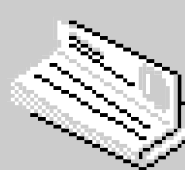
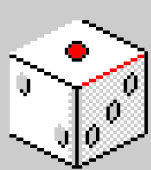
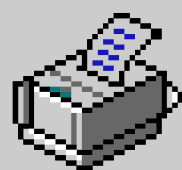
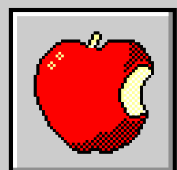


OSTORAGE JUICEFS



Evaluation of a distributed
storage technology based
on JuiceFS



11:11AM



IoT Simulator



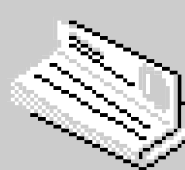
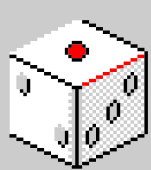
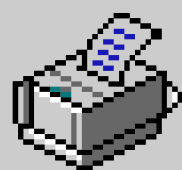
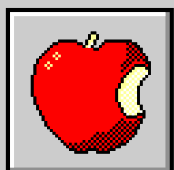
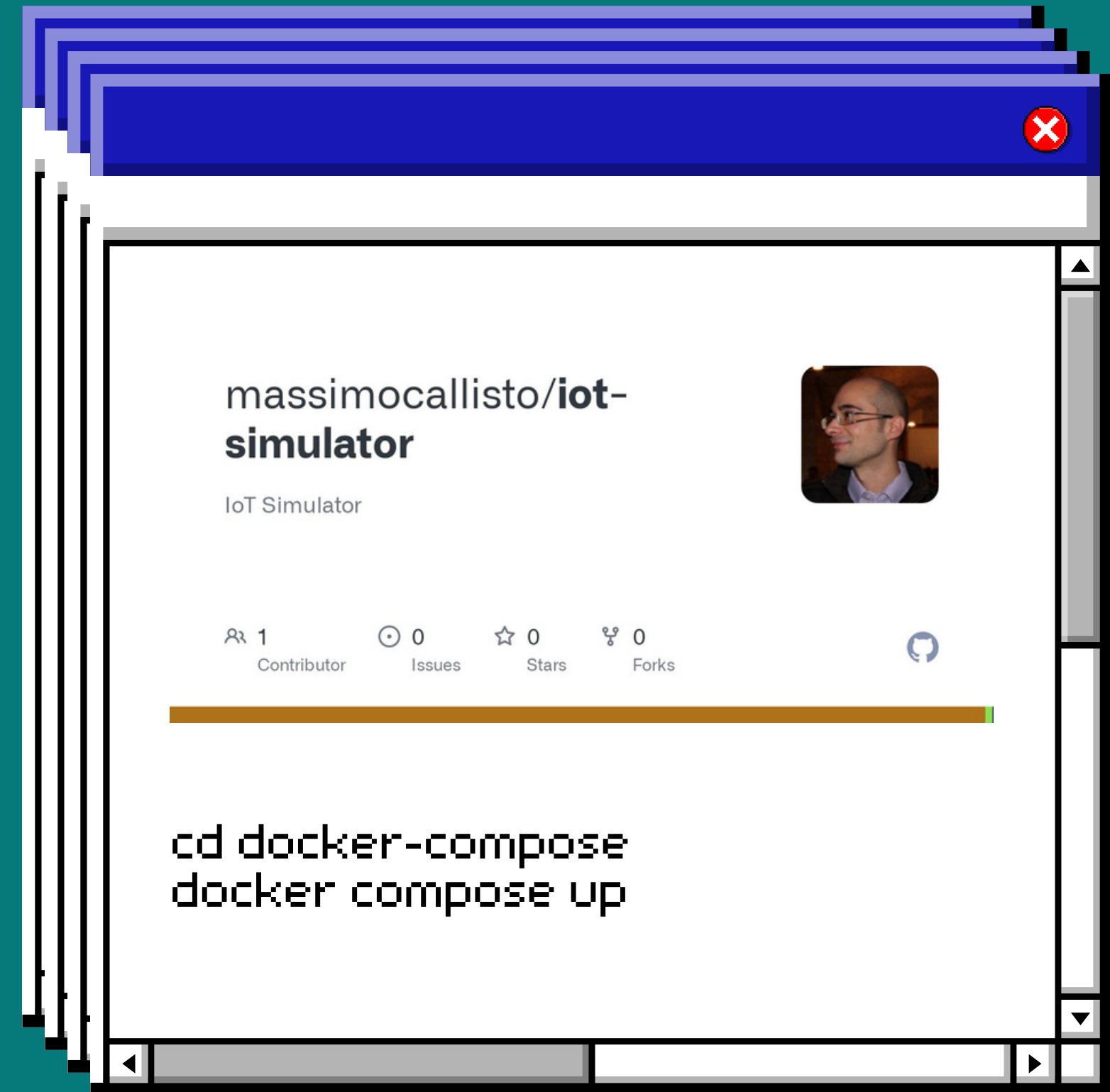
[Back to Architecture Page](#)



IoT Simulator



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



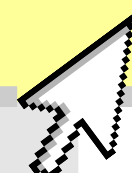
[Back to Architecture Page](#)



