|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра прикладной математики (ПМ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Методы анализа данных»

**Практическое занятие № 5**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИМБО-02-22, Ким Кирилл Сергеевич* | (подпись) | |
| Преподаватель | *Морошкин Никита Андреевич, преподаватель* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2024 г.

Практическая работа №6

Цель работы

В этой практической работе выполняются базовые операции с созданием, удалением и изменением таблицы в оболочке HBase. Мы будем использовать оболочку для размещения и получения данных в HBase.

Выполнения практической работы

Начнем с оболочки HBase

1. Запустите оболочку HBase. Выполните команду справки и просмотрите основную информацию об использовании оболочки HBase.

sudo /usr/local/hbase/hbase-2.3.5/bin/start-hbase.sh

hbase shell

hbase(main):001:0> help

Примечание: Команды HBase пишутся после символа-приглашения ">".

2. Отображение версии и состояния для базового использования

hbase(main):001:0> version

hbase(main):001:0> status

Вы можете увидеть версию Hbase (2.3.5) и автономное выполнение (1 активный мастер, 1 сервер).

3. Используйте команду create для создания новой таблицы. Необходимо указать имя таблицы и имя семейства столбцов.

3.1.1 Имя таблицы: tbl\_authors, Семейство столбцов (Column Family): cf1

hbase(main):001:0> create ‘tbl\_authors’, ‘cf1’

Примечание: Имена таблиц, строки, столбцы должны быть заключены в кавычки.

4. Посмотреть список созданных таблиц можно с помощью команды list:

hbase(main):002:0> list

5. Используйте команду describe для просмотра подробных сведений, включая настройки по умолчанию.

hbase(main):001:0> describe ‘tbl\_authors’

6. Удалите только что созданную таблицу. Выведите все таблицы, чтобы убедиться, что таблица была успешно удалена

hbase(main):001:0> drop ‘tbl\_authors’

Примечание: Получислоь? Дело в том, что таблица tbl\_authors включена. Сначала отключите её, чтобы затем удалить её.

7. Отключение таблицы tbl\_authors для её удаления.

hbase(main):003:0> disable ‘tbl\_authors’

hbase(main):004:0> drop ‘tbl\_authors’

hbase(main):005:0> list

8. Создайте тестовую таблицу и используйте команду put. Здесь мы вставляем три значения, по одному за раз.

8.1.1 Входные данные представлены в следующем формате.

Column family

Rowkey cf1:a cf1:b cf1:c

rk1 A

rk2 B

rk3 C

8.1.2 Имя таблицы: temp, Семейство столбцов (Column Family): cf1

hbase(main):003:0> create 'temp', 'cf1'

hbase(main):004:0> put 'temp', 'rk1', 'cf1:a', 'A'

hbase(main):005:0> put 'temp', 'rk2', 'cf1:b', 'B'

hbase(main):006:0> put 'temp', 'rk3', 'cf1:c', 'C'

9. Подсчет количества строк во временной таблице HBase, созданной в предыдущем шаге

hbase(main):003:0> count ‘temp’

10. Добавим еще значений. Каким будет общее количество строк в таблице temp после выполнения следующих команд?

hbase(main):003:0> put 'temp', 'rk4', 'cf1:d', 'D'

hbase(main):003:0> put 'temp', 'rk1', 'cf1:b', '1B'

hbase(main):003:0> put 'temp', 'rk5', 'cf1:d', 'E'

hbase(main):003:0> scan ‘temp’

Ниже приведен результат сканирования данных в таблице TEMP. Общий список — 6, но фактическое количество строк — 5.

Примечание: В cf1 есть две строчки с ROWKEY rk=1.

11. Измените атрибут version cf1 в таблице temp на 3 и проверьте, правильно ли отражено изменение.

hbase(main):003:0> alter ‘temp’, {NAME=>'cf1', VERSIONS=>3}

hbase(main):004:0> desc ‘temp’

12. Измените временную таблицу, добавив семейство столбцов cf2 и cf3 с помощью команды ALTER.

hbase(main):003:0> alter 'temp', 'cf2', 'cf3'

13. Выполните команду desc, чтобы проверить, добавлены ли cf2 и cf3.

