

# Apache Geode

## 1. История развития СУБД

Apache Geode изначально появился, как версия продукта с открытым исходным кодом под названием GemFire, первоначально разработанного GemStone Systems. GemFire нашел свое первое промышленное применение несколько десятилетий назад в качестве механизма транзакционных данных с малой задержкой для торговли на Уолл-Стрит. В мае 2010 года GemStone была приобретена SpringSource. Вскоре после этого он стал ключевой частью Pivotal. В 2015 году, Pivotal предложила Apache Software Foundation (ASF) создать инкубационное сообщество Apache Geode, чтобы способствовать его развитию, и оно это сделало. Поддержка сообщества вокруг проекта резко возросла, и к концу 2016 года он стал проектом Apache высшего уровня. Сообщество Geode теперь публикует новую версию ветки GitHub ежеквартально, в частности, в первый понедельник 1 февраля, 1 мая, 1 августа и 1 ноября. Сегодня Spring Boot для Apache Geode и Pivotal GemFire предоставляет любому клиенту удобство подхода Spring Boot с использованием автоматической настройки, с мощными абстракциями Spring Framework и высокосогласованной моделью программирования.

## 2. Инструменты для взаимодействия с СУБД

- a. gfsH - предоставляет единый мощный интерфейс командной строки, из которого вы можете запускать, управлять и отслеживать процессы, данные и приложения Geode.
- b. Gemcached - позволяет клиентам Memcached взаимодействовать с кластером серверов Geode, как если бы эти серверы были серверами memcached. Memcached — это решение для кэширования с открытым исходным кодом, которое использует распределенную хеш-карту в памяти для хранения пар ключ-значение строковых или объектных данных.
- c. Модули управления HTTP-сессиями Apache Geode обеспечивают быструю, масштабируемую и надежную репликацию сессий для HTTP-серверов без необходимости внесения изменений в приложения.

- d. Geode Pulse - веб-приложение, которое предоставляет графическую панель для мониторинга жизненно важных показателей работоспособности и производительности кластеров Geode, участников и регионов в режиме реального времени.
- e. Интеграция с Apache Lucene® позволяет пользователям создавать индексы Lucene и выполнять поиск Lucene по данным, хранящимся в Geode.
- f. Micrometer - это простой инструмент для мониторинга приложений (например, SLF4J — это простой инструмент для ведения журнала), который позволяет публиковать метрики из Geode в различные инструменты мониторинга производительности приложений (APM).

### **3. Какой database engine используется в вашей СУБД?**

Кэши — это абстракция, описывающая узел в кластере Geode. Архитекторы приложений могут организовать эти узлы в одноранговой или клиент-серверной топологии.

В каждом кэше определяются регионы данных. Регионы данных аналогичны таблицам в реляционной базе данных и управляют данными распределенным образом в виде пар имя/значение. Реплицированная область хранит идентичные копии данных на каждом члене кэша кластера. Разделенная область распределяет данные между членами кэша. После настройки системы клиентские приложения могут получать доступ к распределенным данным в регионах, не зная об архитектуре базовой системы.

### **4. Как устроен язык запросов в вашей СУБД? Разверните БД с данными и выполните ряд запросов.**

Geode предоставляет SQL-like язык запросов, который позволяет вам получать доступ к данным, хранящимся в регионах Geode. Поскольку регионы Geode представляют собой key-value хранилища, в которых значения могут варьироваться от простых массивов байтов до сложных вложенных объектов, Geode использует синтаксис запроса, основанный на OQL (Object Query Language), для запроса данных региона. OQL и SQL имеют много синтаксического сходства, однако у них есть существенные различия. Например, хотя OQL не предлагает всех возможностей

