IoTDB note

IoTDB

資料庫結構：**root.** + **database.** + **device.** + **timeseries.**

standalone docker 執行

按照 Apache IoTDB 的说明，安装和使用 Apache IoTDB。

以下<version>需要依照需求版本進行修改，例如：apache/iotdb:**1.3.3**-standalone。

```
# get IoTDB official image
docker pull apache/iotdb:<version>-standalone
# create docker bridge network
docker network create --driver=bridge --subnet=172.18.0.0/16 --gateway=172.18.0.1
iotdb
# create docker container
docker run -d --name iotdb-service \
              --hostname iotdb-service \
              --network iotdb \
              --ip 172.18.0.6 \
              -p 6667:6667 \
              -e cn_internal_address=iotdb-service \
              -e cn_seed_config_node=iotdb-service:10710 \
              -e cn_internal_port=10710 \
              -e cn_consensus_port=10720 \
              -e dn_rpc_address=iotdb-service \
              -e dn_internal_address=iotdb-service \
              -e dn_seed_config_node=iotdb-service:10710 \
              -e dn_mpp_data_exchange_port=10740 \
              -e dn_schema_region_consensus_port=10750 \
              -e dn_data_region_consensus_port=10760 \
              -e dn_rpc_port=6667 \
              apache/iotdb:<version>-standalone
# execute SQL
docker exec -ti iotdb-service /iotdb/sbin/start-cli.sh -h iotdb-service
```

configuration

cn+: Config Node Configuration

dn_: DataNode Configuration

C# Native API

C# API

步驟 1：建立專案

```
dotnet new console -n IoTDBExample
cd IoTDBExample
```

步驟 2：加入套件

```
dotnet add package Apache.IoTDB --version 1.3.3.1
dotnet add package Apache.IoTDB.Data --version 1.3.3.1
```

步驟 3：編輯 Program.cs

```
// See https://aka.ms/new-console-template for more information
// Console.WriteLine("Hello, World!");
using Apache.IoTDB;
using Apache.IoTDB.DataStructure;
// using Apache.IoTDB.Data;

string host = "localhost";
int port = 6667;
int pool_size = 2;

// Init Session
var session_pool = new SessionPool(host, port, pool_size);

// Open Session
await session_pool.Open(false);


// // Create TimeSeries
await session_pool.CreateTimeSeries("root.test_group.test_device.ts1",
TSDataType.TEXT, TSEncoding.PLAIN, Compressor.UNCOMPRESSED);
await session_pool.CreateTimeSeries("root.test_group.test_device.ts2",
TSDataType.BOOLEAN, TSEncoding.PLAIN, Compressor.UNCOMPRESSED);
await session_pool.CreateTimeSeries("root.test_group.test_device.ts3",
TSDataType.INT32, TSEncoding.PLAIN, Compressor.UNCOMPRESSED);

// Insert Record
var measures = new List<string>{"ts1", "ts2", "ts3"};
var values = new List<object> { "test_text", true, (int)123 };
var timestamp = 4;
var rowRecord = new RowRecord(timestamp, values, measures);
await session_pool.InsertRecordAsync("root.test_group.test_device", rowRecord);

// Insert Tablet
```

```
var timestamp_lst = new List<long>{ timestamp + 1 };
// var value_lst = new List<object> {"iotdb", true, (int) 12};
var value_lst = new List<List<object>> {
    new List<object> { "iotdb", true, 12 }   // ts1 (TEXT)
};
var tablet = new Tablet("root.test_group.test_device", measures, value_lst,
timestamp_lst);
await session_pool.InsertTabletAsync(tablet);

// 刪除整個 ts3
await session_pool.DeleteTimeSeriesAsync("root.test_group.test_device.ts3");

// 刪除 ts3 中的資料
var delDev = new List<string>{"root.test_group.test_device.ts3"};
await session_pool.DeleteDataAsync(delDev, 0, 5);

// Close Session
await session_pool.Close();
```

