Компонент загрузки и преобразования данных в составе Платформы по работе с данными Sber Data Platform (SDP)

SDP Data Flow

Руководство пользователя

Оглавление

1 Краткое описание SDP DataFlow

Компонент загрузки и преобразования данных в составе платформы по работе с данными SberData Platform» (далее SDP DataFlow) – сервис для приема, передачи и обработки данных в режиме реального времени и в режиме пакетной обработки, которая может передавать и управлять передачей данных между различными источниками и системами назначения. Инструмент поддерживает широкий спектр форматов исходных данных, таких как журналы, данные о географическом местоположении, социальные сети и т.д. Он также поддерживает множество протоколов, таких как SFTP, HDFS и KAFKA, и т.д. Так же реализована поддержка основных реляционных СУБД.

2 Основные характеристики

- SDP DataFlow Предоставляет веб-интерфейс пользователя, который обеспечивает плавное взаимодействие между дизайном, управлением, обратной связью и мониторингом;
- Возможность гибких настроек, для достижения масштабирования и повышения пропускной способности;
- Наличие модуля для отслеживания и мониторинга данных от начала до конца потока;
- Возможность создавать свои собственные процессоры и задачи отчетности в соответствии со своими потребностями;
- Поддержка безопасных протоколов, таких как SSL, HTTPS, SSH и других;
- Управление пользователями и ролями, а также может быть настроен с LDAP для авторизации;
- Получение данных с удаленных компьютеров с помощью SFTP и гарантирует передачу данных;
- Поддержка кластеризацию, поэтому он может работать на нескольких узлах с одинаковым потоком, обрабатывая разные данные, что повышает производительность обработки данных;
- Предоставление политики безопасности на уровне пользователя, группы процессов и других модулей;
- Поддержка более 200 процессоров, и пользователь также может создавать собственные плагины для поддержки широкого спектра систем данных.

3 Основные понятия

Основные понятия SDP DataFlow -компонента загрузки и преобразования данных «Платформа по работе с данными Сбера SberData Platform» следующие:

- Группа процессов – это группа потоков NiFi, которая помогает пользователю управлять и поддерживать потоки в иерархическом порядке;
- Поток – создается для соединения разных процессоров для передачи и изменения данных, если это необходимо, из одного источника данных или источников в другие источники данных назначения;
- Процессор – это Java-модуль, отвечающий за выборку данных из системы источников или сохранение их в системе назначения. Другие процессоры также используются для добавления атрибутов или изменения содержимого в потоковых файлов;
- Flowfile – поточный файл– является основным объектом обработки в SDP DataFlow. Он содержит содержимое и атрибуты данных, которые используются процессорами NiFi для обработки данных. Содержимое файла обычно содержит данные, полученные из исходных систем;
- Событие – события представляют изменения потока файла при прохождении потока NiFi;
- Происхождение данных - также имеет пользовательский интерфейс, который позволяет пользователям проверять информацию о потоковом файле и помогает в устранении неполадок, возникающих при обработке потокового файла.

4 Компоненты (репозитории)

SDP DataFlow состоит из веб-сервера, контроллера потока и процессора, который работает на виртуальной машине Java. Он также имеет три репозитория Flowfile Repository, Content Repository и Provenance Repository, как показано на рисунке ниже (см. Рисунок 1).

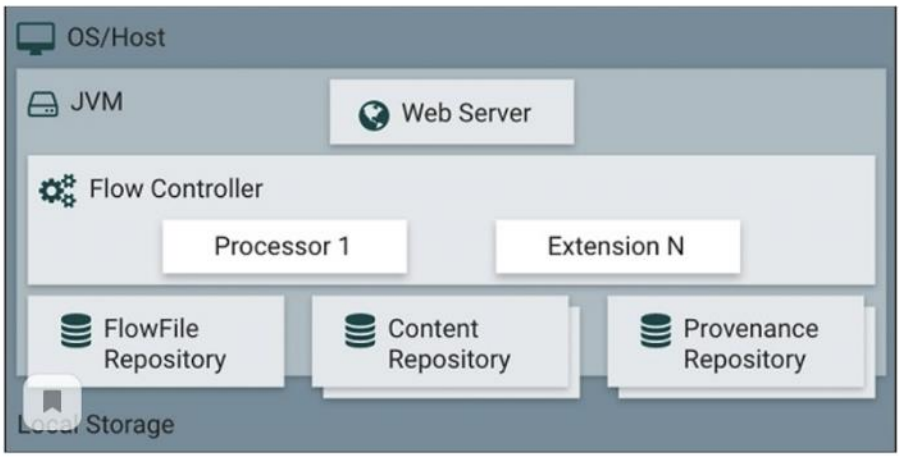

Рисунок 1 - Структура Компонента загрузки и преобразования данных

4.1 Flowfile Repository

Этот репозиторий хранит текущее состояние и атрибуты каждого потокового файла, который проходит через потоки данных SDP DataFlow. Расположение этого хранилища по умолчанию находится в корневом каталоге SDP DataFlow. Расположение этого репозитория можно изменить, изменив свойство с именем «nifi.flowfile.repository.directory».

4.2 Репозиторий контента

Этот репозиторий содержит все содержимое всех потоковых файлов NiFi. Его каталог по умолчанию также находится в корневом каталоге NiFi, и его можно изменить с помощью свойства «org.apache.nifi.controller.repository.FileSystemRepository». Этот каталог занимает много места на диске, поэтому желательно иметь достаточно места на установочном диске.

4.3 Хранилище Provenance

Репозиторий отслеживает и хранит все события всех потоковых файлов, которые поступают в NiFi. Существует два репозитория– изменяемое хранилище (в этом репозитории все данные «Provenance» теряются после перезапуска) и постоянное хранилище. Его каталог по умолчанию также находится в корневом каталоге NiFi, и его можно изменить с помощью свойств:

- «org.apache.nifi.provenance.PersistentProvenanceRepository»;
- «org.apache.nifi.provenance.VolatileProvenanceRepositor»

для соответствующих репозиториев.

5 Пользовательский интерфейс

Пользовательский интерфейс Компоненты предоставляет широкий спектр информации. Как показано на рисунке ниже (см. Рисунок 2), пользователь может получить доступ к информации о следующих атрибутах:

- Количество узлов кластера
- Количество активных потоков;
- Всего данных в очереди;
- Передающие удаленные процессные группы;
- Непередающие удаленные процессные группы ;
- Запущенные компоненты;
- Остановленные компоненты;
- Неисправные компоненты;
- Отключенные компоненты;
- Информация о версионности.

