ALIBABA CLOUD

阿里云

大数据计算服务

数据迁移

文档版本：20220712

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.通用数据上传场景与工具

本文为您介绍如何将数据上传至MaxCompute或从MaxCompute下载数据，包括服务连接、SDK、工具和数据导入导出、上云等常见操作。

背景信息

MaxCompute提供了多种数据上传下载的通道支持，方便您在各种场景下进行技术方案选型时参考。

- 批量数据通道：支持批量上传及下载数据场景。
- 流式数据通道：提供了以流式的方式把数据写入MaxCompute的能力。
- 实时数据通道：DataHub是流式数据（Streaming Data）的处理平台，提供对流式数据的发布（Publish）、订阅（Subscribe）和分发功能，支持流式数据归档至MaxCompute。

功能介绍

- 批量数据通道上传

  使用批量数据通道上传数据时，可以通过单个批量操作将数据上传到MaxCompute中。例如上传数据源可以是外部文件、外部数据库、外部对象存储或日志文件。MaxCompute中批量数据通道上传包含如下方案。

  - Tunnel SDK：您可以通过Tunnel向MaxCompute中上传数据。
  - 数据同步服务：您可以通过数据集成（DataWorks）任务，提取、转换、加载（ETL）数据到MaxCompute。
  - 数据投递：您可以通过DataHub、SLS、Kafka版服务的MaxCompute Sink Connector、Blink将数据投递至MaxCompute。
  - 开源工具及插件：您可以通过Sqoop、Kettle、Flume、Fluentd插件、OGG、MMA将数据上传至MaxCompute。
  - 产品工具：MaxCompute客户端基于批量数据通道的SDK，实现了内置的Tunnel命令，可对数据进行上传，Tunnel命令的使用请参见Tunnel命令。

  > **说明** 对于离线数据的同步，推荐您优先使用数据集成，详情请参见数据集成概述。

- 流式数据通道写入

  MaxCompute流式数据通道服务提供了以流式的方式将数据写入MaxCompute的能力，使用与原批量数据通道服务不同的一套全新的API及后端服务。流式数据写入到MaxCompute的方案如下。

  - SDK接口：提供流式语义API，通过流式服务的API可以方便的开发出分布式数据同步服务。
  - 数据同步服务：您可以通过数据集成实时同步任务实现流式数据写入（StreamX）。
  - 数据投递：您可以通过已集成流式写入API的数据投递模式实现流式数据写入。支持SLS、消息队列Kafka版方式。
  - 数据采集：MaxCompute支持将开源Logstash收集的日志数据流式写入MaxCompute。
  - 插件支持：MaxCompute提供了使用流式数据通道的Flink插件，支持使用Flink在高并发、高QPS场景下写入MaxCompute。

- 数据下载

  MaxCompute提供了多种数据下载通道支持，方便您在各种场景下进行技术方案选型时参考。

  - MaxCompute客户端基于批量数据通道的SDK，实现了内置的Tunnel命令，可对数据进行下载，Tunnel命令的使用请参见Tunnel命令。

- Tunnel SDK是MaxCompute的数据通道，您可以通过Tunnel从MaxCompute中下载数据，支持单线程、多线程接口实现。
- 数据同步服务：您可以通过数据集成实现从MaxCompute下载数据到本地或外部数据源。
- 数据投递：你可以通过SLS实现从MaxCompute下载数据到外部数据源。

基于上述丰富的数据上传、下载的工具，可以满足大部分常见的数据上云场景，后续的章节会对工具本身以及Hadoop数据迁移、数据库数据同步、日志采集等数据上云的场景进行介绍，为您进行技术方案选型时提供参考。

> **说明** 对于上云场景，推荐您参考上云工具说明，详情请参见数据上云工具。

使用限制

- 批量数据通道使用限制说明
  - 批量数据上传
    - UploadSession生命周期：24小时。
    - 单UploadSession写入Block个数：20000个。
    - 单Block写入速度：10 MB/s。
    - 单Block写入数据量：100 GB。
    - 单表创建UploadSession数：每5分钟500个。
    - 单表写入Block数：每5分钟500个。
    - 单表并发提交UploadSession数：32个。
    - 并发写入Block数：受Slot并发数限制，单次Block写入占用一个Slot。
    - 当遇到并发写入时，MaxCompute会根据ACID进行并发写的保障。关于ACID的具体语义请参见ACID语义
  - 批量数据下载
    - DownloadSession生命周期：24小时。
    - InstanceDownloadSession生命周期：24小时，受实例生命周期限制。
    - 单Project创建InstanceDownloadSession数：每5分钟200个。
    - 单表创建DownloadSession数：每5分钟200个。
    - 单次下载请求速度：10MB/s。
    - 并发创建DownloadSession数：受Slot并发数限制，单次创建DownloadSession占用一个Slot。
    - 并发创建InstanceDownloadSession数：受Slot并发数限制，单次创建InstanceDownloadSession占用一个Slot。
    - 并发下载请求数：受Slot并发数限制，单次数据下载请求占用一个Slot。
- 流式数据通道使用限制说明
  - 单Slot写入速度：1MB/s。
  - 单Slot写入请求数：每秒10个。
  - 单表并发写入分区数：64个。
  - 单分区最大可用Slot数：32个。
  - StreamUploadSession占用Slot数：受并发Slot并发数限制，创建StreamUploadSession时指定Slot数。
- DataHub上传数据限制

- 每个字段的大小不能超过这个字段本身的限制，详情请参见数据类型版本说明。

  > 说明 STRING的长度不能超过8 MB。

- 上传的过程中会将多条数据打包成一个Package进行上传。

共享资源说明

下表数据为不同区域下免费共享资源（单位：Slot）Project级最多可用Slot数说明。

Region	城市	Slot（个数）
中国	华东1（杭州）	300
中国	华东2（上海）	600
中国	华东2金融云（上海）	50
中国	华北2（北京）	300
中国	华北2政务云（北京）	100
中国	华北3（张家口）	300
中国	华南1（深圳）	150
中国	华南1金融云（深圳）	50
中国	西南1（成都）	150
中国	中国（香港）	50
亚太	新加坡（新加坡）	100
亚太	澳大利亚（悉尼）	50
亚太	马来西亚（吉隆坡）	50
亚太	印度尼西亚（雅加达）	50
亚太	日本（东京）	50
欧洲与美洲	德国（法兰克福）	50
欧洲与美洲	美国（硅谷）	100
欧洲与美洲	美国（弗吉尼亚）	50
欧洲与美洲	英国（伦敦）	50
中东与印度	印度（孟买）	50
中东与印度	阿联酋（迪拜）	50

> **说明** 如您有临时高资源使用需求，可提工单申请临时上调Slot数，系统会根据可用资源情况确认是否通过工单申请，上调Slot资源可用时间最长不超过一周。

有效状态码

状态码标识	状态码名称
200	HTTP_OK
201	HTTP_CREATED
400	HTTP_BAD_REQUEST
401	HTTP_UNAUTHORIZED
403	HTTP_FORBIDDEN
404	HTTP_NOT_FOUND
405	HTTP_METHOD_NOT_ALLOWED
409	HTTP_CONFLICT
422	HTTP_UNPROCESSABLE_ENTITY
429	HTTP_TOO_MANY_REQUESTS
499	HTTP_CLIENT_CLOSED_REQUEST
500	HTTP_INTERNAL_SERVER_ERROR
502	HTTP_BAD_GATEWAY
503	HTTP_SERVICE_UNAVAILABLE
504	HTTP_GATEWAY_TIME_OUT

注意事项

网络因素对Tunnel上传下载速度的影响较大，正常情况下速度范围为1 MB/s~20 MB/s。当上传的数据量较大时，建议配置Tunnel Endpoint为经典网络或VPC网络相应的Tunnel Endpoint。经典网络或VPC网络需要通过阿里云ECS连通或者通过网络专线开通。如果上传数据速度太慢，可以考虑使用多线程上传方式。

更多Tunnel Endpoint信息，请参见Endpoint。

2.上传数据通用流程

不同网络环境下，您需要选择不同的服务地址（Endpoint）来连接服务，否则将无法向服务发起请求。

DataHub和Tunnel在不同网络环境场景下，所使用的EndPoint会有所区别。您在不同网络环境下，需要选择不同的服务地址（Endpoint）来连接服务，否则将无法向服务发起请求。同时，不同的网络连接也会对您的计费产生影响。

具体的服务连接地址请参见Endpoint。

3.上传数据

3.1. 数据上云场景

MaxCompute平台提供了丰富的数据上传下载工具，可以广泛应用于各种数据上云的应用场景，本文为您介绍三种经典数据上云场景。

Hadoop数据迁移

您可使用MMA、Sqoop和DataWorks进行Hadoop数据迁移。

- 使用DataWorks结合DataX进行Hadoop数据迁移的示例请参见Hadoop数据迁移新手教程，或参见视频教程Hadoop数据迁移到MaxCompute最佳实践。
- Sqoop执行时，会在原来的Hadoop集群上执行MR作业，可以分布式地将数据传输到MaxCompute上，详情请参见Sqoop工具的介绍。
- MMA利用Meta Carrier连接您的Hive Metastore服务，获取Hive Metadata，并利用这些数据生成用于创建MaxCompute表和分区的DDL语句以及用于迁移数据的Hive UDTF SQL。详细信息请参见MMA概述。

数据库数据同步

数据库的数据同步到MaxCompute需要根据数据库的类型和同步策略来选择相应的工具。

- 离线批量的数据库数据同步：可以选择DataWorks，支持的数据库种类丰富，包括MySQL、SQL Server、PostgreSQL等，详情请参见离线同步节点。您也可以参见创建同步任务进行实例操作。
- Oracle数据库数据实时同步时，可以选择OGG插件。
- RDS数据库数据实时同步时，可以选择DataWorks的数据集成，详情请参见配置数据源（来源为MySQL）。

日志采集

日志采集时，您可以选用Flume、Fluentd、Logstash等工具。具体场景示例请参见Flume收集网站日志数据到MaxCompute和海量日志数据分析与应用。

3.2. 数据上云工具

MaxCompute平台支持丰富的数据上传和下载工具（其中大部分工具已经在GitHub公开源代码，以开源社区的方式进行维护）。您可以根据实际应用场景，选择合适的工具进行数据的上传和下载。

阿里云数加产品

- MaxCompute客户端（Tunnel通道系列）
  - 客户端基于批量数据通道的SDK，实现了内置的Tunnel命令，可对数据进行上传和下载，Tunnel命令的使用请参见Tunnel命令的基本使用介绍。
  - 客户端的安装和基本使用方法请参见客户端介绍。

  > **说明** 该项目已经开源，您可进入aliyun-odps-console进行查看 。

- DataWorks数据集成（Tunnel通道系列）

  DataWorks数据集成（即数据同步），是一个稳定高效、弹性伸缩的数据同步平台，致力于为阿里云上各类异构数据存储系统提供离线全量和实时增量的数据同步、集成、交换服务。