14. Удалите только что добавленный cf3 и проверьте результат.

hbase(main):003:0> alter 'temp', {'delete' => 'cf3'}

hbase(main):004:0> desc ‘temp’

15. Удалите временную таблицу. Выведите все таблицы, чтобы убедиться, что таблица была успешно удалена.

hbase(main):005:0> disable ‘temp’

hbase(main):006:0> drop ‘temp’

hbase(main):007:0> list

Выйдите из оболочки HBase с помощью команды quit

hbase(main):004:0> quit

16. Создайте файл с командами HBase (при копировании могут возникнуть проблемы с кавычками).

$vi rubyscript.rb

create 'temp', 'cf1'

put 'temp', 'rk1', 'cf1:a', 'A'

put 'temp', 'rk2', 'cf1:b', 'B'

put 'temp', 'rk3', 'cf1:c', 'C'

put 'temp', 'rk4', 'cf1:d', 'D'

alter 'temp', {NAME=>'cf1', VERSIONS=>3}

alter 'temp', 'cf2', 'cf3'

put 'temp', 'rk1', 'cf1:b', '1B'

put 'temp', 'rk5', 'cf1:d', 'E'

scan 'temp'

get 'temp', 'rk1'

count 'temp'

exit

17. Теперь этот файл можно выполнить из оболочки HBase.

hbase shell

hbase(main):001:0> require ‘./rubyscript.rb’

Примечание: выше приведен экран результатов выполнения rubyscript.rb. Если в скрипте нет оператора exit, HBase остается в состоянии оболочки.

18. Можно также передавать команды в оболочку HBase в неинтерактивном режиме с помощью команды echo и оператора | (pipe).

echo "describe 'temp'" | hbase shell -n

Доступ к данным с помощью команд DML

В этой лабораторной работе вы будете использовать команды для вставки, извлечения, сканирования и удаления строк.

1. Операции CRUD (Вставка, выбор, обновление, удаление).

Запустите оболочку HBase:

hbase shell

1.1. Введите данные со следующими условиями и отобразите результат.

RowKey:rk1, column descriptor : b и значение: F

hbase(main):003:0> put ‘temp’, ‘rk1’, ‘cf1:b’, ‘F’

hbase(main):004:0> scan ‘temp’

1.2. Получение строки с ключом rk1 из временной таблицы. Get ‘’

Hbase(main):003:0> get ‘temp’, ‘rk1’

1.3. Получение двух значений предыдущих версий столбца 'b' в строке rk1.

Hbase(main):003:0> get ‘temp’, ‘rk1’, {COLUMNS => ‘cf1:b’, VERSIONS=>2}

hbase(main):004:0> get ‘temp’, ‘rk1’, {COLUMNS => ‘cf1:b’}

Примечание: Отображается первое входное значение 1B и последнее входное значение F. Если версия не указана, отображается только окончательное значение.

1.4. Просмотр всей таблицы, но отображение только столбцов 'a', 'b', 'c'.

Hbase(main):003:0> scan ‘temp’, {COLUMNS => [‘cf1:b’, ‘cf1:a’, ‘cf1:c’]}

1.5. Удаление ячейки из столбца 'b' из таблицы temp с ключом строки = rk1.

Hbase(main):003:0> delete ‘temp’, ‘rk1’, ‘cf1:b’

1.6. Убедитесь, что указанное значение в столбце 'b' было удалено. Если это не так, повторите команду выше.

Hbase(main):004:0> scan ‘temp’

1.7. Удаление всей строки из временной таблицы в ключе строки rk1.

Hbase(main):012:0> deleteall ‘temp’, ‘rk5’

1.8. Проверьте, что строка с rk1 была удалена из временной таблицы.