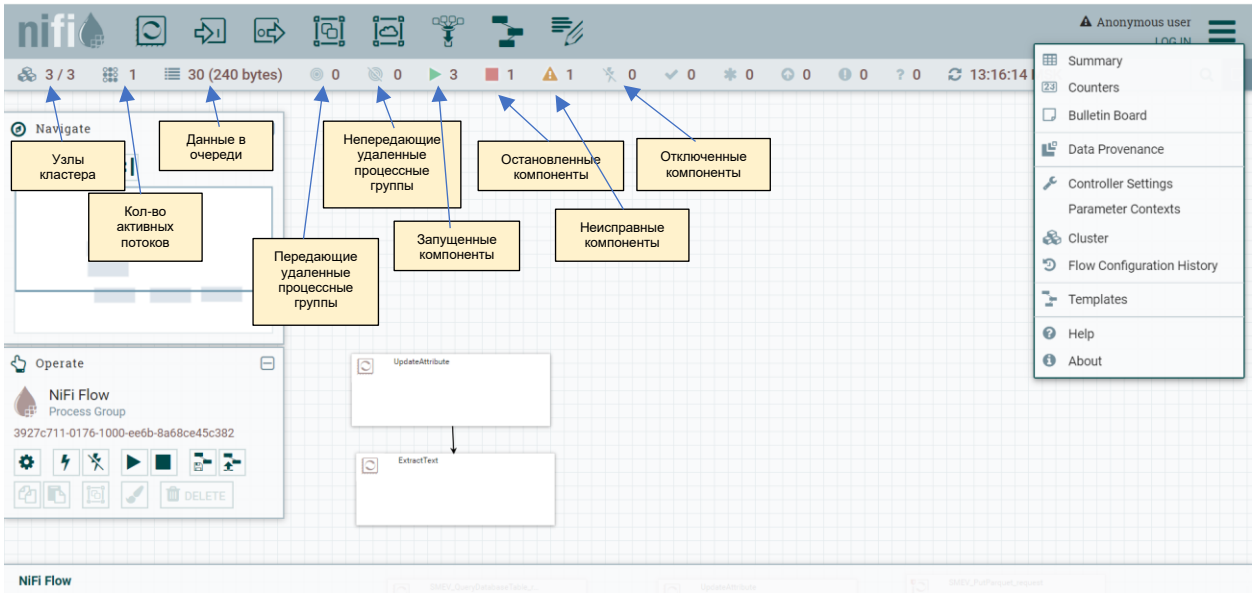

Рисунок 2 —Пользовательский интерфейс

Описания основных элементов SDP DataFlow, которые можно настроить с использованием пользовательского интерфейса приведены ниже.

5.1 Процессоры

Пользователь может перетащить значок процесса на холст и выбрать нужный процессор для потока данных в NiFi.

Процессоры SDP DataFlow являются основными блоками создания потока данных. Каждый процессор имеет разные функциональные возможности, что способствует созданию выходного потокового файла. Поток данных, показанный на изображении ниже (см. Рисунок 3), извлекает файл из одного каталога с использованием процессора GetFile и сохраняет его в другом каталоге с помощью процессора PutFile.

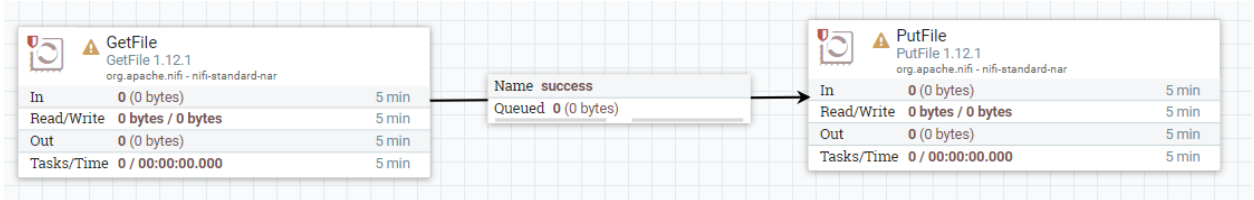

Рисунок 3 — Создание выходного потока данных

5.1.1 Процессор GetFile

Процесс GetFile используется для извлечения файлов определенного формата из определенного каталога (см. Рисунок 4). Он также предоставляет пользователю другие возможности для большего контроля при извлечении.

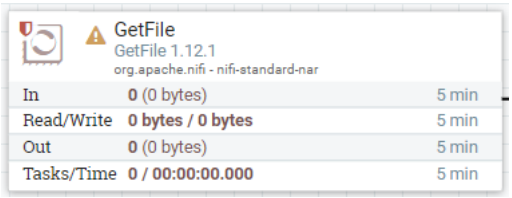

Рисунок 4 — Процесс GetFile

Настройки GetFile

Ниже приведены различные настройки процессора GetFile:

- название - в настройке «Имя» пользователь может определить любое имя для процессоров в соответствии с проектом или тем, что делает имя более значимым;
- включить - пользователь может включить или отключить процессор, используя этот параметр;
- длительность штрафа - этот параметр позволяет пользователю добавить длительность штрафного времени в случае сбоя потока файла.
- уровень бюллетеня - этот параметр используется для указания уровня журнала этого процессора.

Автоматическое завершение соединения

Здесь есть список проверок всех доступных отношений конкретного процесса. Установив флажки, пользователь может запрограммировать процессор на прекращение потока файла для этого события и не отправлять его дальше в потоке (см. Рисунок 5).

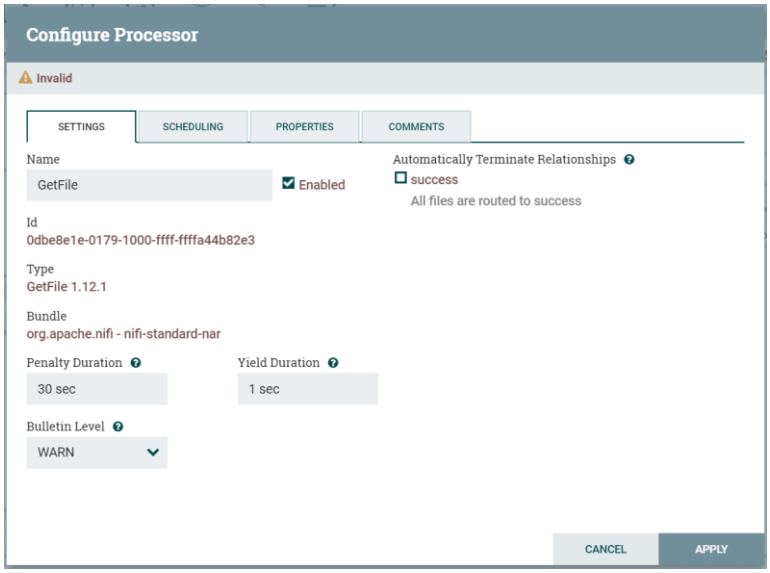

Рисунок 5 —Панель конфигурации процессора

Настройка расписания GetFile

Параметры планирования, предлагаемые процессором GetFile (см. Рисунок 6):

- стратегия планирования - вы можете либо запланировать процесс на основе времени, выбрав время или указанную строку CRON, выбрав опцию драйвера CRON;
- параллельные задачи - эта опция используется для определения расписания одновременных задач для этого процессора;
- выполнение - пользователь может определить, запускать ли процессор во всех узлах или только в основном узле, используя эту опцию;
- расписание запуска - он используется для определения стратегии, основанной на времени, или выражения CRON для стратегии, управляемой CRON.

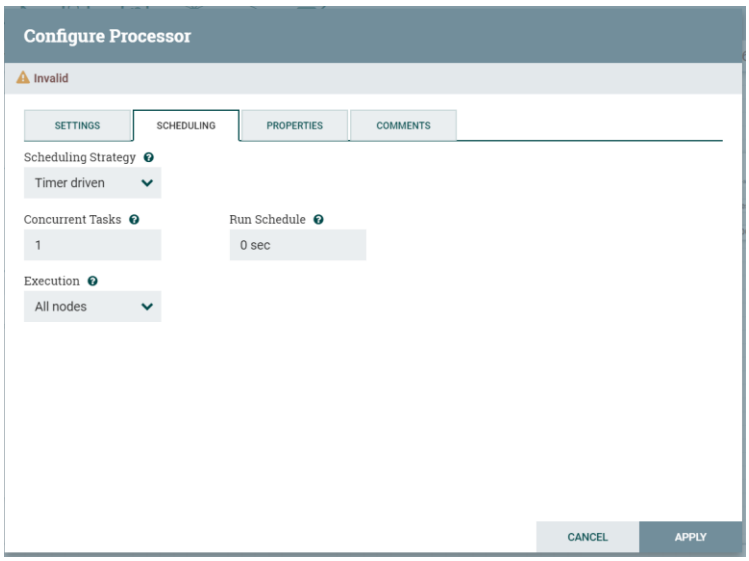

Рисунок 6 — Панель настроек расписания

Свойства GetFile

GetFile предлагает несколько свойств, как показано на рисунке ниже (см. Рисунок 7), а также обязательные свойства, такие как Входной каталог и фильтр файлов, для дополнительных свойств, таких как Path Filter и Maximum File Size. Пользователь может управлять процессом извлечения файлов, используя эти свойства.

