

阿里云

实时计算（流计算） 常见问题

文档版本：20220525

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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1. 计量计费	05
2. 基本概念	08
3. 如何使用	20
4. 分场景排错指引	32
5. 上下游存储	59
6. SQL 常见问题	66
6.1. 语法检查报错	66
6.2. 调试报错	71
6.3. 上线报错	75
6.4. 启动报错	80
6.5. 运行Failover	85

1. 计量计费

本章节汇总了使用实时计算过程中计量计费的常见问题。

- 为什么计算的CU跟显示消耗的CU不一致？
- 阿里云实时计算如何计量和收费？
- 阿里云实时计算是否带来其他费用？
- 为什么已使用CU和资源消耗总和不一致？
- 实时计算到期不续费，作业信息是否会被保留？
- 按量付费和包年包月两种付费模式可以相互转换吗？
- 已经完成购买的独享集群如何进行资源配置变更？
- 非高可用集群如何获得高可用性保障SLA服务？
- 初次购买的独享集群中Slave型号为8核32g，能否将Slave型号升配为16核64g或降配为4核16g？

为什么计算的CU跟显示消耗的CU不一致？

配置界面配置的资源只是单个Operator的计算资源，不包括框架额外消耗的资源。所以，Operator消耗CU数值相加，和最后运行消耗的CU数值不相同。

框架额外消耗的资源包括但不限于：

- 网络资源。
- 为Operator优化所消耗的其他资源。
- 各个JOIN、GROUP BY等节点分配的state_size所消耗的资源。

```
{
  "id": 17,
  "uid": "17",
  "pact": "Operator",
  "name": "JoinTable(.....)",
  "parallelism": 16,
  "maxParallelism": 32768,
  "vcore": 0.02,
  "heap_memory": 8,
  "native_memory": 2,
  "direct_memory": 0,
  "state_size": 1,
  "chainingStrategy": "ALWAYS"
},
```

② 说明 由于state_size=1，需要额外的堆外内存来进行缓存（需要运行的时候才知道缓存的量），这些资源无法体现到资源配置中。

阿里云实时计算如何计量和收费？

实时计算的基本计量单位为CU（Compute Unit），即计算资源。一个CU对应于实时计算底层系统是一个CPU的计算能力。实时计算底层使用虚拟化技术进行资源隔离，保证一个基本的CU消费且最大消费仅能为一个CPU的计算能力。

一个CU描述了一个实时计算作业最小运行能力，即在限定的CPU、内存、I/O情况下对于事件流处理的能力。一个实时计算作业可以指定在1个或多个CU上运行。

当前实时计算定义1CU的处理能力约为1000条数据/秒。

实时计算支持包年包月购买，详情请参见[开通服务和创建项目](#)。

② 说明 一个CU一天的单价是人民币6元，您可以根据业务以月或者年为单位选择购买的时长。例如：购买了10个CU、购买时长是一个月30天，则购买的费用为 $6 \times 10 \times 30 = 1800\text{RMB}$ 。

阿里云实时计算是否带来其他费用？

阿里云实时计算独立于其从中读取数据的流源和向其中写入处理数据的目标。因此您需要分别为输入和输出流相关的上下游付费。例如，您向LogHub写入数据，需要提供相应的存储费用。请查看所涉及到对应存储产品的收费细则。

为什么已使用CU和资源消耗总和不一致？

● 问题描述：

为什么总览界面上已使用CU与项目中所有作业的资源消耗总和不一致？如下图，已使用CU为22.93CU，而所有作业资源消耗总和为24.11CU。



● 解释说明：

作业资源计算方法和已使用CU计算方法不同，导致了二者的显示结果不同。

○ 作业资源计算方法：

CU是真实物理资源的一个抽象， $1\text{CU} = 1\text{核CPU} + 4\text{G MEM}$ （即内存）。一个作业真实消耗的CU数 = $\text{MAX}(\text{作业申请的CPU}, \text{作业申请的MEM}/4\text{G})$ 。例如，一个作业申请了1核CPU+5G MEM，那么实际消耗资源为1.25CU。

○ 已使用CU计算方法：

已使用CU = $\text{MAX}(\text{所有作业申请的CPU总和}, \text{所有作业申请的MEM总和}/4\text{G})$ 。

- 示例说明：

例如，1个项目有2个作业A和B。作业A申请的资源为（1核CPU，6G MEM）。作业B申请的资源为（1核CPU，2G MEM）。

- 作业实际消耗资源总和为2.5CU。单作业实际资源消耗如下：

- 作业A实际消耗资源为1.5CU。
- 作业B实际消耗资源为1CU。

- 整个项目实际的资源申请数为（2核CPU，8GMEM），即项目的已使用CU=2CU。

因此，项目总览界面上已使用CU（2CU）与项目中所有作业的资源消耗总和（2.5CU）有可能不一致。

实时计算到期不续费，作业信息是否会被保留？

实时计算到期释放后，不会保留数据。如果您再次购买实时计算服务后新建项目，可以导入原有项目的作业。

按量付费和包年包月两种付费模式可以相互转换吗？

按量付费和包年包月两种付费模式不可相互转换。

已经完成购买的独享集群如何进行资源配置变更？

已购买独享集群仅支持Master升配和Slave扩容或缩容（Slave数量的增减）。

非高可用集群如何获得高可用性保障SLA服务？

非高可用集群（一台Master）和高可用集群（三台Master）不能够进行相互转换。仅高可用集群（三台Master）支持高可用性保障SLA服务。如果您购买的是非高可用集群（一台Master），需要重新购买高可用集群（三台Master）后，才可以获得高可用性保障SLA服务。

初次购买的独享集群中Slave型号为8核32g，能否将Slave型号升配为16核64g或降配为4核16g？

无法对已购买的独享集群中Slave的型号进行变更，只能对Slave的数量进行调整。需要重新购买集群实现对Slave型号的升降配需求。

2.基本概念

本文汇总了实时计算Flink版基本概念、作业操作、调试运维等方面的常见问题。

基本概念

- 什么是实时计算？
- 实时计算与批量计算相比存在哪些差异？
- 阿里云实时计算Flink版适用哪些场景？
- 阿里云实时计算Flink版使用有哪些限制？
- 什么是流数据？
- 阿里云实时计算Flink版的数据存储有哪些类型？

作业操作

- 实时计算Flink版的全链路示意图是什么样的？
- 实时计算Flink产品全链路使用流程是怎样的？
- 单个阿里云实时计算Flink版作业包含几个部分？
- 实时计算Flink版数据的状态保存时间是多久？
- 如何加载数据到实时计算Flink版？
- 实时计算Flink版处理后的结果数据如何继续应用？
- 实时计算Flink版中一个作业里可以编写多个INSERT语句、插入到多个数据结果表吗？

调试与运维

- 开发调试和生产运维区别是什么？
- 作业暂停恢复和停止重启的区别是什么？
- 什么是启动位点？

其他

- 阿里云实时计算Flink版提供的编程接口是什么？
- 实时计算Flink版SQL作业支持全局ORDER BY吗？
- 无窗口和窗口这两类SQL的业务含义是什么？

什么是实时计算？

数据的业务价值会随着时间的流失而迅速降低，因此在数据发生后必须尽快对其进行计算和处理，目前，对于信息的高时效性和可操作性要求越来越高，这就要求软件系统能够在更短的时间内处理更多的数据。而传统的大数据处理模型将在线事务处理和离线分析从时序上完全分开，对于数据加工均遵循传统的日清日毕模式，即以小时甚至以天为计算周期对当前数据进行累计并处理。显然，传统的大数据处理方式无法满足数据实时计算的需求。数据处理时延造成的影响对要求苛刻的业务场景会非常明显，例如实时大数据分析、风控预警、实时预测和金融交易等领域。

实时计算可以有效地缩短全链路数据流时延、实时化计算业务逻辑、平摊计算成本，最终有效满足实时处理大数据的业务需求。

实时计算具备三大特点：

- 实时（Realttime）且无界（Unbounded）的数据流
实时计算面对的数据是实时且流式的，这些数据按照时间发生顺序被实时计算订阅和消费。例如网站的访问单击日志流，只要网站不关闭，其单击日志流将不停产生并进入实时计算系统。
- 持续（Continuous）且高效的计算

实时计算是一种事件触发的计算模式，触发源为无界流式数据。一旦有新的流数据进入实时计算系统，它就立刻发起并进行一次计算任务，因此整个过程是持续进行的。

- 流式（Streaming）且实时的数据集成

被流数据触发的计算结果，可以被直接写入目的数据存储。例如将计算后的报表数据直接写入阿里云关系型数据库RDS（Relational Database Service）进行报表展示。因此流数据的计算结果可以同流式数据一样，持续被写入目的数据存储。

实时计算与批量计算相比存在哪些差异？

下面从用户和产品层面来理解两类计算方式的区别。

- 批量计算

批量计算是一种批量、高时延、主动发起的计算。目前绝大部分传统数据计算和数据分析服务均是基于批量数据处理模型：使用ETL系统或者OLTP系统进行构造数据存储，在线的数据服务（包括Ad-Hoc查询、DashBoard等）通过构造SQL语言访问上述数据存储并取得分析结果。这套数据处理的方法论伴随着关系型数据库在工业界的演进而被广泛采用。传统的批量数据处理模型如下图所示。



