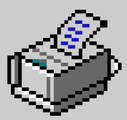






OSTORAGE JUICEFS

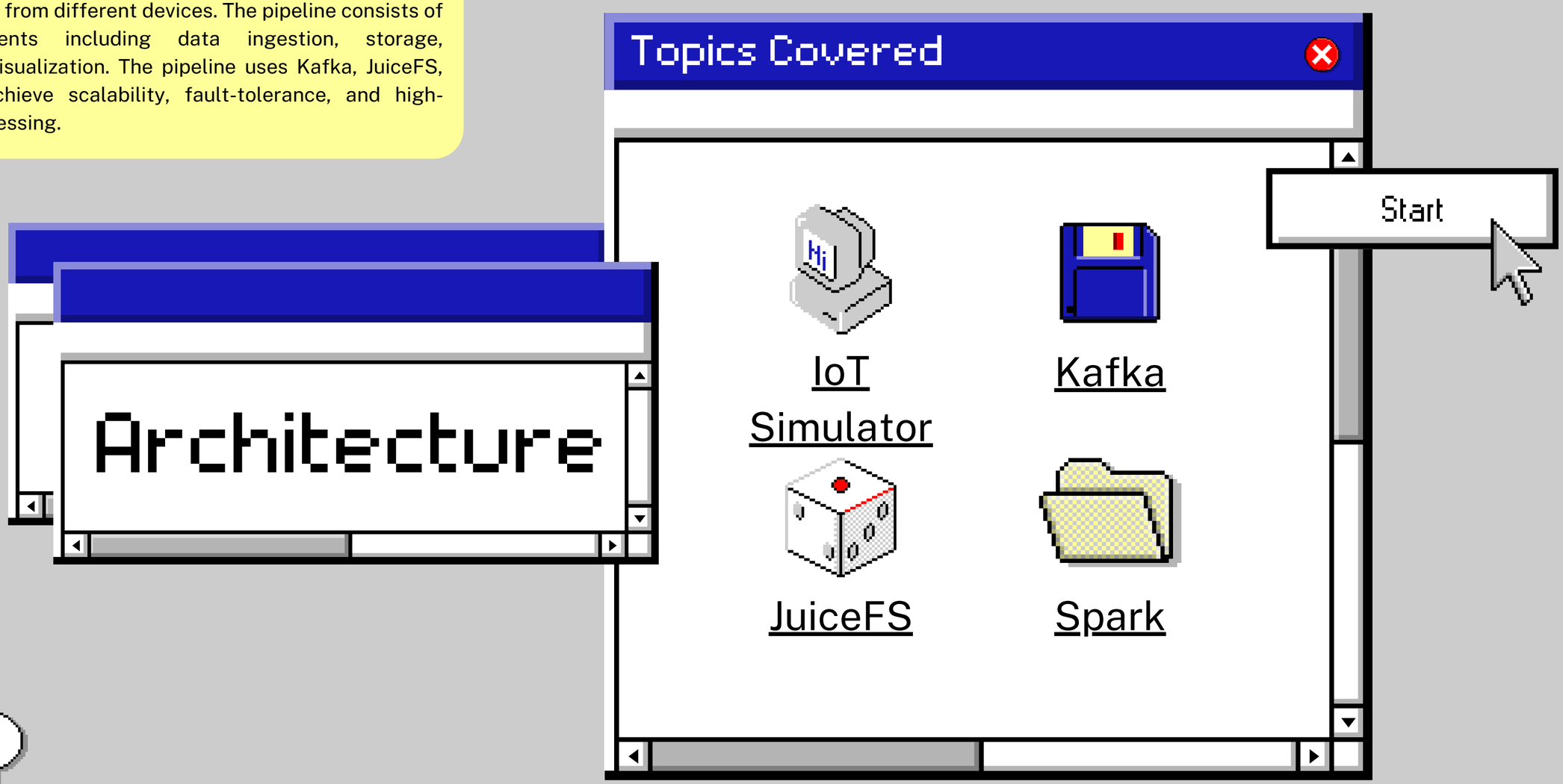
Evaluation of a distributed storage technology based on JuiceFS



11:11AM

This project aims to create a data pipeline for processing and analyzing IoT data from different devices. The pipeline consists of various components including data ingestion, storage, processing, and visualization. The pipeline uses Kafka, JuiceFS, and Spark to achieve scalability, fault-tolerance, and high-performance processing.

