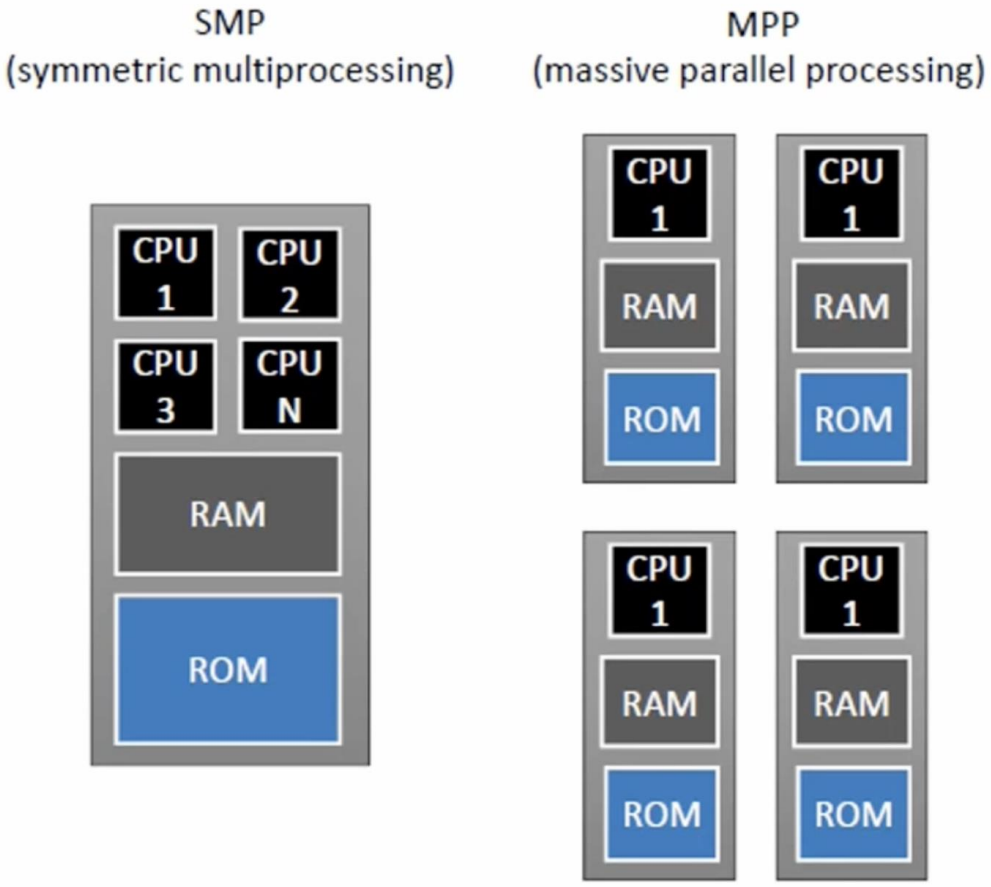
GreenPlum

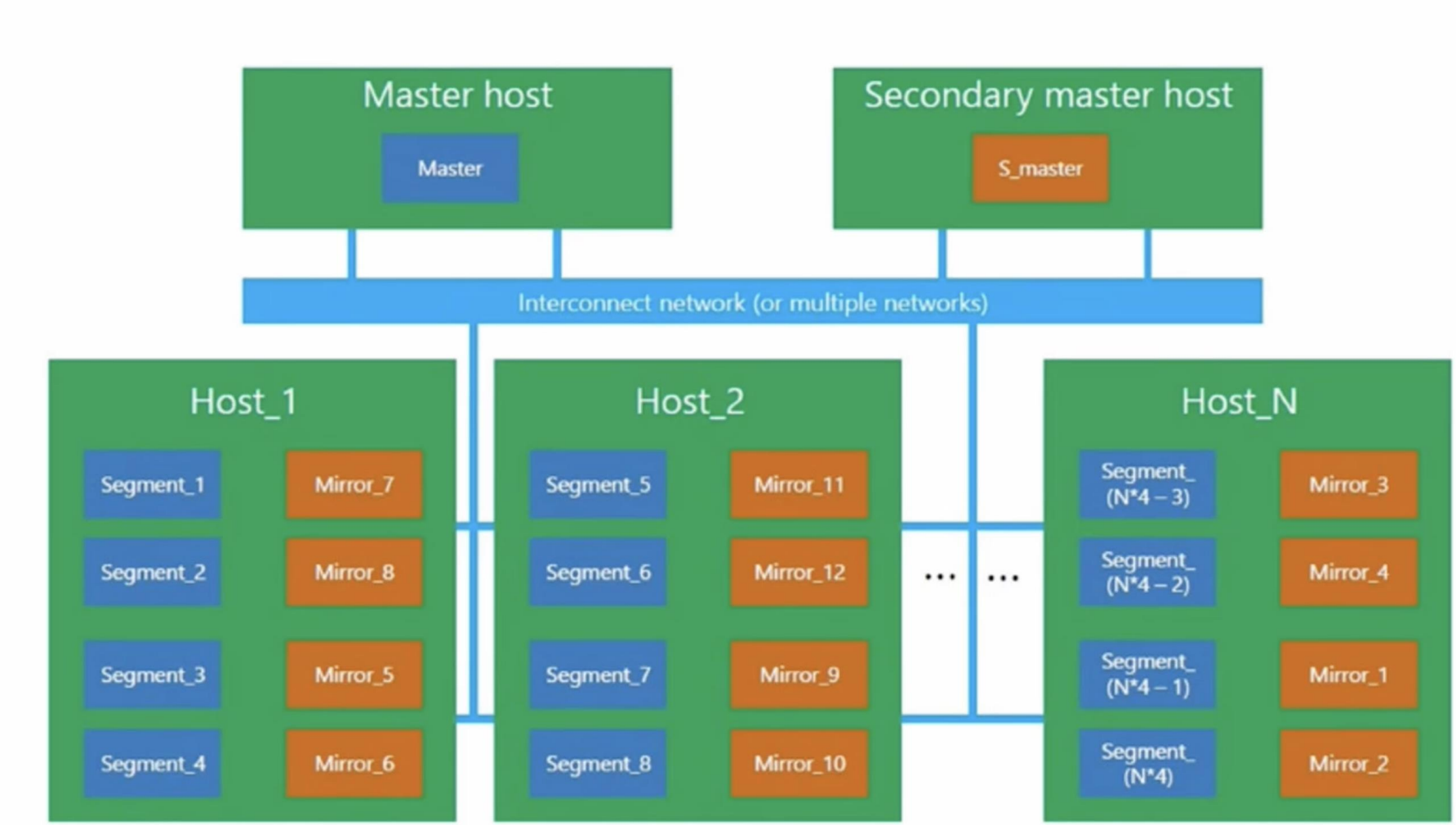
LSML 2025

GreenPlum – MPP on SMP

- Горизонтальная масштабируемость
- Линейный рост производительности
- Отказоустойчивость