- i. 装载数据

对于批量计算，用户需要预先将数据加载到计算系统，您可以使用ETL系统或者OLTP系统装载原始数据。系统将根据自己的存储和计算情况，对于装载的数据进行一系列查询优化、分析和计算。

- ii. 提交请求

系统主动发起一个计算作业（例如MaxCompute的SQL作业，或Hive的SQL作业）并向上述数据系统进行请求。此时计算系统开始调度（启动）计算节点进行大量数据计算，该过程的计算量可能非常大，耗时长达数分钟乃至数小时。由于数据累计处理不及时，上述计算过程中可能就会存在一些历史数据，导致数据不新鲜。

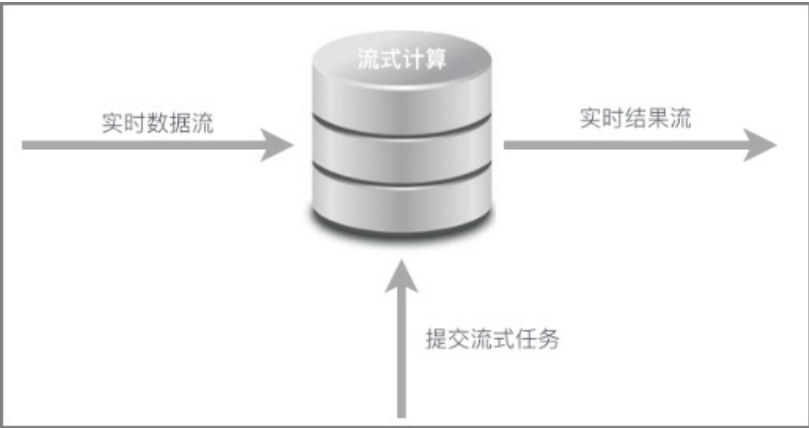
② 说明 您可以根据业务需要随时调整计算SQL后再次提交作业，您甚至可以使用AdHoc查询做到即时修改即时查询。

- iii. 返回结果

计算作业完成后将数据以结果集形式返回给用户，由于保存在数据计算系统中的计算结果数据量巨大，需要用户再次集成数据到其他系统。一旦数据结果巨大，整体的数据集成过程就会漫长，耗时可能长达数分钟乃至数小时。

- 实时计算

实时计算是一种持续、低时延、事件触发的计算作业。相对于批量计算，流式计算整体上还属于比较新颖的计算概念。由于当前实时计算的计算模型较为简单，所以在大部分大数据计算场景下，实时计算可以看做是批量计算的增值服务，实时计算更强调计算数据流和低时延。实时计算数据处理模型如下。



i. 实时数据流

使用实时数据集成工具，将实时变化的数据传输到流式数据存储（例如消息队列、DataHub）。此时数据的传输实时化，将长时间累积的大量数据平摊到每个时间点，不停地小批量实时传输，因此数据集成的时延得以保证。

源源不断的数据被写入流数据存储，不需要预先加载的过程。同时，流计算对于流式数据不提供存储服务，数据持续流动，在计算完成后就被立刻丢弃。

ii. 提交流式任务

批量计算要等待数据集成全部就绪后才能启动计算作业，而流式计算作业是一种常驻计算服务。实时计算作业启动后，一旦有小批量数据进入流式数据存储，流计算会立刻计算并得出结果。同时，阿里云流计算还使用了增量计算模型，将大批量数据分批进行增量计算，进一步减少单次运算规模并有效降低整体运算时延。

从用户角度，对于流式作业，必须预先定义计算逻辑，并提交到流式计算系统中。在整个运行期间，流计算作业逻辑不可更改。用户通过停止当前作业运行后再次提交作业，此时之前已经计算完成的数据是无法重新再次被计算。

iii. 实时结果流

不同于批量计算，结果数据需等待数据计算结果完成后，批量将数据传输到在线系统。流式计算作业在每次小批量数据计算后，无需等待整体的数据计算结果，会立刻将数据结果投递到在线/批量系统，实现计算结果的实时化展现。

使用实时计算的顺序如下：

- i. 提交实时计算作业。
- ii. 等待流式数据触发实时计算作业。
- iii. 计算结果持续不断对外写出。

计算模型差别对比。

对比指标	批量计算	实时计算
数据集成方式	预先加载数据。	实时加载数据到实时计算。
使用方式	业务逻辑可以修改，数据可重新计算。	业务逻辑一旦修改，之前的数据不可重新计算（流数据易逝性）。

对比指标	批量计算	实时计算
数据范围	对加载的所有或大部分数据进行查询或处理。	对滚动时间窗口内的数据或仅对最近的数据记录进行查询或处理。
数据大小	大批量数据。	单条记录或几条记录的微批量数据。
性能	几分钟至几小时的延迟。	大约几秒或几毫秒的延迟。
分析	复杂分析。	简单的响应函数、聚合和滚动指标。

阿里云实时计算Flink版适用哪些场景？

阿里云实时计算Flink版提供标准的Flink SQL语义可以协助您完成流式计算逻辑的处理，但无法满足某些特定场景的业务需求。阿里云实时计算Flink版为部分授信用户提供全功能的UDF函数，帮助授信用户完成业务定制化的数据处理需求。所以，在流数据分析领域，您可以直接使用Flink SQL+UDF即可完成大部分流式数据分析处理逻辑。

阿里云实时计算Flink版优势和限制如下：

- 阿里云实时计算Flink版更擅长于进行流式数据分析、统计和处理。
- 阿里云实时计算Flink版不适合于SQL不能够解决的领域。例如，复杂的迭代数据处理和复杂的规则引擎告警。

阿里云实时计算Flink版使用有哪些限制？

请您评估以下限制对您业务的影响，并做相应的调整。如有疑问，请您[提交工单](#)进行咨询。

- 阿里云实时计算Flink版购买指导，请参见[如何购买](#)。
- 阿里云实时计算Flink版WebConsole仅支持Chrome浏览器访问。
- 阿里云实时计算Flink版支持华东、华北、华南地域，您可以按区域购买。
- 阿里云实时计算Flink版在简单的流式压测处理场景下（例如过滤、清洗等），一个实时计算CU的处理能力为1000条/秒。在复杂的流式压测处理场景下（例如复杂UDF计算、聚合操作等），一个实时计算CU的处理能力为500条/秒。

 说明 请根据您的业务情况，评估所需的CU数量。

- 阿里云实时计算Flink版对整个Project下属的Task、Task版本、IDE打开Task页面均有不同限制，包括：
 - 一个阿里云账号可以购买多个Project项目。
 - 单个Project下允许最多创建100个job。
 - 单个Project下最多允许50个文件夹，深度最大不超过5层。
 - 单个Project下最多允许50个UDF/JAR。
 - 单个Project下最多允许50个数据存储注册。
 - 单个Task最多允许20个历史版本保存数。

什么是流数据？

所有大数据的产生均可以看作是一系列离散事件，这些离散事件是一条条事件流或数据流。相对于离线数据，流数据的规模普遍较小。流数据是由数据源持续产生的数据，示例如下：

- 使用移动或Web应用程序生成的日志文件。
- 网购数据。

- 游戏内玩家活动信息。
- 社交网站信息。
- 金融交易大厅或地理空间服务数据中心内所连接设备或仪器的遥测数据。
- 地理空间服务信息。
- 设备或仪器的遥测数据。

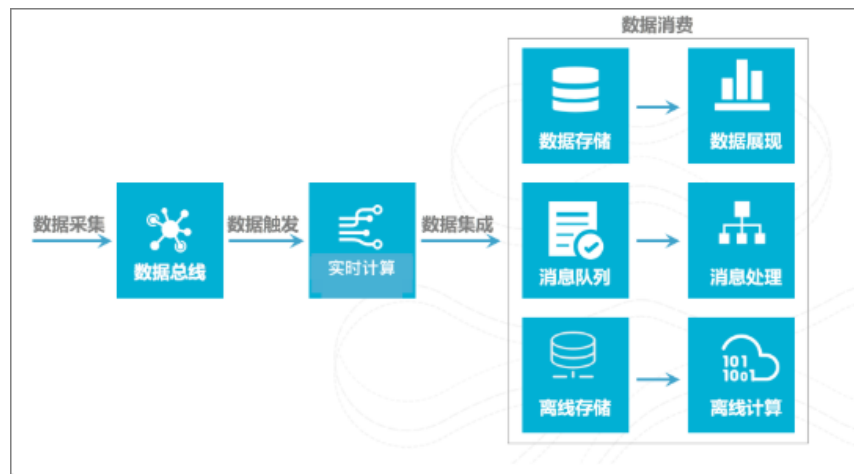
阿里云实时计算Flink版的数据存储有哪些类型？

阿里云实时计算Flink版本本身不带有业务存储，所有的数据均是来自于外部存储系统持有的数据。阿里云实时计算Flink版支持以下几类数据存储类型：

- 流式的数据输入：流式数据的输入会触发和推动实时计算Flink版进行数据处理。每个实时计算Flink版作业必须至少声明一个流式数据输入源。
- 静态数据输入：静态数据输入也被称为维表，对于每条流式数据，可以关联一个外部静态数据源进行查询，为实时计算Flink版提供数据关联查询。
- 结果表输出：实时计算Flink版将计算的结果数据写出到目的数据表，为下游数据继续消费提供各类读写接口。

