



Система управления данными для специализированного вычислительного комплекса SPD Online Filter

Студент

Научный консультант

П.А. Коршунова

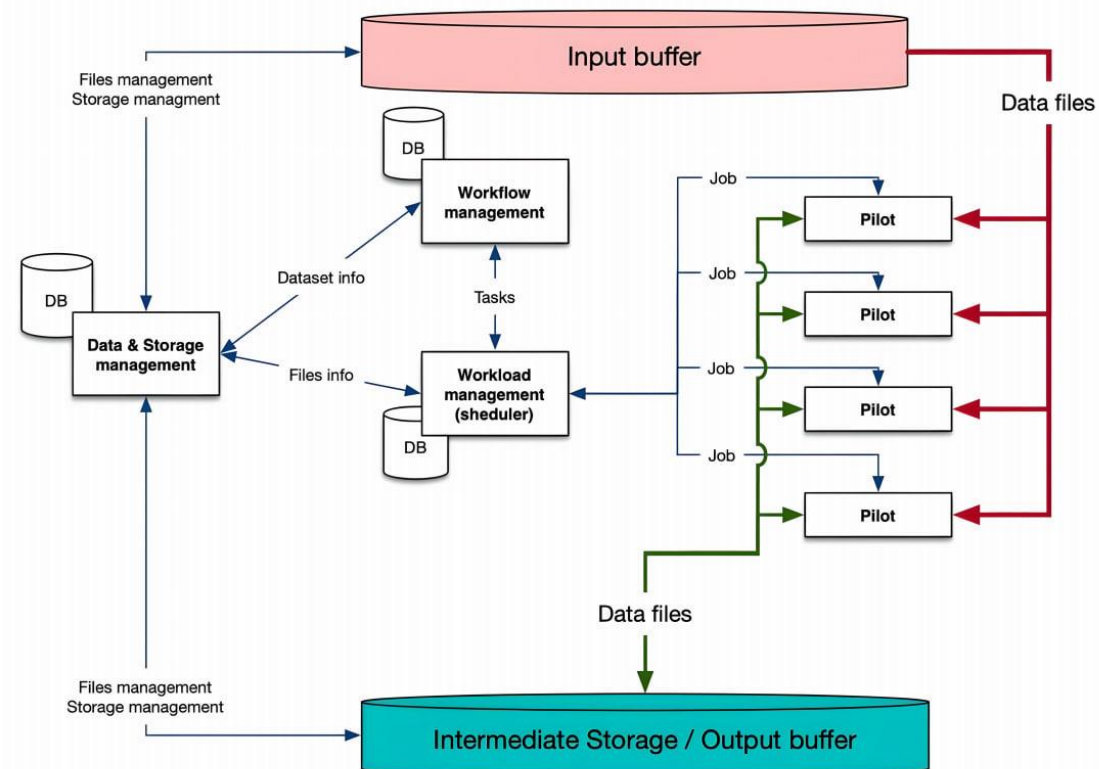
Д.А. Олейник

Москва 2024



SPD Online Filter - это высокопроизводительная вычислительная система для высокопропускной обработки данных

Особенностью **высокопропускной обработки** данных является большой объем данных, как первичных, которые необходимо обработать, так и промежуточных, возникающих в процессе обработки



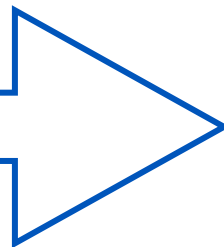
Архитектура SPD Online Filter

Зачем нужна система управления данными?