執行 IoTDB CLI

CLI

在 Docker 中執行 CLI

進入到 docker 的 CLI 環境中，執行以下命令：

```
docker exec -it iotdb-service bash
```

在 docker 中執行 CLI：

```
./start-cli.sh -h 172.18.0.6 -p 6667 -u root -pw root
```

在本機執行 CLI

下載 **apache-iotdb-1.3.4-all-bin.zip**，並解壓縮到本機。
在 **sbin** 資料夾中，執行以下命令：

windows

```
./start-cli.bat
```

```
-h 127.0.0.1 -p 6667 -u root -pw root
---------------------
Starting IoTDB Cli
---------------------
 _____       _________  ______   ______
|_   _|     |  _   _  ||_   _ `.|_   _ \
  | |   .--.|_/ | | \_|  | | `. \ | |_) |
  | | / .'`\ \  | |      | |  | | |  __'.
 _| |_| \__. | _| |_    _| |_.' /_| |__) |
|_____|'.__.' |_____|  |______.'|_______/  version 1.3.3 (Build: ad95a7e)


Successfully login at 127.0.0.1:6667
IoTDB> show databases
+--------------+---------------------+--------------------+------------------
+--------------------+
|
Database|SchemaReplicationFactor|DataReplicationFactor|TimePartitionOrigin|TimePar
titionInterval|
+--------------+---------------------+--------------------+------------------
+--------------------+
|root.test_group|                    1|                   1|
0|           604800000|
+--------------+---------------------+--------------------+------------------
+--------------------+
Total line number = 1
It costs 0.064s
IoTDB>
```

SQL 語法

SQL Manual

```
show timeseries
SHOW TIMESERIES root.ln.** limit 10 offset 10
SHOW TIMESERIES root.ln.** where timeseries contains 'wf01.wt'
```

```
+-------------------------------+-----+--------------+--------+--------+---------
---+----+----------+--------+-----------------+--------+
|                     Timeseries|Alias|      Database|DataType|Encoding|
Compression|Tags|Attributes|Deadband|DeadbandParameters|ViewType|
+-------------------------------+-----+--------------+--------+--------+---------
---+----+----------+--------+-----------------+--------+
|root.test_group.test_device.ts2| null|root.test_group| BOOLEAN|
PLAIN|UNCOMPRESSED|null|      null|    null|             null|    BASE|
|root.test_group.test_device.ts1| null|root.test_group|    TEXT|
PLAIN|UNCOMPRESSED|null|      null|    null|             null|    BASE|
```

```
|root.test_group.test_device.ts3| null|root.test_group|   INT32|
PLAIN|UNCOMPRESSED|null|      null|    null|              null|    BASE|
+------------------------------+-----+--------------+--------+--------+---------
---+----+----------+--------+-----------------+--------+
Total line number = 3
It costs 0.100s
```

```
Select * from root.test_group.test_device
```

```
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|
Time|root.test_group.test_device.ts2|root.test_group.test_device.ts1|root.test_gro
up.test_device.ts3|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|1970-01-01T00:00:00.001Z|                          true|
test_text|                           123|
|1970-01-01T00:00:00.002Z|                          true|
iotdb|                            12|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
Total line number = 2
It costs 0.097s
```

```
Insert into root.test_group.test_device(timestamp, ts1) VALUES (1, 'friday');
```

```
delete from root.test_group.test_device.ts2 where time = 2
```

如果需要**移除某時間的整筆資料**，需要將剛時間的所有 **timeseries** 都刪除

例如：

```
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|
Time|root.test_group.test_device.ts2|root.test_group.test_device.ts1|root.test_gro
up.test_device.ts3|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|1970-01-01T00:00:00.001Z|                          true|
friday|                           123|
|1970-01-01T00:00:00.002Z|                          null|
```

```
iotdb|                              12|
|1970-01-01T00:00:00.003Z|                           null|
monday|                            null|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
```

```
 delete from root.test_group.test_device.ts1 where time = 3
```

```
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|
Time|root.test_group.test_device.ts2|root.test_group.test_device.ts1|root.test_gro
up.test_device.ts3|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
|1970-01-01T00:00:00.001Z|                           true|
friday|                             123|
|1970-01-01T00:00:00.002Z|                           null|
iotdb|                              12|
+-----------------------+------------------------------+-----------------------
-------+------------------------------+
```

SQL　database

```
/*建立資料庫*/
CREATE DATABASE root.ln
/*查看資料庫*/
show databases
show databases root.*
/*刪除資料庫*/
DELETE DATABASE root.ln
/*统计資料庫数量*/
count databases
```

SQL timeseries

```
-- create timeseries root.ln.wf01.wt01.status with datatype=BOOLEAN,encoding=PLAIN
create timeseries root.test_group.test_device.ts3 with
datatype=INT32,encoding=PLAIN

delete timeseries root.test_group.test_device.ts3
```

SQL 數據處理

```
insert into root.ln.wf02.wt02(timestamp,status) values(1,true)
-- 同設備多時間
insert into root.ln.wf02.wt02(timestamp,hardware) values(1, 'v1'),(2, 'v1')
-- 多設備
insert into root.ln.wf02.wt02(timestamp, status, hardware) VALUES (3, false,
'v3'),(4, true, 'v4')
```

調正

如果依文件執行，會出現以下錯誤：

```
./docker_run_iotdb.sh
0272c65e7da40880cc18a85be6c9004a5851b840bf68997abda680c8b9080e51
docker: Error response from daemon: failed to set up container networking: network
workspace not found.
```