实时计算Flink版的全链路示意图是什么样的？

不同于现有的离线或批量计算模型，实时计算Flink版全链路整体上更加强调数据的实时性，包括数据的实时采集、实时计算和实时集成。三大类数据的实时处理逻辑在全链路上保证了流式计算的低时延。全链路实时计算Flink版示意图如下。



1. 数据采集

使用流式数据采集工具将数据实时地采集并传输到大数据消息Pub/Sub系统，该系统将为下游实时计算Flink版提供源源不断的事件源去触发流式计算作业的运行。

2. 数据触发

流数据作为实时计算Flink版的触发源，驱动实时计算Flink版运行。因此，一个实时计算Flink版作业必须至少使用一个流数据作为源。进入的数据流将直接触发实时计算Flink版进行计算。

3. 数据集成

实时计算Flink版将计算的结果数据直接写入目的数据存储，这其中包括多种数据存储系统、消息投递系统，甚至直接对接业务规则告警系统以发出告警信息。不同于批量计算（阿里云MaxCompute或开源Hadoop），实时计算Flink版天生自带数据集成模块，可以将结果数据直接写入到目的数据存储。

4. 数据消费

一旦结果数据从实时计算Flink版投递到目的数据源后，数据消费就和实时计算Flink版完全解耦。您可以使用数据存储系统访问数据，使用消息投递系统进行信息接收，或者直接使用告警系统进行告警。

实时计算Flink产品全链路使用流程是怎样的？

1. 验证网络连通性。

使用网络探测功能验证，详情请参见[存储注册方式](#)中的网络探测。

2. 引用外部存储。

详情请参见[概述](#)文档引用外部存储。

 **说明** 如果不支持注册存储功能，请使用[明文方式](#)。

3. 编写源表、结果表、维表（可选）的DDL和DML语句。

```
CREATE TABLE test_input (  
  id      BIGINT,  
  `name`  VARCHAR,  
  score   INT,  
  `time`  BIGINT,  
  tbirus AS TO_TIMESTAMP(`time`/1000), --使用计算列，将16位时间戳转换为13位时间戳。  
  WATERMARK FOR ts AS WITHOFFSET(ts, 1000) --使用窗口，其中CEP、OVER窗口需要声明。  
)with(  
  type='datahub',  
  ...  
);  
  
CREATE TABLE test_output (  
  window_start TIMESTAMP,  
  window_end   TIMESTAMP,  
  ct           BIGINT  
)with(  
  type='rds',  
  ...  
);  
--简单的滚动窗口示例。  
INSERT INTO  
  test_output  
SELECT  
  TUMBLE_START(ts, INTERVAL '1' MINUTE),  
  TUMBLE_END(ts, INTERVAL '1' MINUTE),  
  count(*)  
FROM test_input  
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE);
```

4. 语法检查。

详情请参见[开发](#)。

5. 调试。

- 调试结果仅供展示，不输出至结果表。
- 部分外部存储支持数据源表抽取功能。例如，DataHub源表、TableStore源表。如果不支持数据源表抽取功能，请下载模板自行上传数据完成调试。
- Kafka中VARBINARY类型不支持调试。
- 部分函数例如EMIT语法不支持调试。

- 调试输出结果中，同时会显示过程数据。

6. 上线运行。

详情请参见[如何利用报错信息快速定位作业问题？](#)

7. 调优。

- 优化SQL逻辑。

详情请参见[高性能Flink SQL优化技巧](#)。

- 添加WITH参数。

- 添加作业参数。

详情请参见：

- [高性能Flink SQL优化技巧](#)。

- [手动配置调优](#)。

- 资源调优。

详情请参见：

- [AutoConf 自动配置调优](#)（实时计算Flink版1.0和2.0版本支持）。

- [AutoScale自动配置调优](#)（实时计算Flink版3.2.2及以上版本支持该功能）。

- [手动配置调优](#)。

单个阿里云实时计算Flink版作业包含几个部分？

- SQL代码：SQL代码是计算作业的核心业务逻辑，其中包括输入表DDL声明（流式输入表、静态输入表）、输出表DDL声明，以及执行业务逻辑的DML语句。
- 参数：参数用来描述作业运行时指标。例如并发量、批处理数据量等信息。
- 属性：作业的业务信息。例如创建人、创建时间等相关记录。
- 资源配置：配置作业的资源。

实时计算Flink版数据的状态保存时间是多久？

实时计算Flink版数据的状态默认保存36小时，如果某个状态超过36小时没有更新，之前的状态可能会被清掉。状态设置步骤如下：

1. 登录[实时计算控制台](#)。
2. 单击页面顶部的开发。
3. 在作业开发区域，双击目标文件夹或目标作业名，进入作业编辑页面。
4. 在作业编辑区域右侧导航栏，单击作业参数。
5. 在作业参数页面输入状态配置参数，示例如下。

注意

- 状态（state）的生命周期（ttl）设置过小，会导致retract时产生NullPointerException的报错。
- 状态（state）的生命周期（ttl）设置过大，会导致子线资源的消耗增大。

- 实时计算Flink版1.0版本。

```
#使用rocksdb作为statebackend。  
state.backend.type=rocksdb  
#rocksdb的数据生命周期，单位为毫秒。  
state.backend.rocksdb.ttl.ms=129600000
```

- 实时计算Flink版2.0及以上版本。

```
#使用niagara作为statebackend。  
state.backend.type=niagara  
#niagara的数据生命周期，单位为毫秒。  
state.backend.niagara.ttl.ms=129600000
```

- 实时计算Flink版3.0及以上版本。

```
#使用Gemini作为statebackend。。  
state.backend.type=gemini  
#gemini的数据生命周期，单位为毫秒。  
state.backend.gemini.ttl.ms=129600000
```

如何加载数据到实时计算Flink版？

实时计算Flink版支持以下两种加载数据的方式：

- 流式数据输入：触发实时计算Flink版进行数据处理，推动实时计算Flink版持续进行数据计算。每个实时计算作业必须至少声明一个流式数据输入源。
- 静态存储：静态数据输入通常是维表，对于每条流式数据，可以关联一个外部静态数据源进行查询，为实时计算Flink版提供了数据关联查询。

实时计算Flink版处理后的结果数据如何继续应用？

实时计算Flink版将计算的结果数据输出到目的数据表，并且为下游数据继续应用提供各类读写接口。

实时计算Flink版中一个作业里可以编写多个INSERT语句、插入到多个数据结果表吗？

- 可以编写存在多个结果表。使用多个INSERT（DML）语句的作业代码，分别插入数据结果表。
- 可以编写存在多个源表的作业代码。多数据源表可以进行双流JOIN、UNION等操作。

 **注意** 不建议将完全不相关的链路写在同一个作业中。如果某条链路出现问题，则会影响整个作业，其他不相关的链路也会受到影响。

开发调试和生产运维区别是什么？

开发调试和生产运维主要有以下两点区别：

- 开发调试对线上作业和数据存储不造成任何影响。

开发调试发生在调试环境，所有的Flink SQL运行将在独立的调试容器运行，且所有的输出将被直接改写到调试结果屏幕，因此不会对线上生产的实时计算Flink版作业、线上生产的数据存储系统造成任何影响。

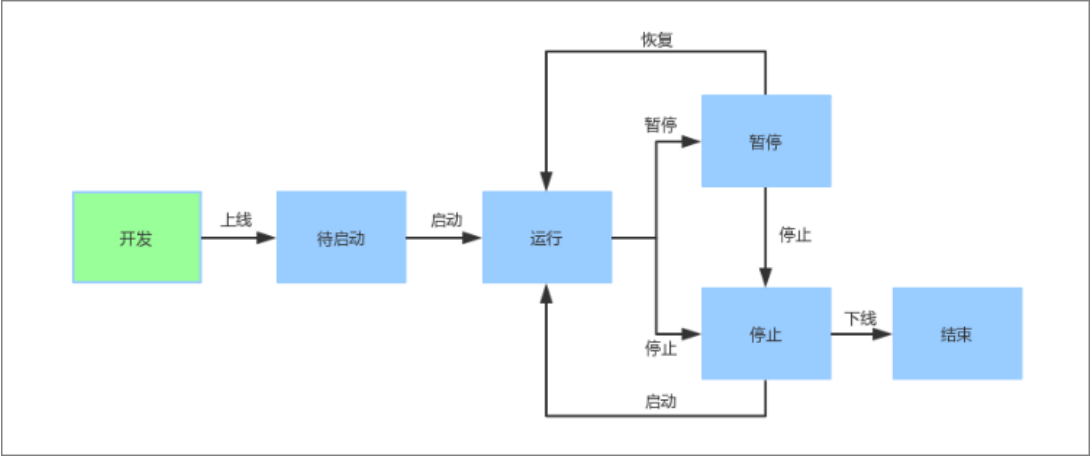
- 开发调试过程无法完全检测部分实际生产过程中的异常。

开发调试过程实际上不是将调试数据写入到外部数据源，而是被实时计算Flink版拦截输出至调试结果页面。因此在实时计算Flink版的调试代码是在调试容器中运行的，所以线上运行过程中可能由于对目标数据源写入格式的不兼容，导致作业运行失败。这类错误在开发调试阶段无法被发现和处理，只能到线上运行后才能发现。