Большой поток данных с
детектора
~ 200 ПБ/год

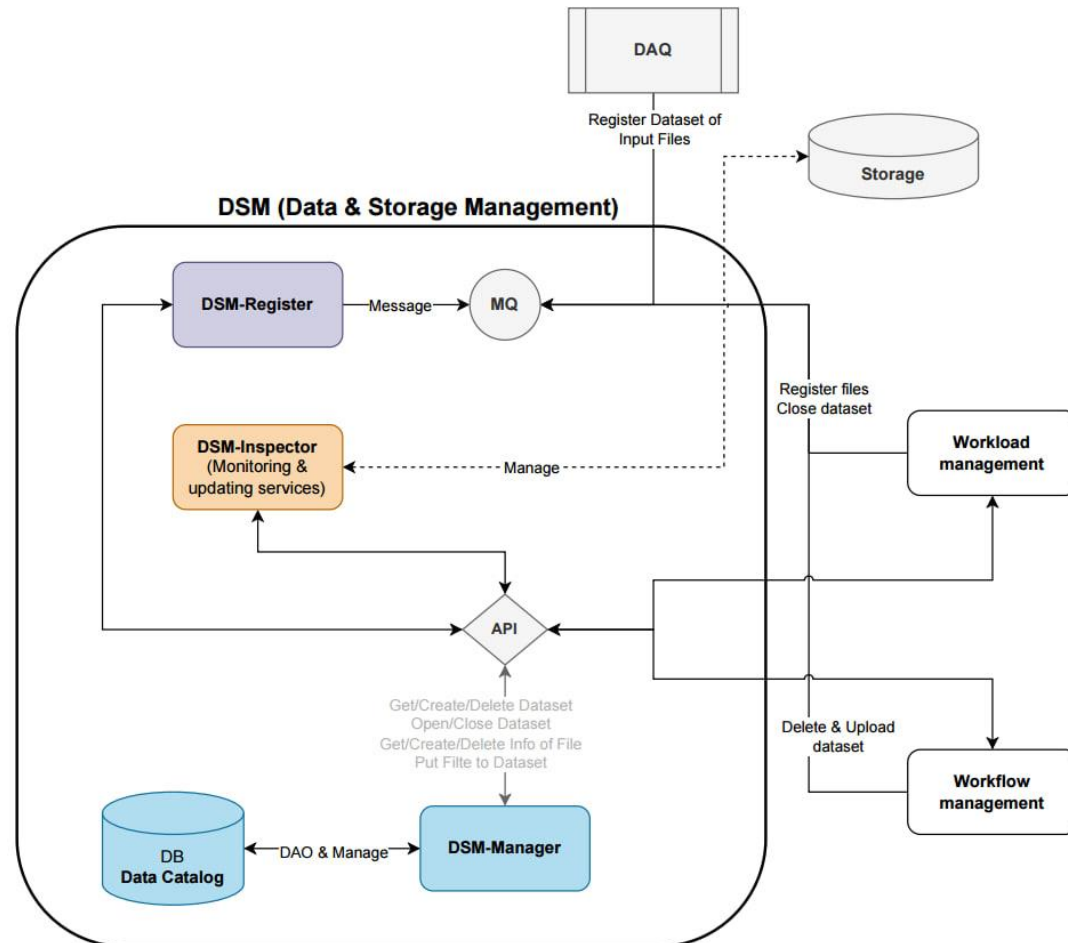


Большой объем
промежуточных данных,
полученных в процессе
обработки



Необходимость
разработки системы
управления данными,
которая должна
обеспечивать контроль
над хранением,
организацией и
целостностью данных

Система управления данными



Архитектура DSM

dsm-register

- сервис, принимающий в асинхронном режиме (через очередь сообщений) заявки на добавление/удаление данных в системе