SQL, таких как агрегаты, OQL позволяет вам выполнять запросы к сложным графам объектов, запрашивать атрибуты объектов и вызывать методы объектов.

```
gfsh>start locator --name=locator --bind-address=localhost
Starting a Geode Locator in /home/ilyas/Apache_Geode/apache-geode-1.14.4/locator
...
.....
Locator in /home/ilyas/Apache_Geode/apache-geode-1.14.4/locator on 127.0.0.1[10334] as locator is currently online.
Process ID: 19716
Uptime: 11 seconds
Geode Version: 1.14.4
Java Version: 1.8.0_312
Log File: /home/ilyas/Apache_Geode/apache-geode-1.14.4/locator/locator.log
JVM Arguments: -Dgemfire.enable-cluster-configuration=true -Dgemfire.load-cluster-configuration-from-dir=false -Dgemfire.launcher.registerSignalHandlers=true -Djava.awt.headless=true -Dsun.rmi.dgc.server.gcInterval=9223372036854775806
Class-Path: /home/ilyas/Apache_Geode/apache-geode-1.14.4/lib/geode-core-1.14.4.jar:/home/ilyas/Apache_Geode/apache-geode-1.14.4/lib/geode-dependencies.jar

Successfully connected to: JMX Manager [host=10.0.2.15, port=1099]

Cluster configuration service is up and running.
```

```
gfsh>start server --name=server1 --server-port=0
Starting a Geode Server in /home/ilyas/Apache_Geode/apache-geode-1.14.4/server1.
..
...
Server in /home/ilyas/Apache_Geode/apache-geode-1.14.4/server1 on 10.0.2.15[46799] as server1 is currently online.
Process ID: 20377
Uptime: 2 seconds
Geode Version: 1.14.4
Java Version: 1.8.0_312
Log File: /home/ilyas/Apache_Geode/apache-geode-1.14.4/server1/server1.log
JVM Arguments: -Dgemfire.start-dev-rest-api=false -Dgemfire.use-cluster-configuration=true -Dgemfire.launcher.registerSignalHandlers=true -Djava.awt.headless=true -Dsun.rmi.dgc.server.gcInterval=9223372036854775806
Class-Path: /home/ilyas/Apache_Geode/apache-geode-1.14.4/lib/geode-core-1.14.4.jar:/home/ilyas/Apache_Geode/apache-geode-1.14.4/lib/geode-dependencies.jar
```

```
gfsh>put --region=test --key="0" --value="Vasya"
Result      : true
Key Class   : java.lang.String
Key         : 0
Value Class : java.lang.String
Old Value   : null
```

```
gfsh>put --region=test --key="1" --value="Petya"
Result      : true
Key Class   : java.lang.String
Key         : 1
Value Class : java.lang.String
Old Value   : null
```

```
gfsh>put --region=test --key="2" --value="Sasha"
Result      : true
Key Class   : java.lang.String
Key         : 2
Value Class : java.lang.String
Old Value   : null
```

```
gfsh>query --query='select e.key, e.value from /test.entries e'
Result : true
Limit  : 100
Rows   : 3
```

key	value
0	Vasya
2	Sasha
1	Petya

## 5. Распределение файлов БД по разным носителям?

Масштабируемость достигается за счет динамического разделения данных между многими элементами и равномерного распределения нагрузки данных по серверам. Для «горячих» данных можно настроить систему на динамическое расширение для создания большего количества копий данных. Также можно настроить поведение приложения для распределенного выполнения в непосредственной близости от необходимых ему данных.

Если необходимо поддерживать высокие и непредсказуемые всплески параллельной клиентской нагрузки, можно увеличить количество серверов, управляющих данными, и распределить данные и поведение между ними, чтобы обеспечить единообразное и предсказуемое время отклика. Клиенты постоянно балансируют нагрузку на сервера на основе постоянной обратной связи от серверов об их загруженности. Благодаря тому, что данные разделены и реплицированы между серверами, клиенты могут

динамически перемещаться на разные серверы для равномерной загрузки серверов и обеспечения наилучшего времени отклика.