例如，您的结果数据输出到RDS系统，其中某些字段输出字符串数据长度大于RDS建表长度最大值，在调试环境下系统无法测试出该类问题，但实际生产运行过程中会引发异常。后续，实时计算Flink版提供将调试结果写入真实数据源的本地调试功能，有效辅助您缩短调试和生产的差距，尽可能在调试阶段发现并解决问题。

作业暂停恢复和停止重启的区别是什么？

实时计算Flink版作业生命周期状态



暂停和停止的区别：

- 停止操作会清除作业的所有状态，停止后再启动相当于全新的一个作业。
- 暂停操作会保留作业的所有状态：
 - 如果状态可以兼容，暂停后会复用暂停前的作业状态。
 - 如果状态无法兼容，则会报错无法恢复。

 **注意** 如果无法兼容，需要停止作业后，启动作业，否则作业状态将会被清理，请谨慎操作。

暂停作业后，再恢复作业，作业状态是否兼容主要取决于实际生成的物理执行图是否一致，和具体的底层优化策略相关。暂停恢复后作业的兼容性如下表所示。

操作变更类型	暂停恢复后状态是否兼容
修改SQL	不兼容
修改资源配置	兼容
修复作业参数	○ 增删作业参数不兼容。 ○ 修改作业参数值一般情况下兼容，但改变部分optimizer参数值，修改statebackend类型等会导致不兼容。
修改WITH参数	○ 增删WITH参数不兼容。 ○ 修改WITH参数值一般情况下兼容，但类似于维表async true <-> false等情况会导致不兼容。
修改资源引用	通常情况下兼容，但类似于用UDF变更影响Schema或者UDAGG则不兼容。
修改作业版本	自Blink 3.2.3以后，最后一位版本号修改可兼容。不兼容其它版本切换。

操作变更生效的前提是作业重新上线，并且恢复时选择按**最新配置恢复**。如果选择按**之前配置恢复**，则所有操作变更均不生效，效果等同于原作业暂停恢复。

什么是启动位点？

启动位点是读取数据的时间点。启动位点的时间点以后产生的数据，都将按照时间的顺序输入至新的作业中进行计算。

阿里云实时计算Flink版提供的编程接口是什么？

阿里云实时计算Flink版提供Flink SQL编程接口编写业务逻辑，为流式数据分析定制多种数据处理函数和操作符。以Word Count为例，通过Flink SQL编写的业务逻辑如下所示。

```
--声明一个流式源表。
create table stream_source(word string) ;
--声明一个目标表。
create table stream_result(word string, cnt bigint) ;
--统计word次数。
insert into stream_result
select t.word,
       count(1)
from stream_source t
group by t.word;
```

实时计算Flink版SQL作业支持全局ORDER BY吗？

实时计算Flink版作业不支持全局ORDER BY，ORDER BY需要在OVER窗口中使用。详情请参见[TopN语句](#)。

无窗口和窗口这两类SQL的业务含义是什么？

- 窗口函数Group Window Aggregation

Group Window Aggregation是一种在窗口上的聚合。Group Window Aggregation在每个窗口结束且无提前的观测值（early firing）时，会发出的一条结果数据，例如MicroBatch。时间窗口有以下几种：

- TUMBLE：滚动窗口
- HOP：滑动窗口
- SESSION：会话窗口

例如，统计过去的1分钟内，多少用户单击了某个网页。在这种情况下，我们可以定义一个窗口，用来收集最近一分钟内的数据，并对这个窗口内的数据进行计算。

测试数据

username (VARCHAR)	click_url (VARCHAR)	ts (TIMESTAMP)
Jark	http://taobao.com/xxx	2017-10-10 10:00:00.0
Jark	http://taobao.com/xxx	2017-10-10 10:00:10.0
Jark	http://taobao.com/xxx	2017-10-10 10:00:49.0
Jark	http://taobao.com/xxx	2017-10-10 10:01:05.0
Jark	http://taobao.com/xxx	2017-10-10 10:01:58.0
Timo	http://taobao.com/xxx	2017-10-10 10:02:10.0

```
INSERT INTO tumble_output
SELECT
  TUMBLE_START(ts, INTERVAL '1' MINUTE),
  TUMBLE_END(ts, INTERVAL '1' MINUTE),
  username,
  COUNT(click_url)
FROM window_input
GROUP BY TUMBLE(ts, INTERVAL '1' MINUTE), username;
```

测试结果

window_start (TIMESTAMP)	window_end (TIMESTAMP)	username (VARCHAR)	clicks (BIGINT)
2017-10-10 10:00:00.0	2017-10-10 10:01:00.0	Jark	3
2017-10-10 10:01:00.0	2017-10-10 10:02:00.0	Jark	2
2017-10-10 10:02:00.0	2017-10-10 10:03:00.0	Timo	1

无窗口函数Group Aggregation

Group Aggregation是一种无限流的聚合。因为没有窗口，所以每到达一条数据就会增量计算一次，并发出更新后的结果。实时计算Flink版状态保存在以下三层存储媒介：

- 第一层：内存，作为缓存来使用。
- 第二层：硬盘，实时计算使用SSD作为存储。
- 第三层：HDFS。

因此，不存在内存占满的问题。

测试数据

Customer	OrderPrice
Bush	1000
Carter	1600
Bush	700
Bush	300
Adams	2000
Carter	100

```
SELECT Customer,SUM(OrderPrice) FROM XXXX  
GROUP BY Customer;
```

测试结果

Customer	SUM(OrderPrice)
Bush	2000
Carter	1700
Adams	2000

3. 如何使用

本章节汇总了实时计算使用、网络连通性、效率提升等相关的常见问题。

使用

- 如何查看作业的系统版本信息？
- 如何查看集群名称、项目名称和作业名称？
- 如何查看AccessKey ID和AccessKey Secret信息？
- 如何查看更新后的SQL是否生效？
- 如何查看维表读取数据的进度？
- 如何切换系统版本？
- 如何在SQL代码中动态获取作业参数？
- 如何处理业务数据中的特殊字符？
- 如何重新获取配置资源？
- 如何使用系统时间进行分组聚合计算？
- 重启作业时，是否能保证数据不丢失且不重复？