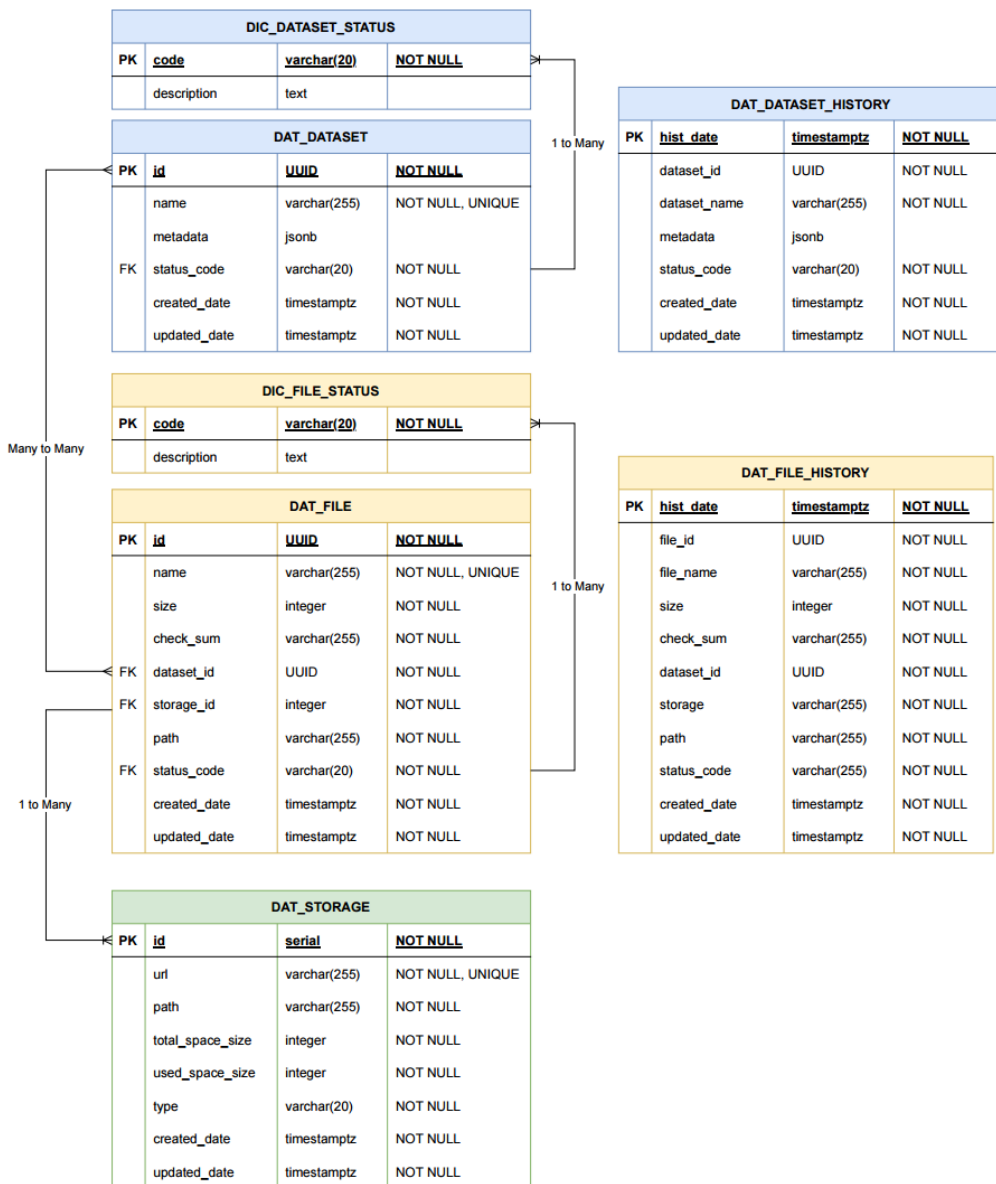
dsm-manager

- сервис, предоставляющий REST API к каталогу данных (размещение данных в каталоге, обращение к каталогу, изменение данных в каталоге)

dsm-inspector

- набор фоновых сервисов для мониторинга и контроля состояния данных в хранилище

Структура БД



Код статуса	Описание
CREATED	Файл добавлен в систему
DAMAGED	Файл повреждён
TO_DELETE	Файл с пометкой «На удалении»
UPLOADING	Файл выгружается
DELETED	Файл удалён из системы

Справочник статусов файлов

Код статуса	Описание
OPEN	Набор открыт
CLOSED	Набор закрыт
FROZEN	Набор временно «заморожен»
TO_UPLOAD	Набор с пометкой «На выгрузку»
UPLOADING	Набор выгружается
TO_DELETE	Набор с пометкой «На удалении»
DELETED	Набор удалён из системы

Справочник статусов датасетов

dsm-register: шлюзы приема и отправки сообщений



Сервис должен слушать очередь сообщений и обрабатывать заявки на добавление/удаление данных в системе.