Apache Kafka



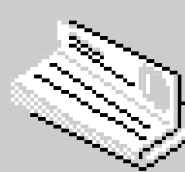
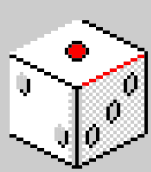
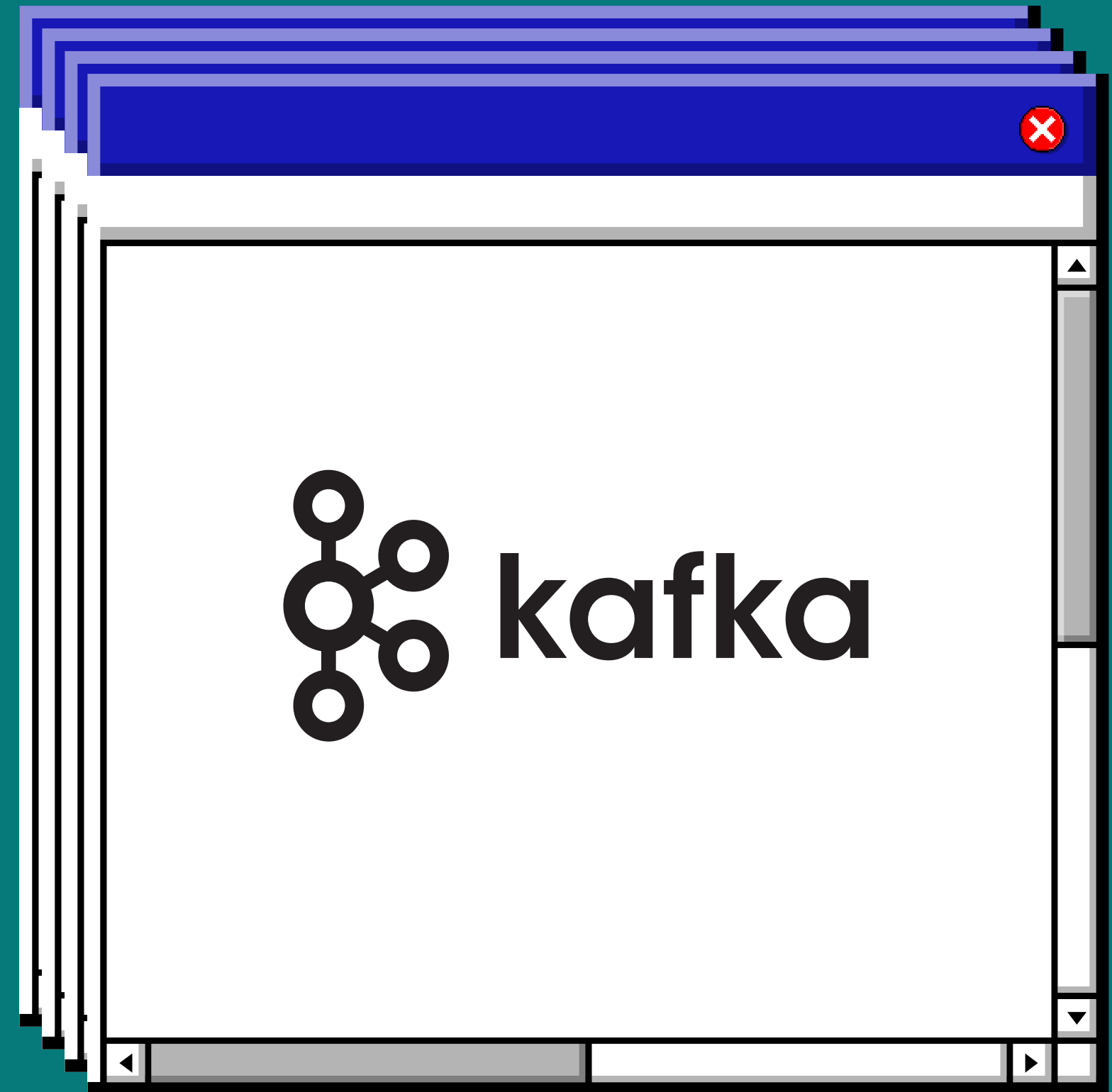
[Back to IoT Page](#)



Apache Kafka



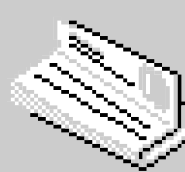
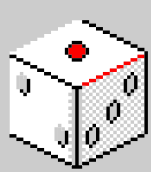
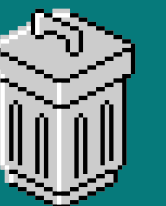
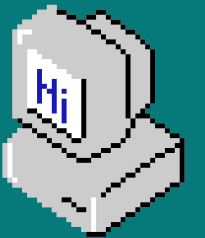
receives the JSON messages from the IoT devices and stores them in topics.



[Back to IoT Page](#)

Apache Kafka

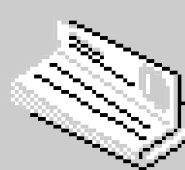
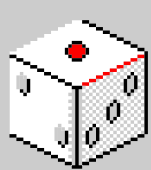
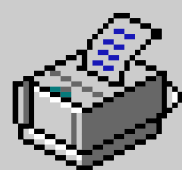
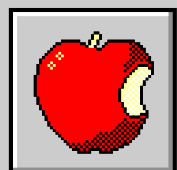
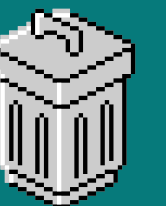
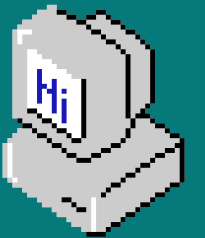
Whether you are looking to process data from IoT devices, to handle data pipelines in big data processing applications, or to provide real-time data feeds for data visualization and analysis, Apache Kafka is a highly effective solution that is well-suited to meet your needs



[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.