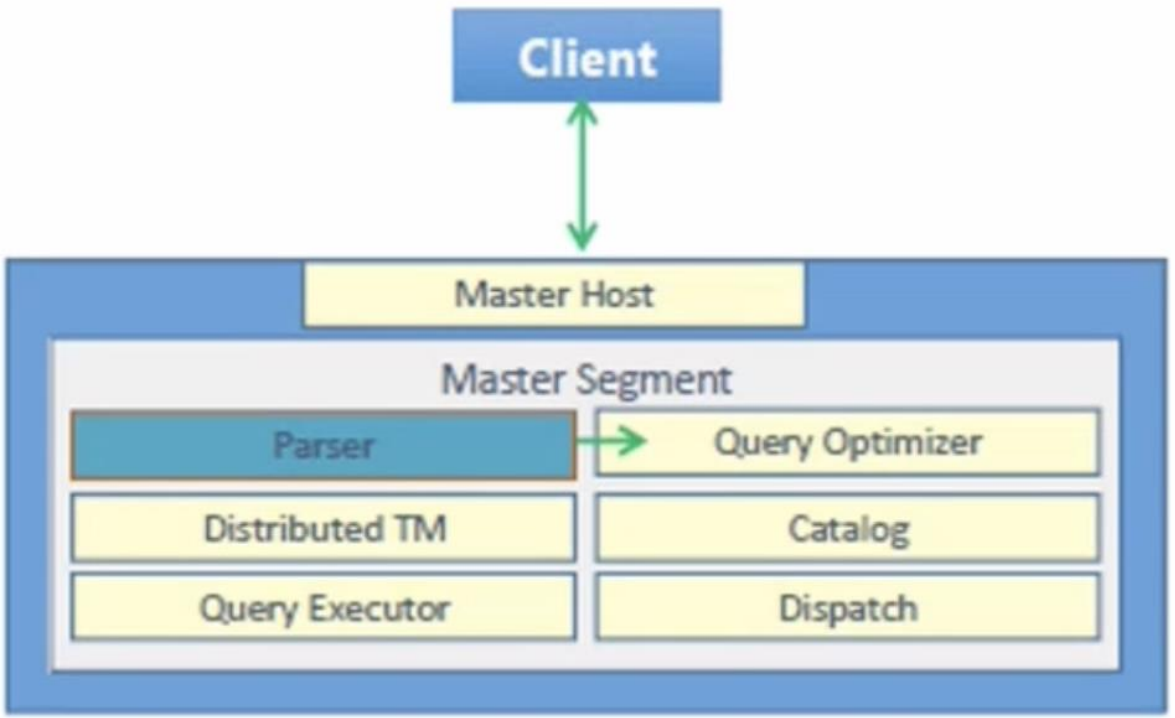
Архитектура GreenPlum



Мастер сегмент

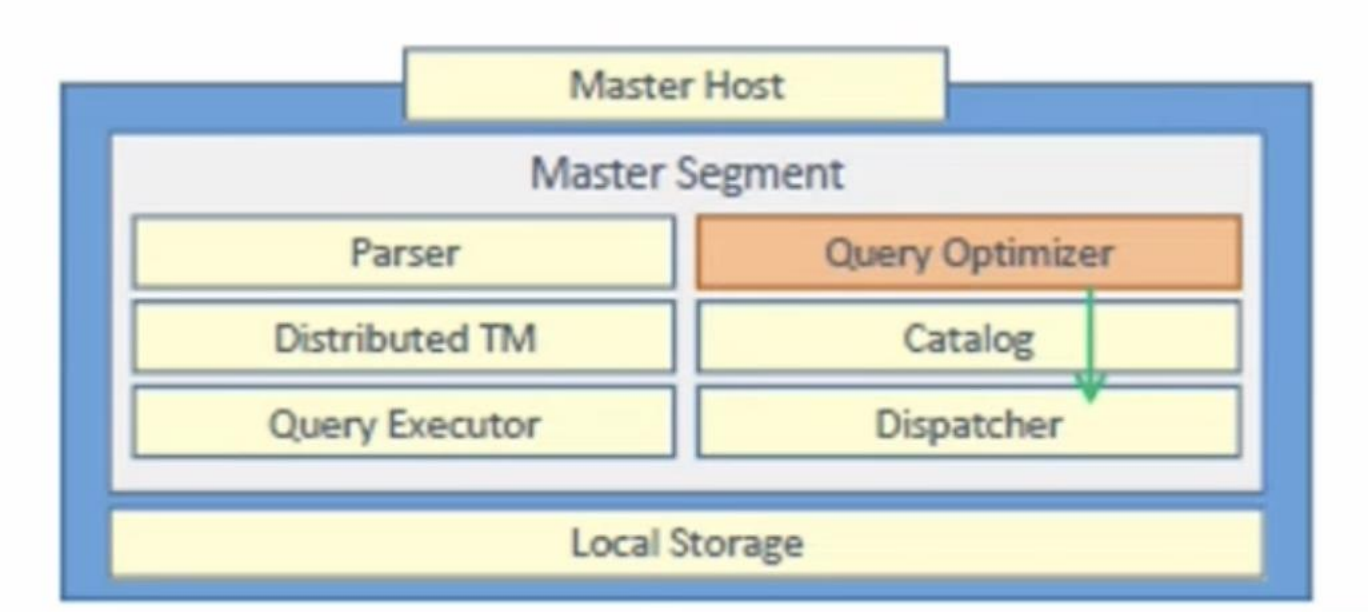
- PostgreSQL listener принимает входящие подключения (дефолтный порт - 5432)
- Query Parser валидирует синтаксис, семантику и генерирует дерево запроса для Query Optimizer
- Мастер не содержит пользовательских данных