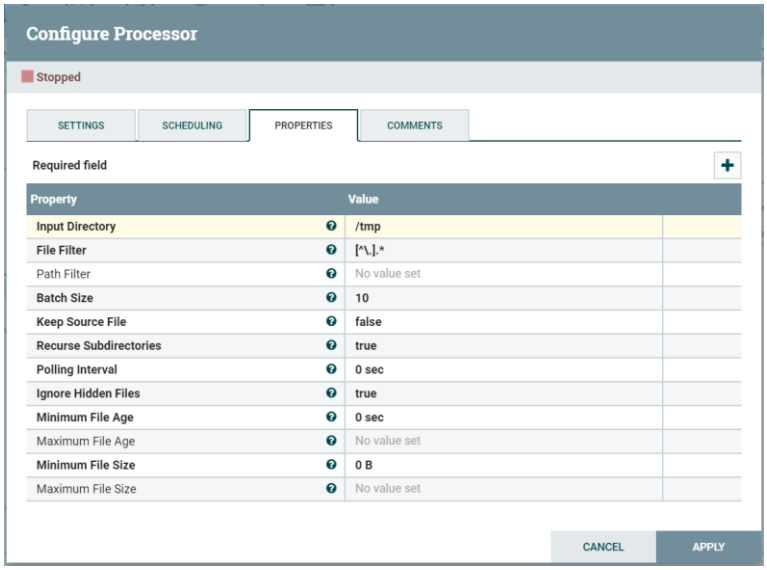

Рисунок 7 — Свойства GetFile

GetFile Комментарии

Этот раздел (см. Рисунок 8) используется для указания любой информации о процессоре.

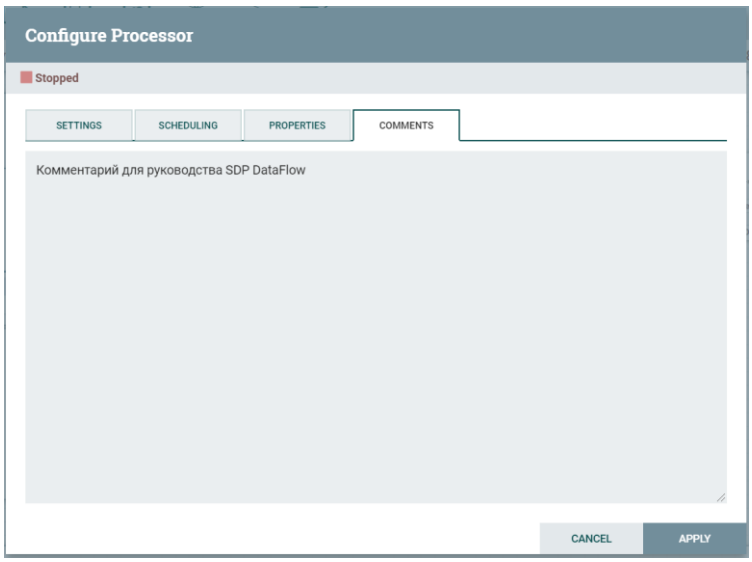

Рисунок 8 — Панель комментарии GetFile

5.1.2 Процессор PutFile

Процессор PutFile используется для хранения файла из потока данных в определенном месте (см. Рисунок 9).

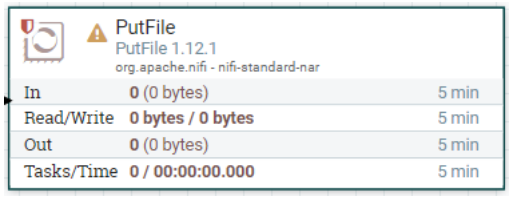

Рисунок 9 — Процессор PutFile

Настройки PutFile

Процессор PutFile имеет следующие настройки:

- название - в настройке «Имя» пользователь может определить любое имя для процессоров в соответствии с проектом или тем, что делает имя более значимым;
- включить - пользователь может включить или отключить процессор, используя этот параметр;
- длительность штрафа - этот параметр позволяет пользователю добавить длительность штрафного времени в случае сбоя потока файла;
- уровень бюллетеня - этот параметр используется для указания уровня журнала этого процессора.

Автоматическое завершение соединения

В этих настройках есть список проверок всех доступных взаимосвязей этого конкретного процесса. Установив флажки, пользователь может запрограммировать процессор на прекращение потока файла для этого события и не отправлять его дальше в потоке (см. Рисунок 10).

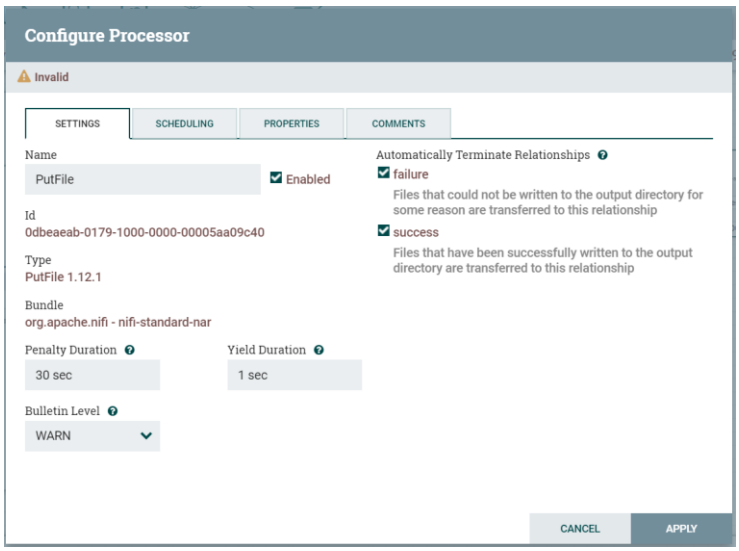

Рисунок 10 — Панель конфигурации процессора

Управление расписанием PutFile

К параметрам планирования процессора PutFile относятся:

- стратегия планирования - вы можете запланировать процесс на основе времени, либо выбрав управляемый таймером, либо указав строку CRON, выбрав опцию драйвера CRON. Существует также экспериментальная стратегия Event Driven, которая запускает процессор при конкретном событии;
- параллельные задач - эта опция используется для определения расписания одновременных задач для этого процессора;
- выполнение - пользователь может определить, следует ли запускать процессор во всех узлах или только в основном узле, используя эту опцию;
- расписание запуска (см. Рисунок 11) - он используется для определения времени для стратегии, управляемой таймером, или выражения CRON для стратегии, управляемой CRON.