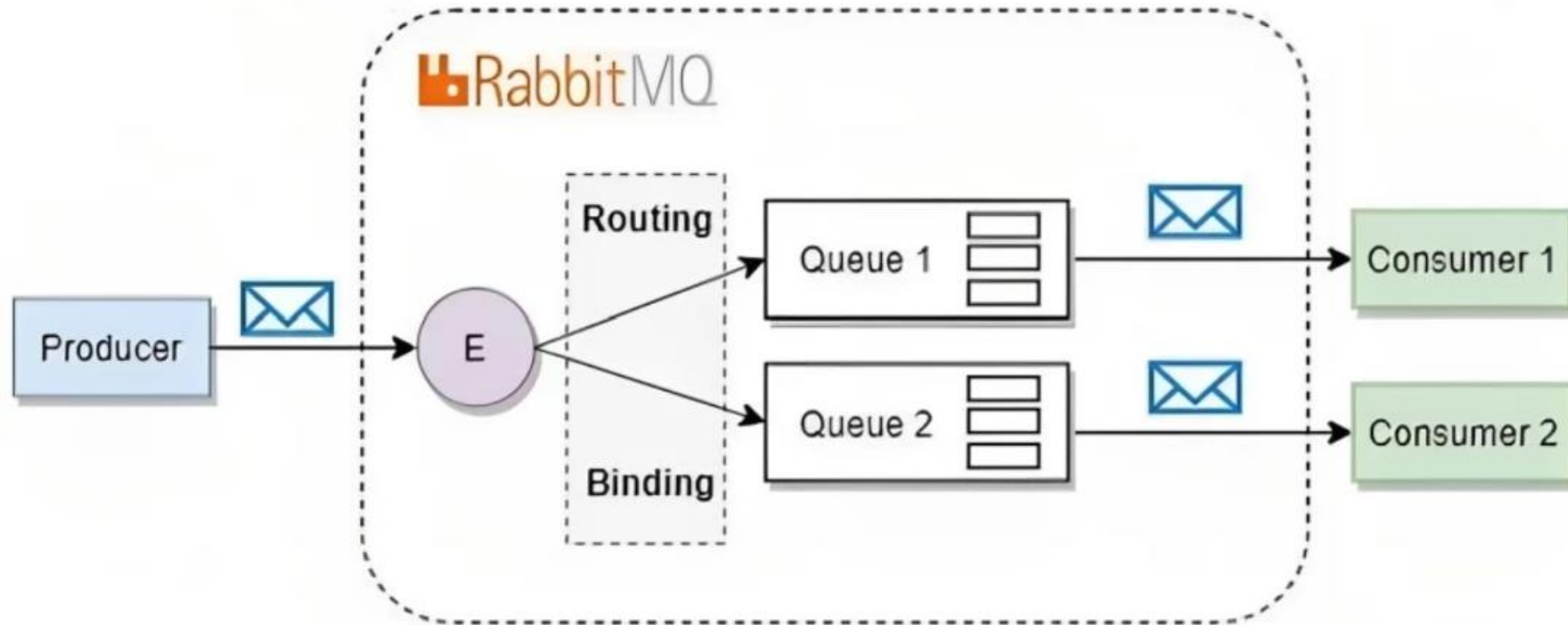
В качестве AMQP-брокера, который осуществляет маршрутизацию и подписку на нужные очереди, используется RabbitMQ.

Exchange	Routing Key	Назначение
	file.input	Приём информации о поступивших файлах на входной буфер
dsm.register (direct)	file.process	Приём информации о новых файлах, полученных в процессе обработки
	dataset.close	Приём заявки на закрытие набора файлов
	dataset.upload	Приём заявки на выгрузку файлов в наборе во внешнее хранилище
	dataset.delete	Приём заявки на удаление файлов в наборе на внутреннем хранилище

Exchange	Routing Key	Назначение
dsm.register (direct)	file.process.reply	Отправка информации о статусе регистрации файлов, полученных в процессе обработки
	dataset.input	Отправка информации о входных датасетах

RabbitMQ

- брокер сообщений, который служит посредником для обмена информацией между различными системами