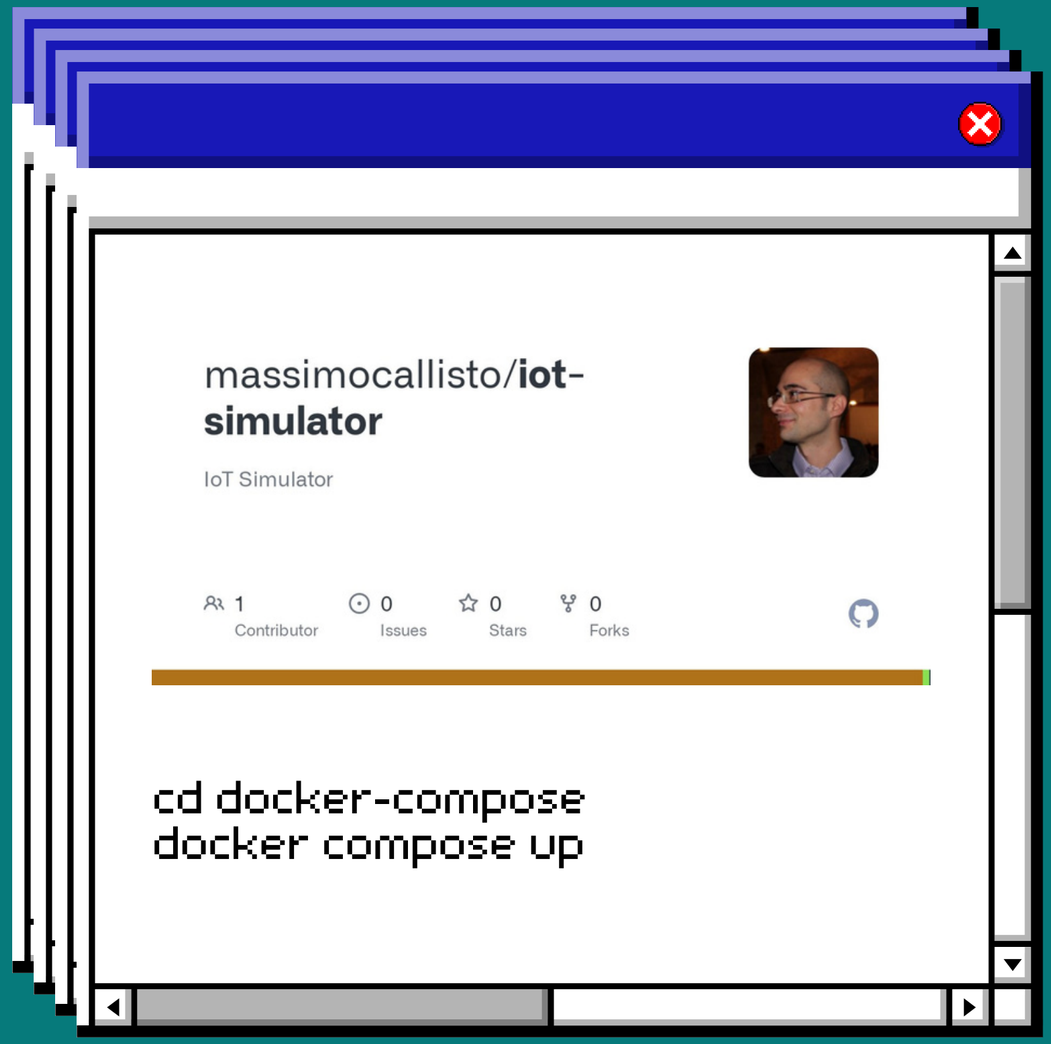


IoT Simulator

Back to Architecture Page

IoT Simulator

used to simulate IoT devices that send JSON messages to the Kafka broker.



Back to Architecture Page

Apache Kafka

Back to IoT Page

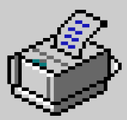
Apache Kafka

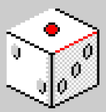
receives the JSON messages from the IoT devices and stores them in topics.

Back to IoT Page

Apache Kafka

Whether you are looking to process data from IoT devices, to handle data pipelines in big data processing applications, or to provide real-time data feeds for data visualization and analysis, Apache Kafka is a highly effective solution that is well-suited to meet your needs

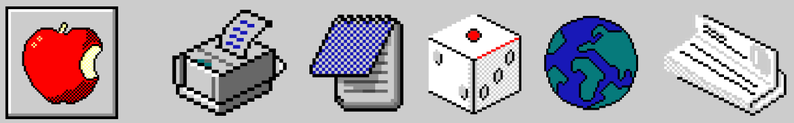




Back to Iot Page

MQTT Protocol

Apache Kafka provides native support for MQTT, allowing MQTT data to be processed and stored using the Kafka platform.