  其中数据同步任务支持的数据源类型包括：MaxCompute、RDS（MySQL、SQL Server、PostgreSQL）、Oracle、FTP、ADS（AnalyticDB）、OSS、Memcache和DRDS，详情请参见数据集成概述。

- DTS（Tunnel通道系列）

  什么是数据传输服务DTS是阿里云提供的一种支持RDBMS（关系型数据库）、NoSQL、OLAP等多种数据源之间数据交互的数据服务。它提供了数据迁移、实时数据订阅及数据实时同步等多种数据传输功能。

  DTS可以支持RDS、MySQL实例的数据实时同步到MaxCompute表中，暂不支持其他数据源类型。详情请参见创建RDS到MaxCompute数据实时同步作业。

开源产品

- Sqoop（Tunnel通道系列）

  Sqoop基于社区Sqoop 1.4.6版本开发，增强了对MaxCompute的支持，可以将数据从MySQL等关系数据库导入或导出到MaxCompute表中，也可以从HDFS或Hive导入数据到MaxCompute表中。详情请参见MaxCompute Sqoop。

  > **说明** 该项目已经开源，您可进入aliyun-maxcompute-data-collectors进行查看。

- Kettle（Tunnel通道系列）

  Kettle是一款开源的ETL工具，纯Java实现，可以在Windows、Unix和Linux上运行，提供图形化的操作界面，可以通过拖拽控件的方式，方便地定义数据传输的拓扑 。详情请参见基于Kettle的MaxCompute插件实现数据上云。

  > **说明** 该项目已经开源，您可进入aliyun-maxcompute-data-collectors进行查看。

- Flume（DataHub通道系列）

  Apache Flume是一个分布式的、可靠的、可用的系统，可高效地从不同的数据源中收集、聚合和移动海量日志数据到集中式数据存储系统，支持多种Source和Sink插件。

  Apache Flume的DataHub Sink插件可以将日志数据实时上传到DataHub，并归档到MaxCompute表中。详情请参见flume_plugin。

  > **说明** 该项目已经开源，您可进入aliyun-maxcompute-data-collectors进行查看。

- Fluentd（DataHub通道系列）

  Fluentd是一个开源的软件，用来收集各种源头日志（包括Application Log、Sys Log及Access Log），允许您选择插件对日志数据进行过滤，并存储到不同的数据处理端（包括MySQL、Oracle、MongoDB、Hadoop、Treasure Data等）。

  Fluentd的DataHub插件可以将日志数据实时上传到DataHub，并归档到MaxCompute表中。详情请参见Fluentd插件介绍。

- LogStash（DataHub通道系列）

  LogStash是一款开源日志收集处理框架，logstash-output-datahub插件实现了将数据导入DataHub的功能。通过简单的配置即可完成数据的采集和传输，结合MaxCompute和StreamCompute可以轻松构建流式数据从采集到分析的一站式解决方案。

  LogStash的DataHub插件可以将日志数据实时上传到DataHub，并归档到MaxCompute表中。具体示例请参见Logstash + DataHub + MaxCompute和StreamCompute 进行实时数据分析。

- OGG（DataHub通道系列）

  OGG的DataHub插件可以支持将Oracle数据库的数据实时地以增量方式同步到DataHub中，并最终归档到MaxCompute表中。详情请参见基于OGG DataHub插件将Oracle数据同步上云。

> **说明** 该项目已经开源，您可进入aliyun-maxcompute-data-collectors进行查看。

- MMA迁移工具

  MMA利用Meta Carrier连接用户的Hive Metastore服务，抓取用户的Hive Metadata，并利用这些数据生成用于创建MaxComputer表和分区的DDL语句以及用于迁移数据的Hive UDTF SQL。详情请参见MMA概述。

3.3. 使用DataWorks（离线与实时）

MaxCompute支持通过DataWorks的数据集成功能将其他数据源的数据以离线或实时方式导入MaxCompute。当您需要将其他数据源的数据导入MaxCompute中执行后续数据处理操作时，您可以使用数据集成功能导入数据。本文为您介绍如何将其他数据源的数据导入MaxCompute。

背景信息

数据集成的导入方式分为离线导入和实时导入两种：

- 离线导入

  您可以通过如下模式将其他数据源的数据导入MaxCompute：

  - 向导模式：创建离线同步节点后，在DataWorks界面以可视化方式配置数据来源、去向及字段的映射关系等信息，完成数据导入操作。
  - 脚本模式：创建离线同步节点后，将DataWorks可视化界面切换至脚本模式，通过脚本配置数据来源、去向及字段的映射关系等信息，完成数据导入操作。

- 实时导入

  您可以通过如下方式将其他数据源的数据导入MaxCompute：

  - 同步单表数据：将其他数据源的数据导入至MaxCompute的某一张表中。
  - 同步整库数据：将其他数据源中全部表数据导入至MaxCompute中。
  - 一键实时同步数据：将实时或离线数据一键同步至MaxCompute中。

前提条件

请确认您已完成如下操作：

- 已准备好待导入MaxCompute的数据源及表。
- 已准备好目标MaxCompute项目。

  更多创建MaxCompute项目操作，请参见创建MaxCompute项目。

使用限制

离线导入场景中，每个离线同步节点仅支持将单张或多张表数据导入至MaxCompute的一张表中。

离线导入

1. 添加MaxCompute数据源。
   i. 登录DataWorks控制台。
   ii. 在左侧导航栏，单击**工作空间列表**。
   iii. 选择工作空间所在地域后，单击相应工作空间后的**进入数据集成**。
   iv. 在左侧导航栏，单击**数据源 > 数据源列表**，进入**工作空间管理 > 数据源管理**页面。
   v. 在**数据源管理**页面，单击右上角的**新增数据源**。

vi. 在**新增数据源**对话框中，选择数据源类型为**MaxCompute（ODPS）**。

vii. 在**新增MaxCompute（ODPS）数据源**对话框中，配置各项参数。

参数	描述
数据源名称	数据源名称必须以字母、数字、下划线（_）组合，且不能以数字和下划线（_）开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择**开发**或**生产**环境。 **说明** 仅标准模式工作空间会显示该配置。
ODPS Endpoint	默认只读，从系统配置中自动读取。
Tunnel Endpoint	MaxCompute Tunnel服务的连接地址，详情请参见Endpoint。
ODPS项目名称	MaxCompute（ODPS）项目名称。
AccessKey ID	访问密钥中的AccessKey ID，您可以进入用户信息管理页面进行复制。

参数	描述
AccessKey Secret	访问密钥中的AccessKey Secret，相当于登录密码。

viii. 选择**资源组连通性**类型为**数据集成**。

ix. 在资源组列表，单击相应资源组后的**测试连通性**。

数据同步时，一个任务只能使用一种资源组。您需要测试每种资源组的连通性，以保证同步任务使用的数据集成资源组能够与数据源连通，否则将无法正常执行数据同步任务。如果您需要同时测试多种资源组，请选中相应资源组后，单击**批量测试连通性**。详情请参见配置资源组与网络连通。

> 说明
> - （推荐）资源组列表默认仅显示独享数据集成资源组，为确保数据同步的稳定性和性能要求，推荐使用独享数据集成资源组。
> - 如果您需要测试公共资源组或自定义资源组的连通性，请在资源组列表右下方，单击**更多选项**，在**警告**对话框单击**确定**，资源组列表会显示可供选择的公共资源组和自定义资源组。

x. 测试连通性通过后，单击**完成**。

2. 添加待导出数据源。

请根据MaxCompute导出的目标数据源类型，完成添加数据源操作。更多添加数据源操作，请参见配置数据源。

3. 创建业务流程。

i. 登录DataWorks控制台。

ii. 在左侧导航栏，单击**工作空间列表**。

iii. 选择工作空间所在地域后，单击相应工作空间后的**进入数据开发**。

iv. 在**数据开发**页面，鼠标悬停至图标，单击**业务流程**。

v. 在**新建业务流程**对话框中，输入**业务名称**和**描述**。

> 注意 业务名称必须是大小写字母、中文、数字、下划线（_）以及小数点（.），且不能超过128个字符。

vi. 单击**新建**。

4. 创建离线同步节点。

i. 展开业务流程，右键单击**数据集成**。

ii. 单击**新建 > 离线同步**。

iii. 在**新建节点**对话框中，输入**节点名称**，并选择**目标文件夹**。

> 注意 节点名称必须是大小写字母、中文、数字、下划线（_）以及小数点（.），且不能超过128个字符。

iv. 单击**提交**。

5. 配置并运行数据同步任务。

- 如果您采用向导模式配置并运行数据同步任务，转6。

- 如果您采用脚本模式配置并运行数据同步任务，转7。

6. 通过向导模式配置并运行数据同步任务。

   i. 通过向导模式配置离线同步任务。

   在数据来源下的**数据源**下拉列表选择数据源类型为待导入的数据源类型及数据源名称，在**表**下拉列表选择待导入数据的表。

   ii. 通过向导模式配置离线同步任务。

   在数据去向下的**数据源**下拉列表选择数据源类型为**ODPS**及目标MaxCompute数据源名称，在**表**下拉列表选择目标表。

   iii. 通过向导模式配置离线同步任务。

   指定数据来源表和数据去向表间字段的映射关系。

iv. 通过向导模式配置离线同步任务。

v. 通过向导模式配置离线同步任务。

在同步任务中配置调度参数进行数据过滤。

vi. 在顶部菜单栏，单击图标后，单击图标，运行离线同步任务。

7. 通过脚本模式配置并运行数据同步任务。

i. 通过脚本模式配置离线同步任务。

在**来源类型**下拉列表选择待导入数据源类型，对应**数据源**为待导入数据源名称。在**目标类型**下拉列表选择**ODPS**，对应**数据源**为创建好的MaxCompute数据源名称。

ii. 通过脚本模式配置离线同步任务。

在脚本中配置离线同步任务读取的数据源，以及需要同步的表信息等。

```
{
            "stepType": "mysql",
            "parameter": {
                "partition": [],
                "datasource": "",
                "envType": 0,
                "column": [
                    "*"
                ],
                "table": ""
            },
            "name": "Reader",
            "category": "reader"
        },
```

- stepType：待导入数据源的类型。
- partition：待导入表的分区信息。
- datasource：待导入数据源的名称。
- column：待导入表的列名称。需要与写入端MaxCompute中配置的列名称建立一一对应关系。
- table：待导入表的名称。
- name和category：取值为Reader，标识数据源为读取端。

iii. 通过脚本模式配置离线同步任务。

在脚本中配置离线同步任务写入的数据源，以及需要写入的表信息等。

```
{
            "stepType":"odps",
            "parameter":{
                "partition":"",
                "truncate":true,
                "datasource":"odps_first",
                "column":[
                    "*"
                ],
                "table":""
            },
            "name":"Writer",
            "category":"writer"
        }
```