Также можно улучшить масштабируемость, внедрив асинхронную «отложенную запись» изменений данных во внешние хранилища данных, таких как база данных. Это позволяет избежать узких мест, ставя в очередь все обновления по порядку. Также можно объединять обновления и распространять их в базе данных в пакетном режиме.

## 6. На каком языке/ах программирования написана СУБД?

Apache Geode написана на Java.

## 7. Какие типы индексов поддерживаются в БД? Приведите пример создания индексов.

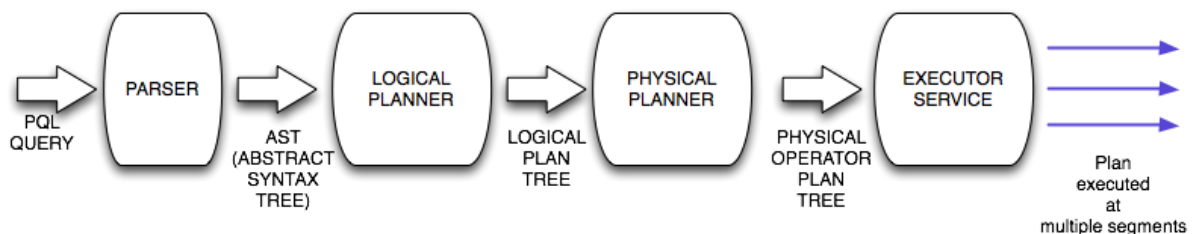
В Apache Geode используются Hash indexes.

```
gfsfsh>create index --name=indexTest --type=key --expression=id --region=/test
```

Member	Status	Message
10.0.2.15(server2:24147)<v1>:41000	OK	Index successfully created

Cluster configuration for group 'cluster' is updated.

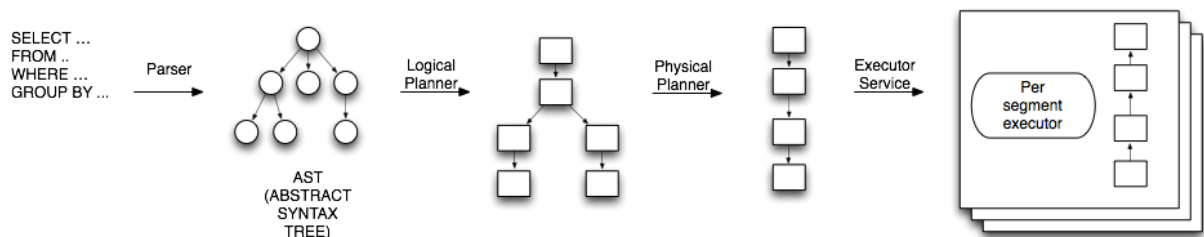
## 8. Как строится процесс выполнения запросов в вашей СУБД?



- Синтаксический анализ запросов: Pinot поддерживает слегка измененную версию SQL, которую мы называем PQL. Мы используем Antlr для анализа запроса в дереве разбора. На этом этапе выполняются все проверки синтаксиса и устанавливаются значения по умолчанию для отсутствующих элементов.
- Фаза логического плана: эта фаза принимает дерево разбора запроса и выводит дерево логического плана. Этот этап является однопоточным и простым и создает соответствующее дерево операторов логического плана на

основе типа запроса (выборка, агрегация, группировка и т. д.) и метаданных, предоставленных источником данных.