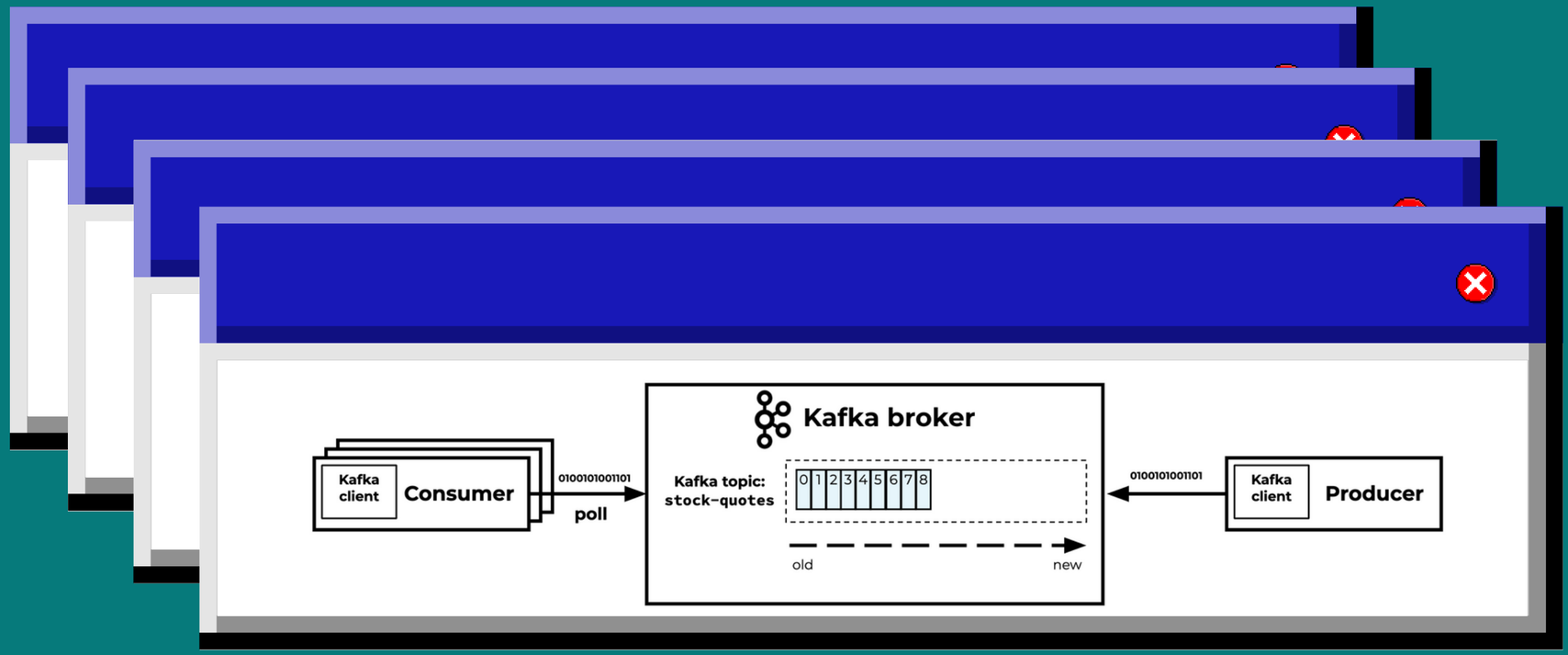
Back to Iot Page

Persistor Prototype

Back to Kafka Page

Persistor Prototype

composed of a producer and consumer that are connected to Kafka.



Back to Kafka Page

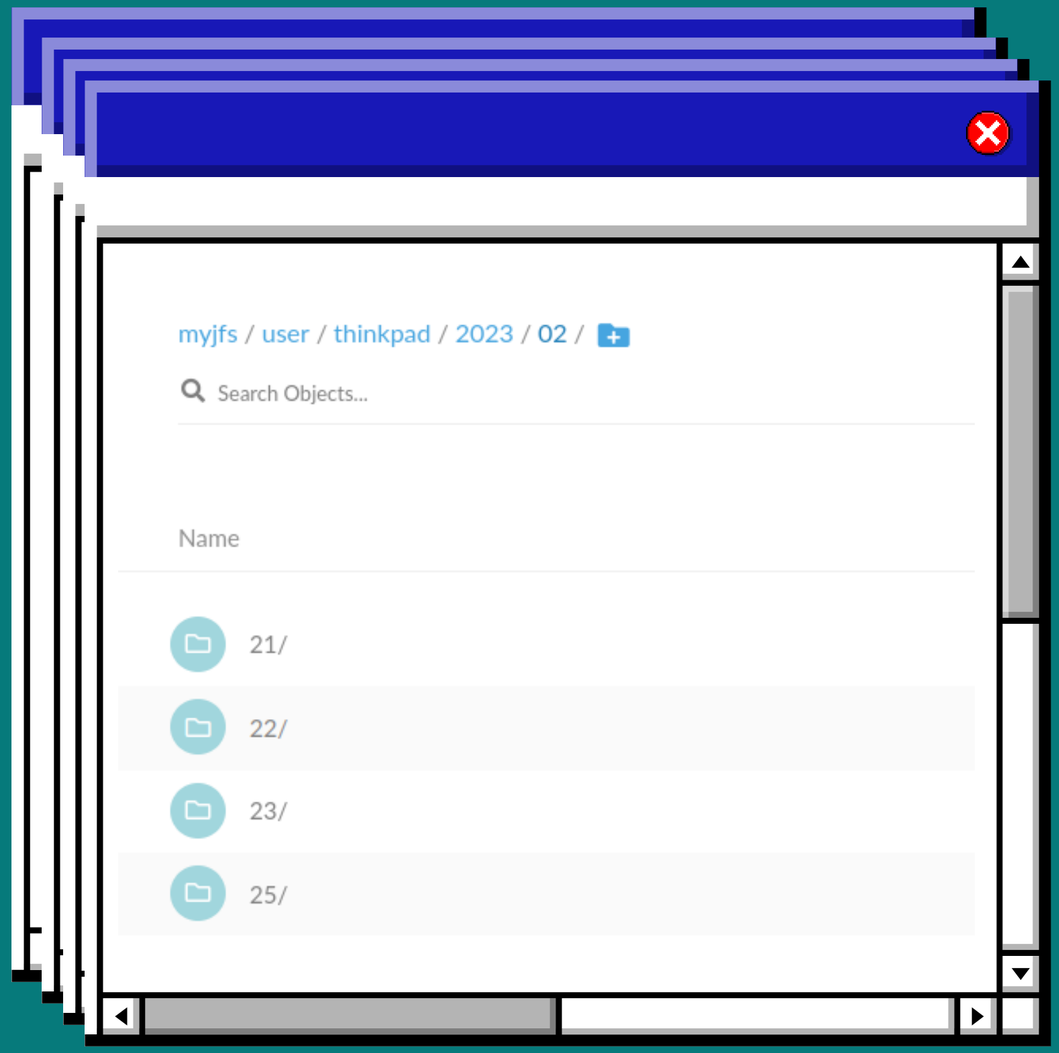
consumer stores the IoT json messages received from Kafka in JuiceFS

