GUIDE FOR INTEGRATION METHODS WITH THE DRUPAL RESTFUL API

МЕТОДИ ЗА ИНТЕГРАЦИЯ С DRUPAL RESTFUL API, ПРЕДИМСТВА И НЕДОСТАТЪЦИ

Съдържание

Интегрирането на Hadoop Distributed File System (HDFS) с RESTful API, включително специфични приложения като Drupal API, обхваща различни подходи. Тези методологии улесняват двупосочния поток от данни и команди между HDFS и RESTful услугите, като използват възможностите на двете системи. По-долу е даден изчерпателен списък на такива техники за интегриране:

1. WebHDFS REST API

WebHDFS осигурява RESTful интерфейс към HDFS, позволявайки на потребителските приложения да взаимодействат с HDFS в клъстер чрез HTTP операции. Тази естествена поддръжка може да се използва за интегриране с всеки RESTful API, включително Drupal API, като позволява директни HTTP повиквания от Drupal към HDFS за файлови операции.

Този API позволява управлението на файлове и директории чрез използване на стандартни HTTP операции.

Функционалност: WebHDFS поддържа множество операции, които могат да се извършват по HTTP, като CREATE, ADD, READ, WRITE, DELETE, LIST и SET PERMISSIONS. Например, за да прочетете данни от HDFS, се издава GET заявка, докато PUT се използва за създаване или добавяне на данни.

Архитектура: Когато клиент изпрати HTTP заявка към WebHDFS REST API, NameNode отговаря с пренасочване към DataNode, съдържащ данните, като гарантира, че трафикът на данни не натоварва NameNode.

Предимства:

- Лесна интеграция с уеб приложения, тъй като използва стандартни HTTP протоколи.
- Позволява HDFS да бъде достъпен от голямо разнообразие от клиентски езици и системи, които поддържат HTTP.
- Не изисква сложна настройка, тъй като идва като част от екосистемата на Hadoop.

Недостатъци:

- HTTP може да въведе значителни режийни разходи, особено за големи трансфери на данни.
- WebHDFS REST API може да не предлага същата производителност като родните

HDFS клиенти поради бездържавния характер на HTTP.

- Сигурността трябва да се управлява на HTTP слоя, а собствените механизми за сигурност на Hadoop като Kerberos не винаги са пряко приложими.

2. HttpFS

HttpFS е услуга, която осигурява REST HTTP шлюз, поддържащ всички HDFS файлови операции. Той служи като прокси сървър на HDFS, което позволява взаимодействие от отдалечени хостове. Това е особено полезно, когато интеграцията с RESTful API трябва да се осъществи извън клъстерната мрежа на Hadoop.

HttpFS предоставя подобна функционалност на WebHDFS по отношение на операциите на файловата система, които поддържа. Въпреки това, той действа като прокси сървър на целия API на файловата система Hadoop, поддържайки не само HDFS, но и други файлови системи, интегрирани с Hadoop.

Архитектура: HttpFS работи като самостоятелна услуга, която прокси заявки към клъстера Hadoop. Той е проектиран да служи като шлюз за отдалечени хостове за взаимодействие с HDFS и изисква отделна инсталация и поддръжка.

Предимства:

- Позволява достъп до HDFS извън клъстерната мрежа на Hadoop, улеснявайки отдалечените взаимодействия.
- Като прокси, той може да служи като единна точка за достъп, опростявайки конфигурациите от страна на клиента.
- Поддържа други файлови системи, съвместими с Hadoop, осигурявайки по-обобщено решение.

Недостатъци:

- Допълнителни режийни разходи за поддържане на отделна услуга единствено за HTTP достъп до HDFS.
- Потенциално увеличение на времето за реакция поради допълнителния HTTP прокси слой между клиента и HDFS.
- Мащабира хоризонтално до известна степен, но може да се превърне в пречка, ако не е правилно оразмерена и настроена за среди с високо търсене.