Основные компоненты RabbitMQ

Сконфигурированные очереди RabbitMQ

Exchange: dsm.register

► Overview

▼ Bindings

This exchange



To	Routing key	Arguments	
dsm.register.dataset.close	dataset.close		Unbind
dsm.register.dataset.delete	dataset.delete		Unbind
dsm.register.dataset.input	dataset.input		Unbind
dsm.register.dataset.upload	dataset.upload		Unbind
dsm.register.file.input	file.input		Unbind
dsm.register.file.process	file.process		Unbind
dsm.register.file.process.reply	file.process.reply		Unbind

Проверка регистрации входных файлов и отправки сообщения с информацией о датасете



Exchange	dsm.register
Routing Key	file.input
Redelivered	o
Properties	
Payload 1406 bytes Encoding: string	{"meta":{"run_number":85},"files":[{"name":"input.test.e847e3c3-e463-455c-bda2-f2b9f937027a.raw","path":"input_18","size":50,"checksum":"d144d83b5405dad60805ac13e52ac2a1"}]}

app-1 | 2024-09-26 11:15:00 INFO: Finish registering files. Dataset ID=ce931b56-86c7-4f11-bf81-f0c823f282fc CLOSED!

Получение сообщения с информацией о входных файлах из очереди dsm.register.file.input

Exchange	dsm.register
Routing Key	dataset.input
Redelivered	•
Properties	
Payload 109 bytes Encoding: string	{"id": "ce931b56-86c7-4f11-bf81-f0c823f282fc", "name": "input.test.cf3d9f65-1341-47aa-823a-3382502a85ab.raw"}

Проверка отправки сообщения с информацией о входном датасете в очередь dsm.register.dataset.input

Проверка успешной регистрации вторичного файла



▼ Publish message

Message will be published to the default exchange with routing key **dsm.register.file.process**, routing it to this queue.

Delivery mode:

Headers: ? = String

Properties: ? =

Payload:

```
{
  "datasetId": "7bac1332-7e23-4a83-a716-4ebd70ffed3d",
  "files": [
    {
      "storageId": "11ae2599-934a-430c-a7d4-5ff7d4ea3602",
      "path": "/home/test_file_process1",
      "size": 100,
      "checksum": "e742438aa8bbf4d034408f07a654308d"
    }
  ]
}
```

Payload encoding:

Publish message

Отправка сообщения в очередь

Exchange	dsm.register
Routing Key	file.process.reply
Redelivered	0
Properties	
Payload	
78 bytes	
Encoding: string	{"status": "SUCCESS", "details": "Registered file = /home/test_file_process1"}

Проверка отправки ответа в очередь

Проверка возникновения ошибки при регистрации файла



Временно приостановим работу сервиса dsm-manager и отправим сообщение в очередь `dsm.register.file.process`. Должна возникнуть ошибка при подключении к сервису. Соответствующее сообщение об ошибке должно быть направлено в очередь `dsm.register.file.process.reply`.

▼ Publish message

Message will be published to the default exchange with routing key **dsm.register.file.process**, routing it to this queue.

Delivery mode:

Headers: ? =

Properties: ? =

Payload:

```
{
  "datasetId": "7bac1332-7e23-4a83-a716-4ebd70ffed3d",
  "files": [
    {
      "storageId": "11ae2599-934a-430c-a7d4-5ff7d4ea3602",
      "path": "/home/test_file_process2",
      "size": 100,
      "checksum": "e742438aa8bbf4d034408f07a654308d"
    }
  ]
}
```

Payload encoding:

Отправка сообщения в очередь

The server reported 0 messages remaining.

Exchange	dsm.register
Routing Key	file.process.reply
Redelivered	○
Properties	
Payload	178 bytes
Encoding: string	
	<pre>{"status": "ERROR", "details": "Error occurs while registering file = /home/test_file_process2. HTTP Exception for http://app:8080/api/v1/file/ - [Errno 111] Connection refused"}</pre>

Проверка отправки ответа в очередь

Сервис должен предоставлять REST API к каталогу данных.