producer sends an ack with the UUID of the iot json file received

consume the messages from the ack topic and create a log file

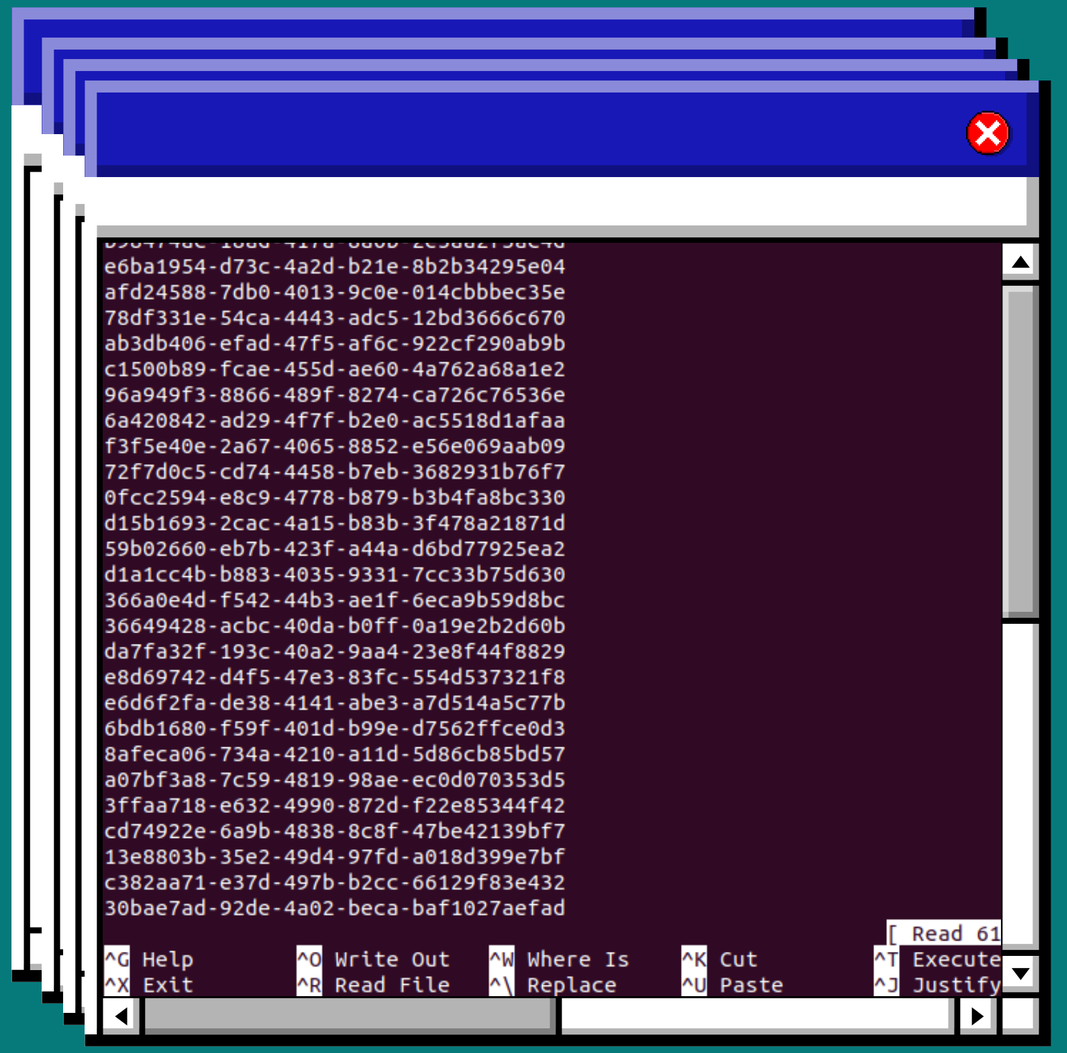
Files Directory

the Consumer creates a path based on the year, month and day.