3. Персонализирана обвивка за RESTful услуги

Специално разработената услуга може да действа като междинен слой между HDFS и външните API. Тази обвивка може да превежда RESTful повиквания в HDFS операции, използвайки API на Hadoop Java. Той може също така да удостоверява и пренасочва заявки, идващи от Drupal API към HDFS.

Тази услуга директно взаимодейства с HDFS с помощта на Java API, осигурявайки REST интерфейс, който може да се справи със сложна логика, обработка на входове и да осигури персонализирани изходи. Той е много адаптивен и може да бъде проектиран да излага само необходимите HDFS операции.

Архитектура: Обикновено разположена като самостоятелно уеб приложение, тази услуга може да бъде хоствана на уеб сървър като Apache Tomcat или Jetty. Той действа като посредник между HDFS и външния REST API, обработвайки HTTP заявки, обработвайки ги и извършвайки съответните HDFS операции чрез Java API.

Предимства:

- Може да бъде съобразена с точните спецификации, предлагайки най-голяма гъвкавост по отношение на функционалността.
- Дава възможност за по-добра оптимизация и контрол върху взаимодействието между HDFS и REST клиентите.
- Логиката за сигурност и валидиране на данни може да бъде разработена по поръчка, за да отговори на специфични изисквания.

Недостатъци:

- Разработването и поддръжката могат да бъдат ресурсоемки, изискващи специално време и опит.
- Отговорността за сигурността и оптимизациите на производителността е изцяло на персонализирания екип за разработка.
- Персонализираната разработка означава, че решението е по-малко общо, което потенциално води до трудности в бъдещата мащабируемост или интеграция с други системи.

4. Hadoop конектори / плъгини за Drupal

Специфични модули или плъгини могат да бъдат разработени за Drupal, които използват API на Hadoop за комуникация с HDFS. Тези плъгини могат да се справят с

удостоверяването, четенето, писането и обработката на данни, съхранявани в HDFS.

Тези модули осигуряват Drupal-центрични начини за взаимодействие с HDFS, като съхранение на файлове, импортиране / експортиране на данни и възможност за управление на HDFS данни чрез интерфейса Drupal. Точните характеристики зависят от конкретния използван модул или плъгин.

Архитектура: Конекторите обикновено са PHP модули или библиотеки, които използват Java API на Hadoop или WebHDFS чрез PHP скриптове. Те са интегрирани в архитектурата на Drupal и се управляват като част от екосистемата на Drupal.

Предимства:

- Безпроблемна интеграция с Drupal, осигуряваща познат интерфейс за потребителите на Drupal.
- Намалено време за разработка, тъй като конекторите са предварително изградени за специфични функции.
- Позволява Drupal удостоверяване и разрешения за контрол на достъпа до HDFS данни.

Недостатъци:

- Функционалността е ограничена до това, което модулът или плъгинът предлага; Персонализирането може да бъде ограничено.
- Разчитане на общността или трети страни за поддържане и актуализиране на модулите.
- Потенциални затруднения в производителността, ако плъгините не са оптимизирани за мащабни HDFS операции.

5. Apache Knox

Apache Knox е REST API Gateway за взаимодействие с HDFS и други услуги на Hadoop. Тя осигурява единна точка за достъп и прилага мерки за сигурност. Друпалните системи могат да се интегрират с HDFS чрез Knox, който прокси RESTful API заявки към HDFS и други услуги в клъстера Hadoop.

Knox предлага унифициран слой за RESTful достъп до услугите на Hadoop. Той опростява сигурността на Hadoop за потребители, които могат да взаимодействат с услуги като HDFS, YARN и Hive чрез общ шлюз, използвайки REST API.

Архитектура: Разположен като самостоятелен сървър, Knox седи на ръба на клъстера

Hadoop и предоставя услуги като удостоверяване, оторизация, одит и пренаписване на URL адреси. Той може да прокси заявки към различни услуги на Hadoop, включително HDFS, като по този начин абстрахира сложността на директната интеграция на HDFS.