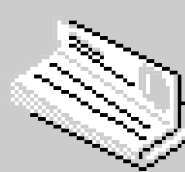
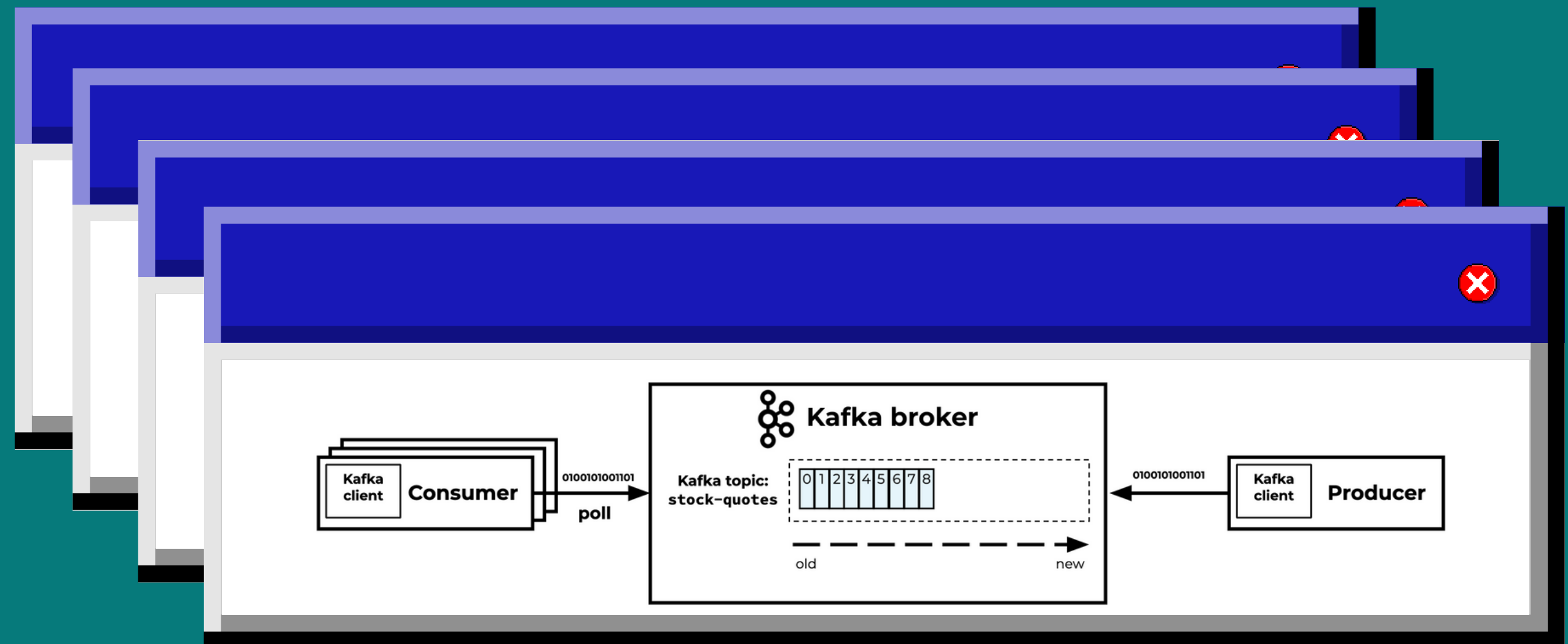
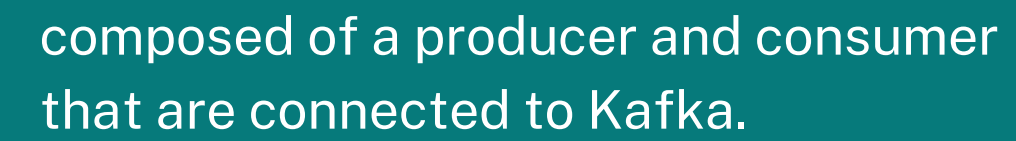
[Back to lot Page](#)



Persistor Prototype

[Back to Kafka Page](#)





[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

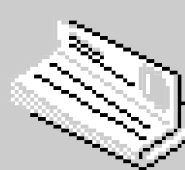
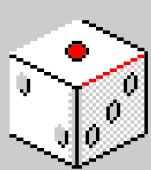
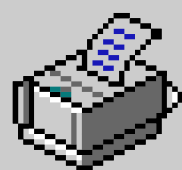
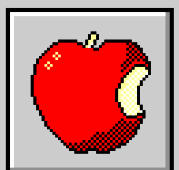


