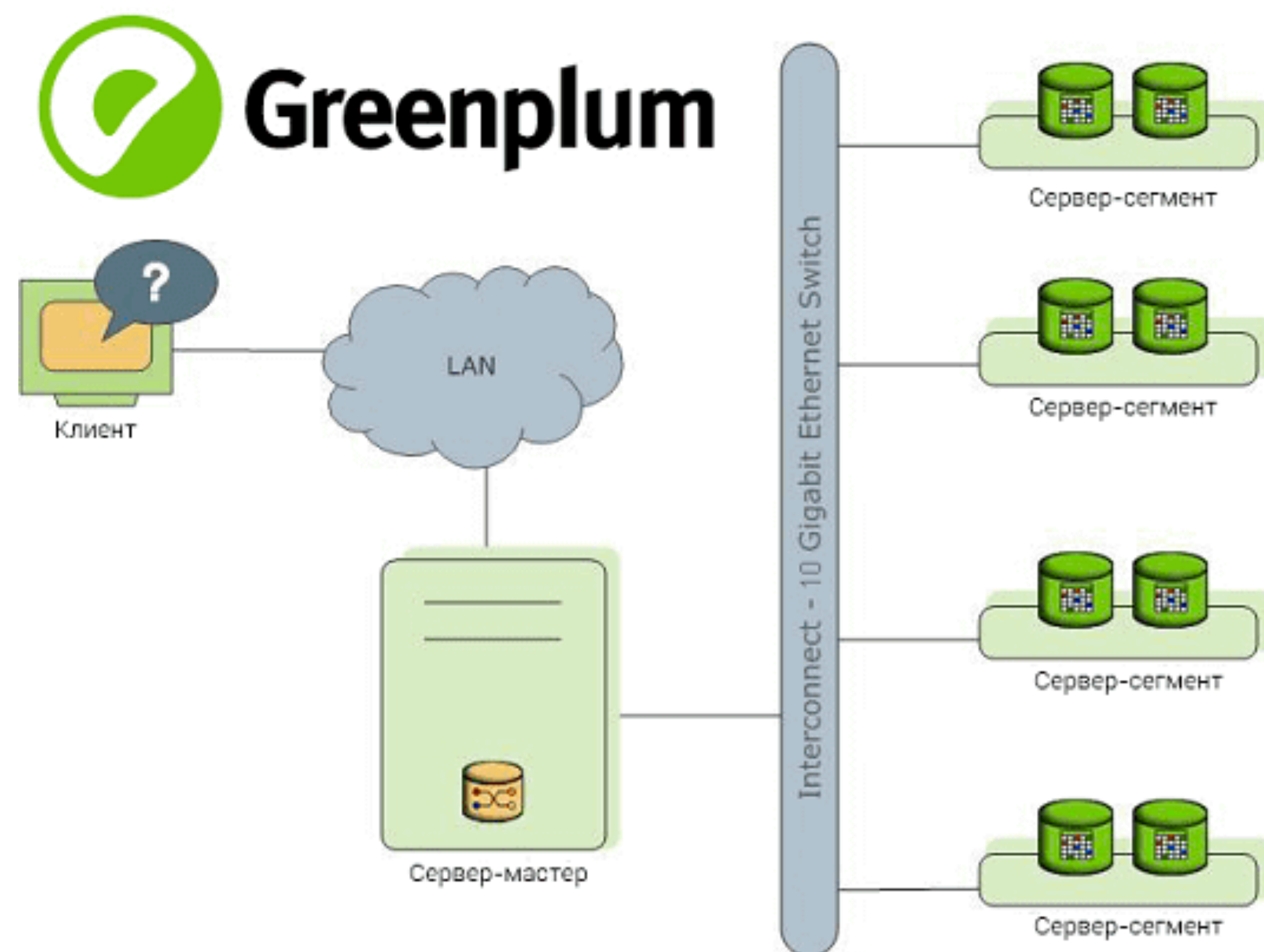


Вне зависимости от того, куда записаны данные — в WOS или в ROS — они доступны сразу. Но из WOS чтение идет медленнее, потому что данные там не сгруппированы.