网络连通性

- 如何查看独享集群白名单和VPC信息？
- 独享集群如何访问公网？
- 如何对网络连通性问题进行排查？
- 如何访问跨VPC里的存储资源？

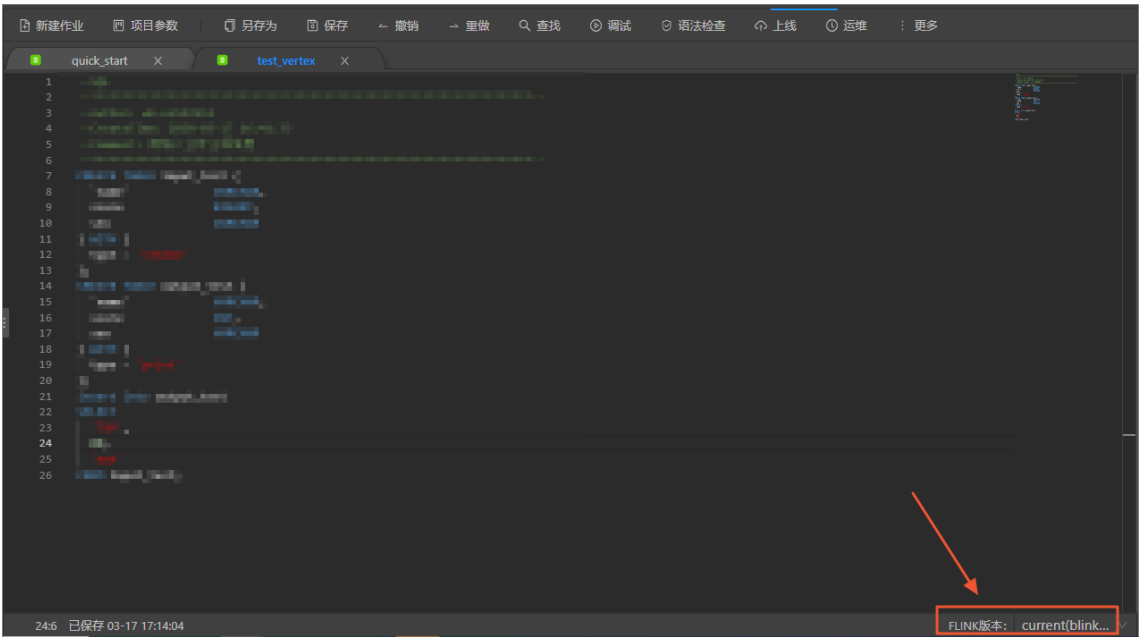
效率提升

- 如何快速编写和测试SQL？
- 如何优化实时计算各个节点资源配置以减少延迟？

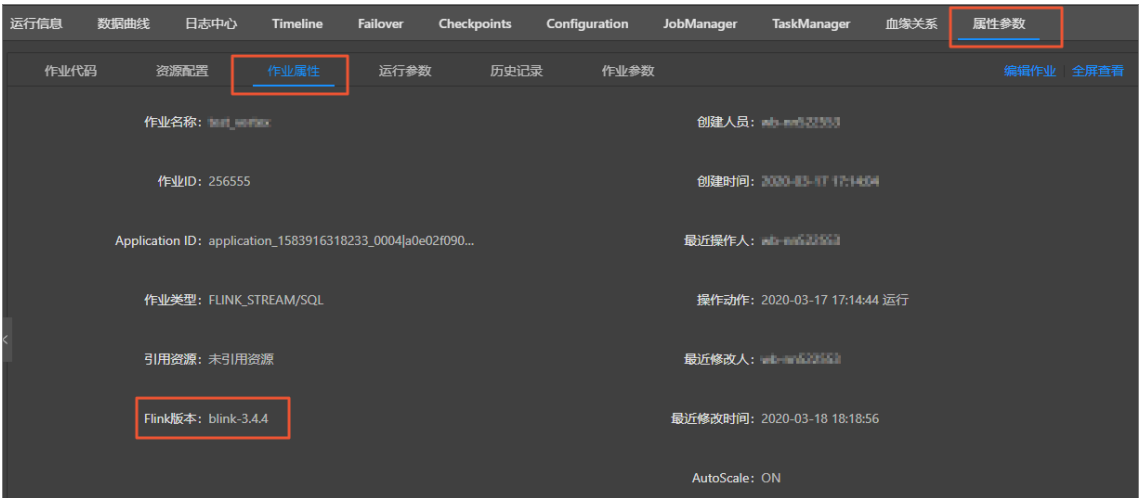
如何查看作业的系统版本信息？

实时计算支持不同的系统版本，您可以在实时计算管理控制台的开发页面或者运维页面查看当前作业的系统版本信息。

- 在开发页面查看系统版本信息
 - i. 登录[实时计算控制台](#)。
 - ii. 单击顶部的**开发**。
 - iii. 在开发页面的右下角的**Flink版本**，查看系统版本信息。



- 在运维页面查看系统版本信息
 - i. 登录[实时计算控制台](#)。
 - ii. 单击顶部的运维。
 - iii. 在运维页面，单击目标作业名称。
 - iv. 在作业运维 > 属性参数 > 作业属性 > Flink版本页面，查看系统版本信息。



如何查看集群名称、项目名称和作业名称？

- 登录[实时计算控制台](#)。
- 将鼠标悬停至页面右上角账号信息位置，选择项目管理。
- 在项目列表页面，单击目标项目名称，进入项目概览页面，查看以下信息：
 - 集群名称