- stepType：目标数据源类型。设置为odps。
- partition：目标表的分区信息。您可以通过 `show partitions <table_name>;` 命令，查看表的分区信息。更多查看分区信息，请参见查看分区。
- datasource：MaxCompute数据源的名称。
- column：目标表的列名称。
- table：目标表的名称。您可以通过 `show tables;` 命令，查看表的名称。更多查看表信息，请参见表操作。
- name和category：取值为Writer，标识数据源为写入端。

iv. 通过向导模式配置离线同步任务。

```
"setting": {
        "errorLimit": {
            "record": "1024"
        },
        "speed": {
            "throttle": false,
            "concurrent": 1
        }
    },
```

- record：脏数据的最大容忍条数。
- throttle：设置是否进行限速。
- concurrent：设置离线同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。

v. 通过向导模式配置离线同步任务。

vi. 在顶部菜单栏，单击图标后，单击图标，运行同步任务。

8. 请前往MaxCompute数据源中确认MaxCompute表中是否已成功导入数据。
   - 如果数据完整无遗漏，则同步完成。
   - 如果数据未同步成功或数据存在遗漏，请参见离线同步常见问题。

同步单表数据（实时导入）

1. 进入**数据开发**页面。
   i. 登录DataWorks控制台。
   ii. 在左侧导航栏，单击**工作空间列表**。
   iii. 选择工作空间所在地域后，单击相应工作空间后的**进入数据开发**。
2. 鼠标悬停至+新建图标，单击**数据集成 > 实时同步**。

   您也可以展开目标业务流程，右键单击**数据集成**，选择**新建 > 实时同步**。
3. 在**新建节点**对话框中，选择同步方式为**单表（Topic）到单表（Topic）ETL**，输入**节点名称**，并选择**目标文件夹**。

   > **注意** 节点名称必须是大小写字母、中文、数字、下划线（_）以及英文句号（.），且不能超过128个字符。
4. 单击**提交**。
5. 在实时同步节点的编辑页面，单击**输出 > MaxCompute**并拖拽至编辑面板，连线已配置好的输入或转换节点。
6. 单击**MaxCompute**节点，在**节点配置**对话框中，配置各项参数。

参数	描述
数据源	选择已经配置好的MaxCompute数据源，此处仅支持MaxCompute数据源。 如果您未配置数据源，请单击右侧的**新建数据源**，进入**工作空间管理 > 数据源管理**页面新建，详情请参见配置MaxCompute数据源。
表	选择当前数据源下需要同步的表名称。 您可以单击右侧的**一键建表**创建新表，也可以单击**数据预览**进行确认。 **注意** 新建目标数据表前，请先连线输入节点，并确认有输出字段。
分区方式	包括**时间自动分区**及**根据字段内容动态分区**。其中**时间自动分区**是根据_*execute_time*_字段进行分区的，详情请参见实时同步字段格式。**根据字段内容动态分区**通过指定源端表某字段与目标MaxCompute表分区字段对应关系，实现源端对应字段所在数据行写入到MaxCompute表对应的分区中。
分区讯息	为您展示MaxCompute分区表的信息。
字段映射	单击**字段映射**，设置源端和目标端字段的映射。同步任务会根据字段的映射关系同步数据。

如果您需要新建表，请单击**一键建表**后，在**新建数据表**对话框中，配置各项参数。

参数	描述
表名称	实时同步写入的MaxCompute表的名称。
生命周期	实时同步写入的MaxCompute表的生命时间长度，详情请参见生命周期。
数据字段结构	实时同步写入的MaxCompute表的字段结构。如果您需要新增字段，请单击**添加**。

参数	描述
分区设置	实时同步写入的MaxCompute表的分区信息。 实时同步写入MCompute表支持**时间自动分区**与**根据字段内容动态分区**两种分区方式 ○ **时间自动分区**：根据_*execute_time*_字段将数据写入到对应时间分区中，详情请参见实时同步字段格式，  注意 ■ 您最少需要设置二级分区（月和年），最多支持设置五级分区（分钟、小时、天、月和年）。 ■ 关于MaxCompute表的介绍可参考文档：分区 ○ **根据字段内容动态分区**：通过指定源端表某字段与目标MaxCompute表分区字段对应关系，实现源端对应字段所在数据行写入到MaxCompute表对应的分区中。  例如：配置MaxCompute表分区字段值来源为源端字段A，当A字段值为aa时，实时同步会将数据写入到MaxCompute表对应的aa分区中，当A字段值为bb时，实时同步会将数据写入到MaxCompute表对应的bb分区中。

7. 单击工具栏中的图标。

同步整库数据（实时导入）

1. 创建实时同步任务。
2. 提交并发布实时同步任务。
3. 执行实时同步任务。

一键实时同步数据（实时导入）

1. 创建同步解决方案任务。
2. 一键实时同步至MaxCompute。

3.4. 使用Kafka（离线与实时）

本文为您介绍如何将消息队列Kafka版数据导入MaxCompute。

背景信息

消息队列Kafka版是阿里云基于Apache Kafka构建的高吞吐量、高可扩展性的分布式消息队列服务，广泛用于日志收集、监控数据聚合、流式数据处理、在线和离线分析等，是大数据生态中不可或缺的产品之一，阿里云提供全托管服务，用户无需部署运维，更专业、更可靠、更安全。

MaxCompute与消息队列Kafka版服务紧密集成，借助消息队列Kafka版服务的MaxCompute Sink Connector，无需第三方工具及二次开发，即可满足将指定Topic数据持续导入MaxCompute数据表的需求。极大简化Kafka消息队列数据进入MaxCompute的集成链路，并显著降低开发和运维成本。

操作方法

您需要创建MaxCompute Sink Connector，并将数据从消息队列Kafka版实例的数据源Topic导出至MaxCompute的表，操作详情请参见创建MaxCompute Sink Connector。

3.5. 使用Logstash（流式数据传输）

MaxCompute支持将开源Logstash收集的日志数据写入MaxCompute。您可以通过Logstash的输出插件 `logstash-output-maxcompute` ，将Logstash收集的日志数据使用MaxCompute流式数据通道（Streaming Tunnel）功能上传到MaxCompute。

前提条件

在执行操作前请确认您已完成如下操作：

- 已安装Logstash并创建Logstash日志收集实例。

  更多信息，请参见Getting Started with Logstash。

- 已创建目标MaxCompute项目。

  更多创建MaxCompute项目信息，请参见创建MaxCompute项目。

背景信息

Logstash是一个开源的服务器端数据处理管道，可以同时从多个数据源获取数据，并对数据进行转换，然后将转换后的数据发送到用户的目标“存储端”。

您需要通过Logstash的 `logstash-output-maxcompute` 插件，将Logstash收集的日志数据使用MaxCompute流式数据通道（Streaming Tunnel）功能上传到MaxCompute。

`logstash-output-maxcompute` 插件基于Logstash v7.8.0版本开发，可以作为输出端口。该插件的特点如下：

- 使用流式数据通道，避免通过批量数据通道导入产生的并发和小文件问题。
- 支持动态分区，可以根据Logstash解析的日志字段产生分区字段，能够自动创建不存在的分区。

`logstash-output-maxcompute` 插件应用于如下场景：

- 需要收集的应用的日志格式在Logstash上有输入插件支持或易于解析，例如NGINX日志。
- 希望根据日志内容自动创建并导入对应分区。

`logstash-output-maxcompute` 插件支持的数据类型为：STRING、BIGINT、DOUBLE、DATETIME和BOOLEAN。

> **说明**
> - 日志中DATETIME类型的字段的格式将自动使用 `ruby Time.parse` 函数推断。
> - 如果日志BOOLEAN字段满足 `.to_string().lowercase() == "true"` ，则结果为True。其他任何值为False。

本文将以收集NGINX日志为例，介绍如何配置和使用插件。

步骤一：下载并安装插件

您可以下载已安装 `logstash-output-maxcompute` 插件的Logstash实例，跳过安装步骤执行下一步。如果需要自行安装，请按照如下步骤操作：

1. 下载logstash-output-maxcompute插件并放置在Logstash的根目录 `%logstash%` 下。
2. 修改Logstash根目录 `%logstash%` 下的 `Gemfile` 文件，将 `source "https://rubygems.org"` 替换为 `source 'https://gems.ruby-china.com'` 。
3. 以Windows系统为例，在系统的命令行窗口，切换至Logstash的根目录 `%logstash%` 下，执行如下命令安装 `logstash-output-maxcompute` 插件。

   ```
   bin\logstash-plugin install logstash-output-maxcompute-1.1.0.gem
   ```

   当返回 `Installation successful` 提示信息时，表示插件安装成功。

   ```
   D:\logstash>
   D:\logstash>bin\logstash-plugin install logstash-output-maxcompute-1.1.0.gem
   Validating logstash-output-maxcompute-1.1.0.gem
   Installing logstash-output-maxcompute
   Installation successful
   ```

4. （可选）运行如下命令验证安装结果。

   ```
   bin\logstash-plugin list maxcompute
   ```

   > **说明** Linux系统需要执行命令 `bin/logstash-plugin list maxcompute` 。

   如果安装成功，会返回 `logstash-output-maxcompute` 信息。如果安装失败，解决方案请参见RubyGems。

   ```
   D:\logstash>bin\logstash-plugin list maxcompute
   logstash-output-maxcompute
   ```

步骤二：创建目标表

通过MaxCompute客户端或其他可以运行MaxCompute SQL的工具执行如下命令，在目标MaxCompute项目中创建目标表，例如 `logstash_test_grokngin x` 。后续会将日志信息以日期为分区导入此表中。

```
create table logstash_test_grokngin x(
 clientip string,
 remote_user string,
 time datetime,
 verb string,
 uri string,
 version string,
 response string,
 body_bytes bigint,
 referrer string,
 agent string
) partitioned by (pt string);
```

步骤三：编写Logstash Pipeline配置文件

在Logstash的根目录 `%logstash%` 下创建配置文件pipeline.conf，并输入如下内容：

```
input { stdin {} }
filter {
        grok {
                match => {
                        "message" => "%{IP:clientip} - (%{USER:remote_user}|-) \[%{HTTPDATE:ht
tptimestamp}\] \"%{WORD:verb} %{NOTSPACE:request} HTTP/%{NUMBER:httpversion}\" %{NUMBER:respon
se} %{NUMBER:body_bytes} %{QS:referrer} %{QS:agent}"
                }
        }
        date {
                match => [ "httptimestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
                target => "timestamp"
        }
}
output {
       maxctunnel {
                aliyun_access_id => "<your_accesskey_id>"
                aliyun_access_key => "<your_accesskey_secret>"
                aliyun_mc_endpoint => "<your_project_endpoint>"
                project => "<your_project_name>"
                table => "<table_name>"
                partition => "pt=$<timestamp.strftime('%F')>"
                value_fields => ["clientip", "remote_user", "timestamp", "verb", "request", "h
ttpversion", "response", "bytes", "referrer", "agent"]
        }
}
```