consume the messages
from the ack topic and
create a log file

Files Directory



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

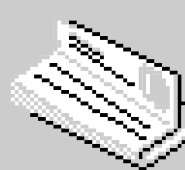
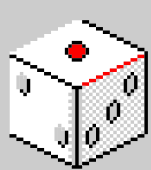
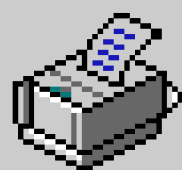
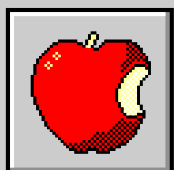
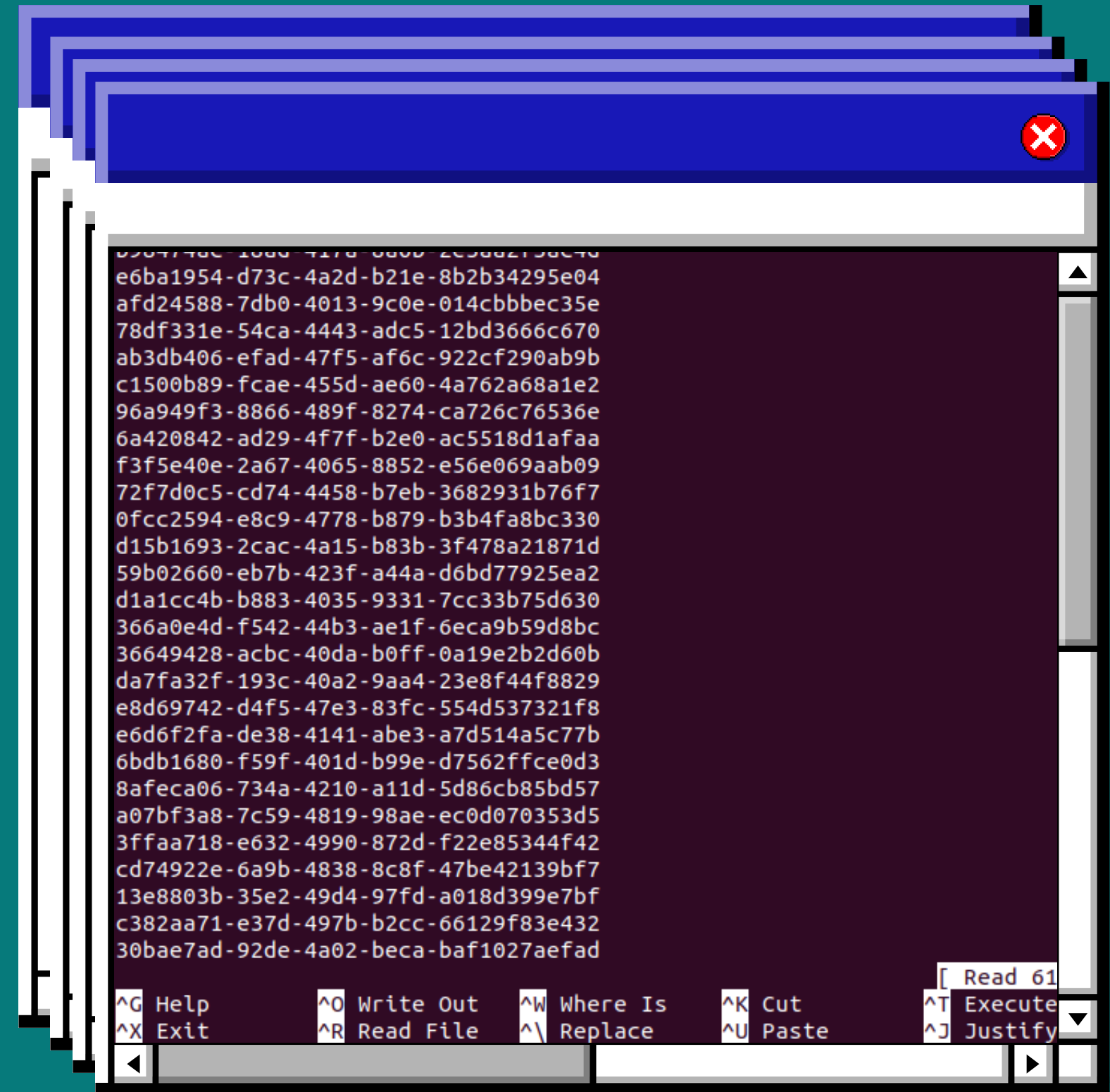


[Back to Kafka Page](#)

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



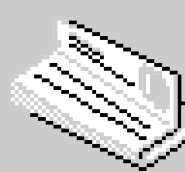
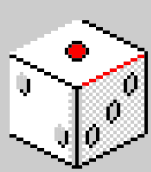
[Back to Persistor Page](#)



JuiceFS



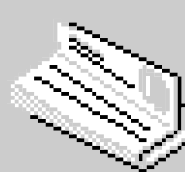
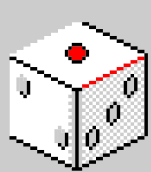
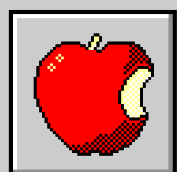
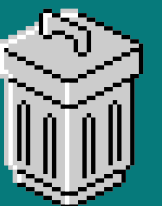
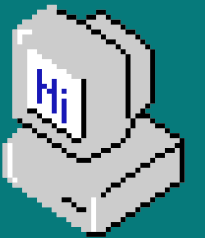
a distributed file system that provides consistent and scalable storage for the IoT data.



[Back to Persistor Page](#)

JuiceFS and IoT

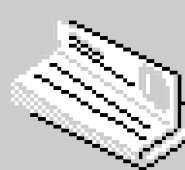
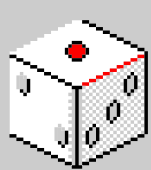
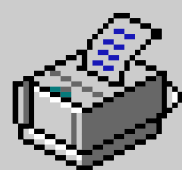
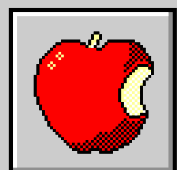
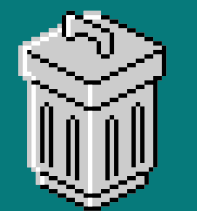
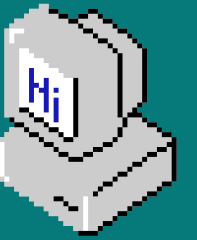
One of the key benefits of JuiceFS is its ability to scale horizontally, allowing it to accommodate growing amounts of data without sacrificing performance. This makes it an ideal choice for use in IoT environments, where the amount of data generated by devices can be substantial.