- Мастер может выполнять некоторые финальные операции с данными - агрегации, сортировки и т.д.
- Большинство административных действий выполняются только на мастере.

Мастер сегмент



Query Optimizer

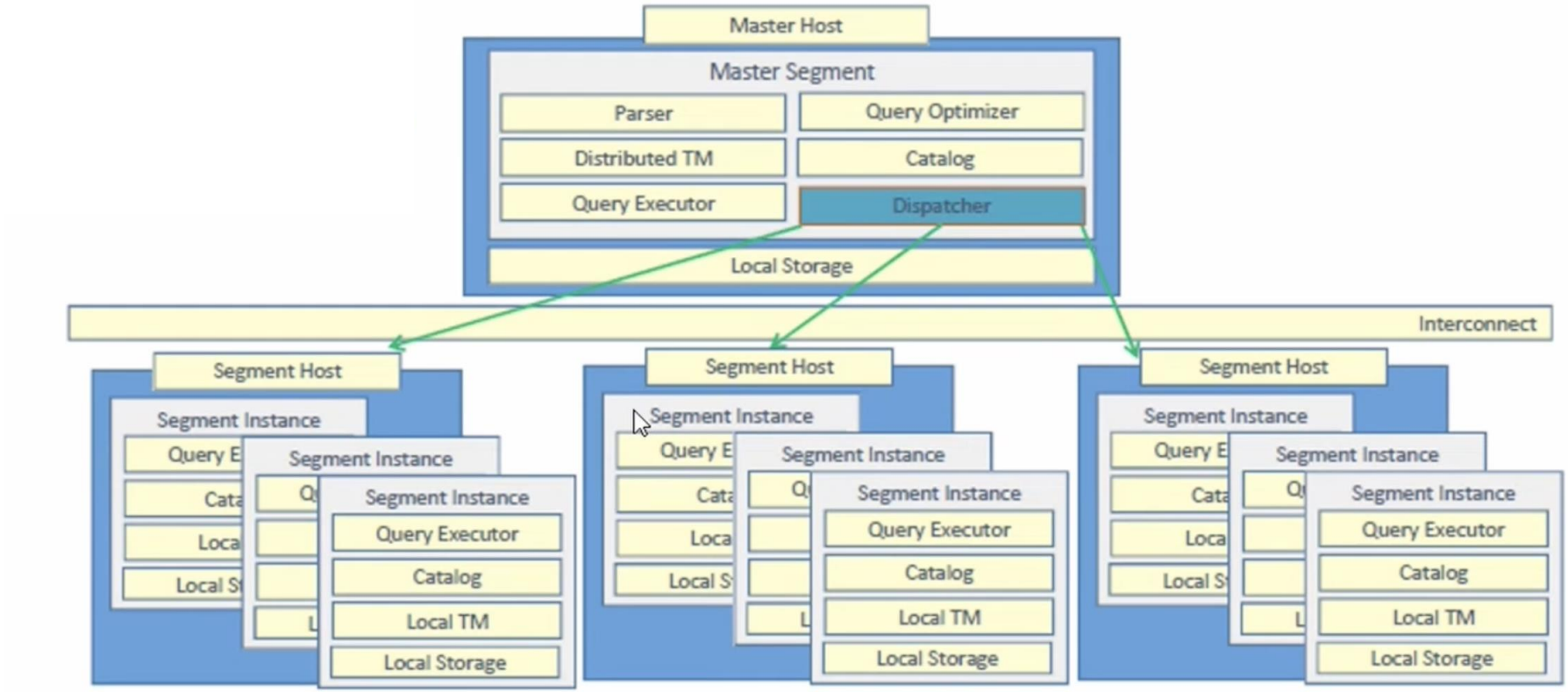
- Генерирует планы запроса
- Каждый план имеет стоимость
- План с меньшей стоимостью передается диспатчеру



