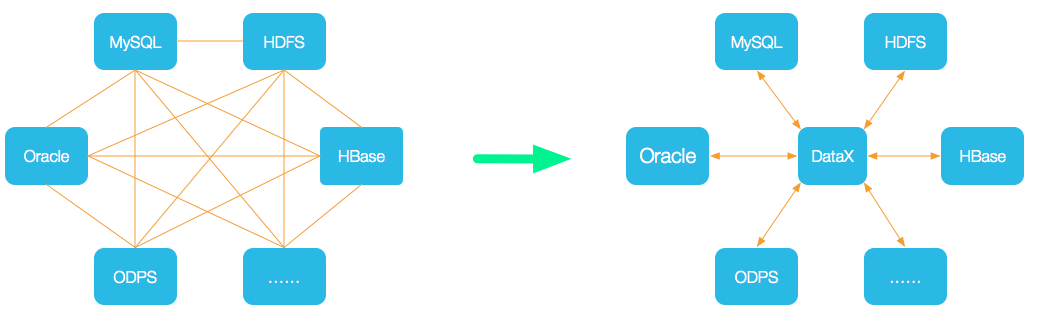
**一. DataX3.0概览**

​ DataX 是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。

[](https://cloud.githubusercontent.com/assets/1067175/17879841/93b7fc1c-6927-11e6-8cda-7cf8420fc65f.png)

* **设计理念**

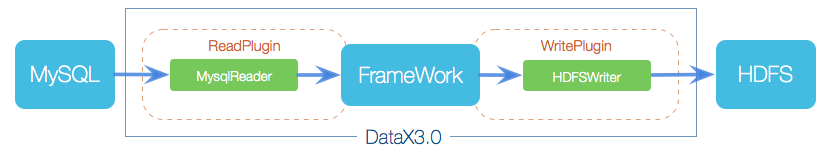
为了解决异构数据源同步问题，DataX将复杂的网状的同步链路变成了星型数据链路，DataX作为中间传输载体负责连接各种数据源。当需要接入一个新的数据源的时候，只需要将此数据源对接到DataX，便能跟已有的数据源做到无缝数据同步。

* **当前使用现状**

DataX在阿里巴巴集团内被广泛使用，承担了所有大数据的离线同步业务，并已持续稳定运行了6年之久。目前每天完成同步8w多道作业，每日传输数据量超过300TB。

此前已经开源DataX1.0版本，此次介绍为阿里云开源全新版本DataX3.0，有了更多更强大的功能和更好的使用体验。Github主页地址：<https://github.com/alibaba/DataX>

**二、DataX3.0框架设计**

[](https://cloud.githubusercontent.com/assets/1067175/17879884/ec7e36f4-6927-11e6-8f5f-ffc43d6a468b.png)

DataX本身作为离线数据同步框架，采用Framework + plugin架构构建。将数据源读取和写入抽象成为Reader/Writer插件，纳入到整个同步框架中。

* Reader：Reader�为数据采集模块，负责采集数据源的数据，将数据发送给Framework。
* Writer： Writer为数据写入模块，负责不断向Framework取数据，并将数据写入到目的端。
* Framework：Framework用于连接reader和writer，作为两者的数据传输通道，并处理缓冲，流控，并发，数据转换等核心技术问题。