[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

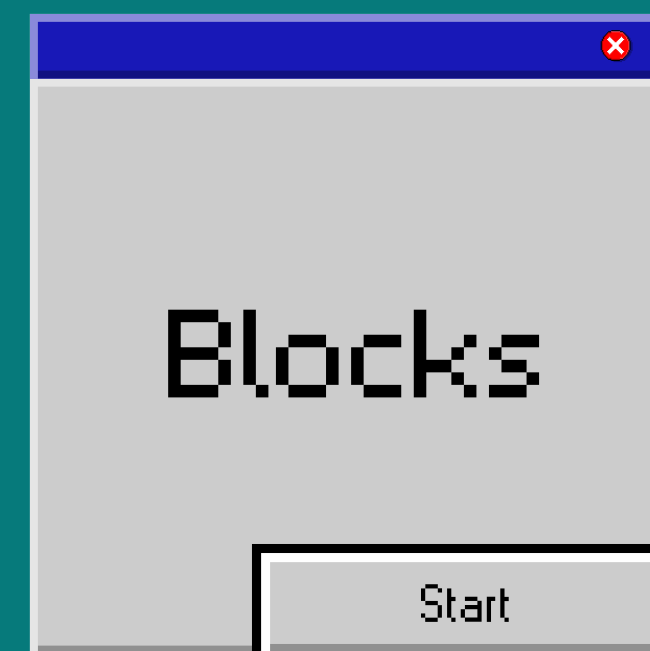
Storage



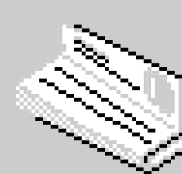
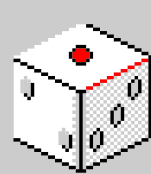
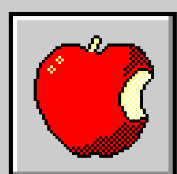
each file is divided
each with a size limit of 64 mb.



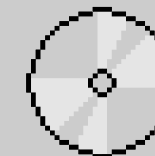
each chunk is divided
serve to optimize different types
of write operations.



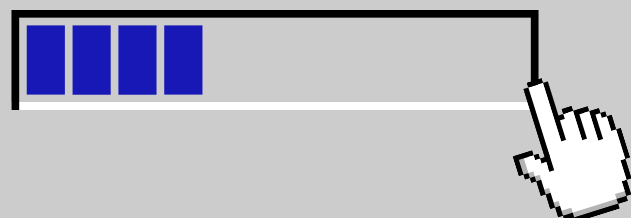
each slice is divided
of size 4 MiB by default.



[Back to JuiceFS Page](#)



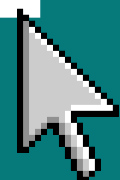
JuiceFS Amazon S3 HDFS



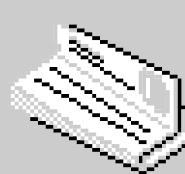
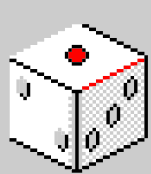
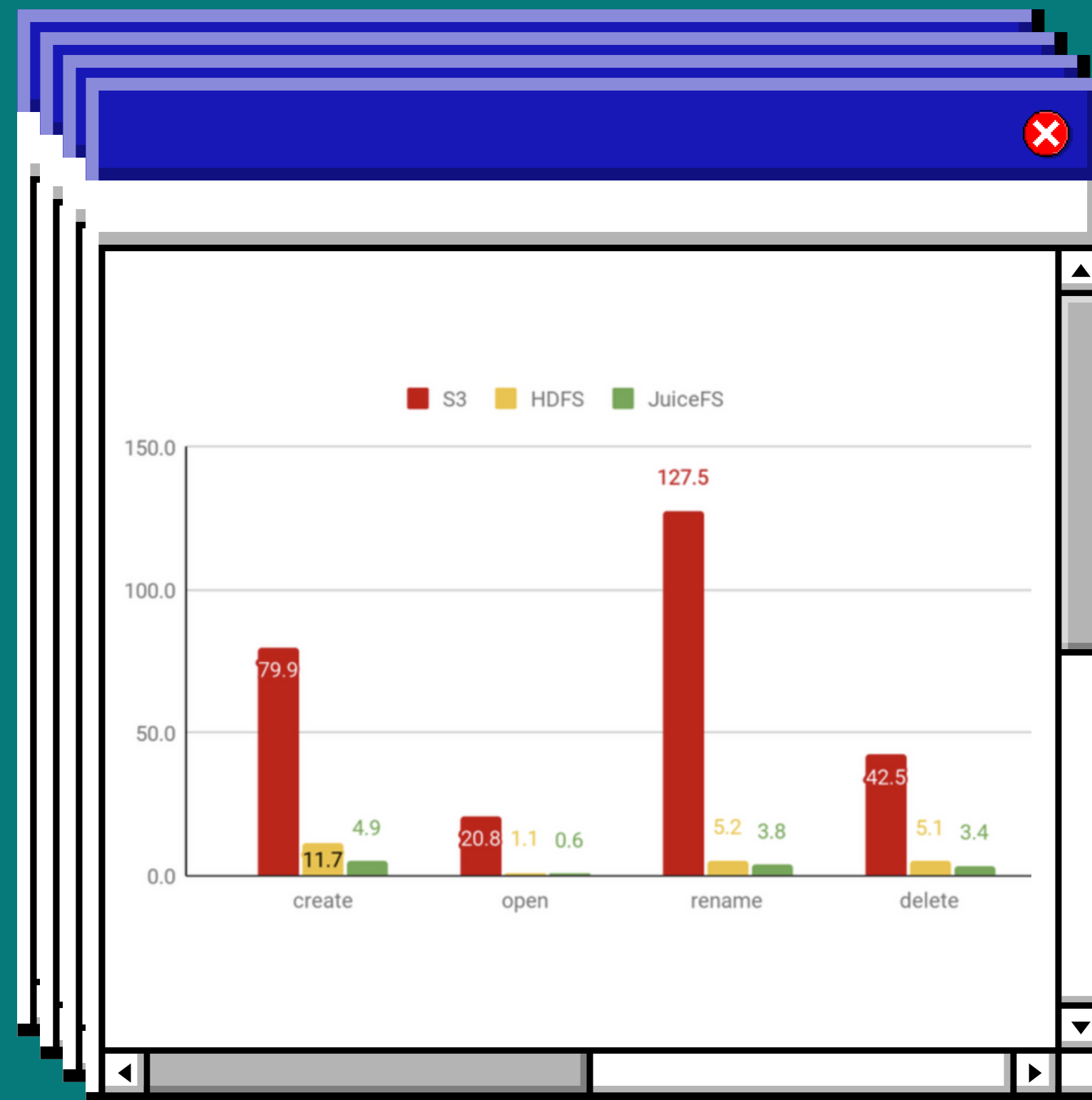
[Back to JuiceFS Page](#)