Tuple Mover — это инструмент-уборщик, который выполняет две операции:

- Moveout — сжимает и сортирует данные в WOS, перемещает их в ROS и создает для них в ROS новые контейнеры.
- Mergeout — подметает за нами, когда мы используем DIRECT. Мы не всегда способны грузить столько информации, чтобы получались большие ROS-контейнеры. Поэтому он периодически объединяет небольшие контейнеры ROS в более крупные, очищает данные с пометкой на удаление, работая при этом в фоновом режиме (по времени, заданному в конфигурации).

# Архитектура GreenPlum

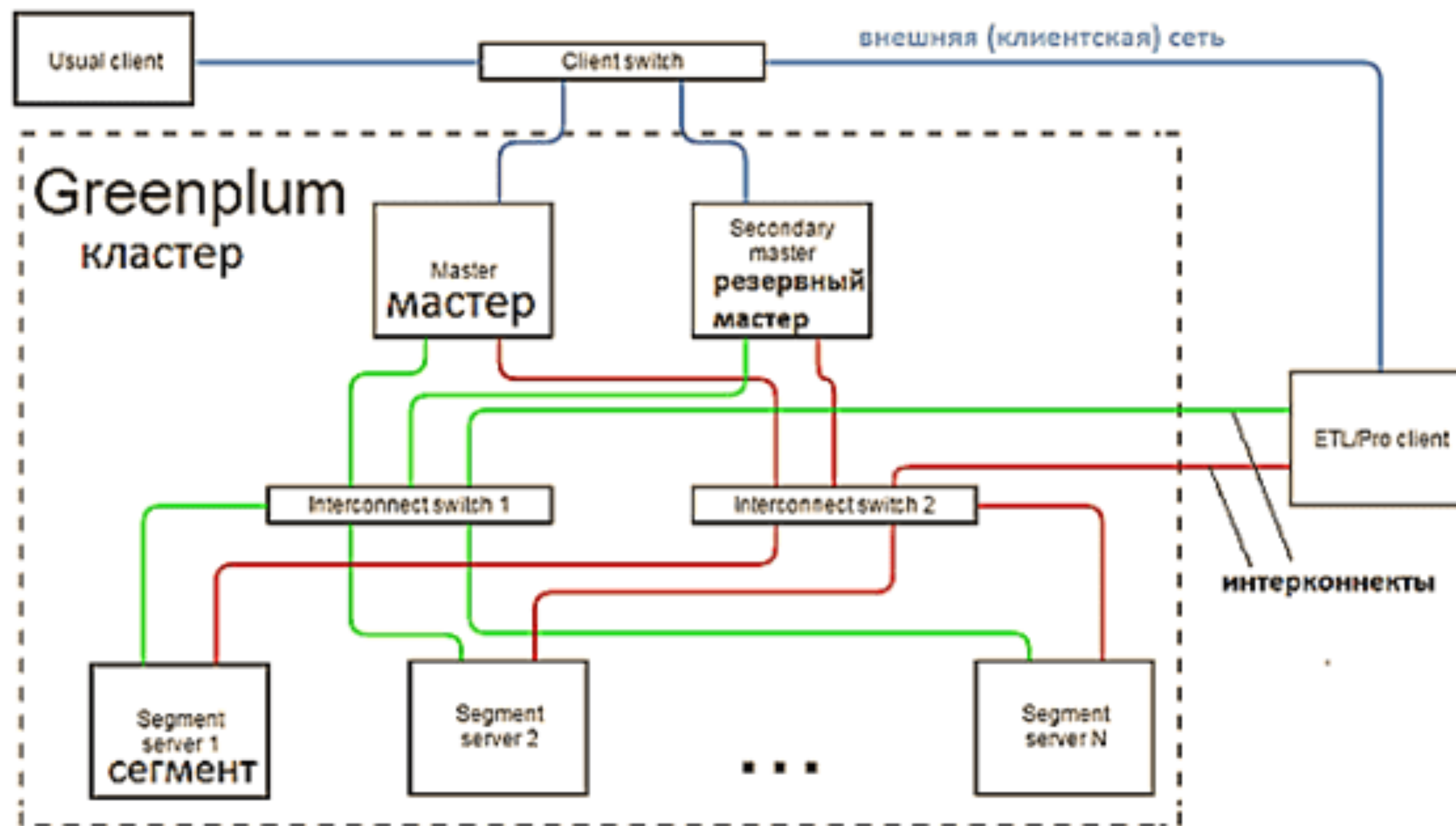


## Архитектура и принципы работы

Таким образом, между компонентами кластер Greenplum существуют следующие отношения:

- Мастер-сервер (Master host), где развернут главный инстанс PostgreSQL (Master instance). Он является точкой входа в Greenplum и представляет собой экземпляр базы данных, к которому подключаются клиенты, отправляя SQL-запросы. Мастер координирует свою работу с сегментами – другими экземплярами базы данных в этой Big Data системе
- Резервный мастер (Secondary master instance) — инстанс PostgreSQL, который вручную включается в работу при отказе основного мастера
- Сервер-сегмент (Segment host), который хранит и обрабатывает данные. Обычно 1 хост-сегмент содержит 2-8 сегментов Greenplum.

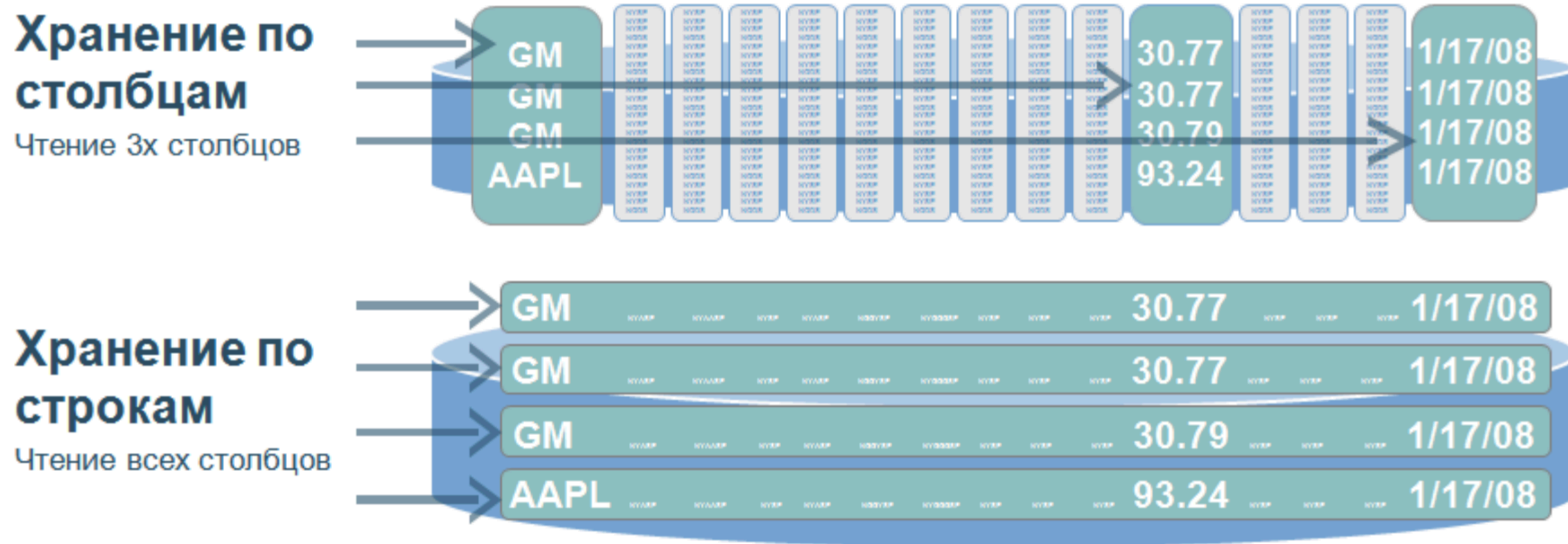
# Архитектура GreenPlum





# Что такое Parquet

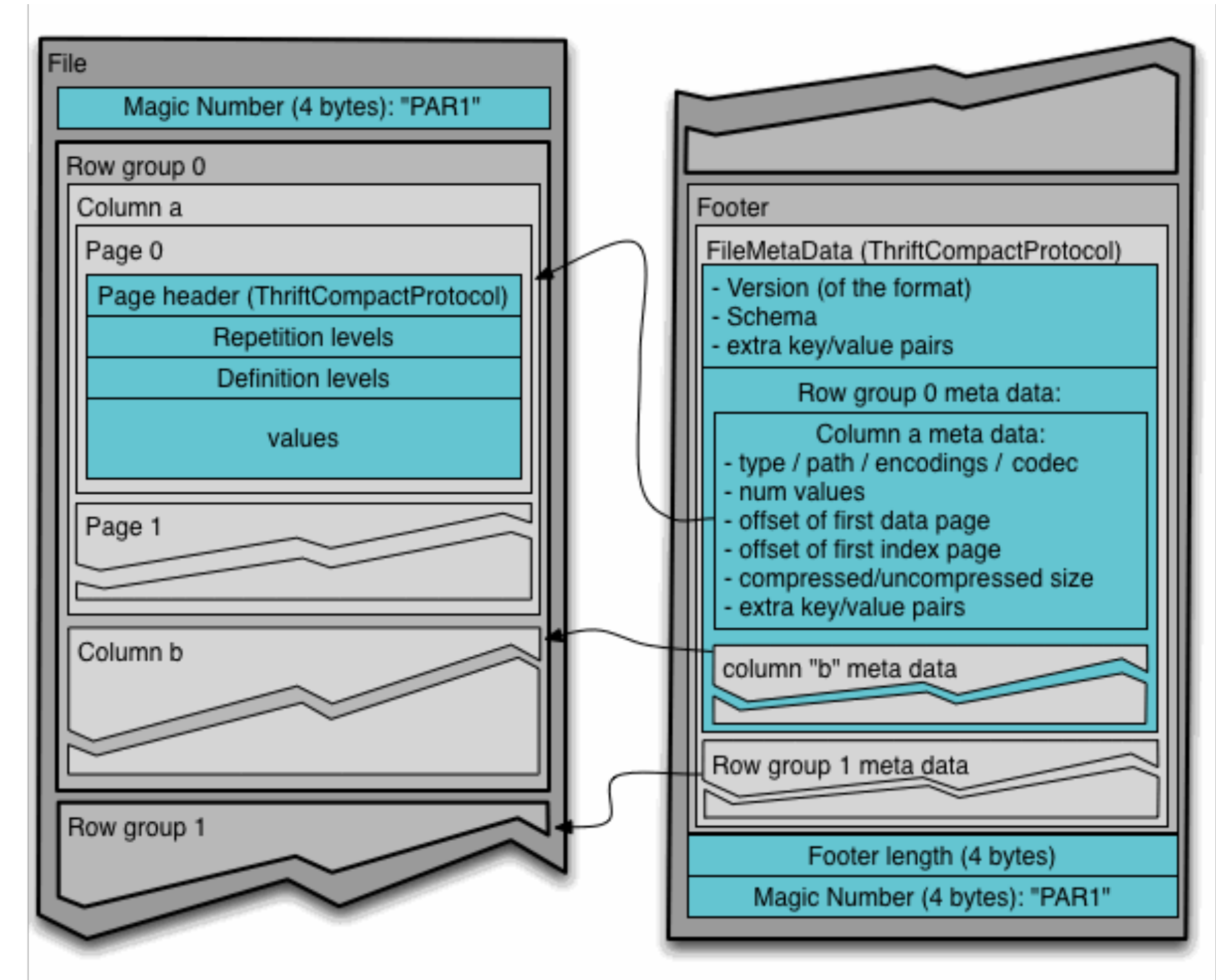
Это бинарный, колоночно - ориентированный формат хранения данных, изначально созданный для экосистемы hadoop. Разработчики уверяют, что данный формат хранения идеален для big data (неизменяемых).



# Как выглядит структура Parquet файлов

Файлы имеют несколько уровней разбиения на части, благодаря чему возможно довольно эффективное параллельное исполнение операций поверх них:

- Row-group — это разбиение, позволяющее параллельно работать с данными на уровне Map-Reduce
- Column chunk — разбиение на уровне колонок, позволяющее распределять IO операции
- Page — Разбиение колонок на страницы, позволяющее распределять работу по кодированию и сжатию



# Хранение данных: CSV vs Parquet

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

Тогда в текстовом файле, скажем, csv мы бы хранили данные на диске примерно так:

A1	B1	C1	A2	B2	C2	A3	B3	C3
----	----	----	----	----	----	----	----	----

В случае с Parquet:

A1	A2	A3	B1	B2	B3	C1	C2	C3
----	----	----	----	----	----	----	----	----