您可以在实时计算控制台的总览页面查看实时计算项目的名称和项目所归属的集群名称。



项目名称

您可以在实时计算控制台的总览页面查看实时计算项目的名称。

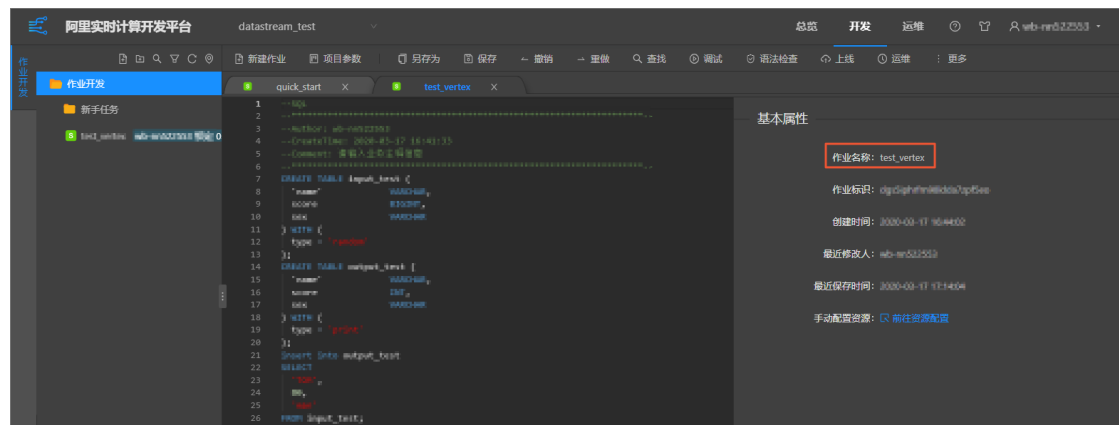


作业名称

您可以在实时计算控制台的开发或运维页面查看实时计算作业的名称。

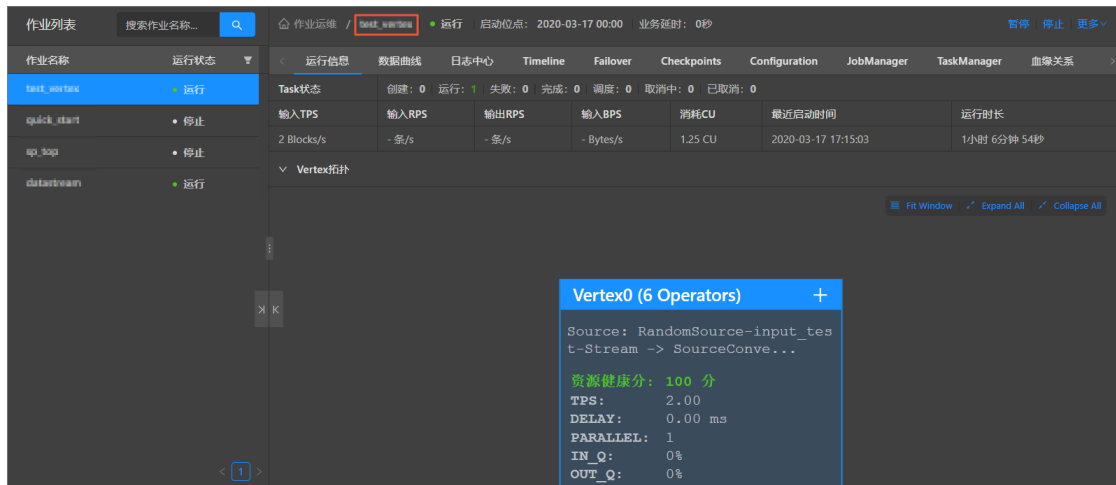
在开发页面查看作业名称

您可以在作业编辑页面右侧的基本属性页面中，查看作业的名称。



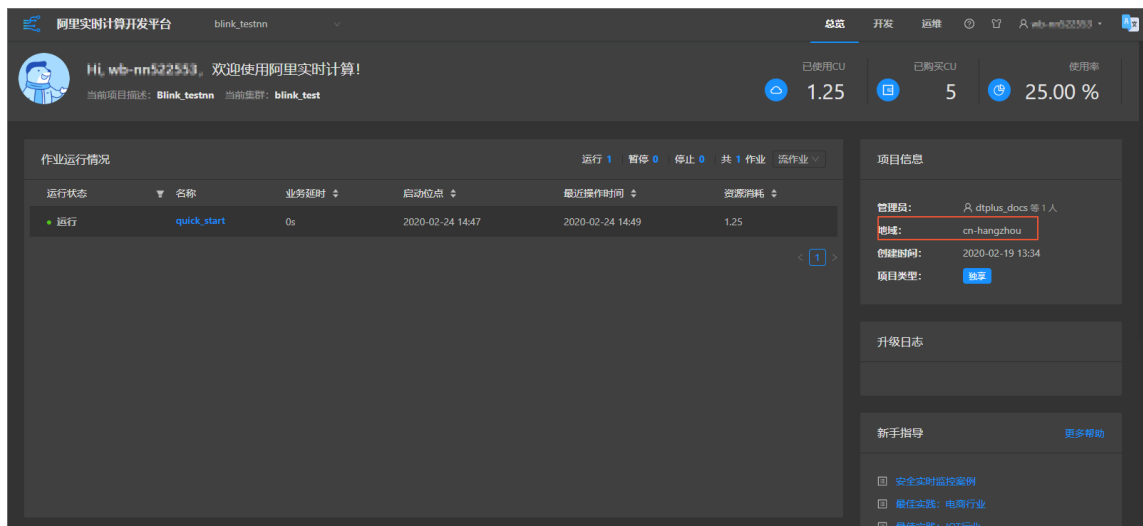
■ 在运维页面查看作业名称

您可以在作业运维页面的顶部查看作业的名称。



○ 地域信息

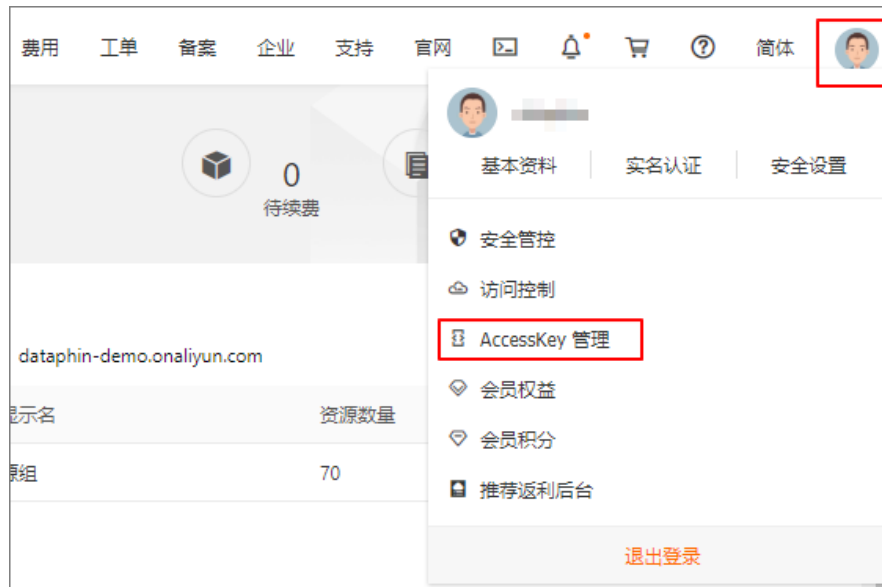
在总览页面右侧项目信息中，查看项目地域信息。



如何查看AccessKey ID和AccessKey Secret信息？

您可以通过阿里云管理控制台获取您的密钥信息，查询步骤如下：

1. 登录[阿里云控制台](#)。
2. 将鼠标悬停至页面右上角的头像，单击Accesskey 管理。

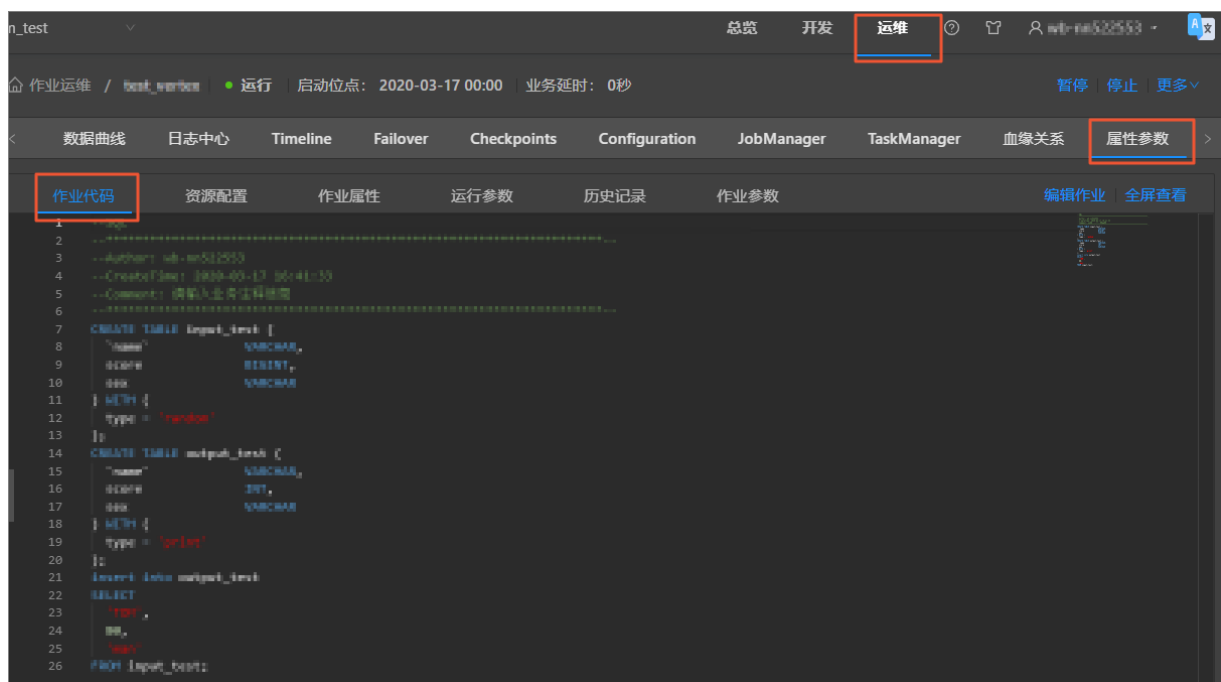


3. 在安全信息管理页面查看您的AccessKey ID和Access Key Secret信息。



如何查看更新后的SQL是否生效？

您可以在运维界面的属性参数 > 作业代码中查看更新后的SQL代码，确认您的变更是否生效。

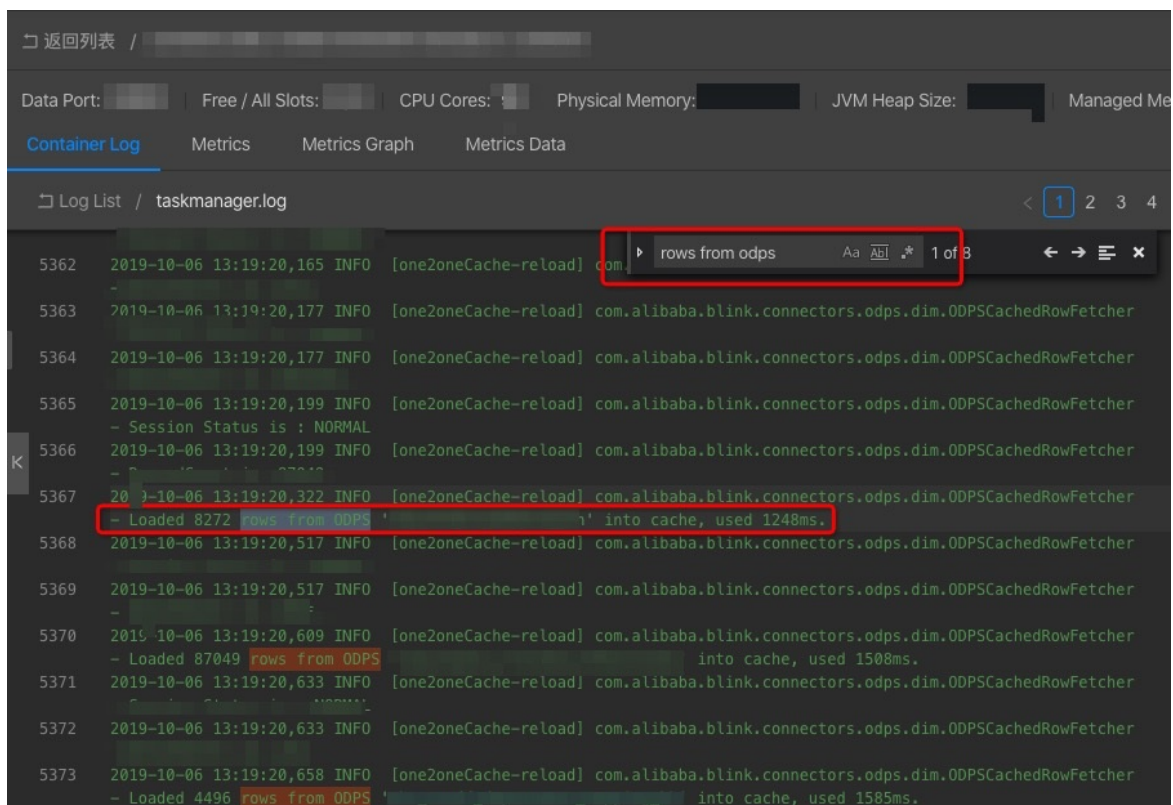


② 说明 查看作业SQL变更生效前，请确保您按照作业生效的完整流程完成了SQL变更：停止作业 > 更新SQL > 上线作业 > 启动作业。

如何查看维表读取数据的进度？

您可以通过实时计算控制台中的taskmanager.log日志信息，查看维表读取数据的进度。具体操作步骤如下：

1. 登录[实时计算控制台](#)。
2. 单击顶部的运维。
3. 在运维页面，单击目标作业名称。
4. 单击对应的JOIN节点拓扑图。
5. 单击SubTasks列表右侧的LOGO，进入taskmanager.log日志。
6. 按下Ctrl + F，检索rows from <维表类型>，例如rows from odps。



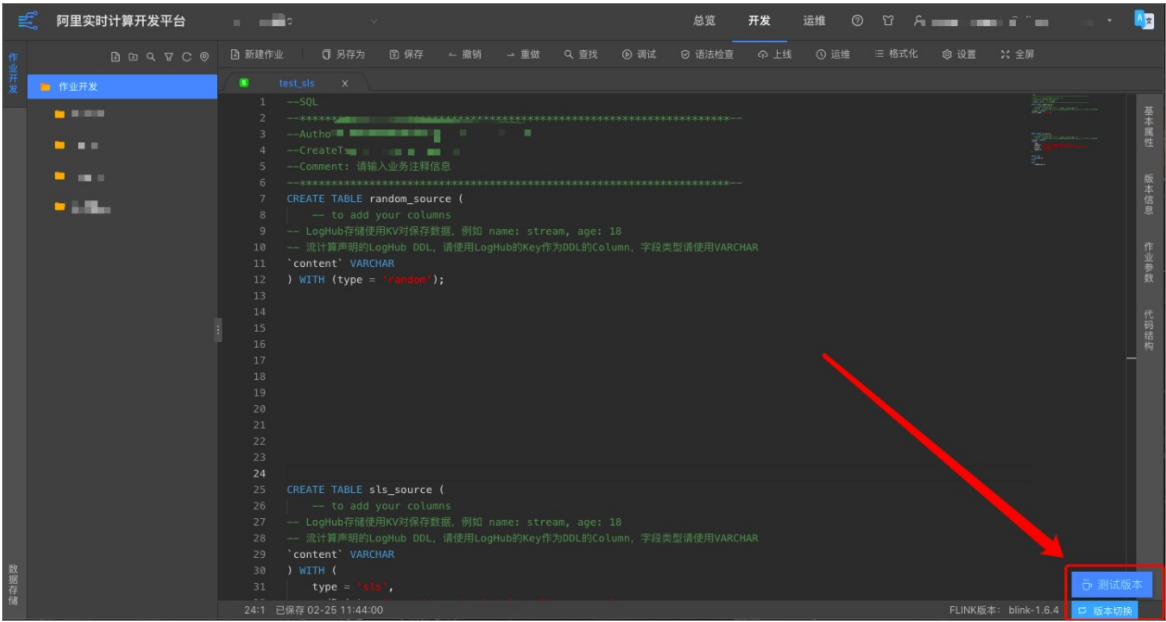
- ② 说明 如果通过以上步骤无法检索到相应的进度信息，请从以下方面进行排查：
- 维表是否具备相应的权限（例如，MaxCompute需要具备download等权限）。
 - WITH参数是否正确。
 - taskmanager.log日志中是否存在异常信息。taskmanager.log日志中异常信息查询方法参见[如何从taskmanager.log中查找脏数据等报错信息？](#)