参数	说明
your_accesskey_id	可以访问目标MaxCompute项目的AccessKey ID。
your_accesskey_secret	AccessKey ID对应的AccessKey Secret。

参数	说明
your_project_endpoint	目标MaxCompute项目所在区域的Endpoint信息。更多Endpoint信息，请参见Endpoint。
your_project_name	目标MaxCompute项目的名称。
table_name	目标表的名称，即步骤二中创建的表。
partition	配置插件如何根据日志字段生成对应的分区信息。如果目标表有多个分区，需要指定到最后一级。配置格式如下： • 如果某个分区的值为常量，格式为 `{分区列名}={常量值}` 。 • 如果某个分区的值为解析后的日志中一个字段的值，格式为 `{分区列名}=$<{日志字段名}>` 。 • 如果某个分区的值为解析后的日志中一个日期时间字段的值，并且需要进行重新格式化，格式为 `{分区列名}=$<{日志字段名}.strftime('{时间格式}')>` 。其中： `{时间格式}` 是重新格式化的格式化字符串。 在本示例中，将格式化到仅保留日期（%F）。如果要按照日期 `date` 作为第一级分区，小时 `hour` 作为第二级分区，配置格式为 `"date=$<timestamp.strftime('%F')>,hour=$<timestamp.strftime('%H')>"` 。 • 多级分区之间用英文逗号（,）连接，分区指定的顺序和建表时的顺序必须一致。
partition_time_format	可选。指定当一个字符串型的日期时间字段被分区信息引用时，该字段的源格式字符串。 在本例中，时间字段 `timestamp` 已经被 `date` 插件转换为时间类型，因此不需指定。 即使未使用 `date` 过滤插件进行转换，亦未指定此配置项的值，在大多数情况下插件仍然可以自动识别内容为日期时间的字符串，并自动完成需要的转换。即只在少数自动识别失败的情况下需要手动指定此项的值。 如果不使用 `date` 过滤插件，而是手动进行转换，则需要配置如下信息： • 手动指定partition_time_format： `partition_time_format => "%d/%b/%Y:%H:%M:%S %z"` 。 • 将分区中引用的字段改为日志中的字符串字段： `partition => "pt=$<httptimestamp.strftime('%F')>"` 。
value_fields	指定目标表中的每个字段对应的日志字段，指定顺序与表中字段的顺序一致。 目标表字段的顺序为 `clientip string、remote_user string、time datetime、verb string、uri string、version string、response string、body_bytes bigint、referrer string、agent string` ，依次对应 `"clientip"、"remote_user"、"timestamp"、"verb"、"request"、"httpversion"、"response"、"bytes"、"referrer"、"agent"` 。
aliyun_mc_tunnel_endpoint	可选。您可以通过此配置项强制指定Tunnel Endpoint，覆盖自动路由机制。
retry_time	失败重试次数。当写入MaxCompute失败时，尝试重新写入的次数。默认值为3。

参数	说明
retry_interval	失败重试间隔。在两次尝试之间最少间隔的时间，单位为秒。默认值为1。
batch_size	一次最多处理的日志条数。默认值为100。
batch_timeout	写入MaxCompute的超时时间，单位为秒。默认值为5。

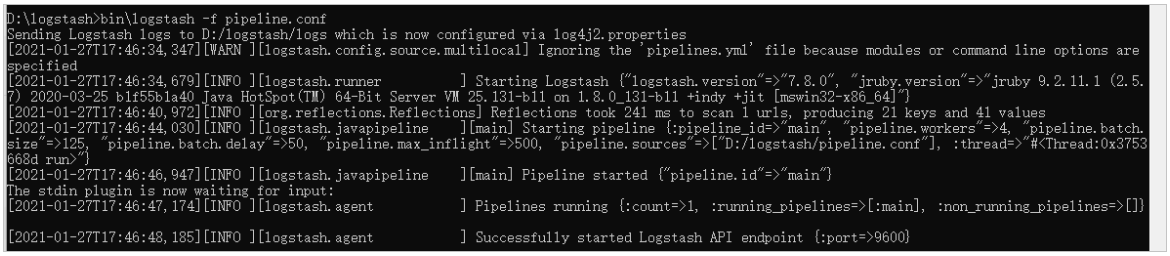
说明 在本配置文件中，指定的日志输入为标准输入 `(input { stdin {} })` 。在实际应用场景中，您可以使用Logstash File输入插件从本地硬盘中自动读取NGINX日志。更多信息，请参见Logstash文档。

步骤四：运行和测试

1. 以Windows系统为例，在系统的命令行窗口，切换至Logstash的根目录 `%logstash%` 下，执行如下命令启动Logstash。

```
bin\logstash -f pipeline.conf
```

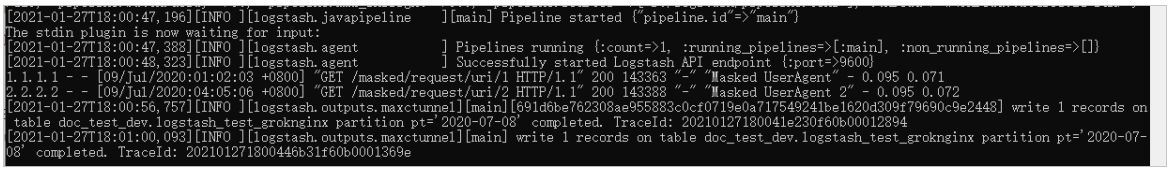
返回 `Successfully started Logstash API endpoint` 信息时，Logstash启动完毕。



2. 在系统的命令行窗口，粘贴如下日志样例，并按下键盘上的Enter键。

```
1.1.1.1 - - [09/Jul/2020:01:02:03 +0800] "GET /masked/request/uri/1 HTTP/1.1" 200 143363 "-" "Masked UserAgent" - 0.095 0.071
2.2.2.2 - - [09/Jul/2020:04:05:06 +0800] "GET /masked/request/uri/2 HTTP/1.1" 200 143388 "-" "Masked UserAgent 2" - 0.095 0.072
```

返回 `write .. records on partition .. completed` 时，表示成功写入MaxCompute。



3. 通过MaxCompute客户端或其他可以运行MaxCompute SQL的工具，执行如下命令，查询数据写入结果。

```
select * from logstash_test_grokngnix;
```

返回结果如下：

```
+--------+-----------+------------+-----------+-----------+-----------+-----------+-
----------+-----------+-----------+-----------+-----------+
| clientip   | remote_user | time       | verb       | uri        | version    | response
| body_bytes | referrer    | agent      | pt         |
+------------+-------------+------------+------------+------------+------------+----------
--+------------+------------+------------+------------+
| 1.1.1.1    | -           | 2020-07-09 01:02:03 | GET        | /masked/request/uri/1 | 1.
1        | 200        | 0          | "-"        | "Masked UserAgent" | 2020-02-10       |
| 2.2.2.2    | -           | 2020-07-09 04:05:06 | GET        | /masked/request/uri/2 | 1.
1        | 200        | 0          | "-"        | "Masked UserAgent 2" | 2020-02-10
|
+------------+-------------+------------+------------+------------+------------+----------
--+------------+------------+------------+------------+
2 records (at most 10000 supported) fetched by instance tunnel.
```

3.6. 使用阿里云Flink（流式数据传输）

实时计算Flink版内置插件支持通过批量数据通道写入MaxCompute，受到批量数据通道并发数及存储文件数影响，内置版本插件会有性能瓶颈。MaxCompute提供了使用流式数据通道的Flink插件，支持使用Flink在高并发、高QPS场景下写入MaxCompute。

前提条件

- 已开通实时计算Flink版的Blink服务并创建Blink项目。

  更多开通Blink及创建Blink项目的信息，请参见开通服务和创建项目。

- 已安装使用流式数据通道的Flink插件。

  更多插件安装信息，请参见自定义函数（UDX）。

背景信息

实时计算Flink版可以调用MaxCompute SDK中的接口将数据写入缓冲区，当缓冲区的大小超过指定的大小（默认为1 MB）或每隔指定的时间间隔时，将数据上传至MaxCompute结果表中。

> **说明** 建议Flink同步MaxCompute并发数大于32或Flush间隔小于60秒的场景下，使用MaxCompute自定义插件。其他场景可以随意选择Flink内置插件和MaxCompute自定义插件。

MaxCompute与实时计算Flink版的字段类型对照关系如下。

MaxCompute字段类型	实时计算Flink版字段类型
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE

MaxCompute字段类型	实时计算Flink版字段类型
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
STRING	VARCHAR
DECIMAL	DECIMAL
BINARY	VARBINARY

使用限制

该功能的使用限制如下：

- 本插件仅支持Blink 3.2.1及以上版本。
- MaxCompute中的聚簇表不支持作为MaxCompute结果表。

语法示例

您需要在Flink控制台新建作业，创建MaxCompute结果表。新建作业操作请参见开发。

> **说明** DDL语句中定义的字段需要与MaxCompute物理表中的字段名称、顺序以及类型保持一致，否则可能导致在MaxCompute物理表中查询的数据为 `/n` 。

命令示例如下：

```
create table odps_output(
    id INT,
    user_name VARCHAR,
    content VARCHAR
) with (
    type ='custom',
    class = 'com.alibaba.blink.customersink.MaxComputeStreamTunnelSink',
    endpoint = '<YourEndPoint>',
    project = '<YourProjectName>',
    `table` = '<YourtableName>',
    access_id = '<yourAccessKeyId>',
    access_key = '<yourAccessKeySecret>',
    `partition` = 'ds=2018****'
);
```

WITH参数

参数	说明	是否必填	备注
type	结果表的类型。	是	固定值为 `custom` 。

参数	说明	是否必填	备注
class	插件入口类。	是	固定值为 `com.alibaba.blink.customersink.MaxComputeStreamTunnelSink`。
endpoint	MaxCompute服务地址。	是	参见各地域Endpoint对照表（外网连接方式）。
tunnel_endpoint	MaxCompute Tunnel服务的连接地址。	否	参见各地域Endpoint对照表（外网连接方式）。 **说明** VPC环境下必填。
project	MaxCompute项目名称。	是	无
table	MaxCompute物理表名称。	是	无
access_id	可以访问MaxCompute项目的AccessKey ID。	是	无
access_key	AccessKey ID对应的AccessKey Secret。	是	无
partition	分区表的分区名称。	否	如果表为分区表则必填： • 固定分区 例如 `` `partition` = 'ds=20180905' `` 表示将数据写入分区 `ds= 20180905`。 • 动态分区 如果不明文显示分区的值，则会根据写入数据中的分区列具体的值，写入到不同的分区中。例如 `` `partition`='ds' `` 表示根据 `ds` 字段的值写入分区。 如果要创建多级动态分区，With参数中Partition的字段顺序和结果表的DDL中的分区字段顺序，必须与物理表一致，各个分区字段之间使用英文逗号（,）分隔。 **说明** ◦ 动态分区列需要显式写在建表语句中。 ◦ 对于动态分区字段为空的情况，如果数据源中 `ds=null` 或 `ds=''`，则会创建ds=NULL的分区。
enable_dynamic_partition	设置是否开启动态分区机制。	否	默认值为False。