2. Использование MIN\_VERSIONS и TTL

2.1. Выведите описание таблицы temp и проверьте значение TTL столбцов из cf2. Значение TTL (Time-To-Live) по умолчанию — FOREVER, что означает, что срок действия версий ячейки никогда не истекает.

Измените значение с FOREVER на 30 секунд. Это означает, что версия будет удалена через 30 секунд после вставки данных в таблицу..

Base(main):005:0> alter ‘temp’, NAME => ‘cf2’, TTL=> 30

2.2. Проверьте, что TTL изменено с FOREVER на 30 секунд.

Вставим данные со следующими условиями и отобразим результаты.

RowKey:rk1, Column Family: cf2, column descriptor : ttl и значение Y

hbase(main):007:0> put ‘temp’, ‘rk1’, ‘cf2:ttl’, ‘Y’

2.3. Подождав не менее 30 секунд, выполните команду сканирования еще раз, чтобы узнать, истек ли срок действия вставленной строки и была ли она удалена.

Установите TTL на 10 секунд и установите MIN\_VERSIONS на 1 дляcf2.

Hbase(main):014:0> alter ‘temp’, NAME => ‘cf2’, TTL=> 10, MIN\_VERSIONS => 1

Примечание: Когда версия истекает с опцией MIN\_VERSION, она не будет удалена, если это единственная оставшаяся версия ячейки.

2.4. Проверим значения TTL и MIN\_VERSIONS в cf2.

Вставим данные со следующими условиями и отобразим результаты.

RowKey:rk5, Column Family: cf2, column descriptor : ttl\_min and value: Z

hbase(main):016:0> put ‘temp’, ‘rk5’, ‘cf2:ttl\_min’, ‘Z’

2.5. Подождав не менее 10 секунд, выполните команду сканирования еще раз, чтобы узнать, истек ли срок действия вставленной строки и осталась ли она.

Самостоятельная работа с HBase

1. Напишите команду для создания таблицы, отвечающей необходимым требованиям

1.1 Имя таблицы: movie

1.2 Содержит описание трех фильмов, каждому фильму соответствует столбец.

1.3 Все столбцы находятся в семействе title.

1.4 Название столбцов соответствуют названиям фильмов (выберите по желанию).

1.5 В ячейках таблицы описание.

1.6 Имя таблицы: ranking

1.7 Содержит информацию об оценках пользователей трех фильмов (опять же, фильмы - столбцы)

1.8 В ячейках таблицы оценки.

1.9 Используйте команду alter, чтобы добавить семейство столбцов (column family) fl\_title для названий фильмов на другом языке.

1.10 Убедитесь, что семейство столбцов fl\_title добавлено.

2. Напишите команды, отвечающие следующим условиям

2.1 Создайте таблицу user по следующим условиям:

1.1.1 Имя таблицы: user

1.1.2 Семейство столбцов: info

1.1.3 Создайте строку с Row key: rk100

1.1.4 Столбец age содержит значение 20

1.1.5 Столбец gender содержит значение F

1.1.6 Столбец zip содержит значение 18730

2.2 Отобразите строку с ключом строки rk100 из пользовательской таблицы.

Кроме того, вставьте строку c rk100 в таблицу user со следующими атрибутами:

1.2.1 Row key: rk100

1.2.2 Семейство столбцов: info

1.2.3 Столбец age - значение 30

1.2.4 Row key: rk100

1.2.5 Семейство столбцов: info

1.2.6 Столбец age - значение 40

2.3 Отобразите строки с ключом строки rk100 из пользовательской таблицы.

В пользовательской таблице получите все старые версии столбца age в строке с ключом rk100.

2.4 Отобразите всю таблицу с помощью команды сканирования.

В таблице пользователей удалите столбец info:age строки с ключом строки rk100.

2.5 Убедитесь, что столбец age удален.

Удалите все строки с ключом строки rk100 из таблицы user.

