



Система управления данными для специализированного вычислительного комплекса SPD Online Filter

Студент

Научный руководитель

П.А. Коршунова

Д.А. Олейник

Москва 2024



Эксперимент SPD

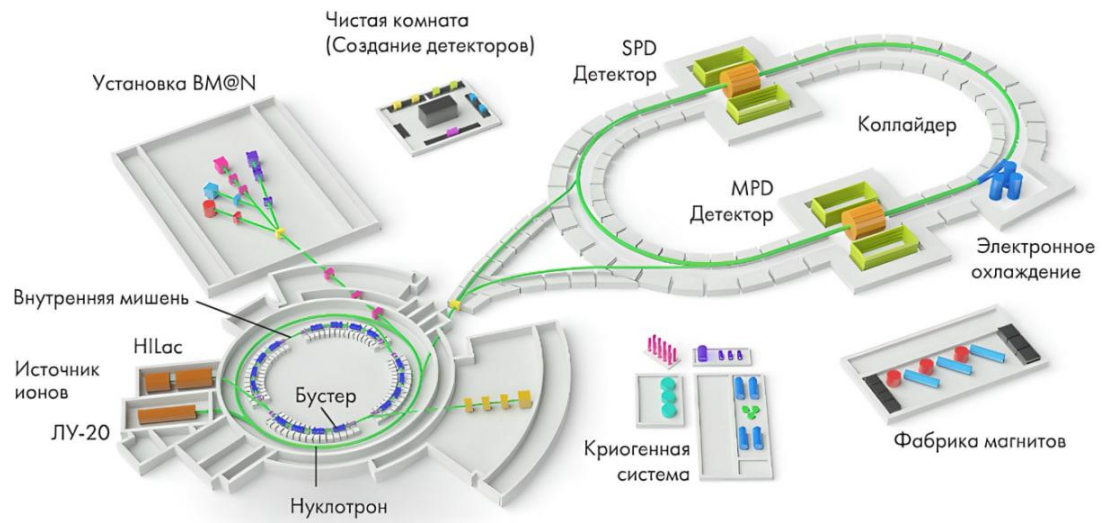


Схема ускорительного комплекса NICA

Детектор SPD будет использоваться для изучения спиновой структуры протона и дейтрона и других связанных со спином явлений

Основная цель эксперимента SPD – всестороннее изучение неполяризованной и поляризованной глюонной составляющей нуклона

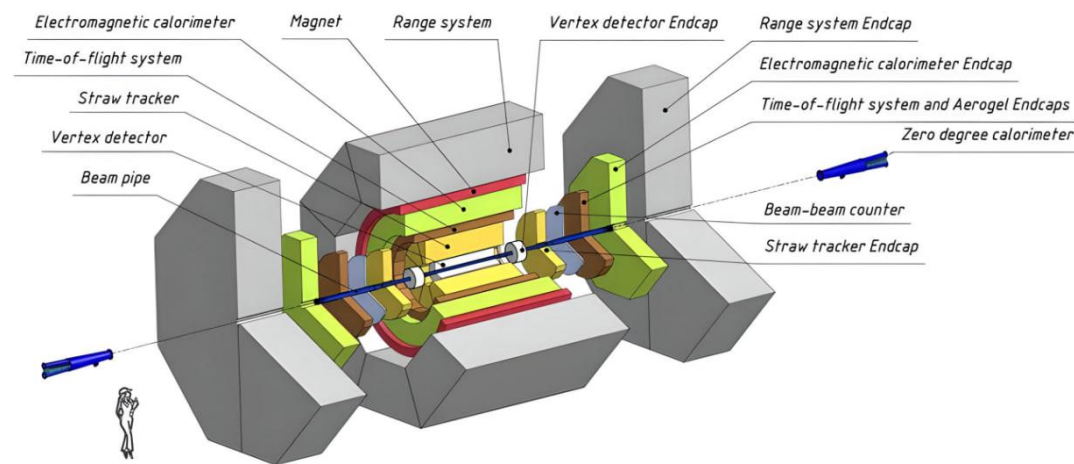


Схема детектора SPD

Зачем нам SPD Online Filter?

