目录

产品概述

金山云智•流计算服务提供面向流式/实时数据进行实时快速计算的解决方案，使您可以更关注业务逻辑处理。支持每秒千万级数据处理，响应延迟达到毫秒级。支持流/实时计算快速入门、快速开发、聚焦业务。

名词解释

- **实时流数据** 实时、持续生成的数据，如业务日志、系统日志等各类日志信息。

- **主题（Topic）** 流计算服务订阅和发布的最小单位，用户可以用Topic表示一种流数据，类似与数据库中的表（Table）。在大数据云服务中，一个流计算服务的数据源对应一个topic，单个topic可以存储一个或多个日志中的流式数据。

- **流连接分区（Partion）** Topic存储数据的最小单元，对于吞吐较高的Topic，可以创建多个分区。

- **状态保存点（SavePoint）** SavePoint由用户手动触发，可以支持程序升级后，继续从升级前的那个点开始执行计算，保证数据不中断。

行业场景与资源

互联网点击流分析

用户在网站浏览时会产生许多的点击行为，对这些点击行为加以分析可精确把握热点趋势。借助于云端流计算服务，能够分钟级构建实时分析，对用户行为数据进行实时汇聚分析，持续地挖掘出有价值信息，帮助更好地做出运营决策，改进用户体验。

金融实时风控

在众多金融风险中，及早探测到风险往往能有效地减少损失，将金融交易大数据与流计算服务相结合，引入特征模型算法，及早地过滤出诸如盗刷卡等异常交易行为，实施风险控制，提升金融安全性。

物联网监控

在工业设备的运转过程中，及早发现潜在故障会极大降低维修成本。借助于云端流计算服务，及时收集设备传感器数据，并进行聚合、分析筛选，可实现秒级设备异常告警，提升设备利用率。

电商精准推荐

在电商交易中，借助于云端流计算服务，实时提取特征变量，及时跟踪用户关注品类，预测用户消费趋势，为精准推荐提供基础能力。从而提升用户购物体验，促进消费行为。

产品优势

高吞吐和低延迟

支持Spark Streaming、Flink等流计算引擎，并且统一一套编程模型。借助多计算引擎支持，流计算服务可以根据需求选择不同的计算引擎以适配高吞吐和低延迟场景需求。

可视化流程编排

支持在Web界面上以可视化的方式进行流计算流程编排，创建、修改大数据处理流程。

技术领先

支持维表解决方案、On Docker等；支持用户自定义插件开发，并在算子、UDF等多层面支持客户进行扩展，支持JDBC、Hive、Redis、Hbase、ES、Kafka等sink。

数据共享

支持用户以写临时表的方式将流计算中间结果进行缓存，供其他流计算流程做为数据源进行再次处理，便于多个流计算工程的数据共享协作。

产品功能

丰富的流计算插件

支持ETL处理、SQL统计分析、Sink类、l Source类等多类型插件，支持用户进行自定义插件开发，通过SDK进行开发扩展支持。

临时表管理

支持用户以写临时表的方式将流计算中间结果进行缓存，供其他流计算流程作为数据源进行再次处理。便于进行多个流计算工程的数据共享协作。

SQL式开发

支持以表的方式进行Kafka等流式数据引擎的数据读写，更好的抽象底层数据，统一开发编程模型，支持直接利用SQL进行实时业务的开发和处理。

高可用支持

框架内的处理节点均支持在线降级，在极端情况下保证高优业务优先执行，保障实时性SLA。

创建数据源和Topic

新建数据源

1. 为了将数据从指定表中读写，需要按照业务需求创建对应的数据源/库/表。在【数据管理】→【数据源管理】新建数据源。
2. serverAddress填写对应Kafka集群地址，可以进行连通性测试，最后点击【确认】。

设计态-新建topic

在【数据管理】→【元数据管理】→【库表管理】中新建数据库和数据表（Kafka不需要新建数据库，直接新建topic即可）。topic时需指定topic归属的项目，即此topic可在归属项目下使用。