Back to Kafka Page

Log Files



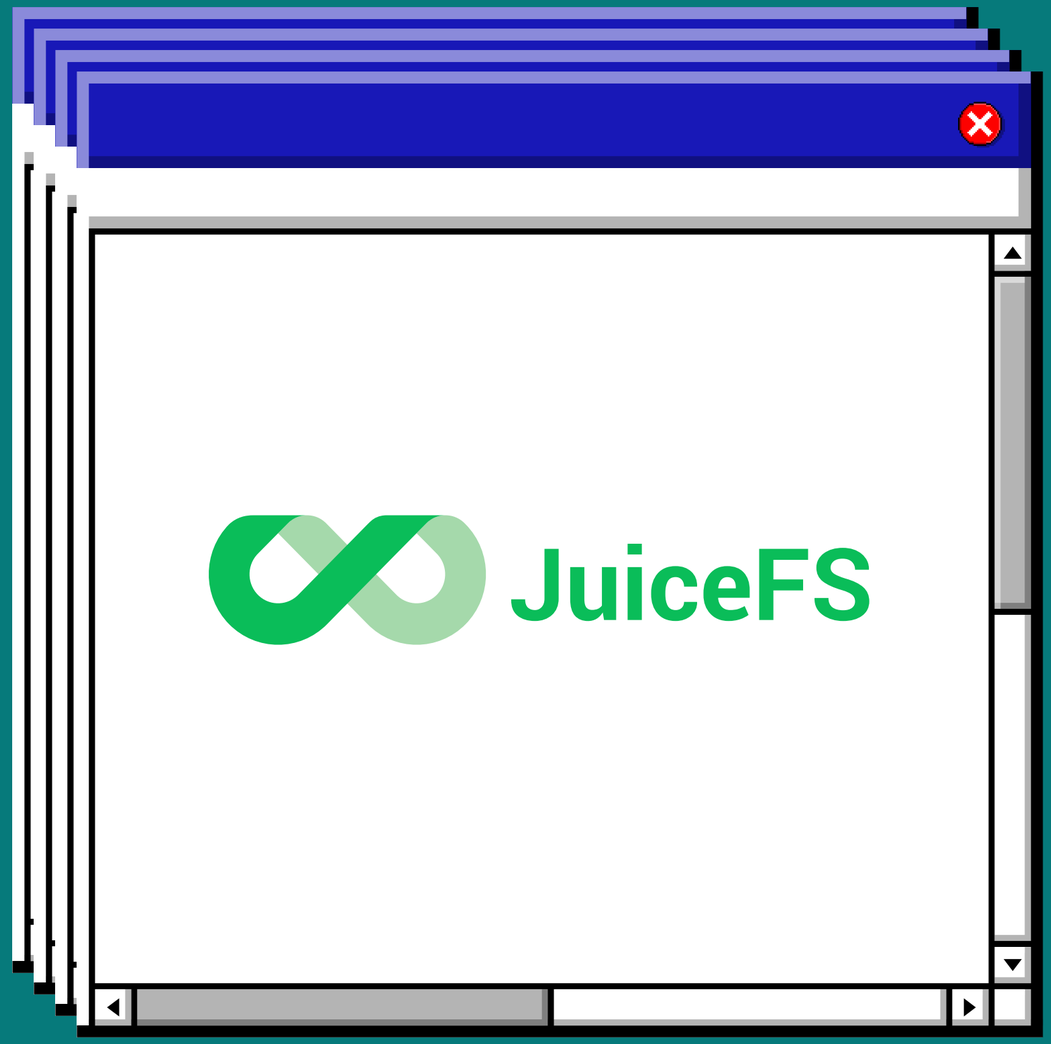
if a message is sent on the topic ack, the Consumer calls a method that generates a log file with all the names of the saved file.

Back to Kafka Page


JuiceFS
Back to Persistor Page

JuiceFS

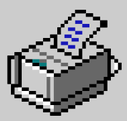
a distributed file system that provides consistent and scalable storage for the IoT data.



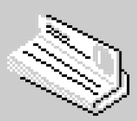
Back to Persistor Page

JuiceFS and IoT

One of the key benefits of JuiceFS is its ability to scale horizontally, allowing it to accommodate growing amounts of data without sacrificing performance. This makes it an ideal choice for use in IoT environments, where the amount of data generated by devices can be substantial.

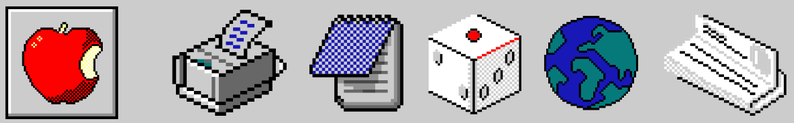




Back to IoT Page

JuiceFS and Persistor Prototype

In the Persistor Prototype, JuiceFS is used to store data that is received from Kafka and processed by the Java consumer.



Back to Persistor Page

where all file I/O happens

data will be split into chunks and stored in object storage

high performance metadata storage uses a multi-engine design

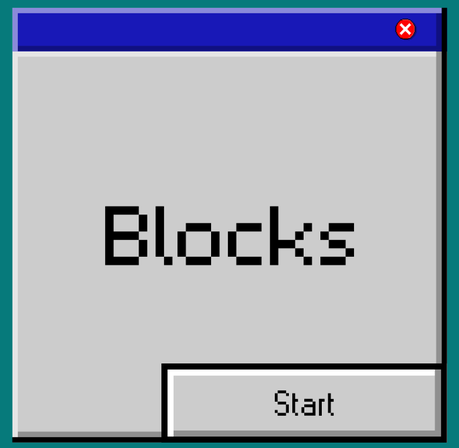
Storage

each file is divided

each with a size limit of 64 mb.

each chunk is divided

serve to optimize different types of write operations.



each slice is divided

of size 4 MiB by default.