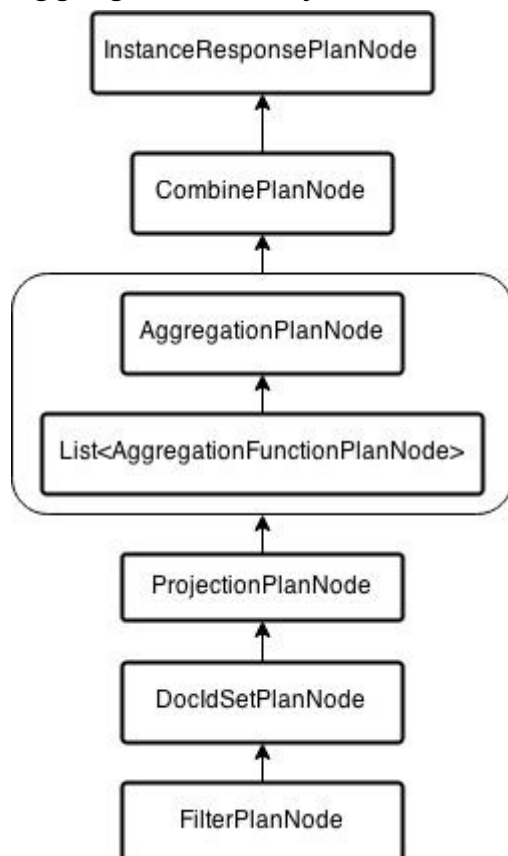
- Этап физического плана: на этом этапе план дополнительно оптимизируется на основе отдельного сегмента. Оптимизация, применяемая на этом этапе, может быть разной для разных сегментов.
- Служба исполнителя: когда у нас есть физическое дерево операторов для каждого сегмента, служба исполнителя берет на себя ответственность за планирование задач обработки запросов для каждого сегмента.



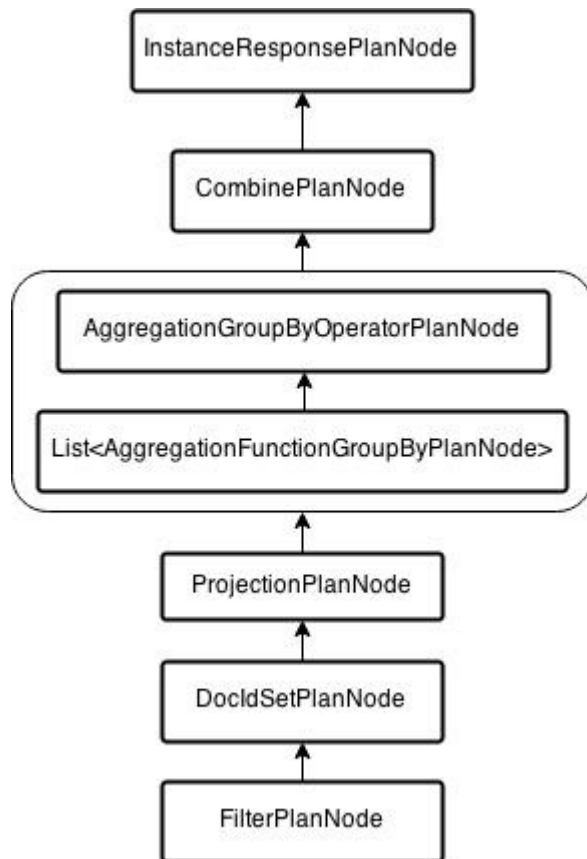
## 9. Есть ли для вашей СУБД понятие «план запросов»? Если да, объясните, как работает данный этап.

Основные планы запросов:

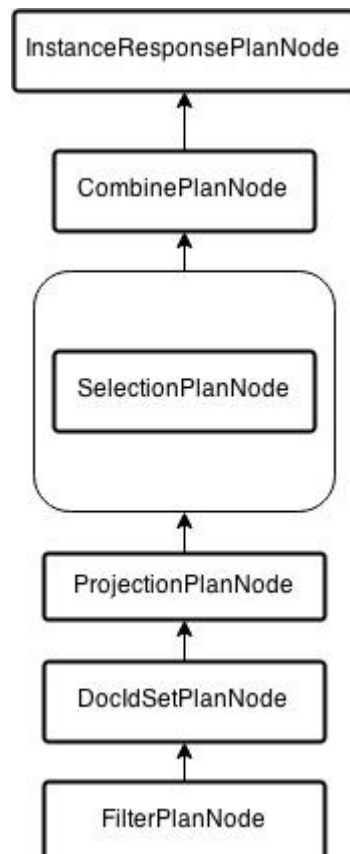
Aggregation Query



## Aggregation Group By Query



## Selection Query



**10. Поддерживаются ли транзакции в вашей СУБД? Если да, то расскажите о нем. Если нет, то существует ли альтернатива?**

Geode предлагает API для клиентских приложений, которые выполняют транзакционные операции. Geode реализует транзакции со знакомыми методами начала, фиксации и отката, которые реализуют те же операции, что и в методах транзакций реляционной базы данных.