参数	说明	是否必填	备注
dynamic_partition_limit	设置最大并发分区数。动态分区模式会为每个分区分配一个缓冲区，缓冲区大小通过flush_batch_size参数控制，所以动态分区模式最大会占用分区数量×缓冲区大小的内存。例如100个分区，每个分区1 MB，则最大占用内存为100 MB。	否	默认值为100。系统内存中会维护一个分区到Writer的Map，如果这个Map的大小超过了dynamicPartitionLimit的值，系统会通过LRU（Least Recently Used）的规则尝试淘汰没有数据写入的分区。如果所有分区都有数据写入，则会出现 `dynamic partition limit exceeded: 100` 报错。
flush_batch_size	数据缓冲区大小，单位字节。缓冲区数据写满后会触发Flush操作，将数据发送到MaxCompute。	否	默认值为1048576，即1 MB。
flush_interval_ms	缓冲区Flush间隔，单位毫秒。 MaxCompute Sink写入数据时，先将数据放到MaxCompute的缓冲区中，等缓冲区溢出或每隔一段时间（flush_interval_ms）时，再把缓冲区中的数据写到目标MaxCompute表。	否	默认值为-1，即不设置主动Flush间隔。
flush_retry_count	数据Flush失败重试次数，在缓冲区Flush失败的场景下自动重试。	否	默认值为10，即重试10次。
flush_retry_interval_sec	Flush失败重试的时间间隔，单位秒。	否	默认值为1，即1秒。

参数	说明	是否必填	备注
flush_retry_strategy	Flush失败重试策略，多次重试的时间间隔增长策略，配合flush_retry_interval_sec使用。包含如下三种策略： • `constant`：常数时间，即每次重试间隔使用固定时间间隔。 • `linear`：线性增长，即每次重试间隔时间线性增长，例如flush_retry_interval_sec设置为1，flush_retry_count设置为5，多次重试时间间隔为1、2、3、4、5秒。 • `exponential`：指数增长。例如flush_retry_interval_sec设置为1，flush_retry_count设置为5，多次重试中间间隔为1、2、4、8、16秒。	否	默认值为`constant`，即常数时间间隔。

类型映射

MaxCompute字段类型	实时计算Flink版字段类型
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP

MaxCompute字段类型	实时计算Flink版字段类型
VARCHAR	VARCHAR
STRING	VARCHAR
DECIMAL	DECIMAL

代码示例

包含MaxCompute结果表的实时计算Flink版作业代码示例如下：

- 写入固定分区

```
create table source (
   id INT,
   len INT,
   content VARCHAR
) with (
   type = 'random'
);
create table odps_sink (
   id INT,
   len INT,
   content VARCHAR
) with (
   type='custom',
   class = 'com.alibaba.blink.customersink.MaxComputeStreamTunnelSink',
   endpoint = '<yourEndpoint>',
   project = '<yourProjectName>',
   `table` = '<yourTableName>',
   accessId = '<yourAccessId>',
   accessKey = '<yourAccessPassword>',
   `partition` = 'ds=20180418'
);
insert into odps_sink
select
   id, len, content
from source;
```

- 写入动态分区

```
create table source (
   id INT,
   len INT,
   content VARCHAR,
   c TIMESTAMP
) with (
   type = 'random'
);
create table odps_sink (
   id INT,
   len INT,
   content VARCHAR,
   ds VARCHAR                          --动态分区列需要显式写在建表语句中。
) with (
   type = 'odps',
   endpoint = '<yourEndpoint>',
   project = '<yourProjectName>',
   `table` = '<yourTableName>',
   accessId = '<yourAccessId>',
   accessKey = '<yourAccessPassword>',
   `partition`='ds'                    --不写分区的值，表示根据ds字段的值写入不同分区。
   ,enable_dynamic_partition = 'true' --启用动态分区。
   ,dynamic_partition_limit='50' --最大并发分区数50。
   ,flush_batch_size = '524288' --缓冲区512 KB。
   ,flush_interval_ms = '60000' --Flush间隔60秒。
   ,flush_retry_count = '5' --Flush失败重试5次。
   ,flush_retry_interval_sec = '2' --失败重试间隔单位2秒。
   ,flush_retry_strategy = 'linear' --连续失败重试时间间隔线性增长。
);
insert into odps_sink
select
   id,
   len,
   content,
   date_dormat(c, 'yyMMdd') as ds
from source;
```

3.7. 使用Datahub（实时数据传输）

本文为您介绍流式数据处理服务DataHub。

DataHub是MaxCompute提供的流式数据处理（Streaming Data）服务，它提供流式数据的发布（Publish）和订阅 （Subscribe）的功能，让您可以轻松构建基于流式数据的分析和应用。

DataHub同样提供流式数据归档的功能，支持流式数据归档至MaxCompute。DataHub实时数据通道的详情请参见DataHub文档。

DataHub提供了Java和Python两种语言的SDK，可供您使用。详情请参见下述文档：

- DataHub Java SDK介绍。
- DataHub Python SDK介绍。

3.8. 使用MMA迁移工具（大批量数据传输）

3.8.1. 版本更新记录

本文为您介绍MMA近期版本的更新说明，基于此您可以了解MMA对应版本中的新增功能、增强功能内容。

MMA近期版本的更新说明如下，详细信息请单击对应版本链接获取。

版本	变更类型	描述
v0.1.0	新功能	• 支持从MaxCompute到OSS迁移。 • 支持从OSS到MaxCompute迁移。 • 添加Hive SQL兼容性检查工具。 • 添加了使用JDBC连接Hive Metasource的方式。
	增强功能	• 在MMA Web界面上添加任务信息显示。 • 支持停止任务。 • 支持重置任务。 • 在数据传输阶段使用MaxCompute的Bearer Token鉴权方式。
v0.0.3	新功能	添加MMA Web UI。 MMA Web UI可帮助用户跟踪其迁移作业的进度并找出失败的可能原因。使用Web UI，用户可以清楚地看到每个动作的进度和运行时信息。例如，作业名称，MC实例ID。
	增强功能	• 优化配置脚本。 • 更新MMA服务器启动脚本，以避免多个MMA服务器进程同时运行。
v0.0.2	新功能	• 支持迁移进度通知。 • 支持Hive 3.x。 • 添加MMA元数据调试工具。
	增强功能	• HMS失败时，MMA通过重建HMS客户端可以不停止服务。 • 优化日志记录配置。

3.8.2. MMA概述

MMA（MaxCompute Migration Assist）是一款MaxCompute数据迁移工具。本文为您介绍MMA的使用概述，帮助您快速了解并使用MMA。

使用向导

参考文档	说明
MMA配置	介绍配置MMA的准备工作以及配置流程。帮助您快速搭建MMA环境。

参考文档	说明
MMA命令	介绍MMA命令行工具，帮助您快速了解配置任务、迁移作业、查看作业状态、SQL兼容性检查等所使用的命令。
MMA Web UI	介绍MMA Web UI常用操作。您可以通过Web UI查看任务作业的状态、进度以及运行过程中出现的错误。
	介绍MMA作业迁移方案架构原理以及其他类型作业迁移方案。帮助您了解MMA的使用场景。
MMA FAQ	介绍MMA在使用过程中的常见问题，帮助您提高迁移效率。

3.8.3. MMA配置

本文以Hive数据迁移至MaxCompute为例，为您介绍如何配置MMA。

前提条件

在配置MMA之前需完成以下准备工作：

- 已下载并安装与Hive版本对应的MMA工具。MMA工具获取途径请参见MMA安装包。

  > **说明** 本文示例对应的MMA版本为v0.1.0，对应的安装包为mma-0.1.0-hive-1.x.zip。

- 已安装JDK1.8及以上版本。
- MaxCompute项目已配置2.0数据类型版本。详情请参见2.0数据类型版本。
- Hive集群各个节点和MaxCompute服务所在地域保持网络连通。

  对于在阿里云上搭建的Hive集群或到阿里云有专线的Hive集群场景，请参考各地域Endpoint对照表（阿里云VPC网络连接方式）；其他场景，请参考各地域Endpoint对照表（外网连接方式）。

  > **说明** 专线场景路由配置说明：
  >
  > 例如，本地IDC通过专线访问MaxCompute的Endpoint，需要在边界路由器（VBR）中将100.64.0.0/10网段的路由条目指向VPC方向的路由器接口，并在本地数据中心的网关设备上将100.64.0.0/10网段的路由指向VBR的阿里云侧互联IP，详情请参见本地IDC通过专线访问云服务器ECS。

配置MMA

1. 进入MMA解压目录，在 `bin` 目录下，执行如下命令运行配置引导脚本 `configure` 。

   ```
   ./configure
   ```

2. Hive configurations配置。配置参数如下表所示：

参数名	参数说明	参数示例
Hive metastore URI(s)	hive-site.xml中 `hive.metastore.uris` 属性值。	thrift://hostname:9083

参数名	参数说明	参数示例
Hive JDBC连接串	通过beeline使用Hive时输入的JDBC连接串，必须为default库，前缀为 `jdbc:hive2` 。	jdbc:hive2://hostname:10000/default
Hive JDBC连接用户名	通过beeline使用Hive时输入的JDBC连接用户名，默认值为Hive。	Hive
Hive JDBC连接密码	通过beeline使用Hive时输入的JDBC连接密码，默认值为空。	无

3. （可选）**Hive security configurations**配置。

   在使用Kerberos的情况下，配置过程需要提供以下Hive Security参数。配置参数如下表所示：

参数名	参数说明	参数示例
jams-gss.conf文件路径	MMA解压后， `conf` 目录下的 `gss-jaas.conf.template` 文件路径。	无
krb5.conf文件路径	通过Hive下的 `etc` 目录获取。	无
Kerberos principal属性	Hive目录下hive-site.xml中 `hive.metastore.kerberos.principal` 的属性值。	hive/_HOST@EXAMPLE.com
Kerberos keytab文件路径	Hive目录下hive-site-xml中的 `hive.metastore.kerberos.keytab.file` 的属性值。	无