Сложность и широта изучаемых процессов



Невозможно использовать классический подход с использованием триггерной системы



Большой поток данных ~ 200 ПБ/год



Проблемы с обработкой и хранением данных

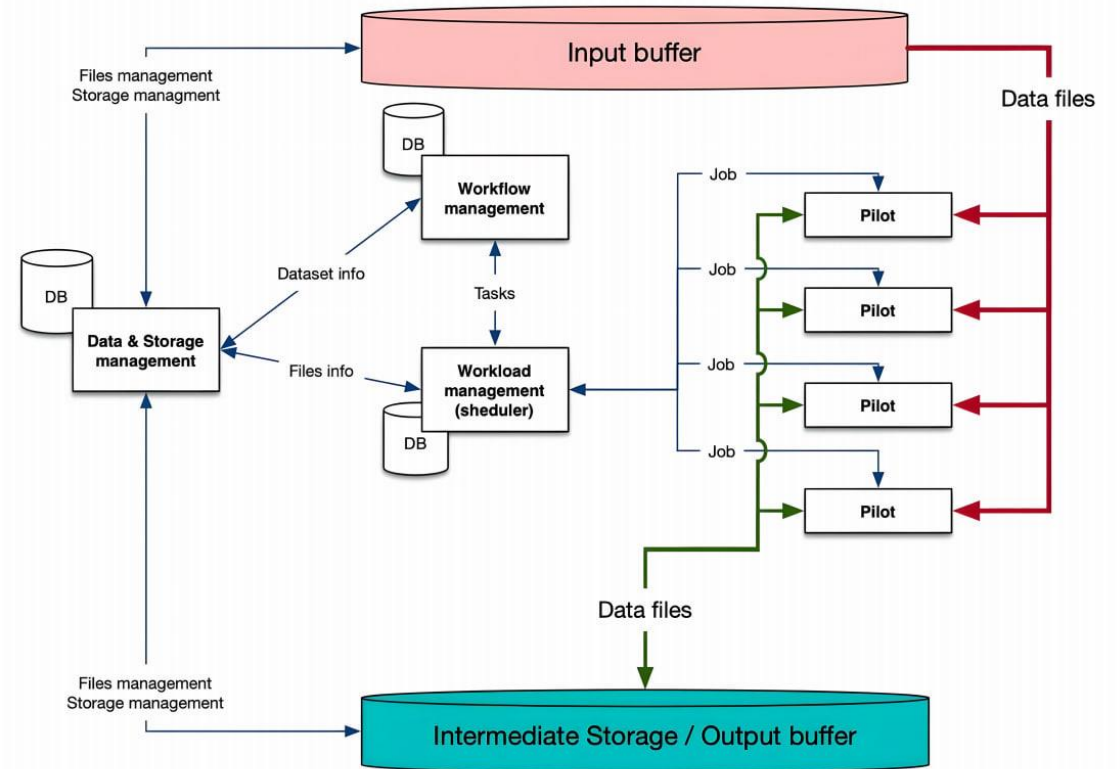


Необходимость разработки специализированной вычислительной системы с целью сокращения объема данных для долговременного хранения