**11. Какие методы восстановления поддерживаются в вашей СУБД. Расскажите о них.**

Если участник системы непреднамеренно выходит из кластера из-за отказа этого участника, хоста или сети, другие участники автоматически адаптируются к потере и продолжают работать. Кластер не испытывает никаких нарушений, таких как тайм-ауты. При планировании стратегии восстановления данных учитывайте следующие факторы:

- Настроен ли регион для избыточности данных — только для секционированных регионов.
- Конфигурация политики потери ролей региона, которая определяет поведение региона после сбоя или сбоя системы — только для распределенных регионов.
- Настроен ли регион для сохранения на диск.
- Настроен ли регион для вытеснения на основе LRU.
- Степень сбоя, независимо от того, задействованы ли несколько участников или сбой сети.
- Конкретные потребности вашего приложения, такие как сложность замены данных и риск работы с несогласованными данными для вашего приложения.
- Когда генерируется предупреждение из-за сетевого раздела или медленного ответа, указывающее, что определенные процессы могут или будут давать сбой.

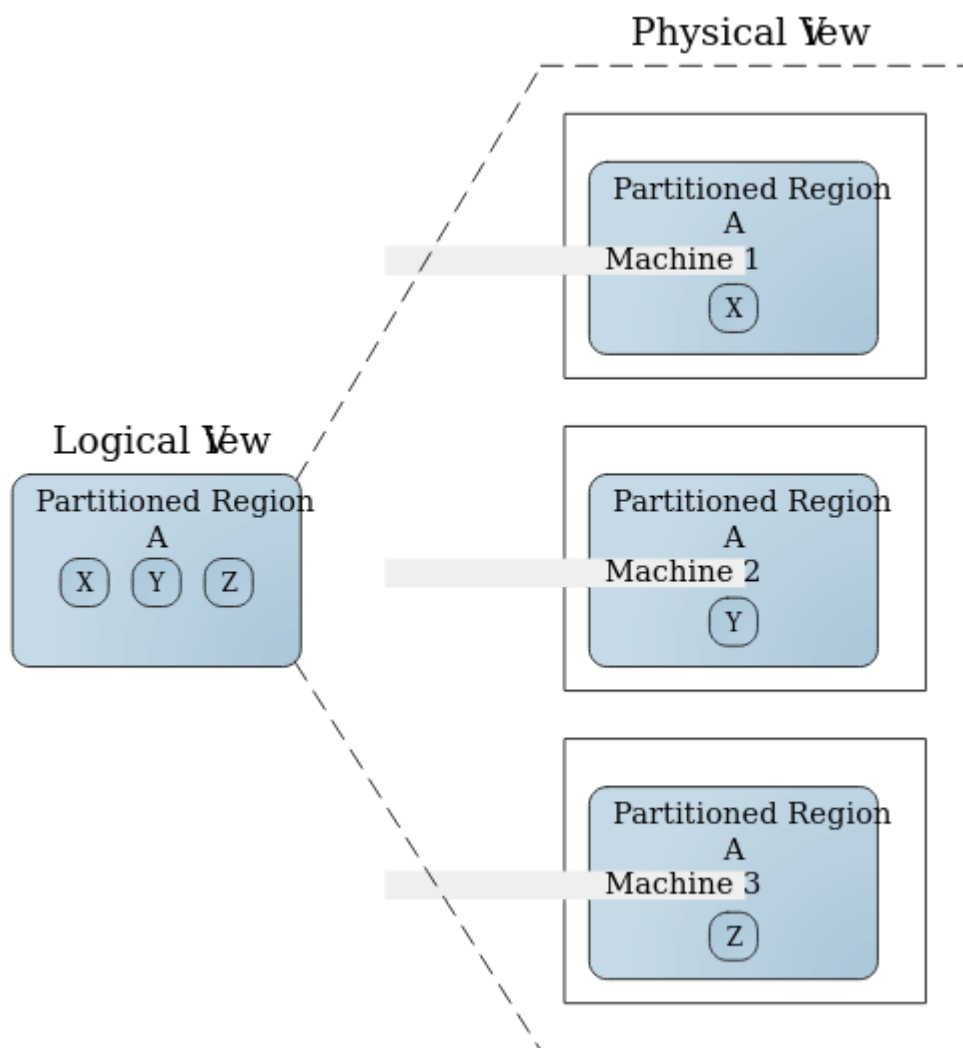
Более подробная информация о возможных ситуациях и как с ними бороться: [geode.apache.org](http://geode.apache.org)