Query Dispatcher

- Передает план запроса сегментам хоста
- Аллоцирует ресурсы на сегментах
- Аккумулирует финальный результат для передачи клиенту

Query Dispatcher

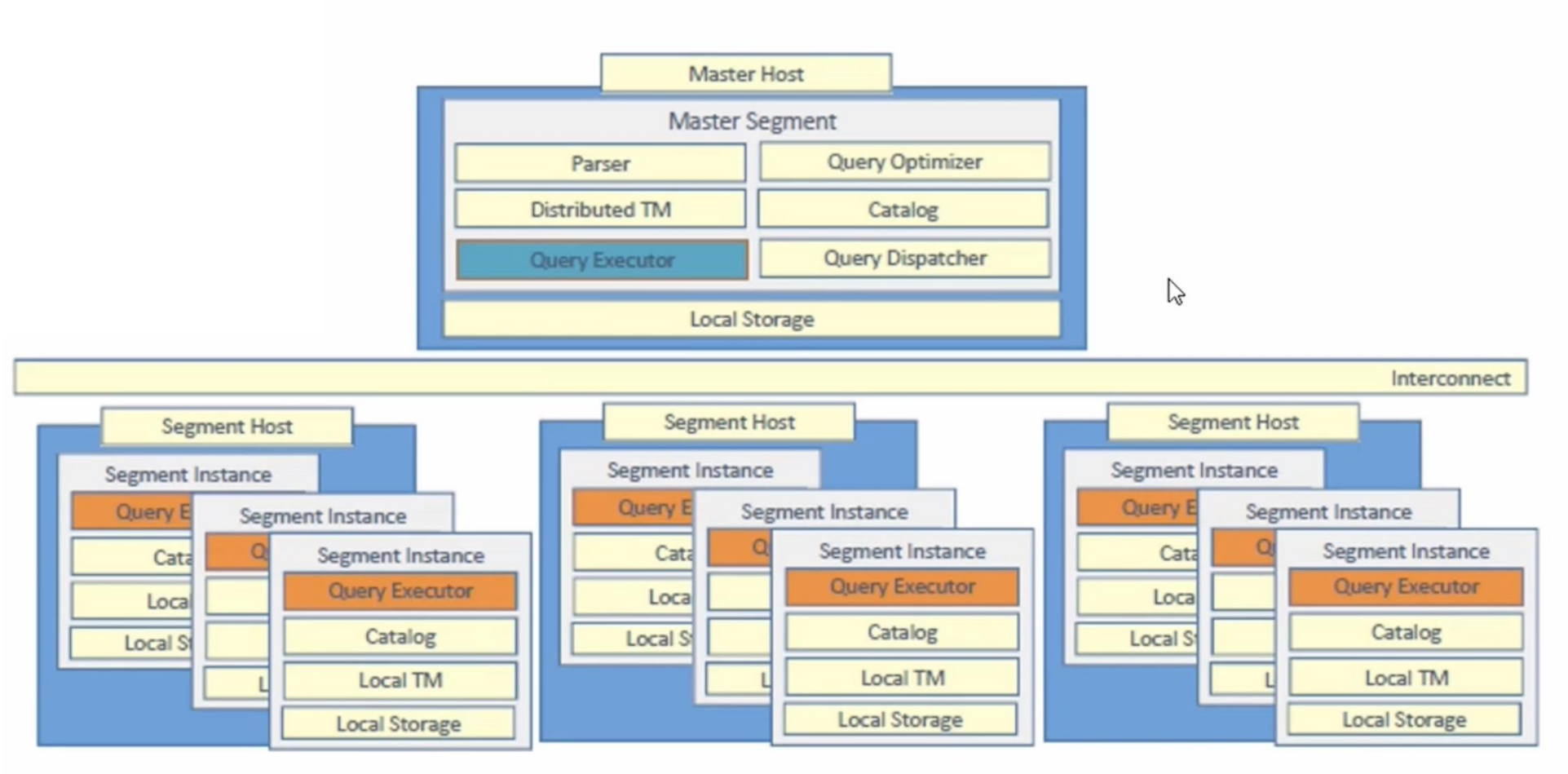


Query Executor

- Выполняет шаги плана
- Обмен промежуточными результатами с другими QE
- Каждый QE запускает для запроса несколько воркеров
- Одинаковые воркеры на сегментах – gangs /