	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

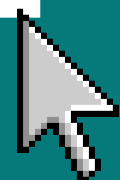


Metadata Latency (less is better).

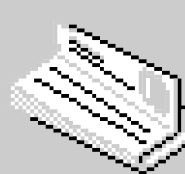
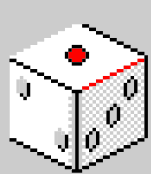
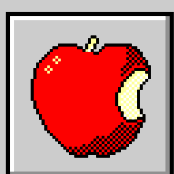
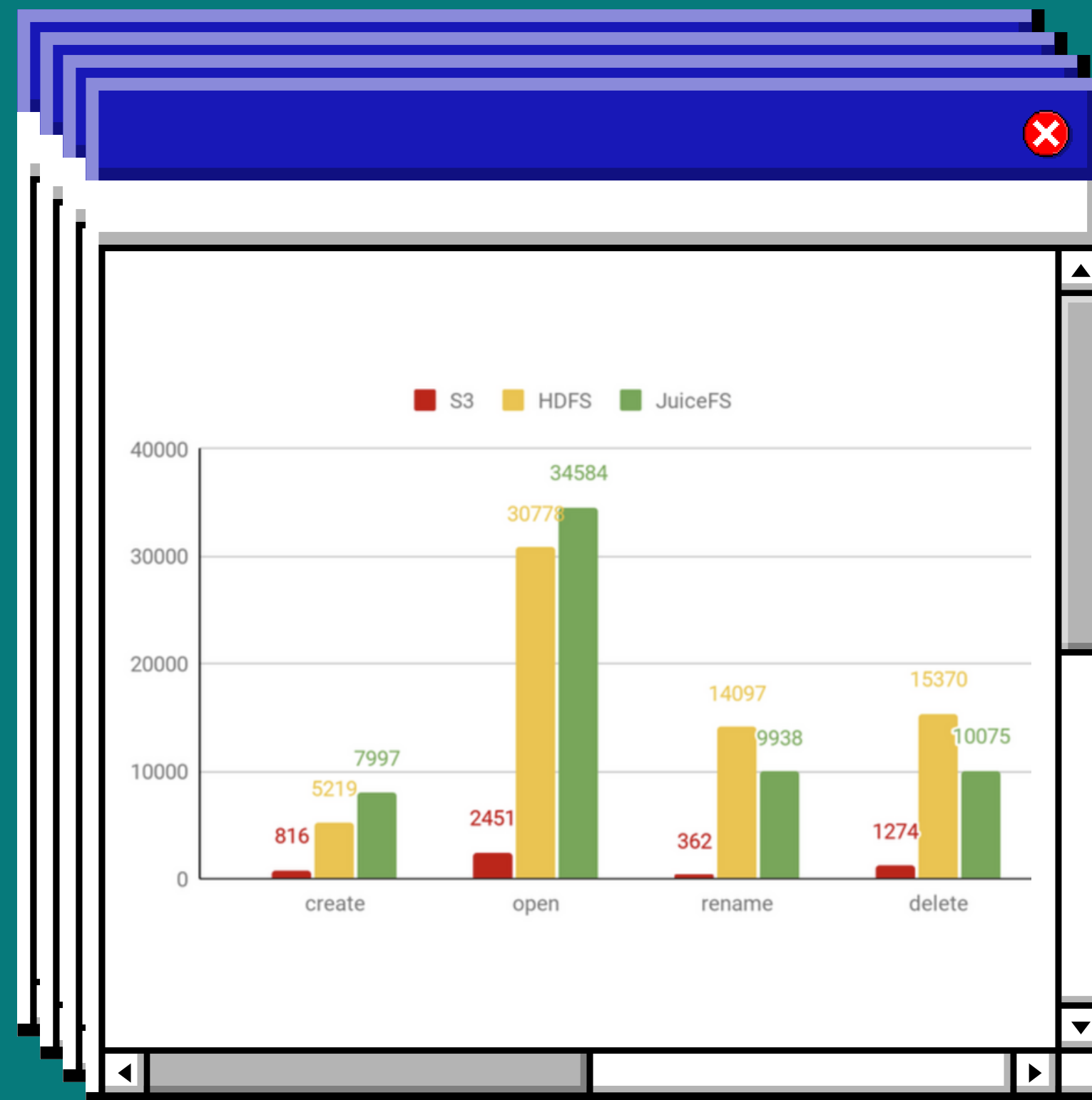


[Back to JuiceFS Page](#)

JuiceFS Amazon S3 HDFS



Metadata Throughput (bigger is better).