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

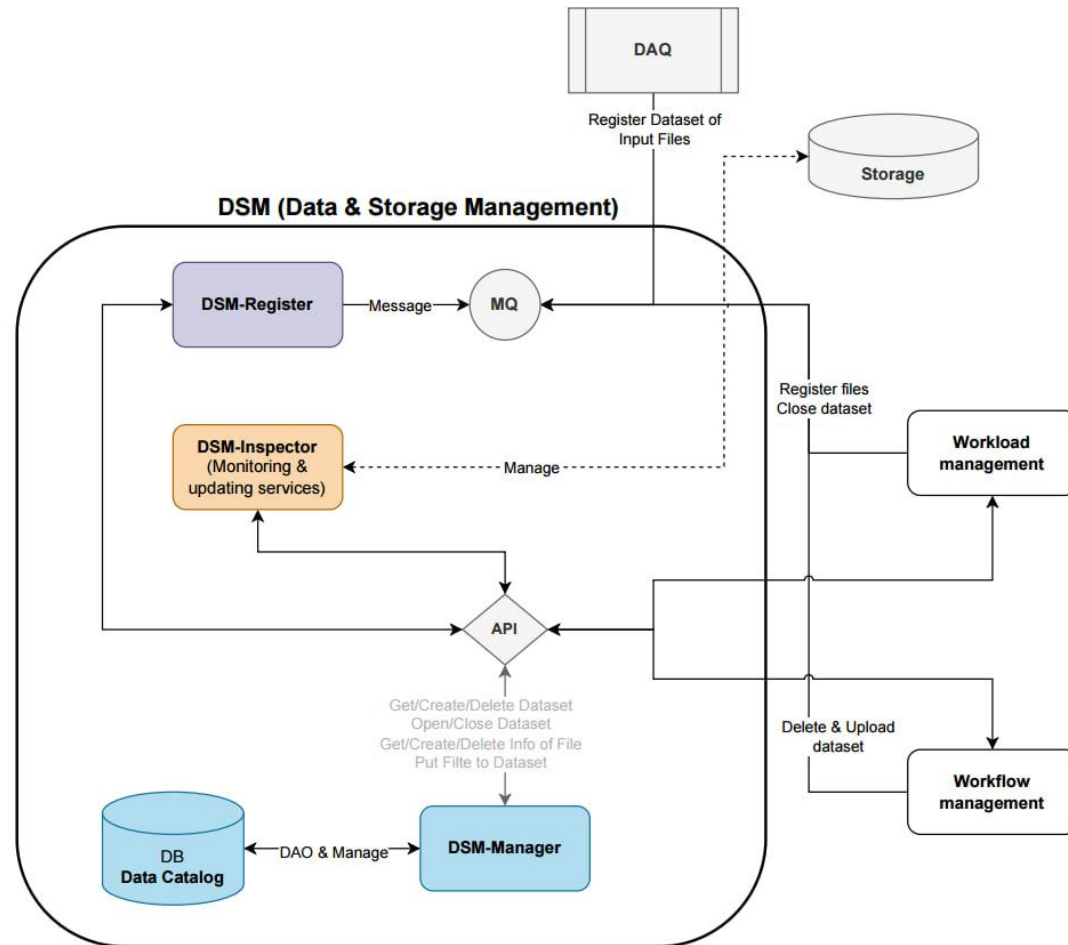
Требования к SPD Online Filter

- декодировать исходные данные, полученные от DAQ
- выделить события
- отфильтровать «скучные» события (в соответствии с заданными физическими критериями)
- упорядочить выходные данные, объединить события в файлы, файлы в наборы данных