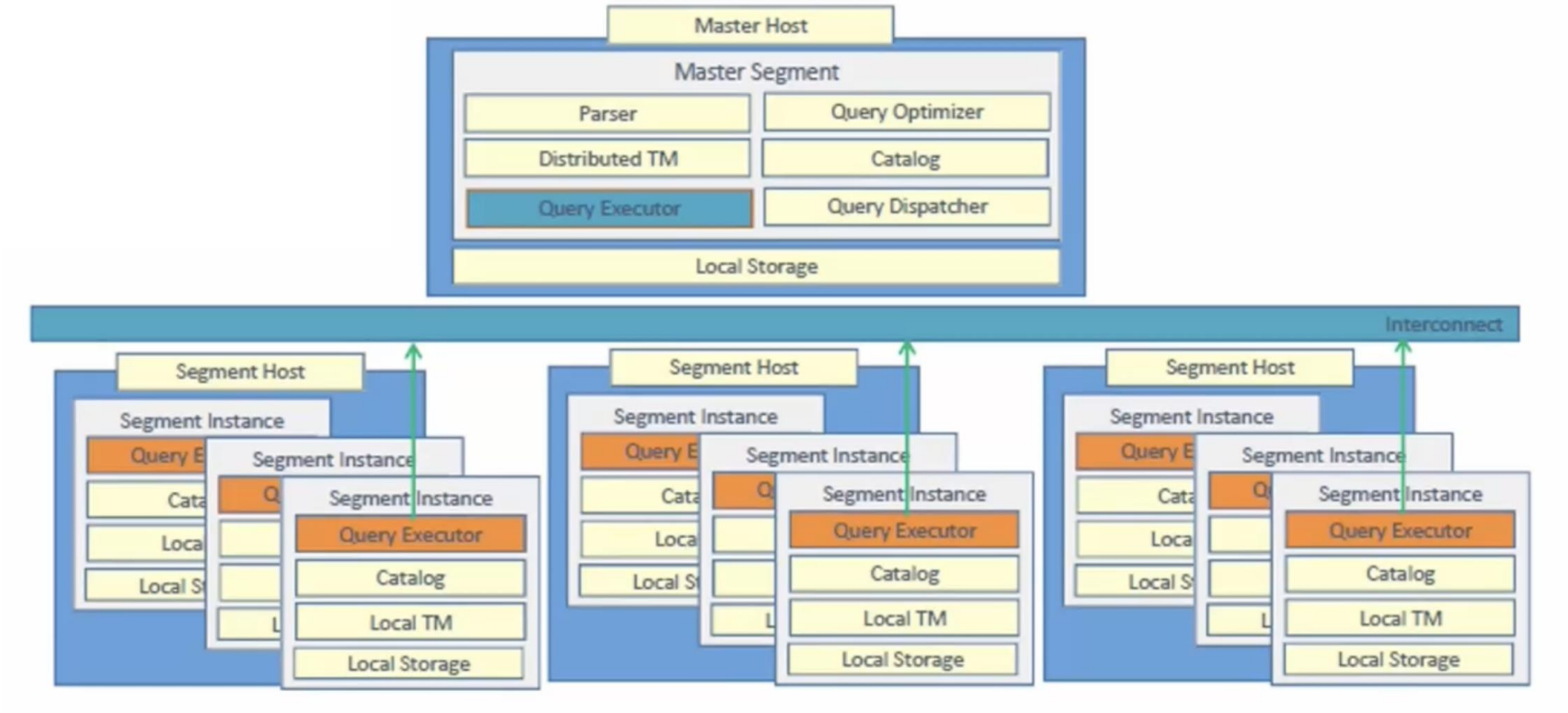
/ выполняют параллельно свою часть запроса

Query Executor



Интерконнекты

- Обмен записями в промежуточных шагах



Системный каталог

- pg_catalog – хранит в себе информацию об объектах базы
- дублируется как на мастере, так и на всех сегментах

Дефолтные схемы

- information_schema - информация об объектах БД согласно стандарту ANSI SQL 2008
- pg_catalog - полная информация о БД и её объектах
- gp_toolkit - информация о размерах таблиц, раздувании таблиц, индексах, логах, спилл-файлах
- public - схема для данных, доступная всем пользователям (мы квотируем эту схему 10 GB)
- diskquota - служебная схема dickquota
- ex_stage - схема для внешних таблиц. В нашей модели администрирования доступна только на чтение etl-пользователями