[Back to JuiceFS Page](#)



Spark



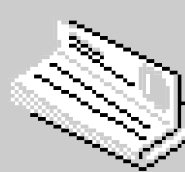
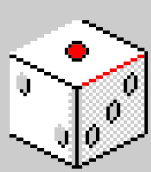
[Back to JuiceFS Page](#)



Spark



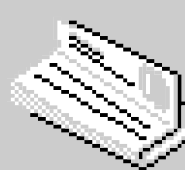
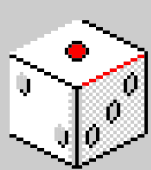
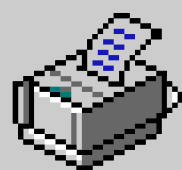
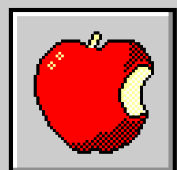
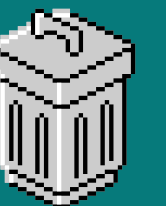
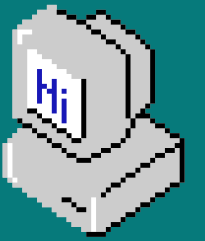
provides fast and flexible ways to analyze large amounts of data in real-time.



[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.



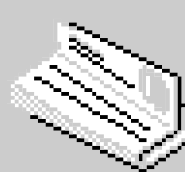
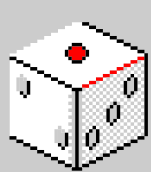
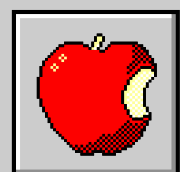
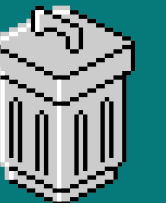
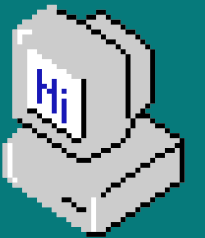
[Back to JuiceFS Page](#)

Spark and Scala

Scala supports object oriented programming, favorable for Java background developers (like us).

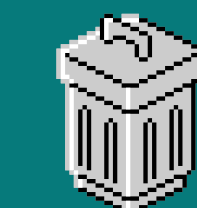
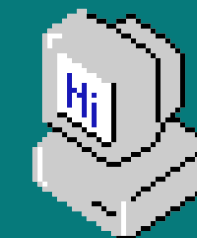
Scala can also reuse a lot of existing Java frameworks.

Scala supports functional programming with functions like:
`.map .flatMap .filter`



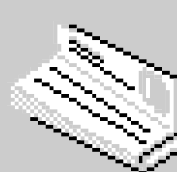
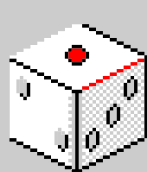
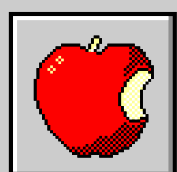
Spark and JuiceFS

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



```
val files = spark.read.json("jfs://myjfs/user/thinkpad/2023/02/28/")  
files.show()
```

