

DataX

第一章 概述

什么是DataX

DataX是阿里巴巴开源的一个异构数据源 **离线同步工具**，致力于实现包括关系型数据库（MySQL、Oracle等）、HDFS、Hive、ODPS、HBase、FTP等各种异构数据源之间稳定高效的数据同步功能。

DataX的设计

为了解决异构数据源同步问题，DataX将复杂的网状的同步链路变成了星型数据链路，DataX作为中间传输载体 **负责连接各种数据源**。当需要接入一个新的数据源的时候，只需要将此数据源对接到DataX，便能跟已有的数据源做到无缝数据的同步。

支持的数据源

DataX目前已经有了比较全面的插件体系，主流的RDMBS数据库、NOSQL、大数据计算系统都已经接入。

第二章 使用案例

从stream流读取数据并打印到控制台

查看配置模板(可选)

```
1 python datax.py -r streamreader -w streamwriter
```

编写配置文件

```
1 vim stream2stream.json
```

```
1  {
2    "job": {
3      "content": [
4        {
5          "reader": {
6            "name": "streamreader",
7            "parameter": {
8              "sliceRecordCount": 10,
9              "column": [
10               {
11                 "type": "long",
12                 "value": "10"
13               },
14               {
15                 "type": "string",
16                 "value": "hello, DataX"
17               }
18             ]
19           }
20         }
21       ]
22     }
23   }
```

```

18         ]
19     }
20 },
21     "writer": {
22         "name": "streamwriter",
23         "parameter": {
24             "encoding": "UTF-8",
25             "print": true
26         }
27     }
28 }
29 }
30 ],
31     "setting": {
32         "speed": {
33             "channel": 1
34         }
35     }
36 }
37 }

```

运行

```
1 /opt/module/datax/bin/datax.py /opt/module/datax/job/stream2stream.json
```

读取MySQL中的数据存放到HDFS

创建student表

```

1 mysql> create database datax;
2 mysql> use datax;
3 mysql> create table student(id int,name varchar(20));

```

插入数据

```
1 mysql> insert into student values(1001,'zhangsan'),(1002,'lisi'),(1003,'wangwu');
```

编写配置文件

```
1 vim /opt/module/datax/job/mysql2hdfs.json
```

```

1 {
2     "job": {
3         "content": [
4             {
5                 "reader": {
6                     "name": "mysqlreader",
7                     "parameter": {
8                         "column": [
9                             "id",
10                            "name"
11                        ],
12                        "connection": [
13                            {
14                                "jdbcUrl": [
15                                    "jdbc:mysql://master:3306/datax"
16                                ],
17                                "table": [

```

```

18         "student"
19     ]
20 }
21 ],
22 "username": "root",
23 "password": "000000"
24 }
25 },
26 "writer": {
27     "name": "hdfswriter",
28     "parameter": {
29         "column": [
30             {
31                 "name": "id",
32                 "type": "int"
33             },
34             {
35                 "name": "name",
36                 "type": "string"
37             }
38         ],
39         "defaultFS": "hdfs://master:9000",
40         "fieldDelimiter": "\\t",
41         "fileName": "student.txt",
42         "fileType": "text",
43         "path": "/",
44         "writeMode": "append"
45     }
46 }
47 }
48 ],
49 "setting": {
50     "speed": {
51         "channel": "1"
52     }
53 }
54 }
55 }

```

执行任务

```
1 bin/datax.py job/mysql2hdfs.json
```

查看HDFS

读取HDFS数据写入MySQL

```
1 vim job/hdfs2mysql.json
```

```

1  {
2      "job": {
3          "content": [
4              {
5                  "reader": {
6                      "name": "hdfsreader",
7                      "parameter": {
8                          "column": ["*"],

```

```

9          "defaultFS": "hdfs://hadoop102:9000",
10         "encoding": "UTF-8",
11         "fieldDelimiter": "\t",
12         "fileType": "text",
13         "path": "/student.txt"
14     }
15 },
16 "writer": {
17     "name": "mysqlwriter",
18     "parameter": {
19         "column": [
20             "id",
21             "name"
22         ],
23         "connection": [
24             {
25                 "jdbcUrl": "jdbc:mysql://hadoop102:3306/datax",
26                 "table": ["student2"]
27             }
28         ],
29         "password": "000000",
30         "username": "root",
31         "writeMode": "insert"
32     }
33 }
34 },
35 ],
36 "setting": {
37     "speed": {
38         "channel": "1"
39     }
40 }
41 }
42 }

```