Кратко про отказоустойчивость

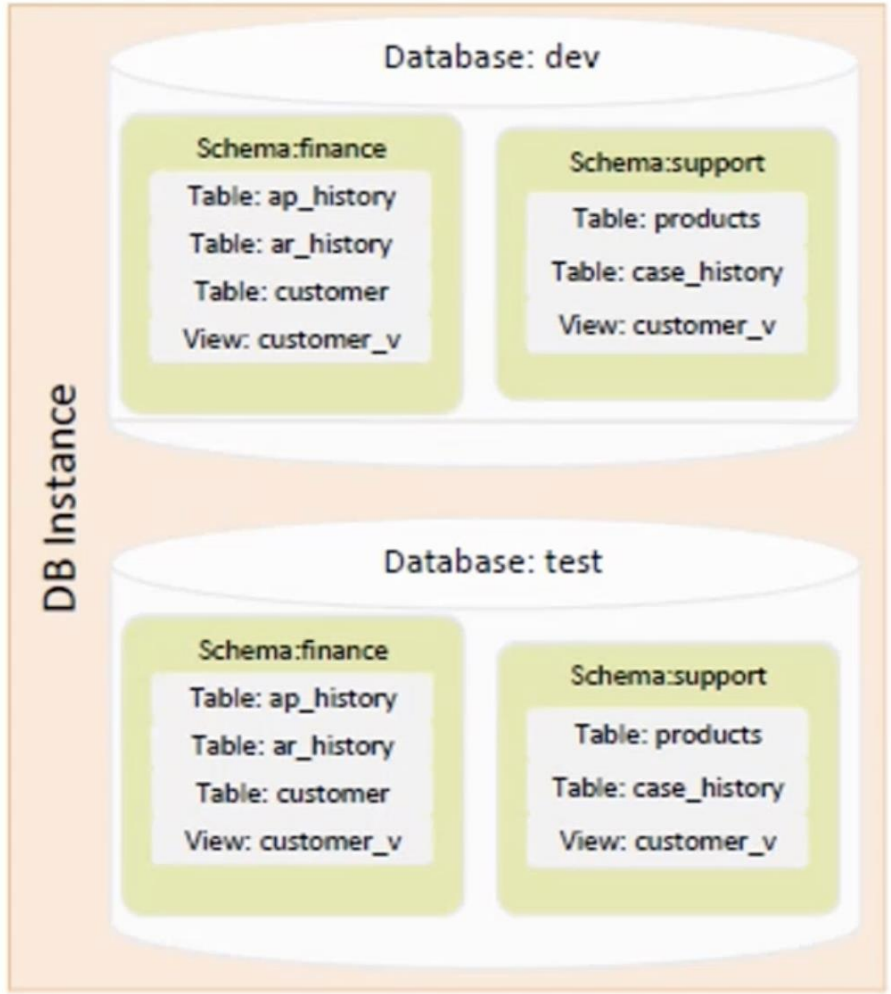
Отказоустойчивость мастера:

- Warm standby (реплика каталога)
- Репликация на основе WAL
- Переключается вручную

Отказоустойчивость сегментов:

- Для каждого основного сегмента создаётся зеркало
- Репликация на основе WAL
- Переключается автоматически
- В случае восстановления основного необходимо выполнить рекавер
- Расположение сегментов и зеркал выбирается на этапе установки СУБД

Иерархия GreenPlum



Пользователи и группы. Ролевая модель

- Роль - сущность в СУБД, которая может владеть объектами и иметь привилегии
- Ролью может быть пользователь или группа. В сущности, ROLE = GROUP
- USER = ROLE с опцией LOGIN
- Роли глобальны для всех БД внутри СУБД
- Роли могут включаться в другие роли, наследуя их привилегии (по-умолчанию)
- public - роль, в которую входят все роли

Пользователи и группы. Ролевая модель

- gpadmin - администратор кластера (команда платформы):
  управляет объектами уровня кластера (не зависят от БД)
  управляет служебными схемами (diskquota, ex_stage etc.)
- projectname_owner - владелец продуктовых схем (роль без авторизации = группа). В неё входят роли пользователей:
  управляет объектами уровня проектных схем;
  выдаёт привилегии на работу с объектами продуктовых схем.
- techuser - технический пользователь. Может быть локальным или LDAP. Подвид: etl-пользователь (имеет права на select из внешних таблиц)
- user - доменный логин пользователя

Пример выдачи роли в коде

alter table [название схемы]. [название таблицы] owner to [какая-то роль];

grant [select – или другие действия] on [название схемы]. [название таблицы] to [какая-то роль на чтение];

Типы join в GP

- Hash Join:
  Строит хеши по меньшей из таблиц и сканирует большую
  Самый быстрый общий способ поджойнить две большие таблицы
- Nested Loop:
  Для каждой строки в большей таблице, просканировать меньшую
  Самый быстрый способ для очень маленьких объёмов
  Используется для кросс-джойнов
- Merge Join:
  Отсортировать обе таблицы и соединить
  Очень быстрый для отсортированных данных
  Очень-очень тяжело добиться использования

Типы join в GP

- Broadcast:

  Каждый сегмент пересылает свои данные каждому;

  У всех сегментов есть полная копия;

- Redistribute:

  Динамическая редистрибуция данных в памяти (или в спилах).

  Опасность SKEW

- Gather:

  Пересылка от сегмента к мастеру

Индексы

- Индекс - объект в СУБД, создаваемый с целью повышения производительности поиска данных;
- Отдельная таблица, содержащая ссылки на строки основной таблицы

Индексы

- Доступны: BTREE, BITMAP, GIST, GIN:
  BTREE - используйте по умолчанию;
  BITMAP - используйте если:
  Колонка(и) содержит(ат) от 100 до 100 000 уникальных значений (низкая кардинальность);
  Чтения много больше, чем UPDATE (не OLTP);
  GIST - очень специфические случаи (https://www.postgresql.org/docs/9.4/indexes-types.html);
  GIN - обратный индекс. Он работает с типами данных, значения которых не являются атомарными, а состоят из элементов.
  Будьте аккуратны с LIKE: '%text' и 'text%" могут вести себя по-разному.