发布到测试态

完成Topic创建动作后，Topic信息可在【库表管理】的发布态查看，点击列表操作中的【发布至测试】，可申请将topic发布到测试环境，并在测试环境使用。 发布测试时，需指定此topic归属的数据源。 发布测试完毕后，可在测试环境中筛选查看。

测试态审批

使用审批账号登陆大数据云平台后，在【数据管理】→【元数据管理】→【发布审批】→【测试环境】→【待我审批】中可以查看待审批的数据库/表，点击审批通过，并在弹出的窗口中，填写审批意见后，点击【确认】，即完成了审批流程。审批后，待发布的数据库/表成功发布至测试环境。 在【测试环境】选项卡下的【我已审批】选项卡下，可以看到所有已审批通过的数据库/表。

发布到生产态

将Topic发布到测试后，点击操作列【显示表】页面下方会显示该数据库中所有的表，其中每个表对应一个topic。同样，点击操作列【发布到生产】，经过同样的审批流程后，即可将该topic发布到生产。 审批通过后，点击【生产环境】，在【我设计的库表】选项卡下，按所属项目进行筛选后，即可看到已发布到生产的库，点击【显示表】，即可看到已发布到生产的topic。

申请数据权限

流计算作业如果需要使用到其他项目的数据表均需要提前通过数据管理组件进行申请和注册。

申请权限

1. 除数据管理员角色以外的其他角色的人员（包括：项目管理员，开发人员，运维人员，业务人员等）进入数据管理模块，点击【元数据管理】下面的【数据目录】。
2. 选中数据源Kafka，搜索表名称，点击对应的表，会展示出表详情。
3. 点击【权限申请】按钮，弹出权限申请的页签，选择申请对象，选择权限，选择申请对象，选择申请项目，点击【确

认】按钮，权限申请成功。

说明：

- 申请类型为列权限时，申请对象只能为个人；申请类型为表权限时，申请对象可以为项目，个人或者项目和个人。
- Kafka类型数据源，没有列权限，只有表权限，不需要选择申请类型；Kafka类型数据源的权限选择只有两个选项：insert和select。
- 申请个人和项目的表权限，会同时生成两条申请记录，审批记录也是两条。

权限审批

1. 使用数据管理员账号登录进入数据管理模块，点击【数据权限管理】，展开下拉列表中点击【权限审批】。
2. 根据申请号选择提交的申请，点击【查看】可查看提交申请的用户，备注中显示提交原因。
3. 点击【通过】（或者驳回）审批该申请。

说明：通过申请，如果申请的时候选择的是项目，该表的项目白名单会新增一条数据，项目管理员对该数据表有对应的权限；如果选择的是个人，用户白名单新增一条数据，申请人对该表有对应的权限。

数据授权

1. 使用数据管理员或者租户的账号登录，进入数据管理模块，点击【数据权限管理】，展开的下拉框中点击【数据授权】，数据空间显示数据授权列表。
2. 选中需要授权的表，点击【白名单设置】按钮，弹出白名单设置标签，选择授权类型，选择权限，选择授权对象，授权项目和个人都支持多选，点击【确认】。 说明：授权类型为表权限时，授权对象可以为项目，个人或者项目和个人；授权对象为列权限时，授权对象只可以为个人；个人和项目均支持多选和搜索。
3. 授权成功后，点击【查看】按钮，可以看到授权的项目白名单和个人白名单。 说明：查看项目白名单和用户白名单，点击【新增白名单】按钮，可以给表增加白名单，操作类似白名单设置。