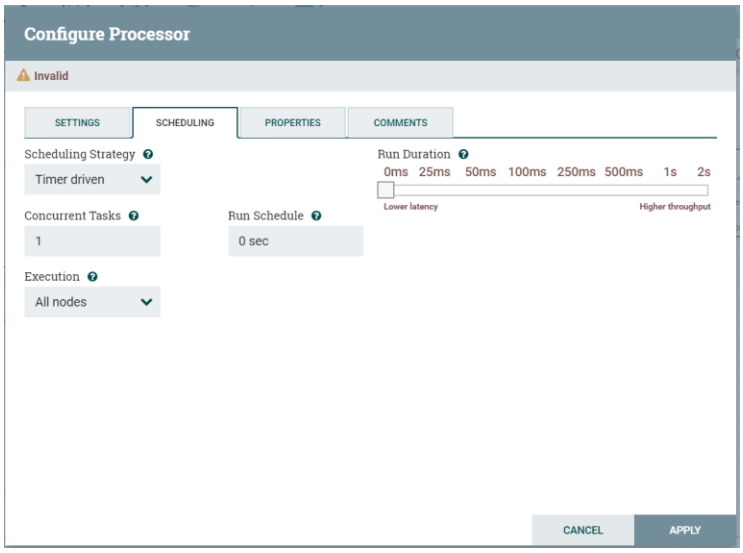

Рисунок 11 — Панель настроек расписания

Свойства PutFile

Процессор PutFile предоставляет (см. Рисунок 12) такие свойства, как Directory, чтобы указать выходной каталог для передачи файлов, а другие – для управления передачей, как показано на рисунке ниже.

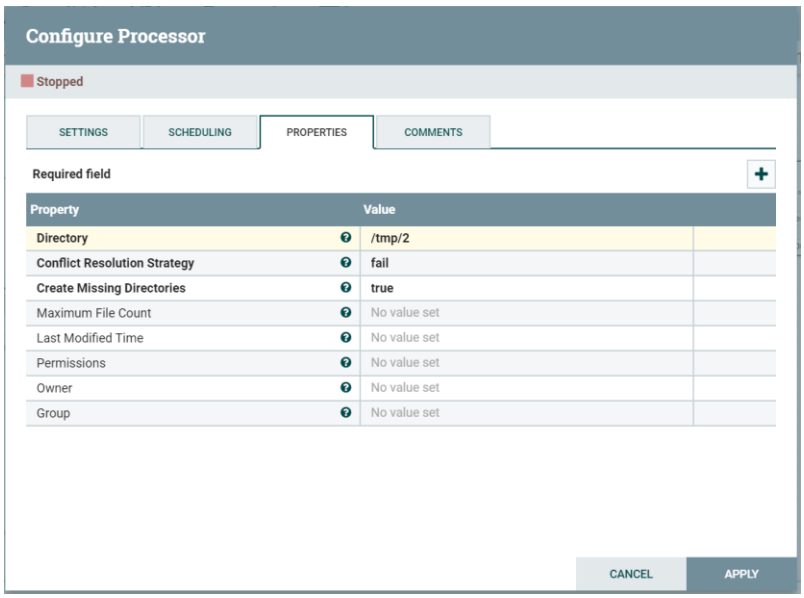

Рисунок 12 — Свойства PutFile

5.2 Входящий порт

Значок ниже (см. Рисунок 13) перетаскивается на холст, чтобы добавить входной порт в любой поток данных.

Входной порт используется для получения данных от процессора, которого нет в этой группе процессов.

Рисунок 13 — Иконка порта ввода

После перетаскивания этого значка NiFi просит ввести имя порта ввода (см. Рисунок 14), а затем оно добавляется на холст NiFi.

Рисунок 14 — Панель ввода имени порта

5.3 Исходящий порт

Значок ниже (см. Рисунок 15) перетаскивается на холст, чтобы добавить выходной порт в любой поток данных.

Выходной порт используется для передачи данных процессору, которого нет в этой группе процессов.

Рисунок 15 — Иконка выходного порта

После перетаскивания этого значка NiFi просит ввести имя выходного порта, а затем он добавляется на холст NiFi.

5.4 Группа процессов

Пользователь использует значок ниже (см. Рисунок 16), чтобы добавить группу процессов на холст NiFi.

Рисунок 16 — Иконка группы процессов

После перетаскивания этого значка NiFi просит ввести имя группы процессов, а затем оно добавляется на холст NiFi.

5.5 Удаленная группа процессов

Этот компонент используется для добавления удаленной группы процессов в холст NiFi. Значок удаленной группы представлен на рисунке ниже (см. Рисунок 17).

Рисунок 17 — Иконка удаленной группы

5.6 Funnel (Воронка)

Воронка используется для передачи выходных данных процессора нескольким процессорам. Пользователь может использовать значок ниже (см. Рисунок 18), чтобы добавить воронку в поток данных NiFi.

Рисунок 18 — Воронка

5.7 Шаблоны

Этот значок (см. Рисунок 19) используется для добавления шаблона потока данных на холст NiFi. Это помогает повторно использовать поток данных в одном и том же или разных экземплярах NiFi.

Рисунок 19 — Иконка добавления шаблона

После перетаскивания пользователь может выбрать шаблоны, уже добавленные в NiFi.

SDP DataFlow предлагает концепцию шаблонов, которая упрощает повторное использование и распределение потоков NiFi. Потоки могут быть использованы другими разработчиками или в других

кластерах NiFi (см. Рисунок 20). Это также помогает разработчикам NiFi делиться своей работой в таких репозиториях, как GitHub.

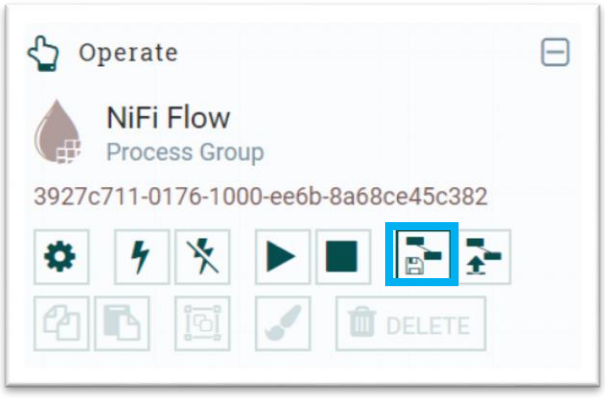

Рисунок 20 — Выбор операции «Создать шаблон»

Создание шаблона

Выделите все компоненты потока с помощью клавиши Shift, а затем щелкните значок создания шаблона в левой части холста NiFi. Вы также можете «Tool box», как показано на рисунке выше. Нажмите на иконку создания шаблона, отмеченную синим, как на картинке выше. Введите имя для шаблона. Разработчик также может добавить описание, которое не является обязательным.

Скачивание шаблона

Затем перейдите к пункту «Шаблоны NiFi» в меню в верхнем правом углу пользовательского интерфейса NiFi, как показано на рисунке ниже (см. Рисунок 21).

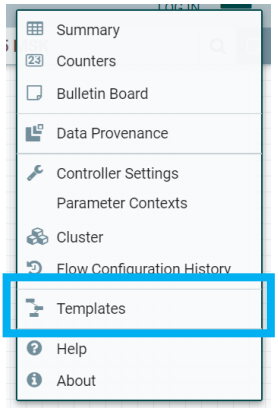

Рисунок 21 — Меню пользовательского интерфейса NiFi

Теперь щелкните значок загрузки (присутствует справа в списке) шаблона, который вы хотите загрузить. Файл XML с именем шаблона будет загружен.