В качестве веб-фреймворка используется асинхронный фреймворк FastAPI

Функциональность	URN	Контракт запроса	Контракт ответа
Внутреннее API			
Управление информацией о наборах в каталоге			
Создать набор	POST /dataset	Информация о наборе	ID набора
Получить набор	GET /dataset/<id>	ID набора	Информация о наборе
Изменить набор	PUT /dataset/<id>	ID набора и изменения	Обновлённая информация о наборе
Удалить набор	DELETE /dataset/<id>	ID набора	-
Управление информацией о файлах в каталоге			
Добавить файл	POST /file	Информация о файле и принадлежности к набору	ID файла
Получить файл	GET /file/<id>	ID файла	Информация о файле
Изменить файл	PUT /file/<id>	ID файла и изменения	Обновлённая информация о файле
Удалить файл	DELETE /file/<id>	ID файла	-
Управление информацией о хранилище			
Добавить хранилище	POST /storage	Информация о хранилище	ID хранилища
Получить хранилище	GET /storage/<id>	ID хранилища	Информация о файле
Изменить хранилище	PUT /storage/<id>	ID хранилища и изменения	Обновлённая информация о хранилище
Удалить хранилище	DELETE /storage/<id>	ID хранилища	-
Получение информации для мониторинга			
Получение списка наборов, к которым принадлежит файл	GET /dataset?fileId=<id>	ID файла	Информация о наборах
Список файлов в определённом статусе	GET /file?status=<>	Статус файла	Информация о файлах
Поиск информации о файле по имени	GET /file?filename=<>	Имя файла	Информация о файлах
Внешнее API			
Взаимодействие с системой управления нагрузкой			
Содержание набора	GET /dataset/<id>/file	ID набора	Общее кол-во файлов в наборе и список с краткой информацией по файлам
Взаимодействие с системой управления процессом			
Список наборов в определенном статусе (с отметкой времени)	GET /dataset?status=<>×tamp=<>	Статус набора и Отметка времени	Информация о наборах
Некоторые API управления наборами (создание выходного набора, получение статуса выходного набора)			

Swagger UI



info		^	
GET	/info/ping	Get Ping	▼
GET	/info/health	Get Health	▼
GET	/info/	Get Database Info	▼
file		^	
GET	/api/v1/file/	Get List	▼
POST	/api/v1/file/	Add	▼
GET	/api/v1/file/{file_id}	Get By Id	▼
PUT	/api/v1/file/{file_id}	Update	▼
DELETE	/api/v1/file/{file_id}	Remove	▼
dataset		^	
GET	/api/v1/dataset/	Get List	▼
POST	/api/v1/dataset/	Add	▼
GET	/api/v1/dataset/{dataset_id}	Get By Id	▼
PUT	/api/v1/dataset/{dataset_id}	Update	▼
DELETE	/api/v1/dataset/{dataset_id}	Remove	▼
storage		^	
GET	/api/v1/storage/	Get List	▼
POST	/api/v1/storage/	Add	▼
GET	/api/v1/storage/{storage_id}	Get By Id	▼
PUT	/api/v1/storage/{storage_id}	Update	▼
DELETE	/api/v1/storage/{storage_id}	Remove	▼

Сервис состоит из набора **фоновых** задач:

- Удаление файлов на хранилищах
 - Контроль выгрузки файлов
 - Проверка целостности файлов
 - Контроль использования хранилища
-
- Реализация в виде **демона**

Алгоритм работы:

1. Получаем список наборов в статусе **CLOSED**
2. По каждому набору получаем его **содержимое** (информацию о файлах)
3. Формируем **список уникальных файлов** (по ID)
4. По каждому файлу:
 - проверяем наличие файла на хранилище
 - проверяем размер файла
 - проверяем контрольную сумму
 - если какая-то из проверок не пройдена, устанавливаем файлу статус **DAMAGED**
5. Если какой-либо файл в наборе **поврежден**, устанавливаем набору статус **FROZEN**

Планы

dsm-inspector:

Реализовать фоновые сервисы для:

- Удаления файлов на хранилищах
- Контроля выгрузки файлов
- Контроля использования хранилища

dsm-register:

Реализовать обработку сообщений из очередей:

- `dsm.register.dataset.closed`
- `dsm.register.dataset.upload`
- `dsm.register.dataset.delete`



**Спасибо за
внимание!!!**