流计算作业开发－在线开发模式

新建作业

1. 进入流计算组件，点击【新建文件夹】和【新建作业】，创建一个新的流计算作业，创建完毕后即进入流计算作业的开发IDE界面。
2. 左侧为本项目流计算任务树，右侧为任务组件和画布区。可以通过拖拽算子和连线轻松编辑流计算作业。
3. 创建文件夹/作业时，支持：在线开发和JAR包上传2种模式，其中在线开发模式支持Flink引擎；JAR包上传模式，依赖线下开发并上传至云平台的JAR包，支持Flink和Spark Streaming两种引擎。

在线开发模式

新建在线开发模式的流计算作业，将进入在线开发的IDE界面，中间的“源表”“数据处理”“结果表”等operator支持拖拽。拖拽后，将自动弹出算子的参数设置栏，完成参数配置，连接各operator后，即可生成流作业。流计算支持以下方式，灵活支持各种业务场景：

- 双流joining（支持2个Kafka数据源）
- 维表管理（支持Kafka数据源关联mysql、oracle等其他数据源 ）
- jar包依赖（通过上传的jar包进行ETL等操作）
- 单元测试（支持线下上传数据包，进行乱序、延时测试）

1. 作业参数配置

新建作业后，会自动弹出作业参数配置弹窗，可先对流作业的信息进行配置：

- 时间属性：支持processing time（系统时间）和event time（时间时间）2种，分别表示流计算应用中，按照什么时间进行数据处理。
- Checkpoint触发间隔：即设置检查点的时间间隔。
- 默认并行数：即流作业的并行数。

- 自定义环境参数：非必须，可配置Flink引擎的其他环境参数，缓存属性等

2. 数据预览

数据预览功能可支持提前对Source、Sink的数据结构和数据进行预览，辅助开发。

3. 添加数据源

在IDE界面拖拽数据源插件“KafkaSource”，双击拖拽的KafkaSource弹出参数设置框（首次拖拽插件时，自动弹出），配置参数后，点击右上角的“X”进行保存。 配置参数说明如下：

- 名称：即插件名称，可修改
- 数据源：可选择有项目下有权限的数据源
- Topic名称：可选择已选数据源下的topic
- 中间表名称：非必填，为将Kafka中的数据进行结构化后注册表的名称
- Operator并发度：非必填，如不指定将按照作业默认并行数执行
- Kafka消费位置：支持Latest（最近时间）、Earliest（最早时间）、custom（自定义选择开始时间）三种方式，默认Latest time执行。

4. SQL operator

在IDE界面拖拽数据源插件“Sql Opertaor”，编写具体的流计算业务逻辑，Sql operator不能独立存在，只能在Source和Sink之间。

双击拖拽的“Sql Opertaor”，即可通过编写Flink SQL进行数据处理，SQL编译框默认提供编写规范，并支持添加预制的模板。

5. 添加目标表

拖拽目标表算子，即可进行添加目标表的操作，流计算服务支持将数据输出至：JDBC（含：Mysql、Oracle、MPP三种数据源）、HBase、Kafka、Redis、ES七种数据源。

JDBCSink、HBaseSink、ESSink中都有数据写入批次概念。数据按照批次写入，有两个条件限制：批次大小batchsize和刷新时间flushIntervalms，只要满足其中一个条件，就会触发写数据操作。batchsize表示这一批次的大小，也就是有多少条数，刷新时间表示这一批次等待多长时间写入。

用户配置RedisSink时，支持多种操作，incr、incrByFloat、set、sadd、zadd、rpush、lpush、hset。

- Key所需字段: Rediskey是从上游表中取的某个或多个字段对应的值，如果输入多个字段时用逗号分隔。
- Value所需字段：Redisvalue的值从上游表中取的某个字段对应的值。
- key连接符: 拼接rediskey时候的连接符

6. 连接Source、Operator、Sink插件

将鼠标悬停在算子上，可浮现连线点，可选择算子的连接顺序，生成作业流。 在IDE界面通过拖拽方式将源表Kafka Source，Sql Operator，Kafka Sink 进行连接，多个作业的时候也可灵活连接。 流计算支持同时对2个Kafka Source进行关联处理（双流joining）。