Загрузка шаблона

Чтобы использовать шаблон в NiFi, разработчик должен загрузить свой XML- файл в NiFi с помощью пользовательского интерфейса. Рядом со значком «Создать шаблон» есть значок «Загрузить шаблон» (помечен синим цветом на изображении ниже) (см. Рисунок 22). После загрузки возможен просмотр XML-файла.

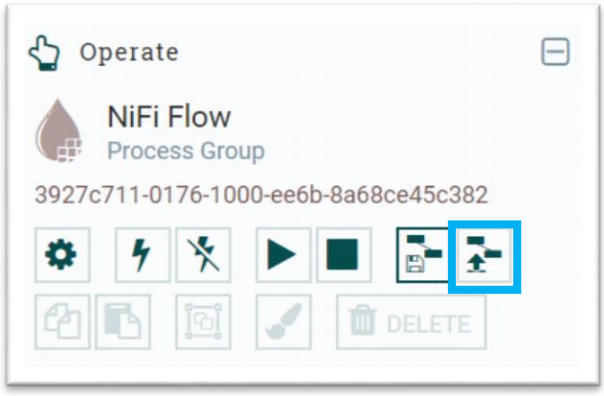

Рисунок 22 — Операция «загрузить шаблон»

Добавление шаблона

На верхней панели инструментов NiFi UI значок шаблона находится перед значком метки. Значок помечен синим цветом, как показано на рисунке ниже (см. Рисунок 23).

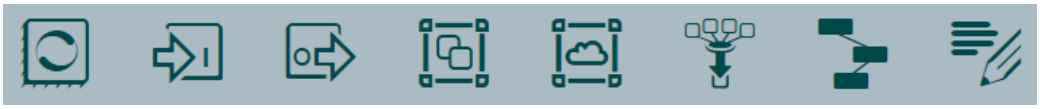

Рисунок 23 — Расположение иконки шаблона на панели инструментов UI

Перетащите значок шаблона, выберите шаблон из выпадающего списка и нажмите «Добавить» Это добавит шаблон к холсту NiFi.

5.8 Метки

Они используются для добавления текста на холсте NiFi о любом компоненте, присутствующем в NiFi. Доступен выбор цвета из палитры, чтобы добавить эстетический смысл (см. Рисунок 24).

Рисунок 24 — Иконка метки

5.9 Очереди

Подключение потока данных SDP DataFlow. SDP DataFlow имеет систему очередей для обработки большого объема данных. Эти очереди могут обрабатывать очень большое количество FlowFiles, чтобы процессор мог обрабатывать их последовательно (см. Рисунок 25).

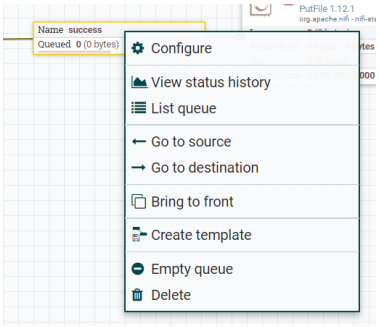

Рисунок 25 — Меню операций с очередью

В очереди на изображении выше (см. Рисунок 25) есть один потоковый файл, переданный через отношения успеха. Пользователь может проверить файл потока, выбрав опцию Список очереди в раскрывающемся списке. В случае любой перегрузки или ошибки пользователь также может очистить очередь, выбрав опцию пустой очереди, а затем пользователь может перезапустить поток, чтобы снова получить эти файлы в потоке данных.

Рисунок 26 — Список потоковых файлов в очереди

Список потоковых файлов в очереди (см. Рисунок 26) состоит из позиции, UUID, имени файла, размера файла, длительности очереди и длительности линии. Пользователь может просмотреть все атрибуты и содержимое потокового файла, щелкнув значок информации, присутствующий в первом столбце списка потокового файла.

5.10 Процессные группы

В SDP DataFlow пользователь может поддерживать разные потоки данных в разных группах процессов. Эти группы могут основываться на разных проектах или организациях, которые поддерживает экземпляр SDP DataFlow.

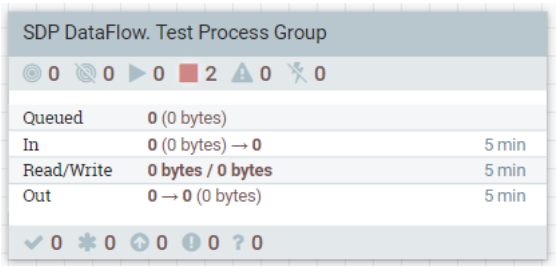

Рисунок 27 — Группа процессов

Четвертый символ в меню в верхней части интерфейса NiFi, как показано на рисунке выше (см.Рисунок 23), используется для добавления группы процессов в холст NiFi. Группа процессов с именем «ProcessGroup» содержит поток данных с четырьмя процессорами, которые в данный момент находятся в стадии остановки, как вы можете видеть на рисунке выше. Группы процессов могут быть созданы иерархически.

В нижнем колонтитуле пользовательского интерфейса NiFi вы можете увидеть группы процессов и вернуться к началу группы процессов, в которой в настоящее время находится пользователь.

Чтобы увидеть полный список групп процессов, представленных в NiFi, пользователь может перейти к сводке, используя меню, представленное в левой верхней части интерфейса NiFi. Таким образом, есть вкладка групп процессов, в которой перечислены все группы процессов с такими параметрами, как состояние версии, перенесено / размер, в / размер, чтение / запись, выход / размер и т.д.

5.11 Ярлыки

SDP DataFlow предлагает ярлыки, позволяющие разработчику писать информацию о компонентах, представленных на холсте NiFI. Крайний правый значок в верхнем меню NiFi UI используется для добавления метки на холсте NiFi (см.Рисунок 23).

Разработчик может изменить цвет метки и размер текста, щелкнув правой кнопкой мыши по метке и выбрав соответствующую опцию в меню.

6 Разработка потоков

6.1 Создание потока

SDP DataFlow предлагает большое количество компонентов, которые помогают разработчикам создавать потоки данных для любых типов протоколов или источников данных. Чтобы создать поток, разработчик перетаскивает компоненты из строки меню на холст и соединяет их, щелкая и перетаскивая мышь из одного компонента в другой.

Как правило, NiFi имеет компонент слушателя в начале потока, такой как GetFile, который получает данные из исходной системы. На другом конце находится компонент-передатчик, такой как PutFile, и между ними есть компоненты, которые обрабатывают данные.

Например, давайте создадим поток, который берет пустой файл из одного каталога, добавляет некоторый текст в этот файл и помещает его в другой каталог (см. Рисунок 28).

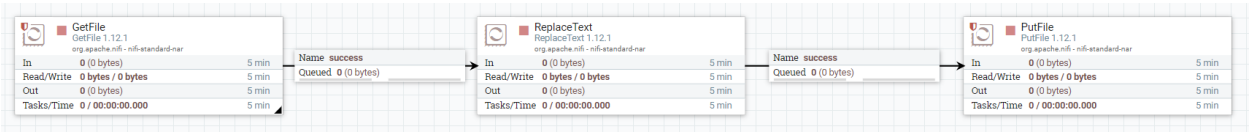

Рисунок 28 — Пример потока

- для начала перетащите значок процессора на холст NiFi и выберите процессор GetFile из списка:
  ⇒ создайте входной каталог, например, c:\inputdir;
  ⇒ щелкните правой кнопкой мыши по процессору и выберите «Настроить», на вкладке свойств добавьте «Входной каталог» (c:\inputdir), нажмите «Применить» и вернитесь на холст;
- перетащите значок процессора на холст и выберите процессор ReplaceText из списка;
- щелкните правой кнопкой мыши по процессору и выберите «Настроить». На вкладке свойств добавьте текст типа «Hello» в текстовое поле «Значение замены» и нажмите «Применить»;
- перейдите на вкладку «Настройки», установите флажок «Ошибка» справа и вернитесь на холст;
- подключите процессор GetFIle к ReplaceText;
- перетащите значок процессора на холст и выберите процессор PutFile из списка:
  ⇒ создайте выходной каталог, например, c:\outputdir;
  ⇒ щелкните правой кнопкой мыши по процессору и выберите «Настроить».На вкладке свойств добавьте каталог (c:\outputdir), нажмите «Применить» и вернитесь на холст;
- перейдите на вкладку «Настройки» и установите флажок «Ошибка и успех» справа, а затем вернитесь на холст;
- подключите процессор ReplaceText к PutFile;
- теперь запустите поток и добавьте пустой файл во входной каталог, и вы увидите, что он переместится в выходной каталог, и текст будет добавлен в файл.