**三. DataX3.0插件体系**

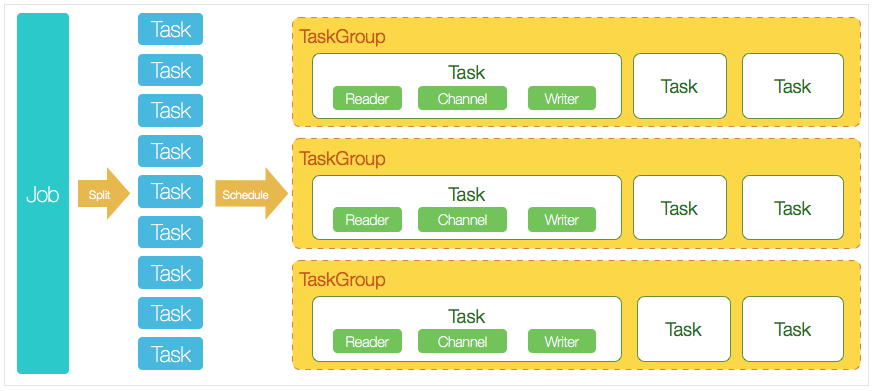
​ 经过几年积累，DataX目前已经有了比较全面的插件体系，主流的RDBMS数据库、NOSQL、大数据计算系统都已经接入。DataX目前支持数据如下：

| **类型** | **数据源** | **Reader(读)** | **Writer(写)** | **文档** |
| --- | --- | --- | --- | --- |
| RDBMS 关系型数据库 | MySQL | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/mysqlreader/doc/mysqlreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/mysqlwriter/doc/mysqlwriter.md) |
|  | Oracle | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/oraclereader/doc/oraclereader.md) 、[写](https://github.com/alibaba/DataX/blob/master/oraclewriter/doc/oraclewriter.md) |
|  | SQLServer | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/sqlserverreader/doc/sqlserverreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/sqlserverwriter/doc/sqlserverwriter.md) |
|  | PostgreSQL | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/postgresqlreader/doc/postgresqlreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/postgresqlwriter/doc/postgresqlwriter.md) |
|  | DRDS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/drdsreader/doc/drdsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/drdswriter/doc/drdswriter.md) |
|  | 达梦 | √ | √ | [读](https://github.com/alibaba/DataX/blob/master) 、[写](https://github.com/alibaba/DataX/blob/master) |
|  | 通用RDBMS(支持所有关系型数据库) | √ | √ | [读](https://github.com/alibaba/DataX/blob/master) 、[写](https://github.com/alibaba/DataX/blob/master) |
| 阿里云数仓数据存储 | ODPS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/odpsreader/doc/odpsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/odpsswriter/doc/odpswriter.md) |
|  | ADS |  | √ | [写](https://github.com/alibaba/DataX/blob/master/adswriter/doc/adswriter.md) |
|  | OSS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/ossreader/doc/ossreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/osswriter/doc/osswriter.md) |
|  | OCS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/ocsreader/doc/ocsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/ocswriter/doc/ocswriter.md) |
| NoSQL数据存储 | OTS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/otsreader/doc/otsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/otswriter/doc/otswriter.md) |
|  | Hbase0.94 | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/hbase094xreader/doc/hbase094xreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/hbase094xwriter/doc/hbase094xwriter.md) |
|  | Hbase1.1 | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/hbase11xreader/doc/hbase11xreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/hbase11xwriter/doc/hbase11xwriter.md) |
|  | MongoDB | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/mongoreader/doc/mongoreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/mongowriter/doc/mongowriter.md) |
|  | Hive | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/hdfsreader/doc/hdfsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/hdfswriter/doc/hdfswriter.md) |
| 无结构化数据存储 | TxtFile | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/txtfilereader/doc/txtfilereader.md) 、[写](https://github.com/alibaba/DataX/blob/master/txtfilewriter/doc/txtfilewriter.md) |
|  | FTP | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/ftpreader/doc/ftpreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/ftpwriter/doc/ftpwriter.md) |
|  | HDFS | √ | √ | [读](https://github.com/alibaba/DataX/blob/master/hdfsreader/doc/hdfsreader.md) 、[写](https://github.com/alibaba/DataX/blob/master/hdfswriter/doc/hdfswriter.md) |
|  | Elasticsearch |  | √ | [写](https://github.com/alibaba/DataX/blob/master/elasticsearchwriter/doc/elasticsearchwriter.md) |