7. 单元测试

单元测试可以快速进行作业测试，支持乱序测试、延迟测试2种方式。 读取线下上传的测试数据包中的数据，代替从Source中读取数据，并且可以复用画板中Source的schema信息，将测试结果数据输出到开发界面，代替了数据输出到Sink。

- 单元测试用于测试数据处理算子的可用性;
- 单元测试需选择指定数据包，数据包在［资源管理］中上传;
- 单元测试使用的数据包将复用画布中拖拽的Kafka Source的schema信息;
- 单元测试Source算子并行度为1;
- 单元测试结果获取会有一定延迟(每3s获取一次，每次上限100条)。

8. 版本管理

支持生成多个版本的流作业（点击【提交】按钮生成新版本），点击【版本管理】tab，可以查看个版本的信息。 当多版本

时，通过点击“流程图对比”或“版本对比”，可对任意2个版本的流程图和参数进行对比，内容不一致的部分会被标注出来。

流计算作业开发－自定义开发模式

自定义开发相关功能包括ETL、Custom Operator、UDF、维表等，提供JAR包上传和在线代码编写两种实现方式。下面将为每一种功能提供详细说明和并附注案例。

ETL operator

ETL功能以算子形式，将stream流暴露给用户，进行细力度的操作。系统提供JAR/在线开发两种实现方式。

（1） JAR开发

1） 代码编写

ETL功能需要继承基类ETLFunction，并重写ETL和tableSchema两个方法。

- params为页面设置的自定义参数，以map形式输入，此处只简单输出。
- with Logging，代码中可直接使用log实例，日志会打印在taskmanager的log信息中。
- ETL函数，重载父类函数。每一条流消息调用一次本方法。本例中将第一列取出，并拼接一个固定列值。实际使用时以具体业务逻辑替换。
- tableSchema函数，重载父类函数。因处理逻辑不确定，所以返回schema需算子自己定义。以标准json schema返回。注意：必须要和ETL函数的返回值对应上，不然后续转换过程和报错。

2） 代码打包

```
mvn clean package -DskipTests -Dcheckstyle.skip=true
```

3） 页面配置

- 在资源管理功能中将打包后的【streaming-flink-test-1.0-SANPSHOT.jar】上传。
- 在作业开发界面选择依赖的jar。
- 拖拽算子生成graph，并设置ETL Operator参数。其中类名称需为全路径，在执行时从依赖jar包中反射注入。

（2）在线开发

1） 代码编写

在线代码编写功能在原ETL算子中，增加了【在线开发】选项，并提供JAVA/SCALA两种语法支持。默认会展示最基本的代码框架，如需业务相关模板可在【是否使用模板】功能中选择。

双击ETL算子后进入编辑页面。按上图进行操作，其中需注意点如下：

- 模板需预先在左侧【模板管理】功能中设置，且公共模板整个租户可见。
- 模板具体参见【Flink自定义开发模板.doc】，只需编写【模板内容】部分，系统会自动填充成【补全后内容】运行。
- 添加参数】输入的参数于ETL功能中使用，通过params: java.util.Map[String, Any]直接获取。

如需在ETL中调用维表，请参见【维表与ETL结合方式】。

Custom Operator

Custom Operator功能提供一个自定义的全新算子，可以充当ETL，也可以充当Source或者Sink。需要自己维护TypeInformation并通过flatMap函数实现业务逻辑，Custom Operator只支持JAR开发模式。

