

Разработка виртуализированного, распределенного хранилища данных

Ибатуллин Е.В. под руководством Мойбенко А.Н.

Saint Petersburg State University

10.04.2025

Что такое BigData?

Новая проблематика

Источники данных

- Датчики
- Iot

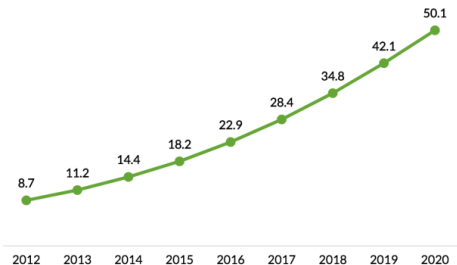


Рис. 1: Число устройств iot в миллиардах

Потребители данных

- ◆ Машинное обучение
- ◆ Визуальный анализ

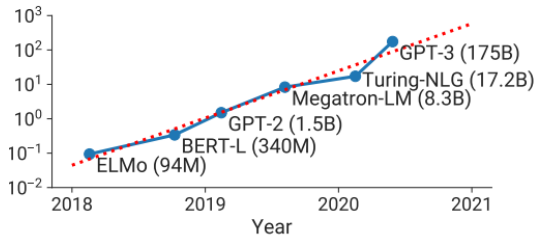


Рис. 2: Кол-во параметров в миллиардах

Тема работы

Исследование развертывания систем хранения на кластерах из контейнеров и дальнейшая организация иерархической структуры этих кластеров.

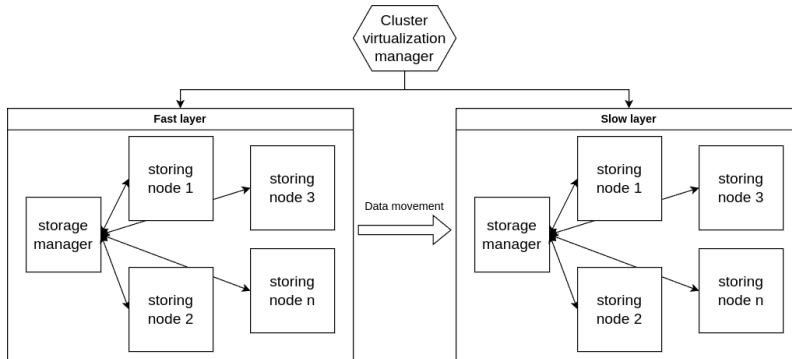


Рис. 3

1. Предложить методы использования виртуализации для систем хранения.
2. Разработать средство интеграции для построения иерархии.
3. Собрать прототип, демонстрирующий использование технологий.

Что работает сейчас?

- Docker образ для сбора контейнера dcache.
- Docker compose для сбора локального кластера в контейнерах.
- **Два** кластера kubernetes для оркестрации виртуальных кластеров на **четырёх** машинах.
- **Две** системы dcache для организации двух уровней хранения на этих кластерах.
- Сервис перемещения данных между уровнями на отдельной машине.

Dcache + Docker + Docker compose

В среде docker compose (локально) производится запуск трех узлов dcache:

1. Центральный узел с базой данных
2. Узел доступа
3. Узел хранения информации

Данное решение позволяет быстро развернуть логически распределенную систему. Это может быть **полезно** для **отладки** и **изучения** системы.

Используется оркестратор контейнеров kubernetes для организации среды запуска контейнеров на нескольких машинах. Затем разворачивается система dcache в виде одного сервиса доступа (door) и сервиса хранения (pool). Также требуется развернуть дополнительные приложения:

- postgresql
- zookeeper
- kafka

Данное решение является виртуальным кластером системы хранения или **слоем** иерархической архитектуры.

Сервис перемещения

Сервис перемещения представляет собой надстройку, которая оперирует метаданными и работает с операциям **чтения, записи и удаления**.