2.6 Убедитесь, что строка с ключом rk100 удалена из таблицы user.

Для того, чтобы сократить нагрузку на нашу виртуальную машину, мы остановим HBase и запустим только Apache Kafka. Перезапускаем Kafka и Zookeeper. Убеждаемся, что zookeeper и kafka запущены (Рисунок 1)

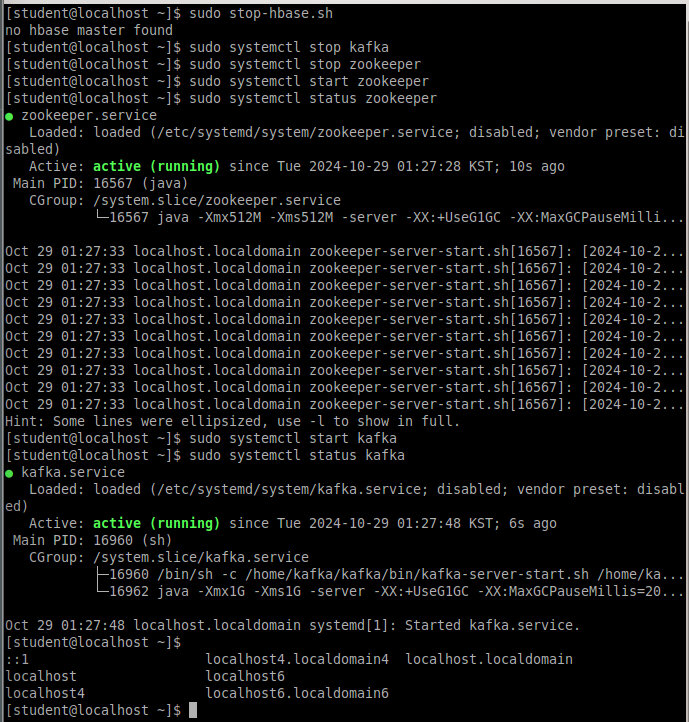


Рисунок — Остановка сервиса HBase и перезагрузка Kafka и Zookeeper

Создайте тему Kafka с названием topic1\_logs, которая будет содержать сообщения, представляющие строки в log файлах. (Рисунок 2)

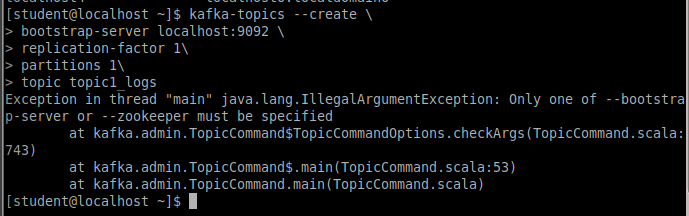


Рисунок 2 — Создание темы

Создайте потребителей и производителей для темы. Представлены на Рисунках 3 и 4.