**12. Расскажите про шардинг в вашей конкретной СУБД. Какие типы используются? Принцип работы.**



В Apache Geode шардирование включено в понятие партиционирования.

Geode автоматически определяет физическое расположение данных в элементах, на которых размещены данные партицированного региона. Geode разбивает данные партицированного региона на единицы хранения, известные как сегменты, и сохраняет каждый сегмент в члене хоста региона. Сегменты распределяются в соответствии с настройками атрибутов региона участника.



**13. Возможно ли применить термины Data Mining, Data Warehousing и OLAP в вашей СУБД?**

Нет.

**14. Какие методы защиты поддерживаются вашей СУБД?**  
Шифрование трафика, модели авторизации и т.п.

Шифрование, безопасная связь SSL, функции аутентификации и авторизации помогают защитить кластер.

Функции безопасности включают в себя:

- Единый интерфейс безопасности для всех компонентов. Единый механизм аутентификации и авторизации упрощает реализацию безопасности. Он просматривает и взаимодействует со всеми компонентами согласованным образом.
- Общесистемный контроль доступа на основе ролей.
- Позволяет настроить соединения на основе SSL, а не на простые соединения через сокеты. Вы можете включить SSL отдельно для одноранговых, клиентских, JMX, шлюзовых отправителей и получателей и соединений HTTP.
- Постобработка данных региона. Возвращаемые значения для операций, которые возвращают значения области, могут быть отформатированы.

**15. Какие сообщества развивают данную СУБД? Кто в проекте имеет права на коммит и создание дистрибутива версий? Расскажите об этих людях и/или компаниях.**

Основным разработчиком является Apache Geode Team. Так же любой желающий может присоединиться к открытому комьюнити и вносить изменения в репозиторий на github

**16. Создайте свои собственные данные для демонстрации работы СУБД.**

```
gfsh>put --region=test --key="0" --value="Vasya"
Result      : true
Key Class   : java.lang.String
Key         : 0
Value Class : java.lang.String
Old Value   : null

gfsh>put --region=test --key="1" --value="Petya"
Result      : true
Key Class   : java.lang.String
Key         : 1
Value Class : java.lang.String
Old Value   : null

gfsh>put --region=test --key="2" --value="Sasha"
Result      : true
Key Class   : java.lang.String
Key         : 2
Value Class : java.lang.String
Old Value   : null
```

- 17. Как продолжить самостоятельное изучение языка запросов с помощью демобазы. Если демобазы нет, то создайте ее.**

Здесь представлены примеры работы с Apache Geode  
<https://github.com/apache/geode-examples>

- 18. Где найти документацию и пройти обучение**  
<https://geode.apache.org/>

- 19. Как быть в курсе происходящего**  
[Github](#)