Архитектура SPD Online Filter

Система управления данными



Архитектура DSM

dsm-register

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

dsm-manager

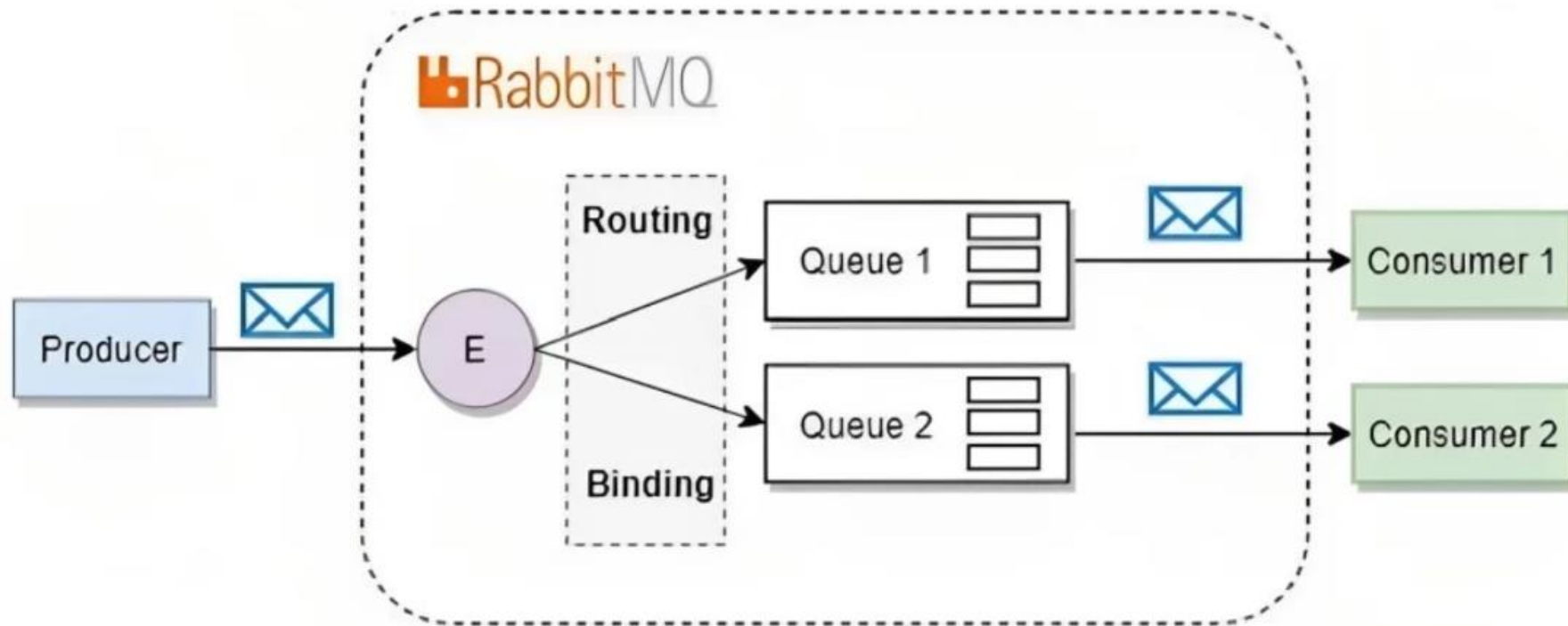
- сервис, предоставляющий REST API к каталогу данных (размещение данных в каталоге, обращение к каталогу, изменение данных в каталоге)

dsm-inspector

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

RabbitMQ

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



Основные компоненты RabbitMQ

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

Доработка взаимодействия `dsm-register` с системой управления нагрузкой



Было необходимо:

- ✓ Создать новый consumer (получатель) для приема сообщений из очереди `dsm.register.file.process`, предназначенной для информации о новых файлах, полученных в процессе обработки
- ✓ Написать обработчик полученных сообщений, а именно осуществить регистрацию промежуточных файлов в системе
- ✓ Создать новую очередь `dsm.register.file.process.reply` для отправки в нее ответов о статусе регистрации промежуточных файлов
- ✓ Настроить механизм отправки сообщений в данную очередь

Проверка успешной регистрации 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	{"status": "SUCCESS", "details": "Registered file = /home/test_file_process1"}
Encoding: string	78 bytes

Проверка отправки ответа в очередь

Parameters Cancel

Name	Description
dataset_id <small>(query)</small>	7bac1332-7e23-4a83-a716-4ebd70ffed3d

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/v1/file/?dataset_id=7bac1332-7e23-4a83-a716-4ebd70ffed3d' \
-M 'accept: application/json'
```

Request URL