4. **MaxCompute configurations**配置。配置参数如下表所示：

参数名	参数说明
MaxCompute endpoint	MaxCompute服务所在地域的Endpoint。各地域及网络对应的Endpoint，请参见Endpoint。
MaxCompute project名	MaxCompute的项目名称。 您可以登录MaxCompute控制台，在**项目管理**页签获取MaxCompute项目名称。 **说明** 建议配置为目标MaxCompute项目，规避权限问题。
阿里云accesskey id	阿里云账号或RAM用户的AccessKey ID。 您可以进入AccessKey管理页面获取AccessKey ID。

参数名	参数说明
阿里云accesskey secret	AccessKey ID对应的AccessKey Secret。 您可以进入AccessKey管理页面获取AccessKey Secret。

说明 Hive configurations和MaxCompute configurations配置完成后，`conf`目录下即可生成`mma_client_config.json`和`mma_server_config.json`配置文件。

5. 创建数据传输所需要的Hive UDTF。
   - 上传Hive UDTF JAR包至HDFS。命令如下：

     ```
     hdfs dfs -put -f <MMA_HOME>/lib/data-transfer-hive-udtf-0.1.0-jar-with-dependencies.jar hdfs:///tmp/
     ```

     MMA_HOME：MMA解压后的根目录。
   - 使用beeline创建Hive永久函数。命令如下：

     ```
     DROP FUNCTION IF EXISTS default.odps_data_dump_multi;
     CREATE FUNCTION default.odps_data_dump_multi as 'com.aliyun.odps.mma.io.McDataTransmissionUDTF' USING JAR 'hdfs:///tmp/data-transfer-hive-udtf-0.1.0-jar-with-dependencies.jar';
     ```
6. 创建完成Hive UDTF后，输入Y即可完成所有配置。

3.8.4. MMA命令

本文为您介绍MMA命令行工具，帮助您快速了解配置任务、迁移作业、查看作业状态、SQL兼容性检查等所使用的命令。

背景信息

MMA命令行工具位于`bin`目录下，包含工具如下：

- `configure`：配置引导工具。详情请参见configure。
- `gen-job-conf`：生成任务配置工具。详情请参见gen-job-conf。
- `mma_client`：客户端命令行工具。详情请参见mma-client。
- `mma_server`：服务端命令行工具。详情请参见mma-server。
- `sql-checker`：SQL兼容性检查。详情请参见sql-checker。

configure

通过运行`bin`目录下`configure`文件进行引导配置MMA。命令如下：

```
./configure
```

gen-job-conf

`gen-job-conf`为生成任务配置工具。

- 表级别任务配置。

i. 进入 `conf` 目录配置 `table_mapping.txt` 文件。该配置文件呈现待迁移表与目标表的对应关系，文件中每一行对应一个源数据库表到目标数据库表的迁移任务。内容如下：

```
source_catalog.source_table1:dest_pjt.dest_table1
```

全名的格式为库名.表名。例如source_catalog.source_table1:dest_pjt.dest_table1表示源表为source_catalog库中的表source_table1，目标表为dest_pjt项目下的表dest_table1。

ii. 进入 `bin` 目录执行以下命令，会根据 `conf` 目录下 `table_mapping.txt` 文件，在 `conf` 目录下生成MMA迁移配置文件 `TABLE-<source_catalog.>.<source_table1>-<dest_pjt>.<dest_table1>-<job_id>.json` 。

```
./gen-job-conf --objecttype TABLE --tablemapping ../conf/table_mapping.txt
```

iii. 在生成的配置文件中添加以下三个属性指定迁移分区。

- 内容示例

```
{
  "mma.filter.partition.begin":"2021/01",
  "mma.filter.partition.end":"2021/05",
  "mma.filter.partition.orders":"lex/lex"
}
```

- 参数说明
  - mma.filter.partition.begin与mma.filter.partition.end：斜线（/）分割的分区值，指定了迁移的分区范围。两者需要满足 `mma.filter.partition.begin<= mma.filter.partition.end` 。
  - mma.filter.partition.orders：斜线（/）分割的分区值排序类型。排序类型有两种lex（普通字典序）和num（数字序），一般使用lex即可。

- 库级别任务配置。
  - 命令格式

```
./gen-job-conf --objecttype CATALOG --sourcecatalog <sourcecatalog_name> --destcatalog <destcatalog_name>
```

  - 参数说明
    - sourcecatalog_name：源数据库名称。例如Hive数据库名。
    - destcatalog_name：目标数据库名称。例如MaxCompute项目名。

mma-client

使用 `mma-client` 工具进行任务的增删改查管理。

- 命令格式

  进入 `bin` 目录执行以下命令查看 `mma-client` 工具命令格式以及参数。

```
./mma-client -h
```

- 命令示例

- 向MMA server提交迁移任务。

```
./mma-client --action SubmitJob --conf <TABLE-<source_db>.<source_table>-<dest_db>.<dest_table>-<job_id>.json>
```

> **说明** TABLE-<source_db>.<source_table>-<dest_db>.<dest_table>-<job_id>.json：MMA迁移配置文件。详情参考gen-job-conf中的表级别任务配置。

- 查看任务状态。

```
./mma-client --action GetJobInfo --jobid <job_id>
```

- 查看迁移任务列表。

```
./mma-client --action ListJobs
```

- 删除迁移任务。

```
./mma-client --action DeleteJob --jobid <job_id>
```

- 重置迁移任务。

```
./mma-client --action ResetJob --jobid <job_id>
```

> **说明**
> - 状态为SUCCEEDED、FAILED、CANCELED三种状态下的任务可以被重置。
> - 当需要增量同步时，重置SUCCEEDED状态下的任务。
> - 当需要重试失败任务时，重置FAILED、CANCELED状态下的任务。

mma-server

`mma-server`为服务端命令行工具。MMA配置完成以后，进入`bin`目录执行以下命令启动MMA server。

```
./mma-server
```

> **说明** MMA server进程在迁移期间应当一直保持运行。若MMA server因为各种原因中断了运行，直接执行以上命令重启即可。MMA server进程在一台服务器最多只能存在一个。

sql-checker

使用`sql-checker`检查SQL脚本的兼容性。

- 命令格式

```
./sql-checker [-d | -f | -q] [-s]
```

- 参数说明
  - -d：包含SQL脚本（特指以.sql结尾的文件）的目录。
  - -f：用于指定SQL脚本文件所在路径。
  - -q：用于直接输入query。
  - -s：用于提供逗号分隔的MaxCompute SQL设置。

> 说明 -d、-f、-q，三个选项互斥，每次执行最多选择其中一个。

- 命令示例
  - 检查单个query。进入MMA `bin` 目录执行以下语句，`sql-checker` 直接输出此query的兼容性信息。

    ```
    ./sql-checker -q "select * test_db.test_tbl;"
    ```

  - 检查SQL脚本中所有query。进入MMA `bin` 目录执行以下语句，`sql-checker` 直接输出指定脚本（script.sql）兼容性总结。

    ```
    ./sql-checker -f /root/mingyou/scripts/script.sql
    ```

  - 检查目录下所有SQL脚本。进入MMA `bin` 目录执行以下语句，`sql-checker` 输出指定目录下各脚本的兼容性总结，并在最后汇总为该目录的兼容性总结。

    ```
    ./sql-checker -d /root/mingyou/scripts/
    ```

  - 指定MCQL settings。通常进行兼容性检查时，通过指定一些MCQL settings可以避免兼容性问题。

    ```
    ./sql-checker -d sample-queries-tpcds/ -s "odps.sql.decimal.odps2=true,odps.sql.type.syst
    em.odps2=true,odps.sql.validate.orderby.limit=false,odps.sql.timezone=UTC"
    ```

    参数说明
    - odps.sql.type.system.odps2=true、odps.sql.decimal.odps2=true、odps.sql.hive.compatible=false：以上三个配置指定使用MaxCompute2.0类型系统。
    - dps.sql.timezone=UTC：提供一个默认Timezone。

3.8.5. MMA Web UI

MMA v0.0.3及以上版本支持Web UI，主要用于查看任务的状态、进度以及作业运行过程中出现的错误等。

背景信息

Web UI运行在MMA Server所在服务器的18888端口，可以通过http://<MMA安装环境所在的服务器地址>:18888地址进行访问。

Web UI的功能主要包括：

- 查看Job信息与状态。详情请参见查看Jobs信息。
- 查看Tasks信息。详情请参见查看Tasks信息。
- 查看MMA Server配置信息。详情请参见查看MMA Server配置信息。