Back to JuiceFS Page

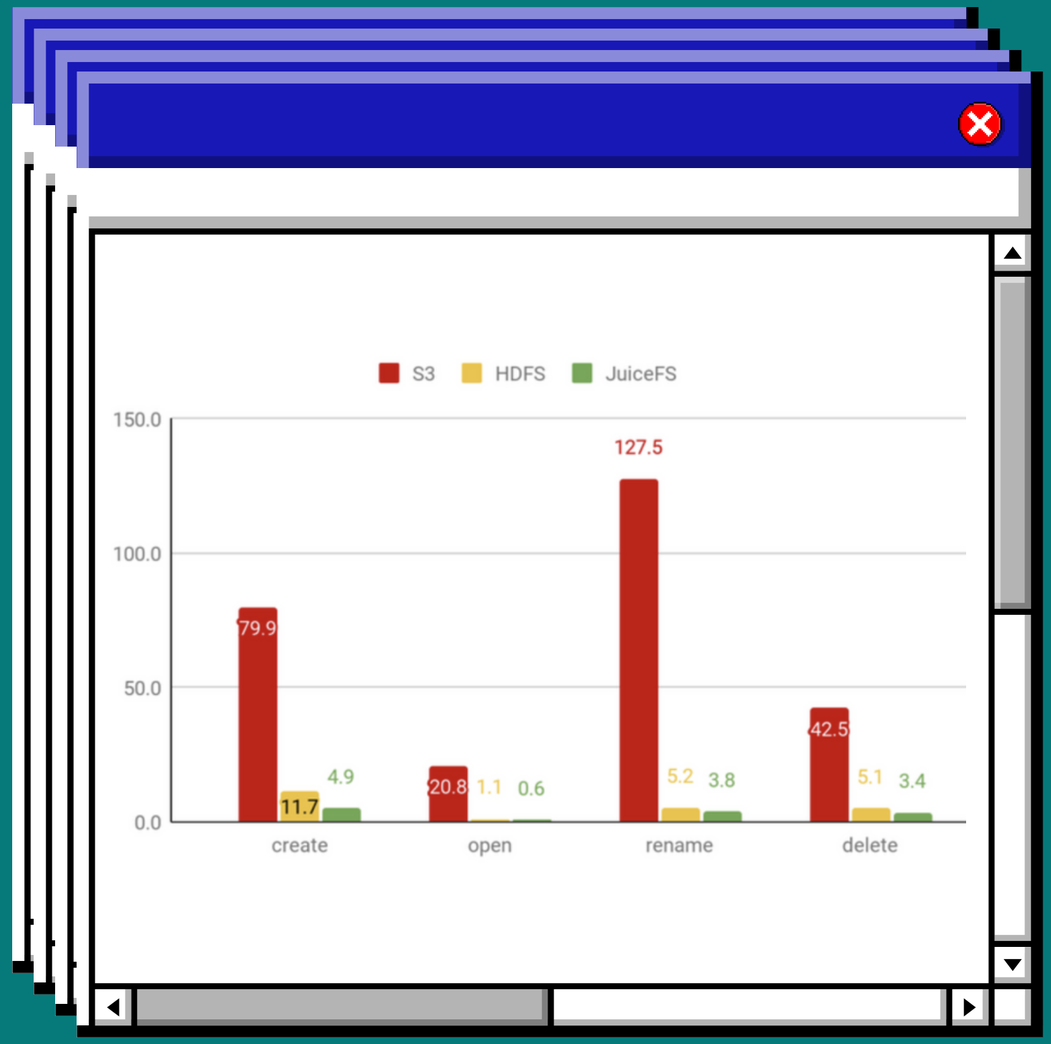
JuiceFS
Amazon S3
HDFS

[Back to JuiceFS Page]

	JUICE FS	S3	HDFS
Support for Advanced Metadata Queries	✓	X	X
Shared Mounts	✓	X	X
Ensuring strong consistency	✓	✓	X
Local caching	✓	✓	X
File grouping	✓	X	X
Atomic operations	✓	X	X
Data compression	✓	X	✓
Object storage	✓	✓	X