Предимства:

- Централизирано прилагане на сигурността, което улеснява управлението на контрола на достъпа и политиките за сигурност.
- Опростява взаимодействието на клиентите с услугите на Hadoop, като извлича сложността зад REST API.
- Осигурява по-сигурна входна точка към HDFS, тъй като може да се интегрира с функции за сигурност на корпоративно ниво.

Недостатъци:

- Изисква допълнителна инфраструктура и настройка, добавяйки сложност към средата на Hadoop.
- Като прокси, той може да въведе допълнителна латентност в достъпа до данни.
- Добавеният слой изисква мониторинг и управление, което може да наложи допълнителни оперативни разходи.

6. Интегриране на HCatalog

HCatalog излага HDFS данни чрез REST API. Въпреки че е предимно за достъп до метаданни на Hive, той може да се използва за абстрактни файлови операции на HDFS. Drupal API може да взаимодейства с REST интерфейса на HCatalog, като по този начин косвено работи с HDFS.

REST API на HCatalog, WebHCat (по-рано Templeton), осигурява RESTful интерфейси към HDFS, Hive, Pig и MapReduce. Това улеснява по-лесния достъп до HDFS данни, без да е необходимо да взаимодействате директно с командите на файловата система Hadoop.

Архитектура: WebHCat превежда RESTful заявките в HCatalog команди, които след това взаимодействат с екосистемата на Hadoop. Това осигурява по-високо ниво на абстракция за достъп до HDFS данни, тъй като потребителите могат да работят с таблични структури, а не с файлове и директории.

Предимства:

- По-високото ниво на абстракция го прави удобен за тези, които са запознати със

SQL и табличните структури.

- Интегрира се добре с Hive, което позволява по-сложни операции с данни като заявки и управление на данни.
- Подходящ за работни потоци, които включват компоненти за съхранение на данни.

Недостатъци:

- Не е директен начин за извършване на файлови операции на HDFS; ограничено до абстракцията, предоставена от HCatalog.
- Основно проектиран за използване с Hive, а директната му полезност за сурови HDFS операции може да бъде ограничена.
- Добавя допълнителен слой сложност и потенциални точки на неуспех в пътя за достъп до данни.

7. Използване на OData (Open Data Protocol)

OData е стандартен протокол за REST API, който може да се използва за излагане на HDFS като услуга на OData. Drupal или други RESTful API клиенти могат след това да взаимодействат с HDFS данни, използвайки стандартни OData заявки.

8. Механизми за прехвърляне на файлове

Въпреки че не е директна интеграция на API, услугите на RESTful могат да взаимодействат с HDFS чрез механизми за прехвърляне на файлове като SFTP, където файловете се качват в зона за постановка и след това се преместват в HDFS, използвайки бекенд скриптове или cron задания.

9. Платформи за интеграция на мидълуер

Платформите за мидълуер като Apache Camel, MuleSoft или Talend могат да осигурят мост между RESTful API и HDFS. Тези платформи могат да организират потока от данни, да извършват трансформации и да посредничат между различни протоколи и услуги.

10. Адаптери за поточно предаване на данни

Инструменти като Apache Flume или Apache NiFi могат да бъдат настроени да предават данни от крайните точки на RESTful API директно в HDFS. Те могат също така да обслужват данни от HDFS до консумирането на RESTful услуги, като действат като непрекъснат тръбопровод за данни.

11. HDFS клиенти с поддръжка на REST

Разработване на HDFS клиенти, които могат да бъдат интегрирани в Drupal и да поддържат RESTful операции. Тези клиенти ще превеждат Drupal API повиквания в HDFS действия с помощта на WebHDFS REST API.