Выполнив вышеуказанные шаги, разработчики могут выбрать любой процессор и другой компонент NiFi, чтобы создать подходящий поток для своей организации или клиента.

6.2 Происхождение данных

SDP DataFlow регистрирует и хранит каждую информацию о событиях, произошедших в загруженных данных в потоке. Хранилище данных хранит информацию о происхождении данных и предоставляет пользовательский интерфейс (см. Рисунок 29) для поиска информации об этом событии. Доступ к данным можно получить как на уровне NiFi, так и на уровне процессора.

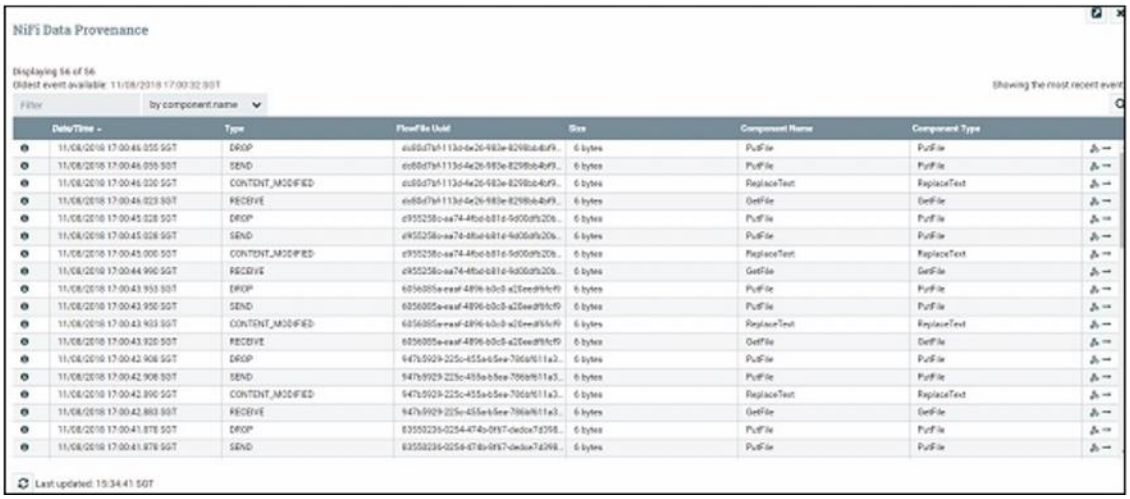

Рисунок 29 — Информация о событиях в потоке

В следующей таблице (Таблица 1) перечислены различные поля в списке событий NiFi Data Provenance:

Таблица 1 — Поля списка событий NiFi Data Provenance

№	Имя поля	Описание
1	Дата / время	Дата и время события.
2	Тип	Тип события, как «CREATE».
3	FlowFileUuid	UUID файла потока, для которого выполняется событие.
4	Размер	Размер потока файла.
5	Имя компонента	Имя компонента, который выполнил событие.
6	Тип компонента	Тип компонента.
7	Показать родословную	В последнем столбце есть значок show lineage, который используется для просмотра линии потока файла, как показано на рисунке ниже.

Схема последовательности событий представлена на рисунке (см. Рисунок 30).

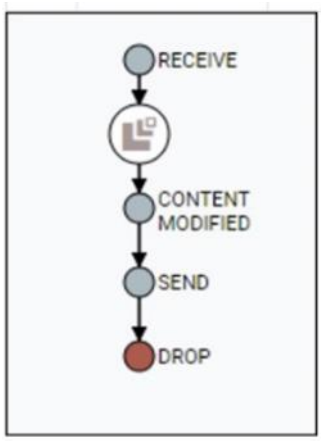

Рисунок 30 – Схема последовательности событий

Чтобы получить больше информации о событии, пользователь может щелкнуть значок информации в первом столбце интерфейса пользователя NiFi Data Provenance.

В файле nifi.properties есть некоторые свойства, которые используются для управления хранилищем данных NiFi Data Provenance.

6.3 Группа удаленных процессоров

SDP DataFlow Remote Process Group или RPG позволяет потоку направлять потоки файлов в поток к различным экземплярам NiFi с использованием протокола Site-to-Site.

Разработчик может добавить RPG с верхней панели инструментов пользовательского интерфейса NiFi, перетащив значок (Таблица 1), на холст. Чтобы настроить RPG, Разработчик должен добавить следующие поля (Таблица 2):

Таблица 2 – Поля настройки RPG

№	Имя поля	Описание
1	URL-адрес	Указать разделенные запятыми URL-адреса удаленных целевых NiFi.
2	Транспортный протокол	Указать транспортный протокол для удаленных экземпляров NiFi. Это либо RAW, либо HTTP.
3	Интерфейс локальной сети	Указать локальный сетевой интерфейс для отправки / получения данных.
4	HTTP прокси-сервер имя хоста	Указать имя хоста прокси-сервера для транспортировки в RPG.
5	Порт прокси-сервера HTTP	Указать порт прокси-сервера для транспортной цели в RPG.
6	Пользователь HTTP-прокси	Это необязательное поле для указания имени пользователя для HTTP-прокси.
7	Пароль прокси HTTP	Это необязательное поле для указания пароля для указанного выше имени пользователя.

Разработчик должен включить его, прежде чем использовать его, как мы запускаем процессоры, прежде чем их использовать.

6.4 Настройка контроллера

SDP DataFlow предлагает общие сервисы, которые могут совместно использоваться процессорами, а задача создания отчетов называется настройкой контроллера. Это как пул соединений с базой данных, который может использоваться процессорами, обращающимися к одной и той же базе данных.

Чтобы получить доступ к настройкам контроллера, используйте раскрывающееся меню в правом верхнем углу интерфейса NiFi, как показано на рисунке ниже (см. Рисунок 31).

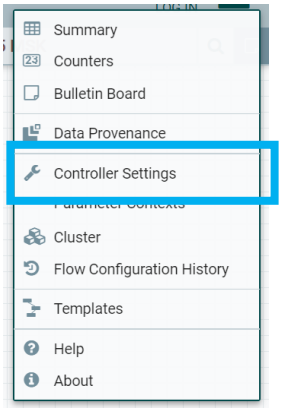

Рисунок 31 — Переход в настройки контроллера

6.5 DBCPConnectionPool

Добавьте знак «плюс» на странице «Настройки Nifi» после выбора параметра «Настройки контроллера». Затем выберите DBCPConnectionPool из списка настроек контроллера. DBCPConnectionPool будет добавлен на главной странице настроек NiFi.