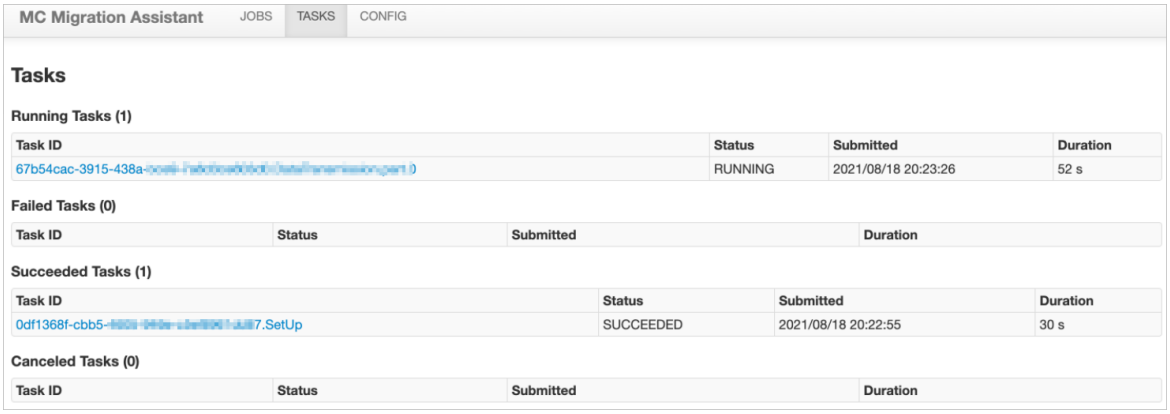
查看Jobs信息

1. 在MMA Web UI界面，单击**JOBS**即可查看所有**Jobs**的信息。

2. 在**JOBS**界面，单击**Job ID**即可查看所有**Sub jobs**的信息以及关联的**Tasks**。

查看Tasks信息

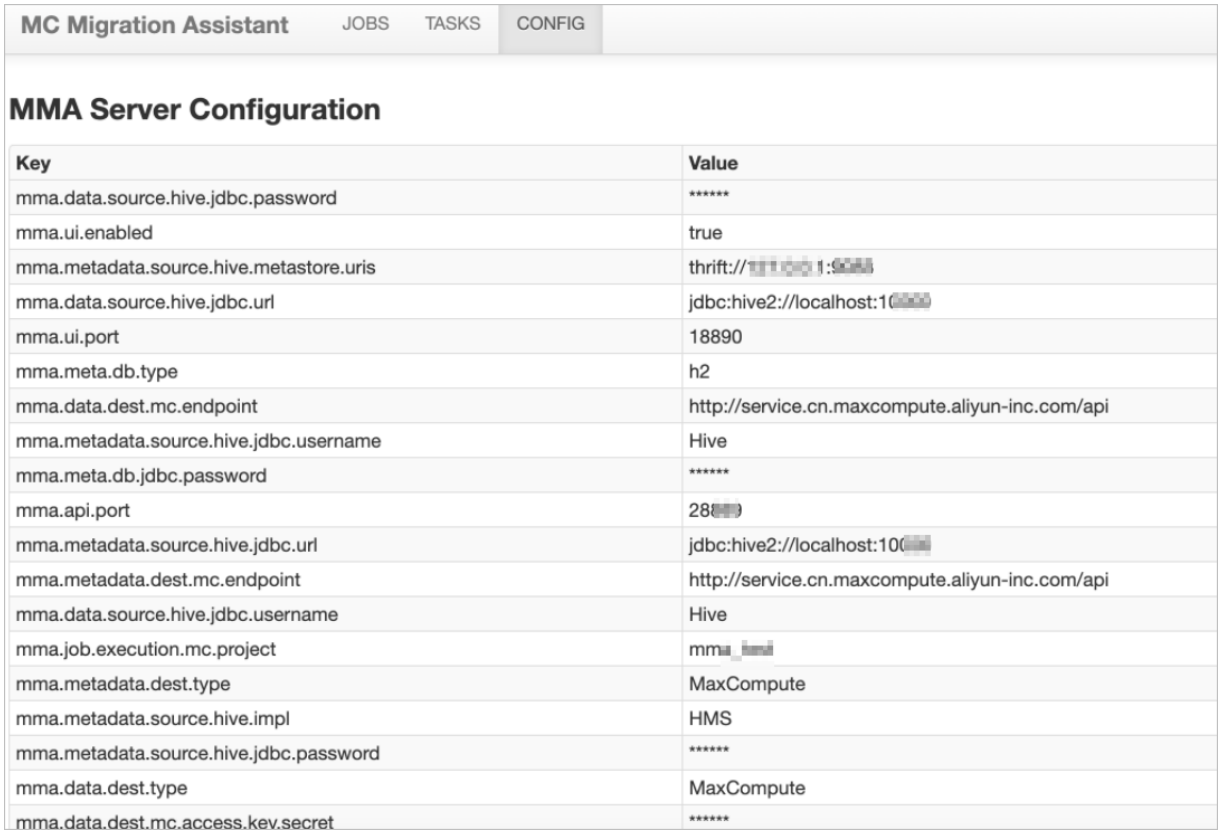
1. 在MMA Web UI界面，单击TASKS即可查看所有Tasks的信息。



2. 在TASKS界面，单击Task ID即可查看Task的执行详情。

查看MMA Server配置信息

在MMA Web UI界面，单击**CONFIG**即可查看MMA Server的配置信息。



Key	Value
mma.data.source.hive.jdbc.password	******
mma.ui.enabled	true
mma.metadata.source.hive.metastore.uris	thrift://
mma.data.source.hive.jdbc.url	jdbc:hive2://localhost:1
mma.ui.port	18890
mma.meta.db.type	h2
mma.data.dest.mc.endpoint	http://service.cn.maxcompute.aliyun-inc.com/api
mma.metadata.source.hive.jdbc.username	Hive
mma.meta.db.jdbc.password	******
mma.api.port	28
mma.metadata.source.hive.jdbc.url	jdbc:hive2://localhost:10
mma.metadata.dest.mc.endpoint	http://service.cn.maxcompute.aliyun-inc.com/api
mma.data.source.hive.jdbc.username	Hive
mma.job.execution.mc.project	mm
mma.metadata.dest.type	MaxCompute
mma.metadata.source.hive.impl	HMS
mma.metadata.source.hive.jdbc.password	******
mma.data.dest.type	MaxCompute
mma.data.dest.mc.access.key.secret	******

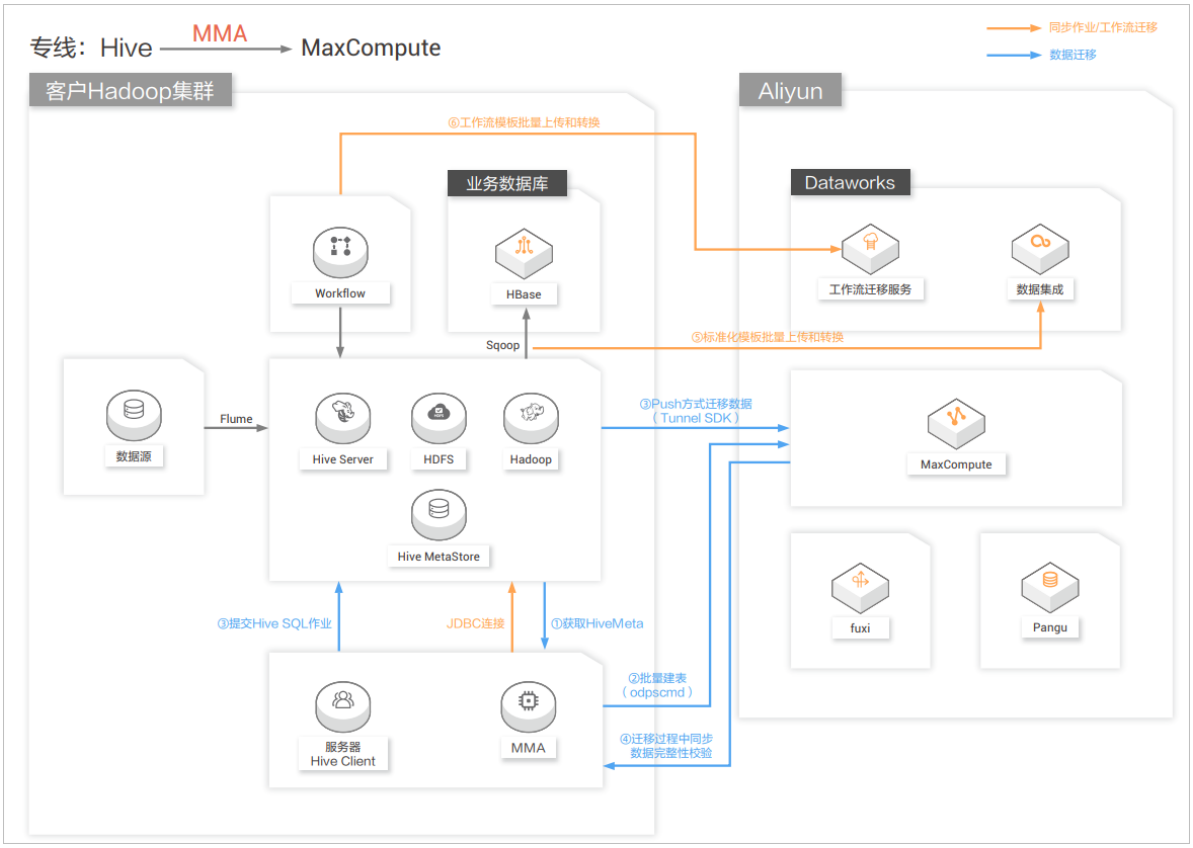
3.8.6. MMA迁移作业方案

本文为您介绍Hadoop数据迁移至MaxCompute的两种迁移方案。

以下是Hadoop数据迁移至MaxCompute的两种迁移方案，您可以根据实际情况进行选择。

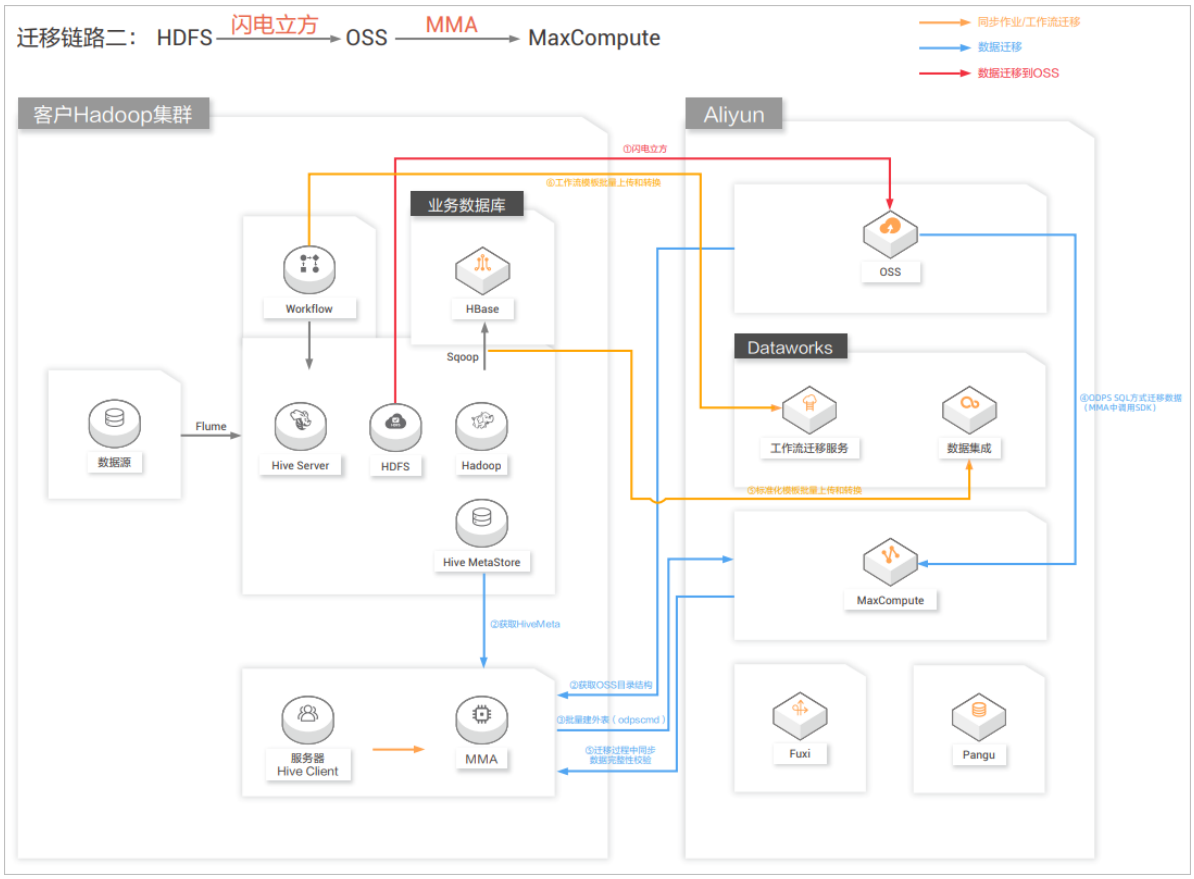
- 迁移链路一

专线场景下，支持通过MMA迁移Hive数据至MaxCompute。迁移方案如下图所示。



- 迁移链路二

无专线场景下，支持通过闪电立方迁移HDFS数据至OSS，再通过MMA将数据写入MaxCompute。迁移方案如下图所示。



3.8.7. MMA FAQ

本文介绍在使用MMA迁移数据过程中的常见问题。

- 如何升级MMA？。
- 如何将迁移数据导入到多个MaxCompute项目？。
- 数据迁移过程中进度条一直不动如何处理？。

如何升级MMA？

MMA会不断更新功能，并修复已知问题，提高稳定性。建议您按照如下步骤升级MMA。

1. 下载解压新版本MMA。详情请参见MMA下载。
2. 作业运行完后，在MMA所在服务器使用`kill mma-server`命令停止老版本MMA。
3. 将已有的MMA根目录下的元数据文件.MmaMeta.mv.db复制到新版本MMA的根目录下。
4. 启动新版本MMA。详情请参见MMA命令。