JuiceFS
Amazon S3
HDFS

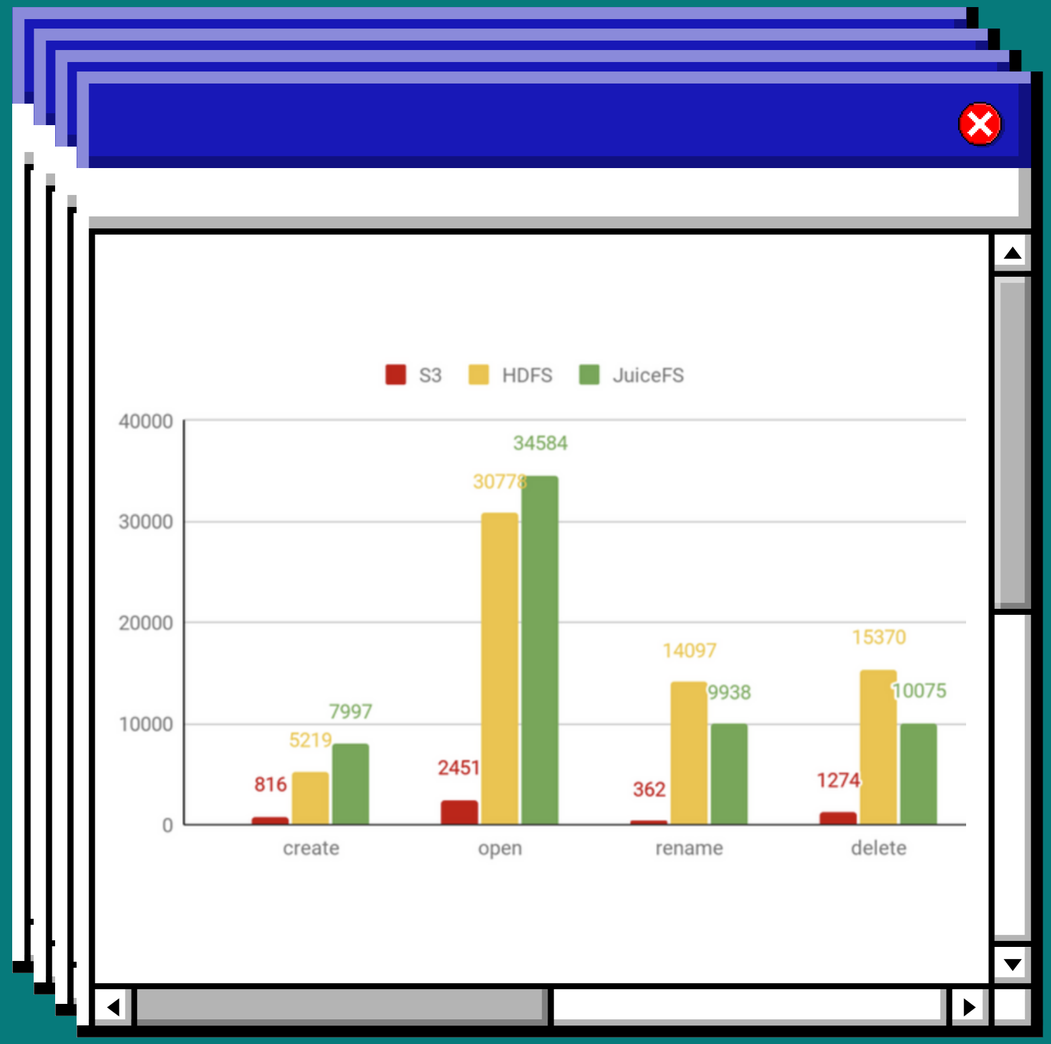
Metadata Latency (less is better).



Back to JuiceFS Page

JuiceFS
Amazon S3
HDFS

Metadata Throughput (bigger is better).



Back to JuiceFS Page

Spark

Back to JuiceFS Page

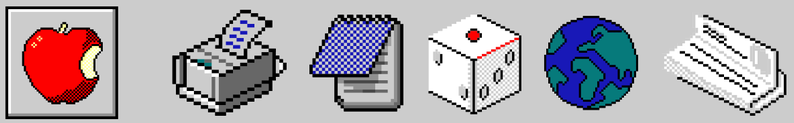
Spark

provides fast and flexible ways to analyze large amounts of data in real-time.

Back to JuiceFS Page

Spark and JuiceFS

once we have configured the Spark session, we can use the spark.read method to load the data from JuiceFS.