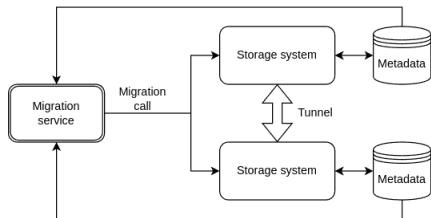
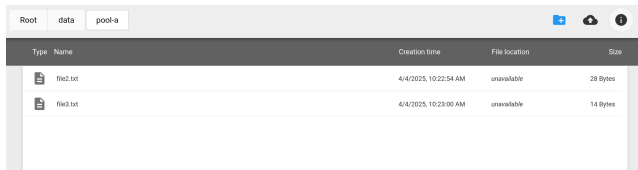


Рис. 4: Сервис перемещения

Пример работы (Dcache on 10.220.18.166:30003)

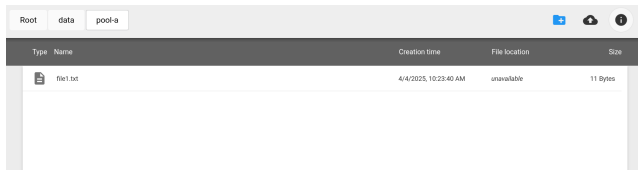
В прототипе используются две системы dcache, которые не имеют способов коммуникации между собой:



The screenshot shows the web interface of Dcache 1. At the top, there are three tabs: 'Root', 'data', and 'pool-a'. To the right of the tabs are three icons: a blue plus sign, a cloud, and a circle with an 'i'. Below the tabs is a table with the following columns: 'Type', 'Name', 'Creation time', 'File location', and 'Size'. The table contains two rows of data:

Type	Name	Creation time	File location	Size
	file2.txt	4/4/2025, 10:22:54 AM	unavailable	28 Bytes
	file3.txt	4/4/2025, 10:23:00 AM	unavailable	14 Bytes

Рис. 5: Dcache 1



The screenshot shows the web interface of Dcache 2. At the top, there are three tabs: 'Root', 'data', and 'pool-a'. To the right of the tabs are three icons: a blue plus sign, a cloud, and a circle with an 'i'. Below the tabs is a table with the following columns: 'Type', 'Name', 'Creation time', 'File location', and 'Size'. The table contains one row of data:

Type	Name	Creation time	File location	Size
	file1.txt	4/4/2025, 10:23:40 AM	unavailable	11 Bytes

Рис. 6: Dcache 2

Пример работы (сервис)

```
• .venvmchin@m15:~/Documents/projects/jinx$ curl -X POST -F "file=@file1.txt"
http://localhost:6000/upload
{
  "file_id": 1,
  "message": "File uploaded successfully"
}
• .venvmchin@m15:~/Documents/projects/jinx$ curl -X POST -F "file=@file2.txt"
http://localhost:6000/upload
{
  "file_id": 2,
  "message": "File uploaded successfully"
}
• .venvmchin@m15:~/Documents/projects/jinx$ curl -X POST -F "file=@file3.txt"
http://localhost:6000/upload
{
  "file_id": 3,
  "message": "File uploaded successfully"
}
• .venvmchin@m15:~/Documents/projects/jinx$ curl -X POST -H "Content-Type: app
lication/json" -d '{"file_id": 1, "target_storage": "storage-2"}' http://loc
alhost:6000/move
{
  "message": "File moved successfully",
  "path": "file1.txt"
}
```

Рис. 7: Запись файлов и перемещение

```
• .venvmchin@m15:~/Documents/projects/jinx$ curl -X GET http://localhost:6000/files
[
  {
    "id": 3,
    "path": "file3.txt",
    "size": 14,
    "storage": "storage-1",
    "upload_time": "2025-04-04T07:23:00.894738"
  },
  {
    "id": 2,
    "path": "file2.txt",
    "size": 28,
    "storage": "storage-1",
    "upload_time": "2025-04-04T07:22:55.112442"
  },
  {
    "id": 1,
    "path": "file1.txt",
    "size": 11,
    "storage": "storage-2",
    "upload_time": "2025-04-04T07:22:49.657377"
  }
]
```

Рис. 8: Чтение метаданных

Почему такая архитектура

Преимущества контейнеризации

- ▶ Скорость работы.
- ▶ Защита от ошибок.
- ▶ Стандартность.
- ▶ Возможности транспортировки среды.

Преимущества иерархического подхода

- ▶ Гибкое использование ресурсов
- ▶ Значительная масштабируемость
- ▶ Независимость модулей (слоев)

Улучшение системы

- Подготовка к автоматическому развертыванию
- Интеграция новых систем хранения (eos)
- Добавление новых протоколов в сервис перемещения данных
- Разработка алгоритмов перемещения данных

Следующие этапы

- ◇ Автоматическое предоставление виртуальных кластеров/систем хранения
- ◇ Интеграция различных областей данных (архитектура data-mesh)

Основные моменты:

- Новые требования в области BigData.
- Актуальные методы в облачных технологиях.
- Иерархический подход и его реализация.
- Прототип и его преимущества.
- Переход к системам data-mesh.