Всеки от тези подходи варира по сложност, последици за производителността и ниво на контрол на достъпа. Изборът на конкретен метод за интеграция ще зависи от няколко фактора, като например изискванията за сигурност, обема и скоростта на данните, както и специфичните случаи на използване на взаимодействието HDFS-RESTful API. Освен това, когато става въпрос за чувствителни данни, интегрирането следва да гарантира спазването на стандартите за управление на данните и неприкосновеността на личния живот, което може да повлияе на проектирането и изпълнението на интеграционния слой.

Следващата таблица предоставя обобщение на плюсовете и минусите на всеки подход за интегриране на HDFS с RESTful API, включително Drupal API:

Подход	**Предимства**	**Недостатъци**
WebHDFS REST API	- Native HDFS поддръжка - Директен HTTP достъп до HDFS	- Ограничен от HDFS функции за сигурност - Мрежови разходи за HTTP
HttpFS	- Подходящ за отдалечен достъп - Поддържа всички HDFS операции	- Допълнителен слой може да добави латентност - Изисква отделно управление
Персонализирана обвивка за услуги RESTful	- Гъвкави, персонализирани функции - Може да се оптимизира за конкретни случаи на употреба	- Режийни разходи за разработка и поддръжка - Възможни рискове за сигурността, ако не се прилагат правилно
Hadoop конектори /	- Безпроблемна интеграция с	- Може да не покрива всички

плъгини за Drupal	Drupal - Drupal-ориентирано управление	функции на HDFS - Зависи от общността на приставката за актуализации
Apache Knox	- Функции за сигурност като удостоверяване - Унифициран API шлюз за Hadoop услуги	- Сложна настройка - Допълнителен компонент за управление
Интегриране на HCatalog	- Опростява достъпа до HDFS данни - Добър за операции, свързани с Hive	- Не е директен начин за взаимодействие с HDFS - По-подходящ за Hive от сурови HDFS
Използване на OData	- Стандартен протокол, широко поддържан - Дава възможност за гъвкави заявки	- Режийни разходи на OData сервизен слой - Може да се въведе производителност хит за големи набори от данни
Механизми за прехвърляне на файлове	- Лесен за изпълнение - Работи със съществуваща инфраструктура	- Партида-ориентирани, а не в реално време - Изисква ръчна или скриптова обработка на файлове
Платформи за интеграция на мидълуер	- Мощни функции за интегриране на данни - Може да се справи със сложни потоци от данни	- Може да бъде скъпо - Често изискват специализирани знания
Адаптери за поточно предаване на данни	- Интеграция на данни в реално време - Гъвкава и мащабируема	- Настройката може да бъде сложна - Може да е прекалено много за прости случаи на употреба
HDFS клиенти с	- Съобразени с Drupal	- Разходи за разработка за

поддръжка на REST	взаимодействия - Може да се възползва WebHDFS директно	персонализиран клиент - Нуждае се от актуализации, за да остане в синхрон с промените в HDFS

Всяка от тези стратегии за интеграция предлага уникален набор от предимства и недостатъци, а оптималният избор ще зависи от специфичните изисквания и ограничения на въпросната система. Наложително е да се вземат предвид фактори като очаквания обем на данните, изискванията за латентност, политиките за сигурност и нивото на контрол, необходимо за операциите с данни, когато се избира подходящият подход за интегриране на HDFS с RESTful API.

Литература

1. WebHDFS REST API; https://hadoop.apache.org/docs/r1.0.4/webhdfs.html
2. WebHDFS FileSystem APIs; 2022; https://learn.microsoft.com/en-us/rest/api/datalakestore/webhdfs-filesystem-apis
3. Hadoop HDFS over HTTP - Documentation Sets; https://hadoop.apache.org/docs/stable/hadoop-hdfs-httpfs/index.html
4. Apache Knox Gateway 2.0.x User's Guide; https://knox.apache.org/books/knox-2-0-0/user-guide.html
5. Walker Rowe; What is Apache HCatalog? HCatalog Explained; 2017; https://www.bmc.com/blogs/what-is-apache-hcatalog-hcatalog-explained/