```
http://localhost:8080/api/v1/file/?dataset_id=7bac1332-7e23-4a83-a716-4ebd70ffed3d
```

Server response

Code	Details
200	<pre>Response body { { "name": "test_file_process", "path": "/home/", "storageId": "11ae2599-934a-430c-a7d4-5ff7d4ea3602", "size": 100, "checksum": "e742438aa8bbf4d034408f07a654308d", "statusCode": "CREATED", "id": "a9ff3db8-ac46-4df4-8897-571abee88219" }, { "name": "test_file_process1", "path": "/home/", "storageId": "11ae2599-934a-430c-a7d4-5ff7d4ea3602", "size": 100, "checksum": "e742438aa8bbf4d034408f07a654308d", "statusCode": "CREATED", "id": "e3870540-74cb-4fac-95f8-0464a2581b1c" } }</pre>

Download

Проверка регистрации файла в нужном наборе

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

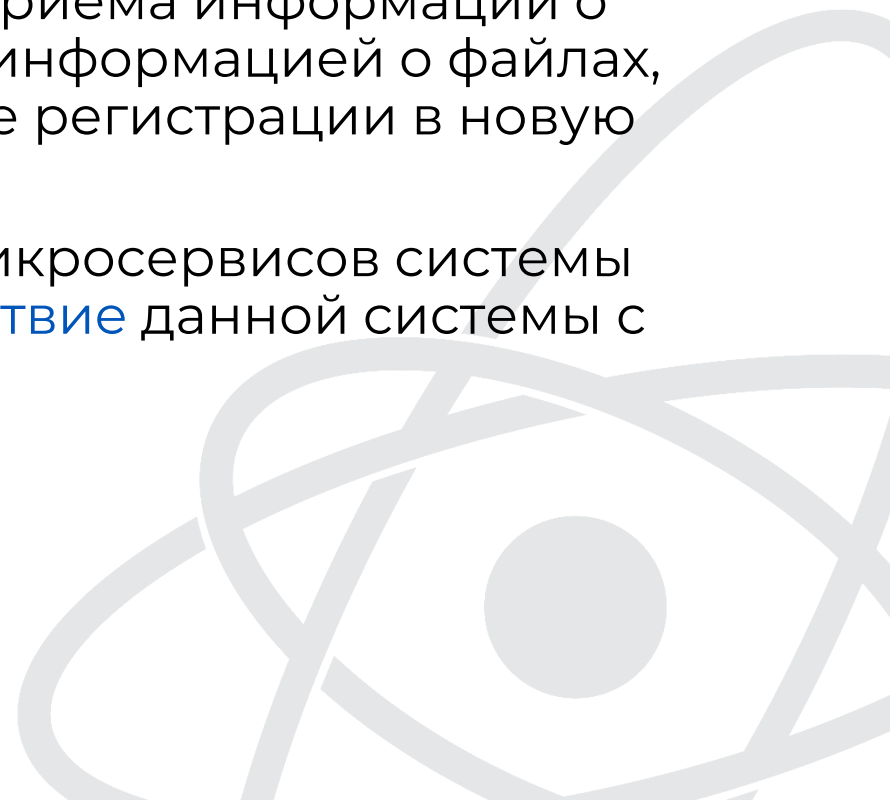
Заключение

В работе было изучено, что такое эксперимент SPD и система Online Filter, а также зачем нужна и как устроена система управления данными.

Также был подробно разобран функционал сервиса dsm-register, который принимает заявки на добавление/удаление данных в системе.

Для данного сервиса была реализована логика обработки сообщений из очереди dsm.register.file.process (предназначена для приема информации о вторичных файлах). Сюда входит прием сообщений с информацией о файлах, их регистрация в системе и отправка ответов о статусе регистрации в новую очередь dsm.register.file.process.reply.

В дальнейшем планируется закончить реализацию микросервисов системы управления данными, а также доработать взаимодействие данной системы с системами управления процессами и нагрузкой.



Дополнительные слады



Промежуточное ПО

Система управления данными

- регистрация, каталогизация, контроль целостности

Система управления процессами обработки

- формирование и контроль исполнения этапов обработки данных

Система управления нагрузкой

- реализация этапов обработки (формирование задач, отправка задач пилотам)

Pilot – приложение, работающее на вычислительном узле и исполняющее задачи