Нажмите на иконку конфигурации и заполните необходимые поля. Поля перечислены в таблице ниже (Таблица 3).

Таблица 3 – Необходимы поля

№	Имя поля	Значение по умолчанию	Описание
1	URL соединения с базой данных	пустой	Уазать URL-адрес подключения к базе данных.
2	Имя класса драйвера базы данных	пустой	Указать имя класса драйвера для базы данных.
3	Максимальное время ожидания	500 мс	Указать время ожидания данных от соединения с базой данных.
4	Макс. Всего подключений	8	Указать максимальное количество выделенного соединения в пуле соединений с базой данных.

Чтобы остановить или настроить параметры контроллера, сначала необходимо остановить все подключенные компоненты NiFi.

NiFi также добавляет область действия в настройках контроллера для управления его конфигурацией. Следовательно, не будут затронуты только те, которые имеют одни и те же настройки и будут использовать те же настройки контроллера.

7 Подключение к SDP Hadoop

Ниже описаны действия по настройке подключения NiFi к SDP Hadoop с помощью Kerberos-аутентификации, но если у вас нет Kerberos, то игнорируйте создание keytab и настройку krb5.conf, остальные действия похожи.

7.1 Настройка Kerberos

1 Зайдите на сервер с помощью SSH.
2 Выполните команды

```
kinit -p username@SDPLAB.LOCAL
cd /tmp
ipa-getkeytab -p username@SDPLAB.LOCAL -k username.keytab
```

Если вы генерируете keytab для пользовательской учетной записи, добавьте «-P» и введите пароль, совпадающий с текущим паролем этой учетной записи.

3 Скопируйте получившийся keytab из папки /tmp на все серверы кластера nifi, например, в папку /Hadoop. Убедитесь, что пользователь из-под которого запускается NiFi имеет доступ к keytab.
4 В файле nifi/conf/nifi.properties на всех серверах кластера необходимо прописать путь до krb5.ini (Windows) или krb5.conf (Linux):

```
nifi.kerberos.krb5.file=krb5.ini
```

7.2 Получение файлов конфигурации Hadoop

1 Зайдите в Ambari (система управления SDP Hadoop)
2 Перейти в раздел HDFS
3 Скачайте клиентскую конфигурацию, нажав, Download Client Configs из меню Service Action
4 Скопируйте файлы core-site.xml и hdfs-site.xml на сервер с установленным nifi.

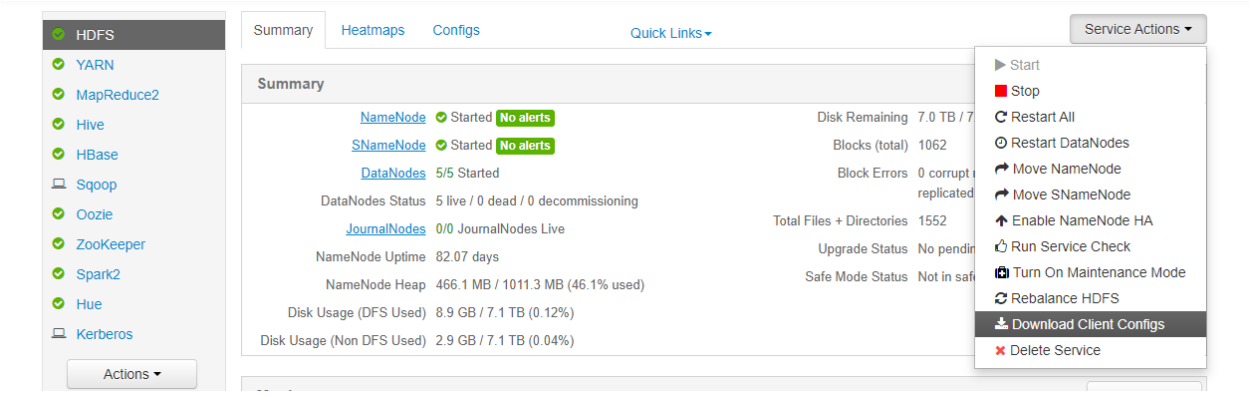

Рисунок 32. Интерфейс Ambari

7.3 Пример потока для работы с SDP Hadoop

1 Зайдите в пользовательский интерфейс NiFi
2 Нажмите правой кнопкой на свободное место и выберите пункт контекстного меню Variables.
3 Заполните переменные процесса и нажмите APPLY:

⇒ hadoopConf=<путь к core-site.xml>,<путь к hdfs-site.xml>
⇒ keytab=<путь к keytab>
⇒ principal=username@SDPLAB.LOCAL

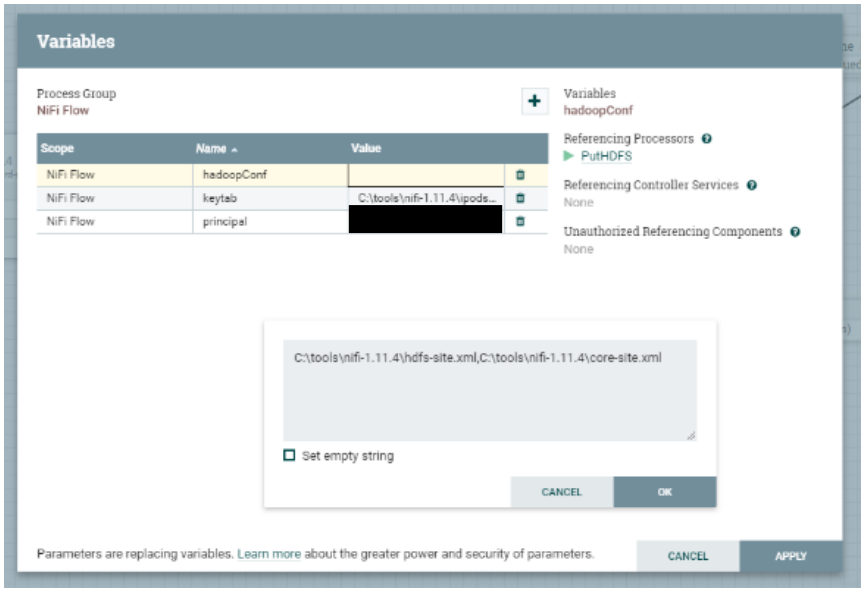

Рисунок 33. Глобальные переменные NiFi

4 Добавьте процессор GenerateFlowFile.
5 Откройте конфигурацию процессора и отредактируйте. Нажмите APPLY

⇒ Run Schedule=60 sec
⇒ Custom Text=AnyText

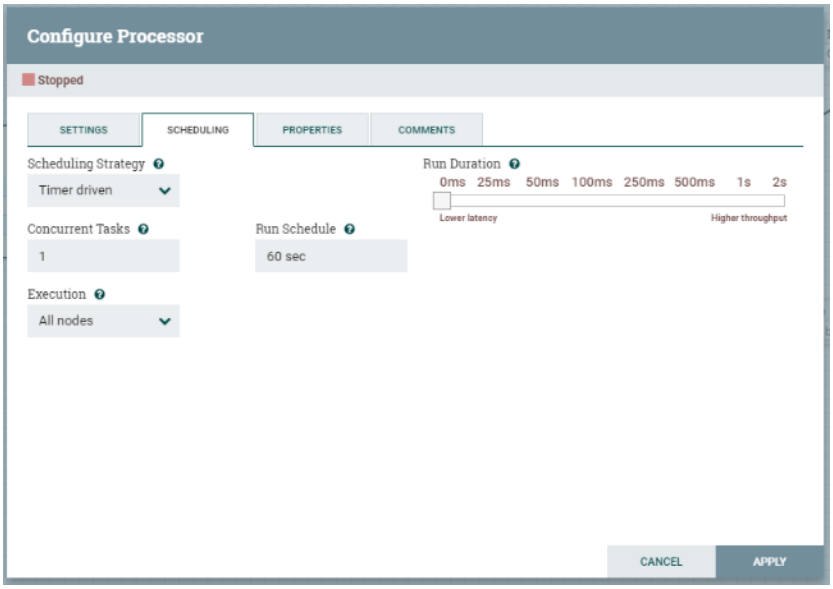

Рисунок 34. Настройка расписания

6 Добавьте процессор PutHDFS, настройте и нажмите APPLY:

⇒ Hadoop Configuration Resources=${hadoopConf}
⇒ Kerberos Principal=${principal}
⇒ Kerberos Keytab=${keytab}

Directory=/home/username

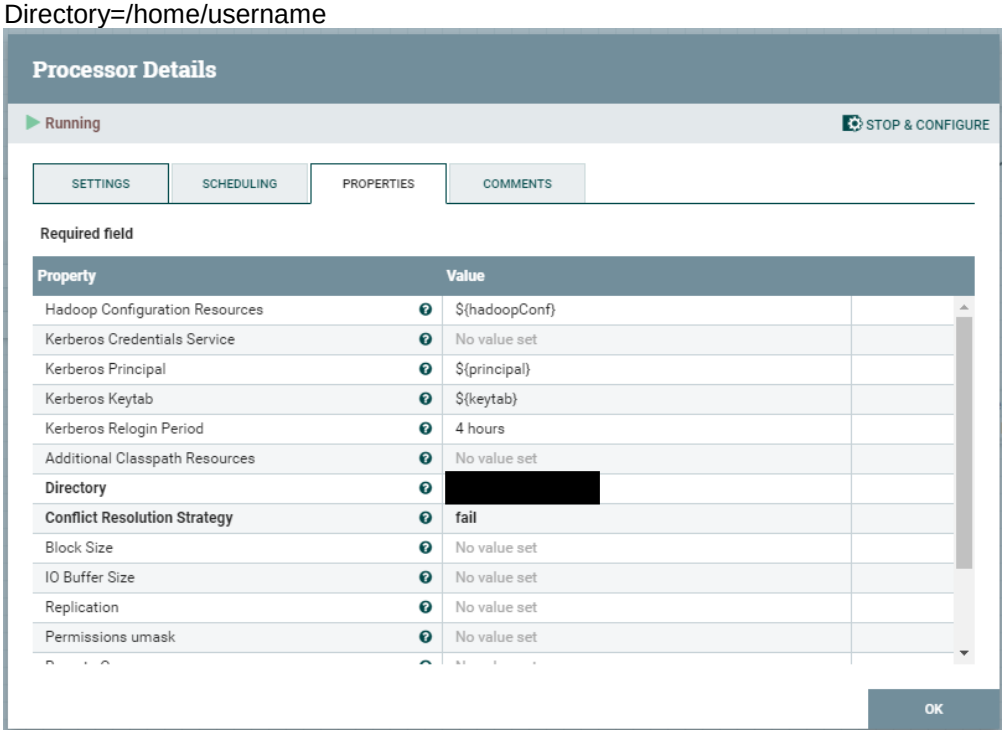

Рисунок 35. Настройка свойств

7 Добавьте процессор LogMessage
8 Подключите success и failure выходы PutHDFS к LogMessage
9 Подключите выход GenerateFlowFile к PutHDFS.

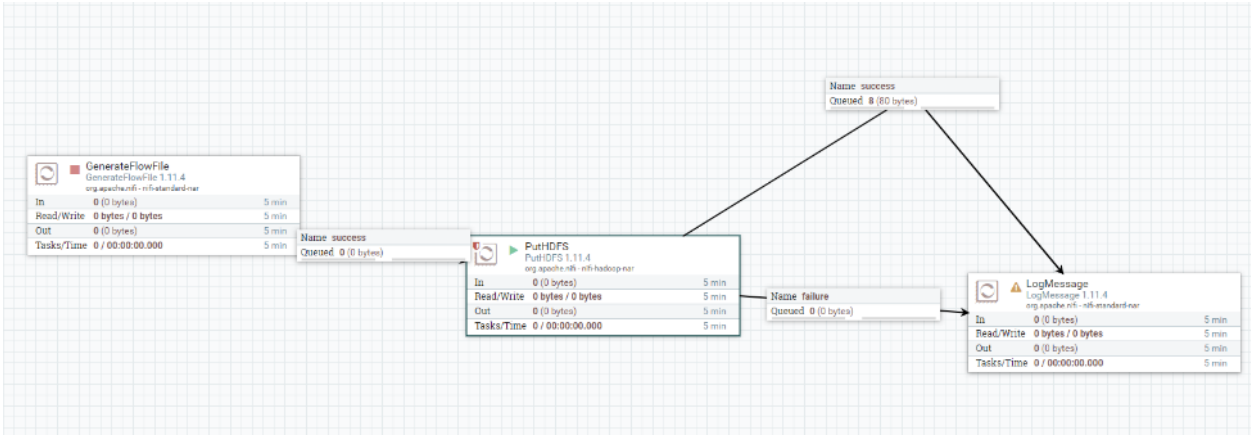

Рисунок 36. Готовый поток для работы с Hadoop

10 Запустите каждый процессор на две минуты.
11 Проверьте через Hue, что в HDFS лежат файлы.

7.4 Создание пользовательских процессоров

SDP DataFlow является платформой с открытым исходным кодом и дает разработчикам возможность добавить свой собственный процессор в библиотеку NiFi. Выполните следующие действия, чтобы создать собственный процессор:

- Загрузите последнюю версию Maven по ссылке https://maven.apache.org/download.cgi
- добавьте переменную среды с именем M2_HOME и задайте значение в качестве установочного каталога maven;
- загрузите Eclipse IDE по ссылке https://www.eclipse.org/downloads/ ;
- откройте командную строку и выполните команду Maven Archetype;

- ищите тип nifi в проектах архетипов;
- выберите org.apache.nifi: проект nifi-процессор-пакет-архетип;
- затем из списка версий выберите последнюю версию;
- введите groupId, artifactId, версию, пакет, artifactBaseName и т. д.;
- будет создан проект с каталогами:

⇒ nifi- <artifactBaseName> -processors;
⇒ nifi- <artifactBaseName> -nar;

- запустите приведенную далее команду в каталоге nifi- <artifactBaseName> -processors, чтобы добавить проект в eclipse;
- откройте затмение и выберите импорт из меню файла;
- затем выберите «Существующие проекты в рабочую область» и добавьте проект из каталога nifi- <artifactBaseName> -processors в eclipse;
- добавьте свой код в публичную функцию void onTrigger (контекст ProcessContext, сеанс ProcessSession), которая запускается, когда запланирован запуск процессора;
- затем упакуйте код в файл NAR, выполнив указанную ниже команду;
- файл NAR будет создан в nifi-nar / целевой каталог;
- скопируйте файл NAR в папку lib SDP DataFlow и перезапустите NiFi;
- после успешного перезапуска NiFi проверьте список процессоров для нового пользовательского процессора;
- на наличие ошибок проверьте файл ./logs/nifi.log.

8 Журналирование и мониторинг

Журналирование и мониторинг осуществляются внешним по отношению к SDP DataFlow сервисами облачной платформы.