DataX Framework提供了简单的接口与插件交互，提供简单的插件接入机制，只需要任意加上一种插件，就能无缝对接其他数据源。详情请看：[DataX数据源指南](https://github.com/alibaba/DataX/wiki/DataX-all-data-channels)

**四、DataX3.0核心架构**

DataX 3.0 开源版本支持单机多线程模式完成同步作业运行，本小节按一个DataX作业生命周期的时序图，从整体架构设计非常简要说明DataX各个模块相互关系。

[](https://cloud.githubusercontent.com/assets/1067175/17850849/aa6c95a8-6891-11e6-94b7-39f0ab5af3b4.png)

**核心模块介绍：**

1. DataX完成单个数据同步的作业，我们称之为Job，DataX接受到一个Job之后，将启动一个进程来完成整个作业同步过程。DataX Job模块是单个作业的中枢管理节点，承担了数据清理、子任务切分(将单一作业计算转化为多个子Task)、TaskGroup管理等功能。
2. DataXJob启动后，会根据不同的源端切分策略，将Job切分成多个小的Task(子任务)，以便于并发执行。Task便是DataX作业的最小单元，每一个Task都会负责一部分数据的同步工作。
3. 切分多个Task之后，DataX Job会调用Scheduler模块，根据配置的并发数据量，将拆分成的Task重新组合，组装成TaskGroup(任务组)。每一个TaskGroup负责以一定的并发运行完毕分配好的所有Task，默认单个任务组的并发数量为5。
4. 每一个Task都由TaskGroup负责启动，Task启动后，会固定启动Reader—>Channel—>Writer的线程来完成任务同步工作。
5. DataX作业运行起来之后， Job监控并等待多个TaskGroup模块任务完成，等待所有TaskGroup任务完成后Job成功退出。否则，异常退出，进程退出值非0

**DataX调度流程：**

举例来说，用户提交了一个DataX作业，并且配置了20个并发，目的是将一个100张分表的mysql数据同步到odps里面。 DataX的调度决策思路是：

1. DataXJob根据分库分表切分成了100个Task。
2. 根据20个并发，DataX计算共需要分配4个TaskGroup。
3. 4个TaskGroup平分切分好的100个Task，每一个TaskGroup负责以5个并发共计运行25个Task。

**五、DataX 3.0六大核心优势**

* **可靠的数据质量监控**
  + 完美解决数据传输个别类型失真问题

DataX旧版对于部分数据类型(比如时间戳)传输一直存在毫秒阶段等数据失真情况，新版本DataX3.0已经做到支持所有的强数据类型，每一种插件都有自己的数据类型转换策略，让数据可以完整无损的传输到目的端。

* + 提供作业全链路的流量、数据量�运行时监控

DataX3.0运行过程中可以将作业本身状态、数据流量、数据速度、执行进度等信息进行全面的展示，让用户可以实时了解作业状态。并可在作业执行过程中智能判断源端和目的端的速度对比情况，给予用户更多性能排查信息。

* + 提供脏数据探测

在大量数据的传输过程中，必定会由于各种原因导致很多数据传输报错(比如类型转换错误)，这种数据DataX认为就是脏数据。DataX目前可以实现脏数据精确过滤、识别、采集、展示，为用户提供多种的脏数据处理模式，让用户准确把控数据质量大关！

* **丰富的数据转换功能**

DataX作为一个服务于大数据的ETL工具，除了提供数据快照搬迁功能之外，还提供了丰富数据转换的功能，让数据在传输过程中可以轻松完成数据脱敏，补全，过滤等数据转换功能，另外还提供了自动groovy函数，让用户自定义转换函数。详情请看DataX3的transformer详细介绍。

* **精准的速度控制**

还在为同步过程对在线存储压力影响而担心吗？新版本DataX3.0提供了包括通道(并发)、记录流、字节流三种流控模式，可以随意控制你的作业速度，让你的作业在库可以承受的范围内达到最佳的同步速度。