Back to JuiceFS Page

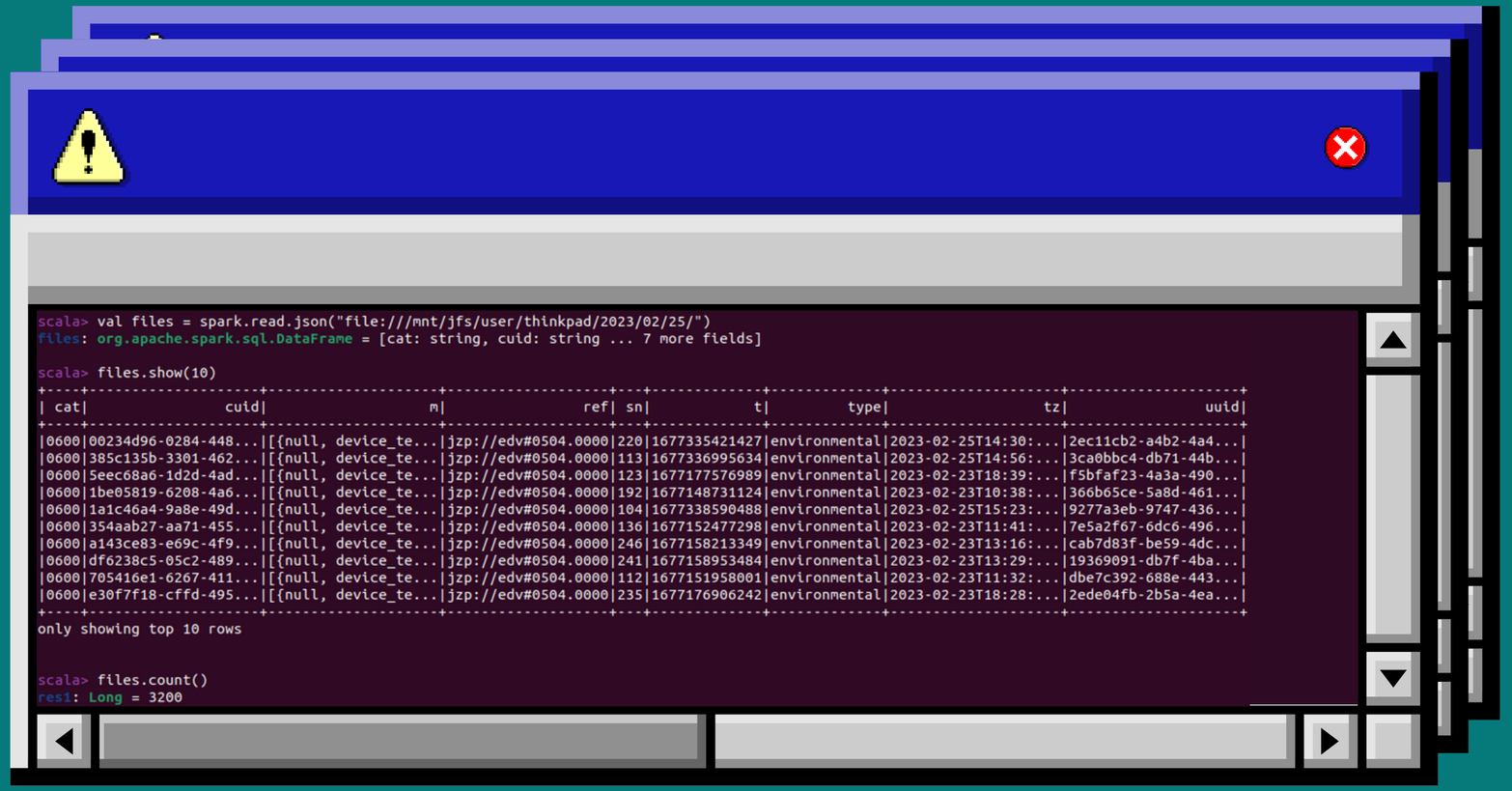
Spark and JuiceFS

loaded the data into Spark, we can start querying it using Scala.

Back to JuiceFS Page

Spark Queries

the result of a Spark Query in Scala on JuiceFS.



```
scala> val files = spark.read.json("file:///mnt/jfs/user/thinkpad/2023/02/25/")
files: org.apache.spark.sql.DataFrame = [cat: string, cuid: string ... 7 more fields]

scala> files.show(10)
+----+--------------------+--------------------+--------------------+---+-------------+-------------+--------------------+--------------------+
| cat|                cuid|                   m|                ref| sn|            t|         type|                  tz|                uuid|
+----+--------------------+--------------------+--------------------+---+-------------+-------------+--------------------+--------------------+
|0600|00234d96-0284-448...|[{null, device_te...|jzp://edv#0504.0000|220|1677335421427|environmental|2023-02-25T14:30:...|2ec11cb2-a4b2-4a4...|
|0600|385c135b-3301-462...|[{null, device_te...|jzp://edv#0504.0000|113|1677336995634|environmental|2023-02-25T14:56:...|3ca0bbc4-db71-44b...|
|0600|5eec68a6-1d2d-4ad...|[{null, device_te...|jzp://edv#0504.0000|123|1677177576989|environmental|2023-02-23T18:39:...|f5bfaf23-4a3a-490...|
|0600|1be05819-6208-4a6...|[{null, device_te...|jzp://edv#0504.0000|192|1677148731124|environmental|2023-02-23T10:38:...|366b65ce-5a8d-461...|
|0600|1a1c46a4-9a8e-49d...|[{null, device_te...|jzp://edv#0504.0000|104|1677338590488|environmental|2023-02-25T15:23:...|9277a3eb-9747-436...|
|0600|354aab27-aa71-455...|[{null, device_te...|jzp://edv#0504.0000|136|1677152477298|environmental|2023-02-23T11:41:...|7e5a2f67-6dc6-496...|
|0600|a143ce83-e69c-4f9...|[{null, device_te...|jzp://edv#0504.0000|246|1677158213349|environmental|2023-02-23T13:16:...|cab7d83f-be59-4dc...|
|0600|df6238c5-05c2-489...|[{null, device_te...|jzp://edv#0504.0000|241|1677158953484|environmental|2023-02-23T13:29:...|19369091-db7f-4ba...|
|0600|705416e1-6267-411...|[{null, device_te...|jzp://edv#0504.0000|112|1677151958001|environmental|2023-02-23T11:32:...|dbe7c392-688e-443...|
|0600|e30f7f18-cffd-495...|[{null, device_te...|jzp://edv#0504.0000|235|1677176906242|environmental|2023-02-23T18:28:...|2ede04fb-2b5a-4ea...|
+----+--------------------+--------------------+--------------------+---+-------------+-------------+--------------------+--------------------+
only showing top 10 rows


scala> files.count()
res1: Long = 3200
```

Back to JuiceFS Page


Thank you!
thevladdo - raphlat - muno1