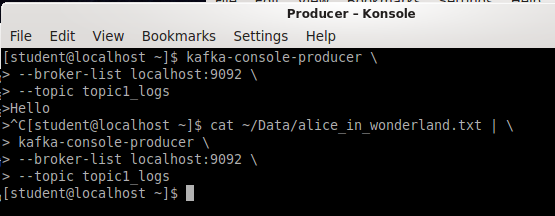
****

Рисунок 3 — Создание производителя

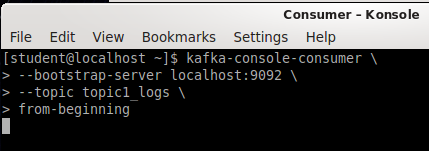
****

Рисунок 4 — Создание потребителя

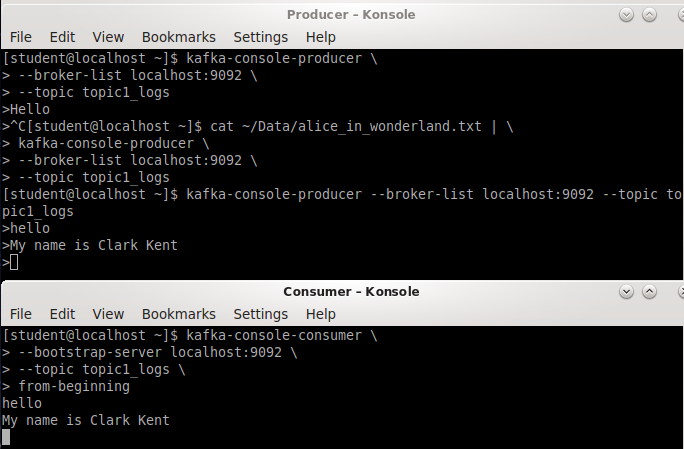
****

Рисунок 5 — Создание сообщения в пакетном режиме

Создайте конвейер Flume для сбора текста в потоковом режиме. Перейдите в папку /home/student/Labs/C3U4 в терминале, а затем создайте новую папку и назовите ее spooldir. Создадим файл конфигурации агента spooldir.conf и заполним его.

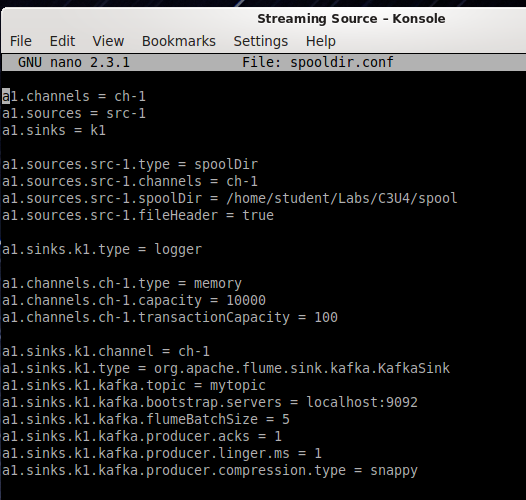


Рисунок 6 — Создание агента spooldir.conf

Запустите производителя Flafka через агента Flume

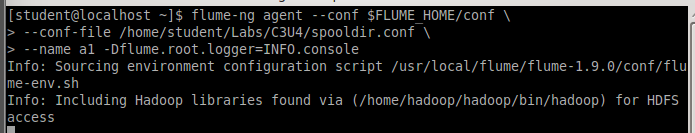


Рисунок 7 — Запуск Flafka

Запустите симулятор потокового источника

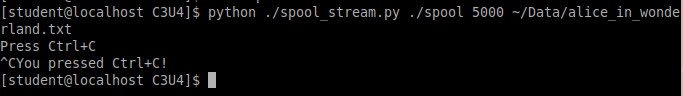


Рисунок 8 — Запуск симулятора

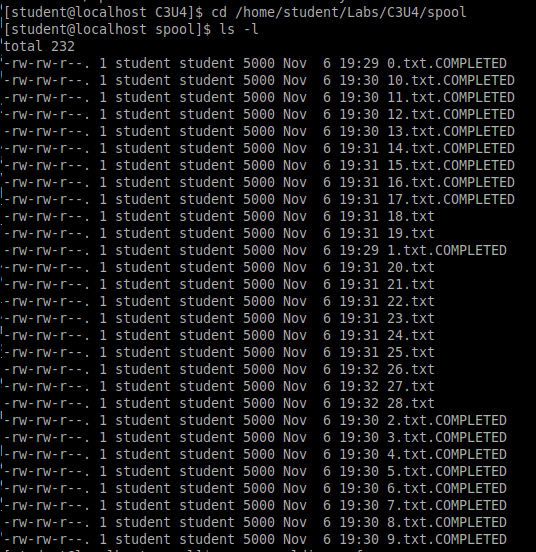


Рисунок 9 — Просмотр созданных файлов

**Вывод:** в ходе выполнения практической работы 5 мы научились использовать Kafka для создания производителей, потребителей и передачи данных через них.