ClickhouseFile

Clickhouse's Tens of Billions of Data Synchronization Through SeaTunnel

Apache SeaTunnel Committer FanJia

2022-07-02

CONTENT

Next-generation high-performance, distributed, massive data integration framework

https://seatunnel.apache.org/

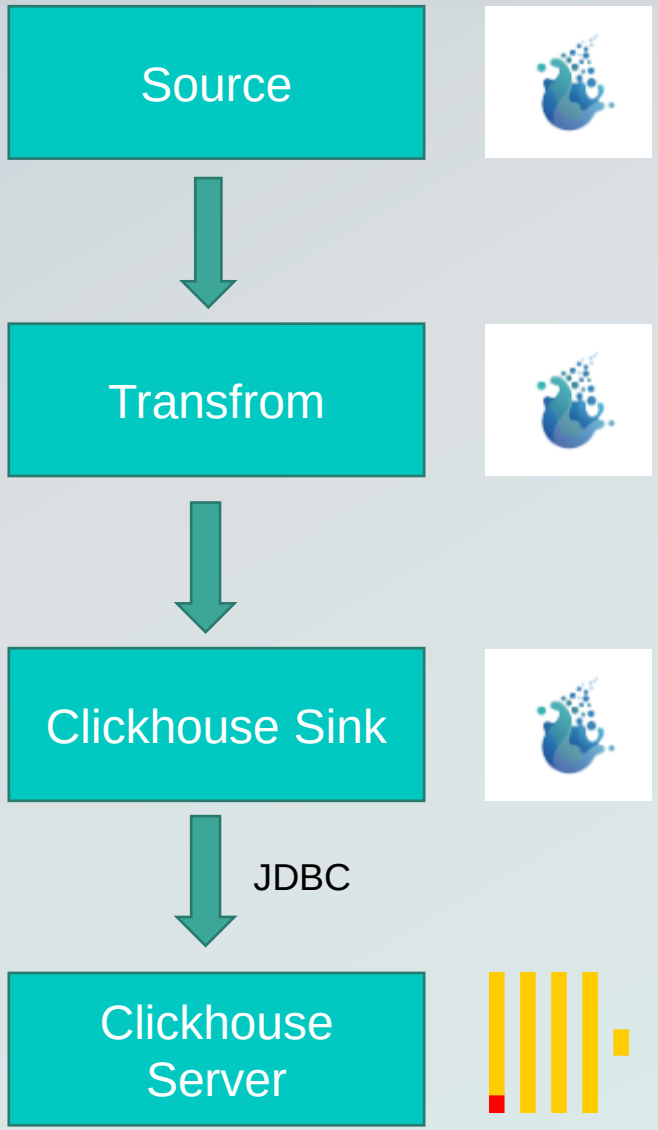
Why implement the ClickhouseFile Connector

01

Clickhouse Sink Status

Clickhouse Sink Status

SeaTunnel Basic process

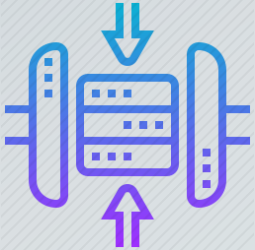


Clickhouse Sink Status

Clickhouse Sink Status

Write through JDBC interface, HTTP mode

Is there a faster way than HTTP? TCP? Or else?



Data compression is not possible

Isn't it a waste of resources to have a lot of duplicate data in the data?
Why can't it be compressed first and then transmitted?

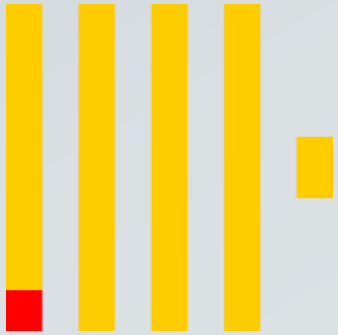
Too much data can easily cause to OOM

02

Weak Scene for Clickhouse Sink

Weak Scene For Clickhouse Sink

The amount of data is large and cannot be processed through HTTP



Server stress, too many INSERT requests

Weak Scene For Clickhouse Sink

Is there a way to meet the following requirements?

- Data compression can be done on the SeaTunnel side
- Not increase the resource load of the server when writing data
- Write massive amounts of data quickly

03

ClickhouseFile Connector

ClickhouseFile Connector

Key：Clickhouse-local

The clickhouse-local program enables you to perform fast processing on local files, without having to deploy and configure the ClickHouse server.

clickhouse-local uses the same core as ClickHouse server, so it supports most of the features and the same set of formats and table engines.