如何切换系统版本？

实时计算支持多种系统版本，您可以在实时计算管理控制台的开发页面对系统版本进行切换，操作步骤如下：

1. 登录[实时计算控制台](#)。

- 2. 单击顶部的开发。
- 3. 在开发页面的右下角，选择对应的版本号。



- 4. 单击确定，完成切换。



说明

- 如果您使用的是独享集群，且只安装了一个系统版本，您需要先安装新版本再进行切换。详情请参见[管理独享集群Blink版本](#)。
- 系统版本切换完成后，需要进行重新上线（生成新的planJSON）和启动作业操作，才能使更新后的系统版本在作业中生效。

如何在SQL代码中动态获取作业参数？

您可以对作业中指定的参数进行配置，以达到作业更新后，作业SQL代码中WITH参数信息自动同步的目的。操作步骤如下：

- 1. 登录[实时计算控制台](#)。
- 2. 单击顶部的开发。
- 3. 在作业编辑页面，单击右侧的[作业参数](#)。
- 4. 在[作业参数](#)页面，输入WITH参数的动态配置信息。

```
1 #checkpoint模式: EXACTLY_ONCE 或者 AT_LEAST_ONCE
2 #blink.checkpoint.mode=EXACTLY_ONCE
3
4 #checkpoint间隔时间, 单位毫秒
5 #blink.checkpoint.interval.ms=180000
6
7 #rocksdb的数据生命周期, 单位毫秒
8 #state.backend.rocksdb.ttl.ms=129600000
9
10 status.topic=...
11 status.accessId=...
12 status.accessKey=g...
13 status.consumerGroup=...
14 status.endPoint=...
```

5. 在对应的作业编辑页面，在作业代码中，使用 `${}` 声明WITH参数。

```
1 ) WITH (
2
3   type = 'mq',
4   topic = '${status.topic}',
5   pullIntervalMs = '100',
6   consumerGroup = '${status.consumerGroup}',
7   endPoint = '${status.endPoint}',
8   accessId = '${status.accessId}',
9   accessKey = '${status.accessKey}'
10 );
```

如何处理业务数据中的特殊字符？

如果您的业务数据中存在特殊字符（即系统使用的字符或者SQL关键字），则需要使用REGEXP_REPLACE函数对特殊字符进行转义处理。详情请参见[REGEXP_REPLACE](#)。

常见的特殊字符和转义方式如下表所示。

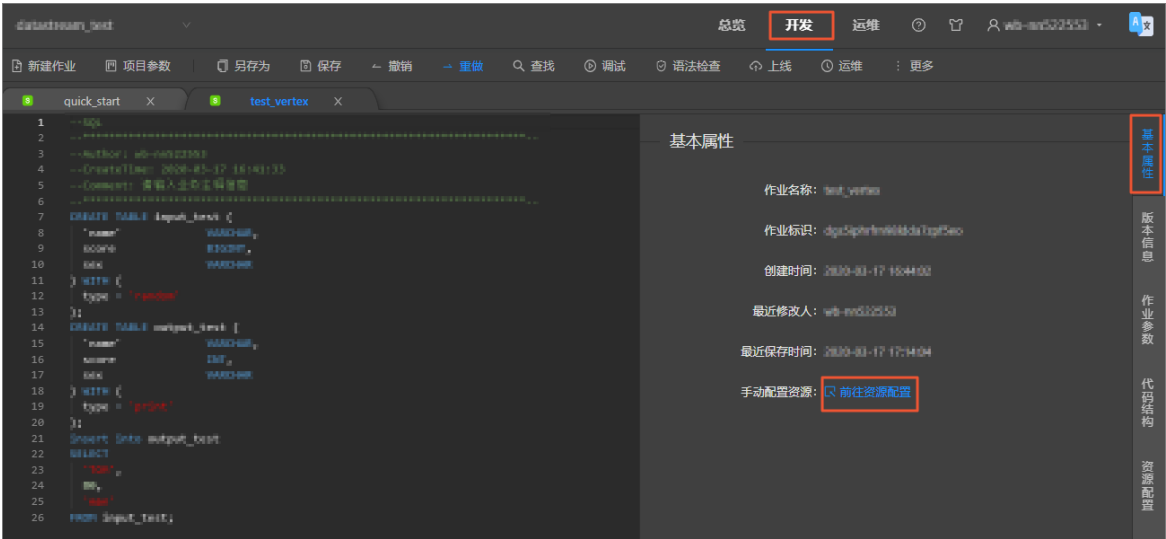
特殊字符	字符的转义方式
'	\'
\	\\
.	\\.
\$	\\\$

如何重新获取配置资源？

对作业的配置进行更新前，您需要进行重新获得获取配置资源的操作。操作步骤如下：

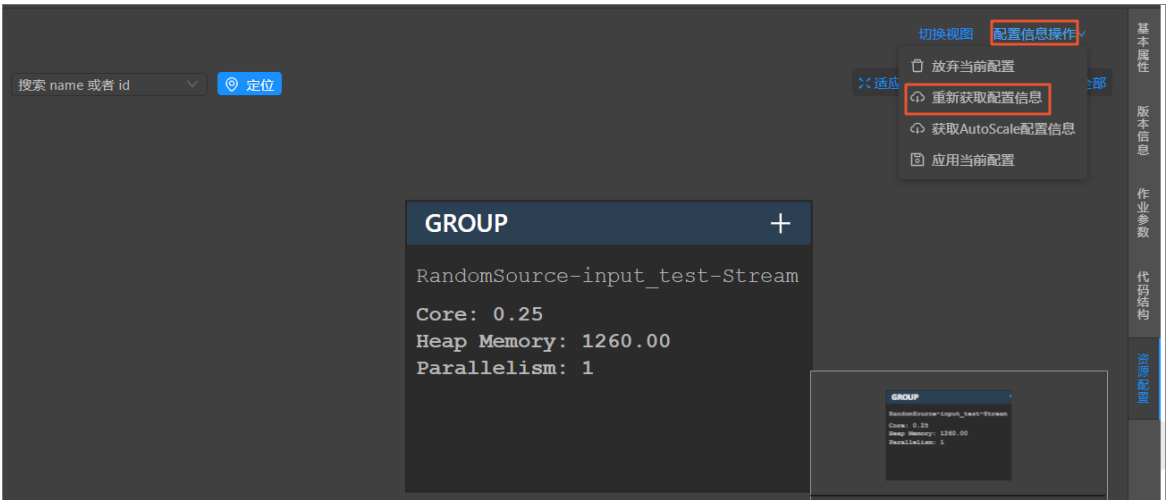
- 1. 登录[实时计算控制台](#)。

- 2. 单击页面底部的开发。
- 3. 在作业开发页面，单击基本属性 > 前往资源配置。



❓ 说明 如果在操作中出现报错，请参见[重新获取配置资源报错，该如何处理？](#)。

- 4. 单击配置信息操作 > 重新获取配置信息。



如何使用系统时间进行分组聚合计算？

如果您有使用系统时间做分组聚合计算的需求，建议您使用GROUP BY DATE_FORMAT(TO_TIMESTAMP(NOW()*1000), 'yyyyMMdd') as `day` 的方式来代替GROUP BY CURRENT_DATE。

因为，如果使用GROUP BY CURRENT_DATE方式进行分组聚合，运行结果可能不符合您的预期。

重启作业时，是否能保证数据不丢失且不重复？

- 如果您是手动停止作业后，再重启作业，您可以在重启设置启动位点时，调整启动时间到停止时间之前，并建议您使用带主键唯一特性的数据库做结果表，即使上游存储有部分重复消费的数据，最终结果也能接近准确。因为实时计算独享和共享模式本身不支持Savepoint，所以手动停止并重启的位点无法做到非常精准。
- 如果系统出现故障后重启作业，您无需担心数据丢失问题。因为实时计算存在容错机制。容错机制的核心是持续创建分布式数据流及其状态的快照。当系统出现故障时，这些快照充当可以回退的一致性检查点