"speed": {

"channel": 5,

"byte": 1048576,

"record": 10000

}

* **强劲的同步性能**

DataX3.0每一种读插件都有一种或多种切分策略，都能将作业合理切分成多个Task并行执行，单机多线程执行模型可以让DataX速度随并发成线性增长。在源端和目的端性能都足够的情况下，单个作业一定可以打满网卡。另外，DataX团队对所有的已经接入的插件都做了极致的性能优化，并且做了完整的性能测试。性能测试相关详情可以参照每单个数据源的详细介绍：[DataX数据源指南](https://github.com/alibaba/DataX/wiki/DataX-all-data-channels)

* **健壮的容错机制**

DataX作业是极易受外部因素的干扰，网络闪断、数据源不稳定等因素很容易让同步到一半的作业报错停止。因此稳定性是DataX的基本要求，在DataX 3.0的设计中，重点完善了框架和插件的稳定性。目前DataX3.0可以做到线程级别、进程级别(暂时未开放)、作业级别多层次局部/全局的重试，保证用户的作业稳定运行。

* + 线程内部重试

DataX的核心插件都经过团队的全盘review，不同的网络交互方式都有不同的重试策略。

* + 线程级别重试

目前DataX已经可以实现TaskFailover，针对于中间失败的Task，DataX框架可以做到整个Task级别的重新调度。

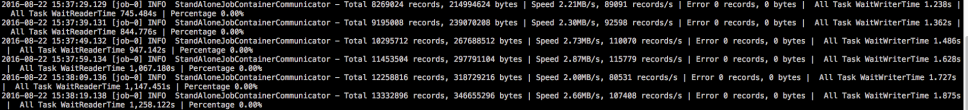
* **极简的使用体验**
  + 易用

下载即可用，支持linux和windows，只需要短短几步骤就可以完成数据的传输。请点击：[Quick Start](https://github.com/alibaba/DataX/wiki/Quick-Start)

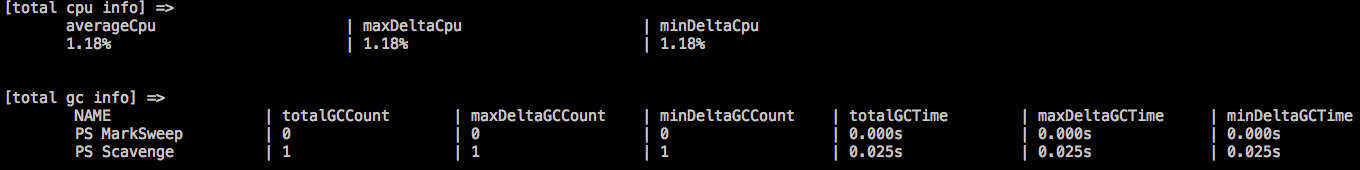
* + 详细

DataX在运行日志中打印了大量信息，其中包括传输速度，Reader、Writer性能，进程CPU，JVM和GC情况等等。

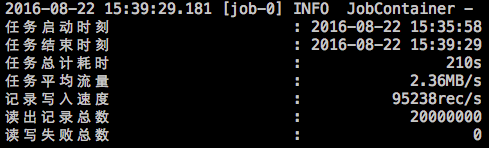
* + - 传输过程中打印传输速度、进度等

[](https://cloud.githubusercontent.com/assets/1067175/17850877/d1612c0a-6891-11e6-9970-d6693c15ef24.png)

* + - 传输过程中会打印进程相关的CPU、JVM等

[](https://cloud.githubusercontent.com/assets/1067175/17850903/ee63c2fe-6891-11e6-9056-97d7e3d13d8d.png)

* + - 在任务结束之后，打印总体运行情况

[](https://cloud.githubusercontent.com/assets/1067175/17850930/0484d3ac-6892-11e6-9c1d-b102ad210a32.png)