ClickhouseFile Connector

Key：Clickhouse-local

```
% echo -e "1\n2\n3" |  ./Clickhouse local -S "id Int64" -N "test_table"
-q "CREATE TABLE result_table
    (id Int64,CreateTime Date)
    ENGINE = MergeTree()
    PARTITION BY toYYYYMM(CreateTime) ORDER BY id;
    INSERT INTO TABLE result_table SELECT id,'2022-04-16' FROM test_table;"
--path /tmp/spark/clickhouse_file/data
```

1. Use Linux pipes to pass data to the test_table table of the Clickhouse-local program

2-5. Create a result_table table to receive data

6. Move data from test_table to result_table table

7. Defines the disk path for data processing

ClickhouseFile Connector

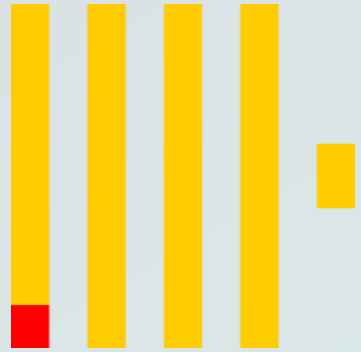
Implementation principle

Before

Apache SeaTunnel

SeaTunnel: Data is given to you, process it yourself

Clickhouse: I'm tired

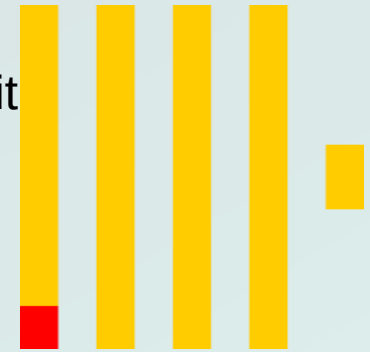


Now

Apache SeaTunnel

SeaTunnel: Generated the file for you, you can use it

Clickhouse: OK

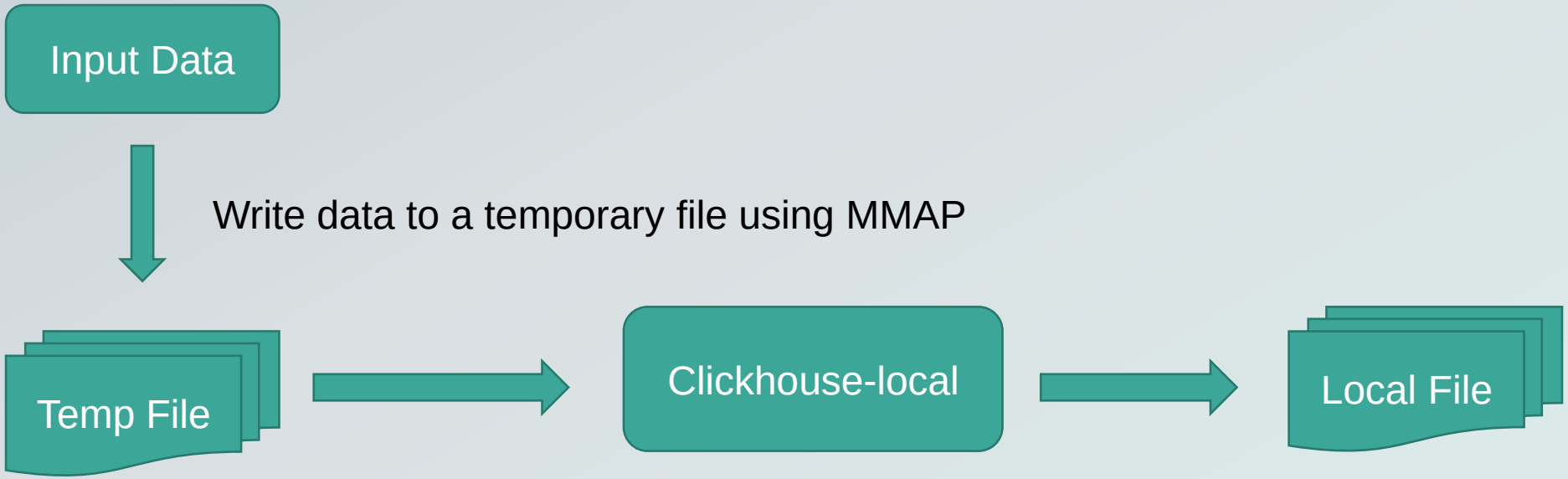


04

Core Technology Point

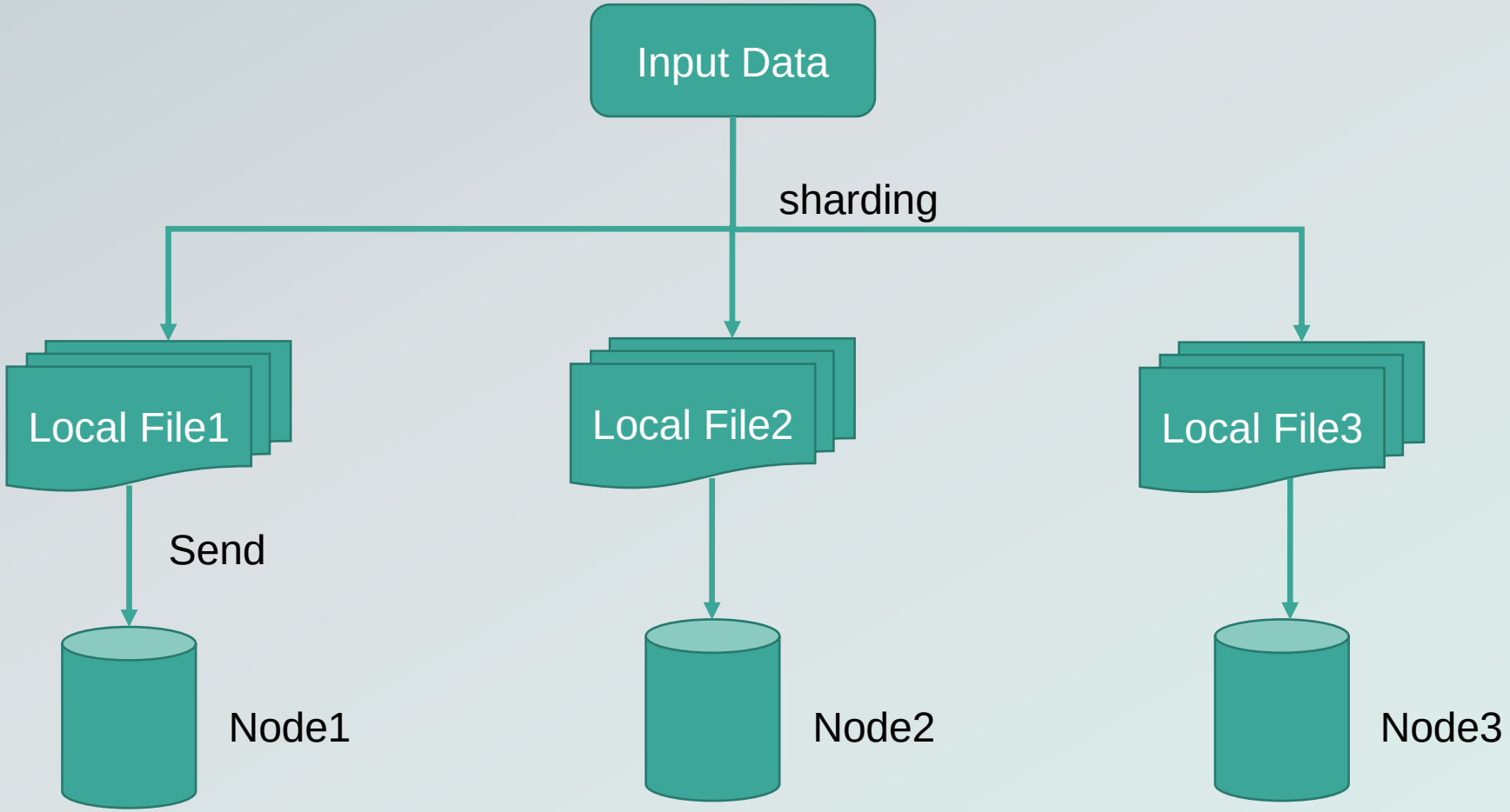
Core Technology Point

Temporary Files



Core Technology Point

Sharding Support



Core Technology Point

File Transfer

SCP

- Safety
- Universal
- Out of the box

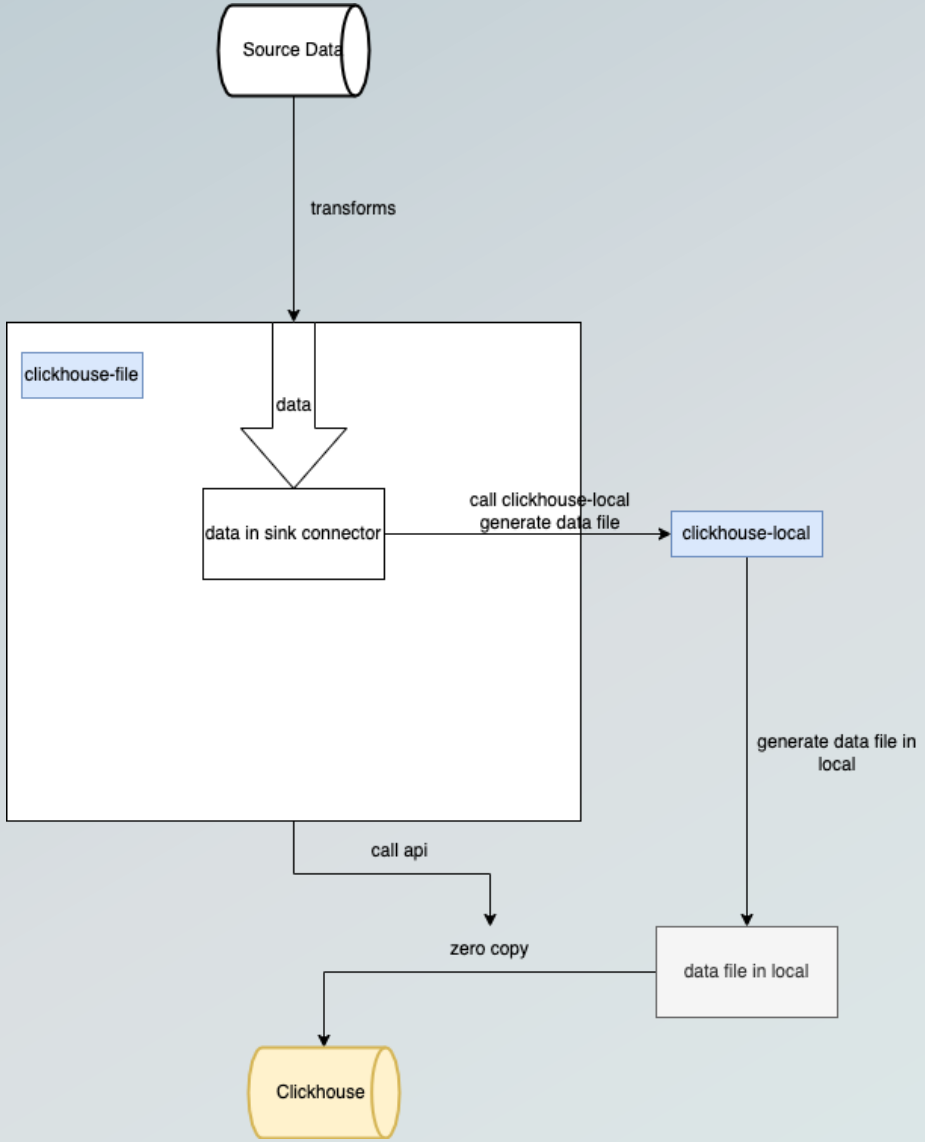