1） 代码编写

ETL功能需要继承基类TableOperateProcessor，并重写innerbuild函数。举例CustomTestSimple为一个Custom Operator简单实现：

a） 构造器中定义了四个参数：

- name:本算子的name，通常设置为在jobGraph中显示名称。

- childs:子算子集合
- configs: Map格式参数集合
- sm: StreamingMate可获取环境变量、维表等全局参数

b) with Logging，代码中可直接使用log实例，日志会打印在taskmanager的log信息中。

c) innerBuild函数，重载父类函数。将dataStream直接暴露给用户，本例中不做处理直接转给下一个算子。实际使用时以具体业务逻辑替换。

d) dataStream.dataType.asInstanceOf[RowTypeInfo]获取输入流的schema

e) 注意：必须要隐式声明implicit val tpe: TypeInformation[Row]，供下游算子使用

2) 代码打包 同ETL

3) 页面配置 同ETL

自定义UDF

Udf功能采用Flink自身Udf语法，详细说明参见https://ci.apache.org/projects/flink/flink-docs-release-1.7/dev/table/udfs.html

（1）JAR开发

1）代码编写

Udaf功能需要继承基类AggregateFunction，并重写createAccumulator、getValue、accumulate、getResultType四个方法（有些可选方法见官网）。

2）代码打包

mvn clean package -DskipTests -Dcheckstyle.skip=true

3）页面配置

在资源管理功能中将打包后的【streaming-flink-test-1.0-SANPSHOT.jar】上传。

（2）在线代码编写模式

1）在左侧菜单栏添加了【函数管理】功能，支持UDF/UDAF的添加和修改。

2）双击新增按钮后进入编辑页面。按上图进行操作，其中需要注意点如下：

- 函数名为最终register的函数名称，sql中使用这个名称。
- 函数属性用以展示，在选择函数时使用。
- 模板具体参见【Flink自定义开发模板.doc】，只需编写【模板内容】部分，系统会自动填充成【补全后内容】运行。

维表配置

（1）功能说明

当前Flink版本只支持流式数据源，当流数据需要关联外部数据库（如mysql、oracle、redis等），需要采用维表的形式支持。 数据库的维表查询请求，有大量相同 key 的重复请求。如何减少重复请求？本地缓存是常用的方案。本方案目前提供两种缓存方案：LRU 和 ALL。

- cache = ALL（默认）: 全量内存缓存
- cacheTTLMs:缓存的过期时间(ms)
- cache = LRU: LRU内存缓存
- cacheSize: 缓存的条目数量
- cacheTTLMs:缓存的过期时间(ms)

（2）与ETL结合方式

本版本维表功能只支持在ETL算子中使用。本文提供mysql版的维表实现案例，如需其他版本请用户自定义实现。

- 代码编写

dac.getConfigByKey(MySQLUpdater.key)为从运行时环境获取缓存对象，其中MySQLUpdater.key为对应实现的key。

- 代码打包

同ETL

- 页面配置

其中数据源非必选参数，如使用外部数据源，可完全由自定义参数配置数据库连接。维表可设置多个。

测试与运行

发布测试

完成作业开发配置后，点击【保存】按钮保存当前任务，点击【提交】按钮生成新版本的作业，点击【运行测试】将作业发布到“运维中心”进行测试运行。

测试运行

提交作业后，会在【运维中心-测试实例-流计算作业】生成一条流作业数据，可进行：【启动】【终止】【创建savepoint】【查看savepoint】【查看flink UI】等操作。

- 【启动】：点击启动，并选择启动方式（可选择立即启动或savepoint启动）、配置作业参数后，可启动作业。
- 【终止】：点击终止，可停止作业；
- 【创建savepoint】：创建后，再次启动作业时，可选择从savepoint标识的点启动作业；
- 【查看savepoint】：可以查看所有的savepoint列表
- 【查看FlinkUI】 备注：查看Flink_UI前，需要提前配置部署集群的hosts，否则页面无法正常跳转。

发布生产

提交作业后，会在【运维中心-测试实例-流计算作业】生成一条流作业数据，可进行：【启动】【终止】【创建savepoint】【查看savepoint】【查看flink UI】等操作，对于执行正常的作业，点击【发布生产】，可进入发布生产的审批流程。