如何将迁移数据导入到多个MaxCompute项目？

在阿里云账号（主账号）或RAM用户（子账号）具备admin权限的情况下，配置`table_mapping.txt`时即可以选择将数据导入MaxCompute不同项目。详情请参考权限列表和MMA命令中的表级别任务配置。

数据迁移过程中进度条一直不动如何处理？

进度条是基于完成的分区数量显示进度的，因此会出现跳变的情况（对于非分区表会直接从0跳到100）。目前可以通过Web UI监控进度。详情请参见MMA Web UI。

3.8.8. 其他类型作业迁移说明

本文为您介绍Hive之外其他类型作业迁移时的注意事项。

UDF和MapReduce迁移

- 支持相同逻辑的UDF和MapReduce输入、输出参数的映射转换，但UDF和MapReduce内部逻辑需要您自行维护。
- 不支持在UDF、MapReduce中直接访问文件系统、网络访问、外部数据源连接。
- Hive UDF兼容示例，请参见兼容Hive UDF。

外表迁移

- 原则上数据会全部迁到MaxCompute内部表。
- 如果必须通过外表访问外部文件，建议先将文件迁移到OSS，然后在MaxCompute中创建外部表，实现对文件的访问。
- MaxCompute外部表支持的格式包括ORC、PARQUET、SEQUENCEFILE、RCFILE、AVRO和TEXTFILE。

Spark作业迁移

- 如果作业无需访问MaxCompute表和OSS，可直接运行Jar包，请参见《MaxCompute Spark开发指南》准备开发环境和修改配置。

  > **说明** 对于Spark或Hadoop的依赖必须设成provided。

- 如果作业需要访问MaxCompute表，请参见《MaxCompute Spark开发指南》中访问MaxCompute表所需依赖编译Datasource并安装到本地Maven仓库，在*pom.xml*中添加依赖后重新打包即可。
- 如果作业需要访问OSS，请参见《MaxCompute Spark开发指南》中OSS依赖，在*pom.xml*中添加依赖后重新打包即可。

4.数据集成导出数据

MaxCompute支持通过DataWorks的数据集成功能将MaxCompute中的数据以离线方式导出至其他数据源。当您需要将MaxCompute中的数据导出至其他数据源执行后续数据处理操作时，您可以使用数据集成功能导出数据。本文为您介绍如何将MaxCompute的数据导出至其他数据源。

背景信息

数据集成的导出方式有如下两种：

- 向导模式：创建离线同步节点后，在DataWorks界面以可视化方式配置数据来源、去向及字段的映射关系等信息，完成数据导出操作。
- 脚本模式：创建离线同步节点后，将DataWorks可视化界面切换至脚本模式，通过脚本配置数据来源、去向及字段的映射关系等信息，完成数据导出操作。

前提条件

请确认您已完成如下操作：

- 已在MaxCompute上准备好待导出至其他数据源的表数据。

  更多创建表及写入数据操作，请参见表操作和插入或覆写数据（INSERT INTO | INSERT OVERWRITE）。
- 已准备好目标数据源及目标表。

使用限制

每个离线同步节点仅支持将单张表数据导出至其他数据源。如果您需要导出多张表数据，需要创建多个离线数据同步节点。

操作流程

通过数据集成导出MaxCompute数据的流程如下：

1. 添加MaxCompute数据源

   将MaxCompute数据源添加至DataWorks的数据源列表。
2. 添加目标数据源

   将MaxCompute数据源导出至的目标数据源添加至DataWorks的数据源列表。
3. 创建业务流程

   在DataWorks上创建业务流程，为创建离线同步任务做准备。
4. 创建离线同步节点

   在创建的业务流程基础上，创建离线同步任务。
5. 通过向导模式配置并运行数据同步任务或通过脚本模式配置并运行数据同步任务

   以可视化或脚本模式配置并运行离线同步任务。
6. 确认同步结果

   在目标数据源侧确认数据同步结果。

添加MaxCompute数据源

1. 进入**数据源管理**页面。
   i. 登录DataWorks控制台。
   ii. 在左侧导航栏，单击**工作空间列表**。

iii. 选择工作空间所在地域后，单击相应工作空间后的**进入数据集成**。

iv. 在左侧导航栏，单击**数据源 > 数据源列表**，进入**工作空间管理 > 数据源管理**页面。

2. 在**数据源管理**页面，单击右上角的**新增数据源**。
3. 在**新增数据源**对话框中，选择数据源类型为**MaxCompute（ODPS）**。
4. 在**新增MaxCompute（ODPS）数据源**对话框中，配置各项参数。

参数	描述
数据源名称	数据源名称必须以字母、数字、下划线（_）组合，且不能以数字和下划线（_）开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择**开发**或**生产**环境。 **说明** 仅标准模式工作空间会显示该配置。
ODPS Endpoint	默认只读，从系统配置中自动读取。

参数	描述
Tunnel Endpoint	MaxCompute Tunnel服务的连接地址，详情请参见Endpoint。
ODPS项目名称	MaxCompute（ODPS）项目名称。
AccessKey ID	访问密钥中的AccessKey ID，您可以进入用户信息管理页面进行复制。
AccessKey Secret	访问密钥中的AccessKey Secret，相当于登录密码。

5. 选择**资源组连通性**类型为**数据集成**。
6. 在资源组列表，单击相应资源组后的**测试连通性**。

   数据同步时，一个任务只能使用一种资源组。您需要测试每个资源组的连通性，以保证同步任务使用的数据集成资源组能够与数据源连通，否则将无法正常执行数据同步任务。如果您需要同时测试多种资源组，请选中相应资源组后，单击**批量测试连通性**。详情请参见配置资源组与网络连通。

   > 说明
   > - （推荐）资源组列表默认仅显示独享数据集成资源组，为确保数据同步的稳定性和性能要求，推荐使用独享数据集成资源组。
   > - 如果您需要测试公共资源组或自定义资源组的连通性，请在资源组列表右下方，单击**更多选项**，在**警告**对话框单击**确定**，资源组列表会显示可供选择的公共资源组和自定义资源组。

7. 测试连通性通过后，单击**完成**。

添加目标数据源

请根据MaxCompute导出的目标数据源类型，完成添加数据源操作。更多添加数据源操作，请参见配置数据源。

通过向导模式配置并运行数据同步任务

1. 通过向导模式配置离线同步任务。

   在数据来源下的**数据源**下拉列表选择数据源类型为**ODPS**及创建好的MaxCompute数据源名称，在**表**下拉列表选择待导出数据的表。如果为分区表需要配置分区信息。

2. 通过向导模式配置离线同步任务。

   在数据去向下的**数据源**下拉列表选择目标数据源类型及目标数据源名称，在**表**下拉列表选择目标表。

3. 通过向导模式配置离线同步任务。

   指定数据来源表和数据去向表间字段的映射关系。

4. 通过向导模式配置离线同步任务。

5. 通过向导模式配置离线同步任务。

   在同步任务中配置调度参数进行数据过滤。

6. 在顶部菜单栏，单击图标后，单击图标，运行离线同步任务。

通过脚本模式配置并运行数据同步任务

1. 通过脚本模式配置离线同步任务。

   在**来源类型**下拉列表选择**ODPS**，对应**数据源**为创建好的MaxCompute数据源名称。在**目标类型**下拉列表选择目标数据源类型，对应**数据源**为目标数据源名称。

2. 通过脚本模式配置离线同步任务。

   在脚本中配置离线同步任务读取的数据源，以及需要同步的表信息等。

```
{
            "stepType": "odps",
            "parameter": {
                "partition": [],
                "datasource": "odps_first",
                "envType": 0,
                "column": [
                    "*"
                ],
                "table": ""
            },
            "name": "Reader",
            "category": "reader"
        },
```

   - stepType：数据源类型。设置为odps。
   - partition：表的分区信息。您可以通过 `show partitions <table_name>;` 命令，查看表的分区信息。更多查看分区信息，请参见查看分区。
   - datasource：MaxCompute数据源的名称。
   - column：待导出数据的表的列名称。
   - table：待导出数据的表的名称。您可以通过 `show tables;` 命令，查看表的名称。更多查看表信息，请参见表操作。
   - name和category：取值为Reader，标识数据源为读取端。

3. 通过脚本模式配置离线同步任务。

   在脚本中配置离线同步任务写入的数据源，以及需要写入的表信息等。

```
{
   "stepType":"mysql",
   "parameter":{
    "partition":"",
    "truncate":true,
    "datasource":"",
    "column":[
     "*"
    ],
    "table":""
   },
   "name":"Writer",
   "category":"writer"
  }
```

- stepType：目标数据源类型。
- partition：目标表的分区信息。
- datasource：目标数据源的名称。
- column：目标表的列名称。需要与中配置的列名称建立一一对应关系。
- table：目标表的名称。
- name和category：取值为Writer，标识数据源为写入端。

4. 通过向导模式配置离线同步任务。

```
"setting": {
        "errorLimit": {
            "record": "1024"
        },
        "speed": {
            "throttle": false,
            "concurrent": 1
        }
    },
```

- record：脏数据的最大容忍条数。
- throttle：设置是否进行限速。
- concurrent：设置离线同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。

5. 通过向导模式配置离线同步任务。
6. 在顶部菜单栏，单击图标后，单击图标，运行同步任务。

确认同步结果

请前往目标数据源中确认MaxCompute表中的数据是否已成功导入目标表中：

- 如果数据完整无遗漏，则同步完成。
- 如果数据未同步成功或数据存在遗漏，请参见离线同步常见问题。

5.迁移示例

本文为您介绍MaxCompute相关迁移案例，为您执行数据迁移操作提供指导。

本文档已为您提供相关数据迁移最佳实践，请参见数据迁移。