RSYNC

- Fast and efficient
- Resume from beak point

05

Implementation Analysis

Implementation Analysis



- Cache data
- Call the local clickhouse-local generate file
- Send data to server
- Execute the ATTACH command

06

Ability Comparison

Ability comparison

Feature	Clickhouse	ClickhouseFile
Mass data transfer	✓	✓ ✓
Environment complexity	😎	☹
Universal	😎	☹
Server pressure	😎	😎 😎

07

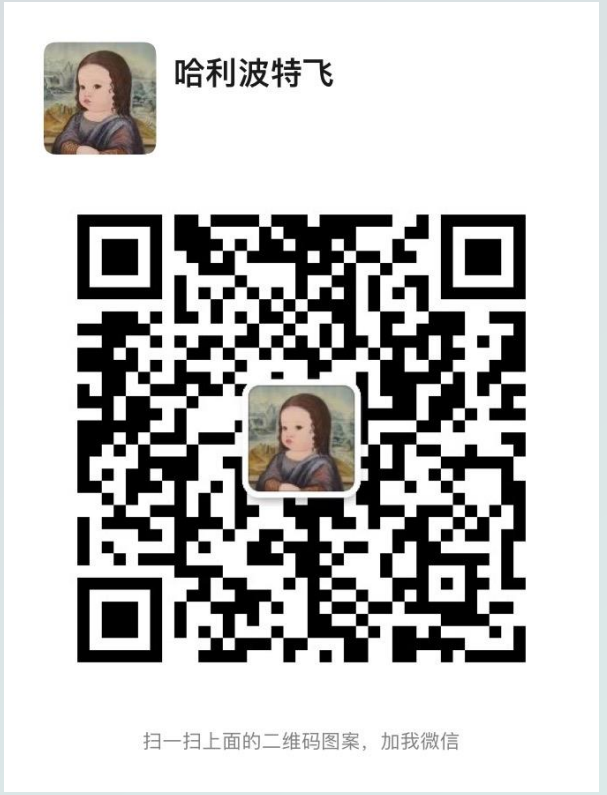
Next Step

Next Step

- Exactly-Once Support
- Support Zero Copy Transfer File
- More Engine Support

Join Us

Welcome to join our WeChat group



Thanks

https://seatunnel.apache.org/