Start

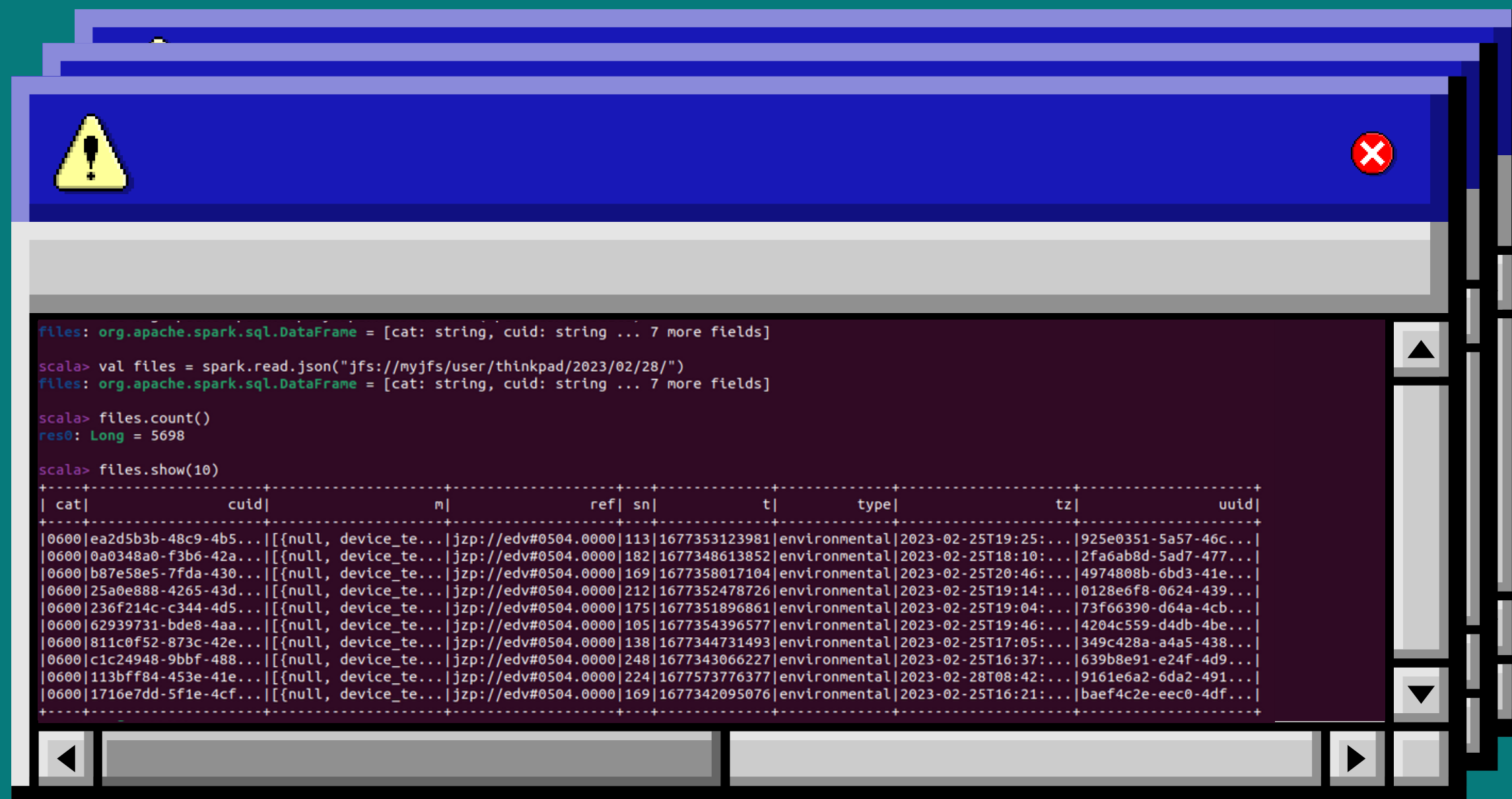


[Back to JuiceFS Page](#)

Spark Queries



the result of a Spark Query in Scala on JuiceFS.

A pixel-art style terminal window with a blue header bar containing a yellow warning triangle icon on the left and a red close button on the right. The terminal content shows Scala code and the output of a Spark query. The output includes a table with 10 columns: cat, cuid, m, ref, sn, t, type, tz, and uuid. The table contains 10 rows of data, each representing a file entry with various identifiers and timestamps.

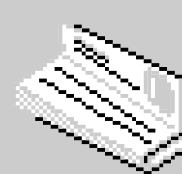
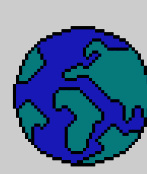
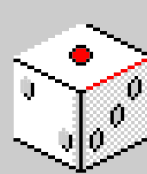
```
files: org.apache.spark.sql.DataFrame = [cat: string, cuid: string ... 7 more fields]

scala> val files = spark.read.json("jfs://myjfs/user/thinkpad/2023/02/28/")
files: org.apache.spark.sql.DataFrame = [cat: string, cuid: string ... 7 more fields]

scala> files.count()
res0: Long = 5698

scala> files.show(10)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| cat|      cuid|      m|      ref| sn|      t|      type|      tz|      uuid|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|0600|ea2d5b3b-48c9-4b5...|[null, device_te...|jzp://edv#0504.0000|113|1677353123981|environmental|2023-02-25T19:25:...|925e0351-5a57-46c...|
|0600|0a0348a0-f3b6-42a...|[null, device_te...|jzp://edv#0504.0000|182|1677348613852|environmental|2023-02-25T18:10:...|2fa6ab8d-5ad7-477...|
|0600|b87e58e5-7fda-430...|[null, device_te...|jzp://edv#0504.0000|169|1677358017104|environmental|2023-02-25T20:46:...|4974808b-6bd3-41e...|
|0600|25a0e888-4265-43d...|[null, device_te...|jzp://edv#0504.0000|212|1677352478726|environmental|2023-02-25T19:14:...|0128e6f8-0624-439...|
|0600|236f214c-c344-4d5...|[null, device_te...|jzp://edv#0504.0000|175|1677351896861|environmental|2023-02-25T19:04:...|73f66390-d64a-4cb...|
|0600|62939731-bde8-4aa...|[null, device_te...|jzp://edv#0504.0000|105|1677354396577|environmental|2023-02-25T19:46:...|4204c559-d4db-4be...|
|0600|811c0f52-873c-42e...|[null, device_te...|jzp://edv#0504.0000|138|1677344731493|environmental|2023-02-25T17:05:...|349c428a-a4a5-438...|
|0600|c1c24948-9bbf-488...|[null, device_te...|jzp://edv#0504.0000|248|1677343066227|environmental|2023-02-25T16:37:...|639b8e91-e24f-4d9...|
|0600|113bff84-453e-41e...|[null, device_te...|jzp://edv#0504.0000|224|1677573776377|environmental|2023-02-28T08:42:...|9161e6a2-6da2-491...|
|0600|1716e7dd-5f1e-4cf...|[null, device_te...|jzp://edv#0504.0000|169|1677342095076|environmental|2023-02-25T16:21:...|baef4c2e-eec0-4df...|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```



[Back to JuiceFS Page](#)



Thank you!

thevladdo - raphlat - muno1