作业审批

发布生产后，会在【发布管理】中查看，【发布管理】中有3个页面，功能分别如下：

- 【发布管理-发布列表】：可查看待发布生产的流作业列表，支持对作业进行测试、发布、下线操作。
- 【发布管理-发布审批】：由系统管理员对待发布生产的流作业进行审批。
- 【发布管理-已发布列表】：可查看已发布的流作业。

上线启动

已通过审批的作业流可以在【运维中心-任务管理】页面中看到，并可以进行相应操作，已拒绝的作业流不可以在【运维中心-任务管理】中看到。 点击【上线启动】按钮，需要对待上线的作业分配运行资源。完成资源分配的作业，将提交至【运维中心-生产实例-流计算作业】中，进行调度运行：启动、停止、创建savepoint、查看savepoint、查看flink ui等。

生产运行

- 【启动】：选择启动位点，点击按以上配置启动，作业即提交至计算集群运行。流计算作业启动时候您可以指定启动时间。表示从源头数据存储的指定时间点开始读取数据。例如设置启动时间为当前时间或者一个小时之前，也可以立即启动
- 【终止】：点击终止，可停止作业；
- 【创建savepoint】：创建后，再次启动作业时，可选择从savepoint标识的点启动作业；
- 【查看savepoint】：可以查看所有的savepoint列表。
- 【查看FlinkUI】 备注：查看Flink_UI前，需要提前配置部署集群的hosts，否则页面无法正常跳转。

常见问题

1. **什么是实时流计算？**

“实时”指实时处理，计算框架支持按消息时间逐条处理； “流”指数据如水流，一个接着一个；“计算”指数学运算、数据分析、算法模型执行等。 “实时流计算”指实时处理当下正在发生的数据流，逐条大数据分析或算法运算。

2. **目前要流计算引擎支持哪几种？在使用上有什么区别？**

目前支持Flink、Spark Streaming两种引擎。其中Flink支持上传jar包、通过IDE拖拽插件的方式进行流计算作业的开发，Spark Streaming只支持上传jar包的方式。

3. **目前要流计算支持哪几种数据源？**

Source支持：Kafka Sink支持：Kafka、Mysql、Oracle、MPP、ES、HBase、Redis

4. **如何快速测试流计算作业是否有问题？**

- 在IDE开发页面进行在线开发，开发完毕后，可点击“单元测试”快速进行作业测试；
- 单元测试使用线下上传的测试数据包充当数据源，支持对数据进行延迟、乱序测试，并快速生成结果；
- 单元测试无误后，可将作业发布到测试环境，进行试运行。经过数据验证后，可将该流计算作业发布至生产环境进行正式运行。

5. **流计算的作业怎么更新？**

- 流计算作业支持多版本，版本生成动作通过开发IDE界面的“提交”进行触发
- 作业提交前，可以在流计算的开发IDE页面进行作业编辑。
- 作业提交后，为保障已有版本作业的稳定，以后版本不可再进行编辑操作，但可以通过生成新版本的方式替换老版本，实现流作业的更新。

6. **SavePoint有什么作用，如何创建？**

SavePoint可以支持程序升级后，继续从升级前的那个点开始执行计算， 保证数据不中断。SavePoint需要用户手动创建。

7. **是否支持自定义函数，自定义函数的作用范围多大？**

流计算支持用户自定义函数，函数统一作用在租户级别。

8. **流计算是否支持用户自定义扩展插件？**

流计算支持Kafka Source、SQL operator、ETL operator、JDBC Sink、HBase Sink、ES Sink、Redis Sink、Kafka Sink 这8个插件，可灵活满足用户的各种需求。插件由平台统一维护，不支持租户自定义扩展。

9. **流计算有哪些使用限制？**

Flink Sql不支持TopN、Emit等语法，可采用ETL实现。 KafkaSource多流Join，不可同时支持多个Kafka 版本。