Создание таблиц

```
CREATE [UNIQUE] INDEX name ON table
       [USING btree|bitmap|gist|spgist|gin]
       ( {column_name | (expression)} [COLLATE parameter] [opclass] [ ASC | DESC ]
         [ NULLS { FIRST | LAST } ] [, ...] )
       [ WITH ( storage_parameter = value [, ... ] ) ]
       [TABLESPACE tablespace]
       [WHERE predicate]
```

Выполнение запросов. Статистика

- ANALYZE - команда для сбора статистики
- Без параметров - запускает сбор статистики по всем таблицам в БД
- С указанием имени таблицы - собирает статистику только по заданной таблице
- Сохраняет статистику в системную таблицу pg_statistic
- pg_stats - представление, которое покажет, какая статистика была собрана для таблицы
- pg_stat_operations - представление, которое подскажет, какие операции и когда были над объектом в том числе и ANALYZE

ANALYZE

- Если таблица очень большая и сбор статистики занимает много времени, можно собрать статистику только для определённых колонок;
- ANALYZE не собирает статистику по внешним таблицам;
- Старайтесь выполнить ANALYZE сразу после того, как произошли значительные изменения с данными в таблице;
- Для партиционированных таблиц, статистика не собирается автоматически, если вставка данных происходит через родительскую партицию;
- Для партиционированных таблиц, оптимизатор ORCA использует статистику на уровне root partition. Legacy оптимизатор использует статистику на уровне child partitions.

MVCC

- Моментальный снимки распределенные и синхронизированные на всех сегментах gp
- Не заменяют блокировки, но гарантируют согласованность пользовательских данных
- Экспорт снимков с помощью функции *pg_export_snapshot()*

Транзакции и блокировки

Global Deadlock Detector

- Позволяет выполнять конкурирующие операции такие как UPDATE, DELETE и SELECT...FOR UPDATE для HEAP-таблиц
- Для включения необходимо поменять значение параметра gp_enable_global_deadlock_detector на on
- Если обнаружится, что случилась глобальная блокировка, то одна из транзакций откатится

Транзакции и блокировки

Ручное управление блокировками

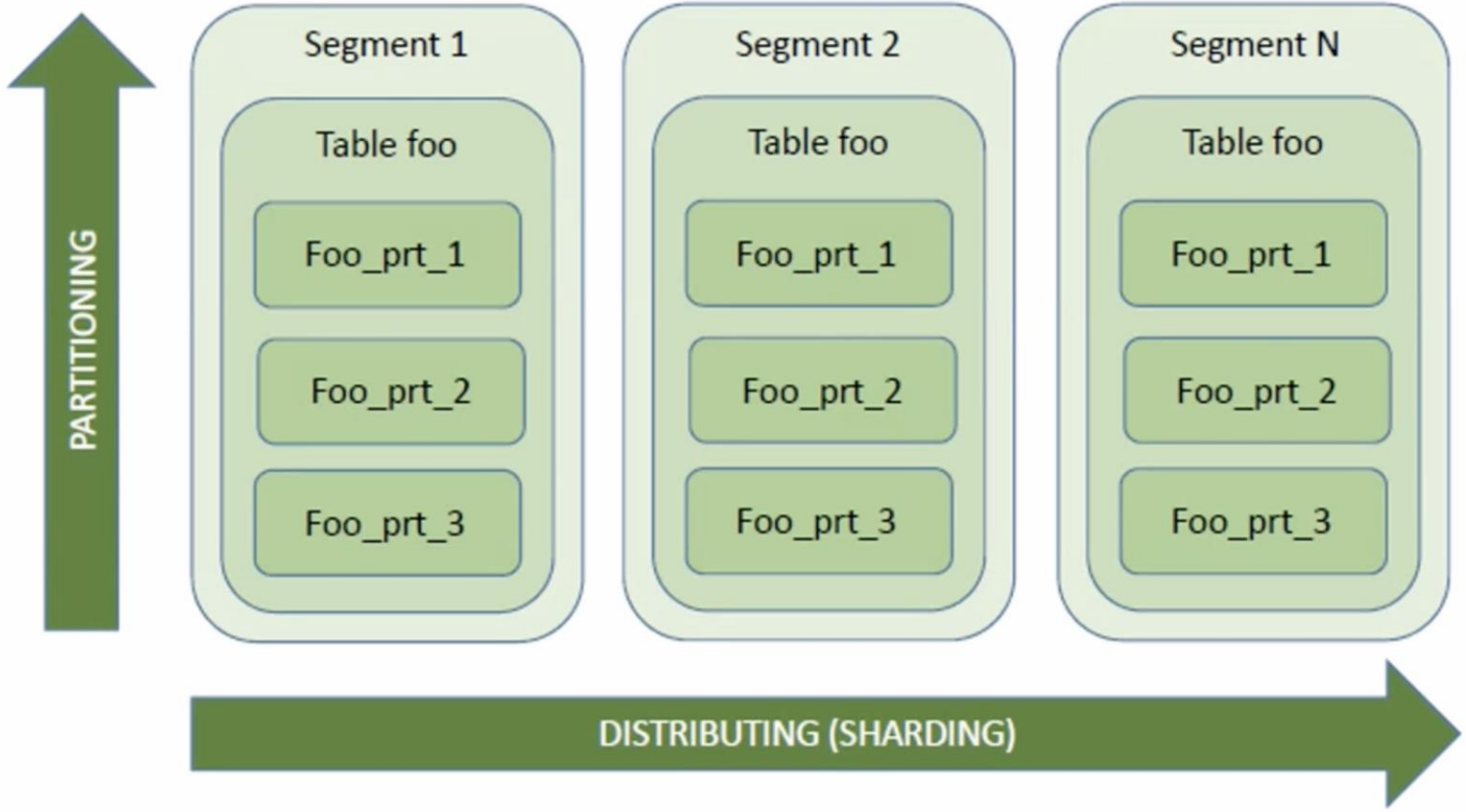
Команда LOCK позволяет получить блокировку в ручном режиме:

LOCK [TABLE] name (, ...] [IN lockmode MODE] [NOWAIT]

where lockmode is one of:

ACCESS SHARE I ROW SHARE | ROW EXCLUSIVE | SHARE UPDATE EXCLUSIVE
I SHARE I SHARE ROW EXCLUSIVE | EXCLUSIVE I ACCESS EXCLUSIVE

Партиционирование таблиц



Сравнение с Hadoop

Характеристика	OLAP	Hadoop	Greenplum
Тип данных	Структурированные, многомерные	Неструктурированные, структурированные	Структурированные (SQL-совместимые)
Обработка	Ориентировано на аналитические запросы	Распределенная обработка больших данных	Аналитика, SQL-запросы, MPP-платформа
Масштабируемость	Ограниченная, зависит от конкретного решения	Высокая масштабируемость (террабайты и петабайты данных)	Высокая масштабируемость (MPP)
Производительность	Высокая для аналитических запросов	Не всегда высокая для аналитики	Высокая для аналитики и SQL-запросов
Применение	Бизнес-аналитика, отчетность	Большие данные, машинное обучение, хранение данных	Бизнес-аналитика, обработка больших объемов структурированных данных
Технологии	Многомерные кубы данных	Hadoop Distributed File System (HDFS), MapReduce	PostgreSQL, распределенная база данных

GreenPlum для аналитики

- Загрузка данных в Greenplum
- SQL
- Создание моделей и таблиц для аналитики
- Использование аналитических функций
- Мониторинг и масштабирование
- Интеграция с BI-инструментами

GreenPlum для машинного обучения

- **MADlib** — это расширение для Greenplum, которое предоставляет набор инструментов для машинного обучения и статистического анализа

- Распараллеливание задач обучения и анализа на кластере Greenplum для обработки больших данных

- Интегрирация Greenplum с Python и R для более сложных алгоритмов и применения внешних библиотек

Сравнение GP и ClickHouse

Критерий	ClickHouse	Greenplum
Описание	Колоночная реляционная СУБД от Яндекса	Платформа аналитической реляционной базы данных на основе PostgreSQL от компании VMWare Tanzu (ранее Pivotal)
Первый выпуск	2016 год	2005 год
Первичная модель управления данными	Реляционная	Реляционная
Вторичные поддерживаемые модели управления данными	СУБД временных рядов, о чем мы подробнее рассказываем здесь	Документо-ориентированная СУБД Пространственная СУБД
Лицензия	Открытый исходный код	Открытый исходный код
Поддерживаемые ОС на сервере	FreeBSD, Linux, MacOS	Linux
Поддерживаемые языки программирования	C#, C++, Elixir, Go, Java, JavaScript (Node.js), Kotlin, Nim, Perl, PHP, Python, R, Ruby, Scala	C, Java, Perl, Python, R
Поддержка хранимых процедур и триггеров	нет	да

Поддержка MapReduce	нет	да
Обеспечение согласованности	Мгновенная согласованность	Мгновенная согласованность
Внешние ключи	нет	да
Соответствие ACID	нет	да
Параллелизм	да	да
Хранение данных в памяти	да	нет
Управление доступом	права доступа для пользователей и ролей	детальные права доступа в соответствии со стандартом SQL
Поддержка схемы данных	да	да
Предопределенные типы данных	да	да
Поддержка XML	нет	да

Первичные индексы	да	да
Вторичные индексы	да, при использовании движка MergeTree	да
Поддержка ANSI SQL	нет	да
API и другие методы доступа	HTTP, gRPC, собственный протокол JDBC, ODBC	JDBC, ODBC
Соединения таблиц	Оператор JOIN поддерживается, но не очень хорошо: правая таблица должна помещаться в память одного сервера. Невозможно соединить 2 таблицы, если они больше памяти одного сервера	Отлично обрабатывает локальные и распределенные запросы с JOIN-операторами
Скорость обработки	Работает очень быстро, время отклика менее 200 миллисекунд	Скорость зависит от объема данных, время отклика до пары десятков секунд
Типовые сценарии использования	Аналитика данных в реальном времени Мониторинг технических метрик Обработка данных с IoT-устройств и промышленных датчиков	Предиктивная аналитика Прогнозирование событий Скоринг событий Предоставление данных для BI-дэшбордов