（Checkpoint）。作业Failover会从Failover前一次成功的Checkpoint位点恢复。

如何查看独享集群白名单和VPC信息？

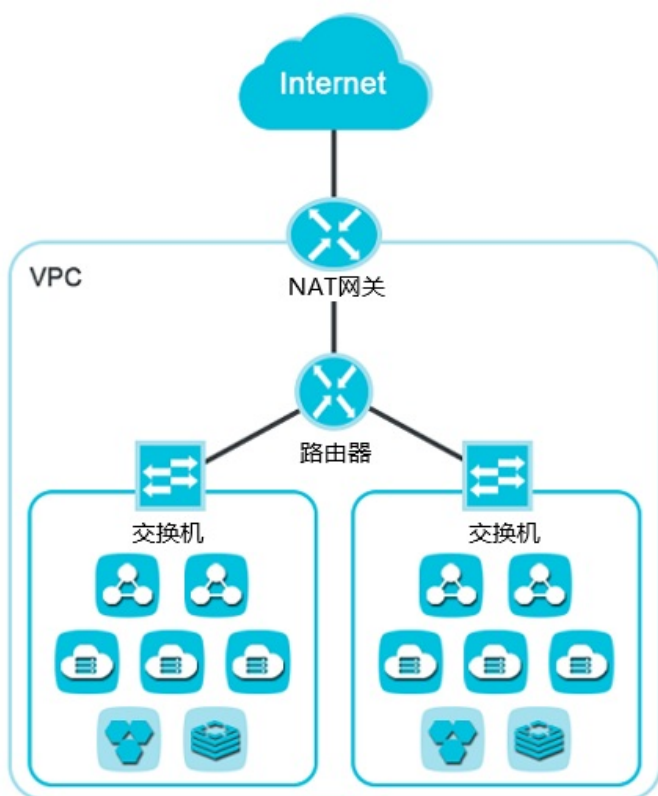
独享集群引用的外部存储必须和实时计算处于相同的VPC。如果引用的数据存储（例如RDS）需要配置白名单，则需要将ENI中的IP添加到外部数据存储的白名单中。

1. 登录[实时计算控制台](#)。
2. 将鼠标悬停至页面右上角账号信息位置，选择项目管理。
3. 单击左侧导航栏中**集群列表**。
4. 在**集群列表**页面，**名称**下单击目标集群名称。
5. 在集群信息窗口查看白名单和VPC信息。
 - **ENI**：引用的外部存储白名单中需要添加的IP地址。
 - **VPC**：集群VPC ID。

独享集群如何访问公网？

● 背景说明

独享集群默认不具备访问公网的能力，但阿里云提供的NAT网关可实现VPC网络与公网网络互通，以满足部分独享集群用户通过UDX或Datastream代码访问公网的需求。



● 解决方案

通过在VPC中创建NAT网关，并创建SNAT条目（如下图所示）将独享集群所在的交换机绑定至弹性公网IP（EIP），即可通过EIP访问公网。具体配置方法请参见：

- [创建NAT网关](#)。
- [创建和管理SNAT条目](#)。

○ 绑定EIP。



如何访问跨VPC里的存储资源？

您可以通过以下几种方式跨VPC访问存储资源：

- 请您[提交工单](#)，产品名称选择VPC，要求通过高速通道或其它产品打通网络，但是此种方式需要付费。
- 退掉存储服务后，重新购买一个与实时计算相同VPC的存储资源服务。
- 释放Blink集群后，重新购买一个和存储服务相同VPC的实时计算服务。
- 开通实时计算的公网访问能力，通过公网访问存储服务。因为在延迟性方面，公网不如内网，所以不推荐使用此方式。独享集群默认不能访问公网，如有需求，请参见[独享集群如何访问公网](#)？

如何对网络连通性问题进行排查？

如果出现网络连通性问题，请先参见[网络探测](#)完成网络连通性的初步验证。本文为您介绍不同场景应该如何处理网络连通性问题：

- 网络连接失败
 - 未设置白名单

请参见[数据存储白名单配置](#)完成上下游数据存储设备的白名单设置。
 - 白名单设置不正确
 - 独享集群

检查相关数据存储和实时计算是否处于不同VPC。如果处于不同VPC，请使用[高速通道](#)等产品进行连通，或者重新购买与实时计算处于相同VPC的存储资源。
 - 共享集群

检查Endpoint填写是否正确。共享模式访问VPC下的存储需要先进行VPC授权，详细步骤请参见[VPC访问授权](#)。

❗ 说明 Kafka的连接信息可以进行VPC授权，但是Meta数据不能进行授权，因此共享模式不能使用Kafka作为上下游数据存储。

- 端口连接失败

请参见[安全组概述](#)完成安全组设置。

● 域名连接失败

如果IP连接正常，但域名连接失败，请检测域名解析是否正确。可以使用[PrivateZone](#)产品对域名进行解析。

如何快速编写和测试SQL？

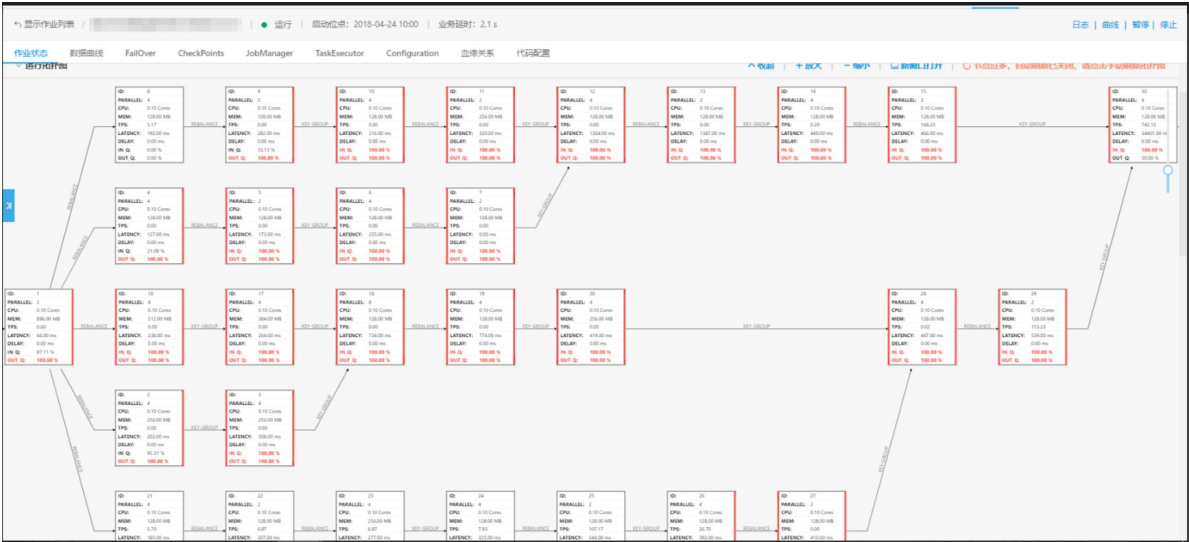
实时计算提供整套开发工具辅助您进行开发测试工作，推荐流程如下：

- 1. 在数据开发页面进行SQL开发，使用SQL编辑器的DDL生成、SQL智能提示、语法检测等工具实现快速开发。
- 2. 在数据开发的在线调试功能，进行模拟容器调试。该调试过程不会影响线上代码和数据上下游存储。
- 3. 将发布调试完成的代码，到生产运维试运行。经过真实业务和数据验证后，该SQL作业进入生产状态。
- 4. 在数据开发页面完成修改代码后，不会影响生产作业，需要您单击上线后，重新启动该作业，该修改才会生效。

如何优化实时计算各个节点资源配置以减少延迟？

● 问题描述

实时计算作业启动后，多个节点的延迟会迅速增加。示例如下图。



● 原因分析

- 资源不足
- 启动作业的时候选择了追历史数据，作业将会有一定的延时。

● 解决方案

- 资源不足：进行[AutoConf自动配置调优](#)或[手动配置调优](#)。
- 当启动作业的时，指定读取数据时间选择当前时间之前：历史数据追取完成后，延时将恢复正常。

4.分场景排错指引

本文汇总了实时计算Flink版服务和作业上线、启动、输出、延时等的常见问题。

购买

- 购买实时计算Flink版服务时，遇到Inventory check failed报错，应该如何处理？
- 控制台提示项目参数错误，项目不存在，应该如何处理？

上线启动

- 作业无法启动，应该如何处理？
- 作业上线后，作业的运维页面看不到拓扑图、瞬时值的指标或曲线指标，应该如何处理？
- 新作业提交后，项目中现存作业出现异常的原因是什么？

数据输出

- 为什么作业没有输出？
- 为什么使用滑动窗口没有数据输出？

作业延时

- 作业延时增加，应该如何处理？
- 作业延迟过大，应该如何处理？
- 业务延迟显示作业无效，应该如何处理？

问题定位思路

- 如何对实时计算Flink版3.0以上版本的作业进行反压检测？
- 如何通过调试模式（Debug）查看作业的输出信息？
- 如何处理调试过程、窗口或计算过程中的脏数据？
- 如何利用报错信息快速定位作业问题？
- 如何从taskmanager.log中查找脏数据等报错信息？
- Vertex排查数据量问题
- 重新获取配置资源报错，该如何处理？
- 在使用UDX函数时，引用JAR包内的本地文件报错
- GROUP BY数据出现热点、数据倾斜
- OpenAPI DEMO报错
- 云监控控制台页面配置监控告警时，显示无数据，应该如何处理？

购买实时计算Flink版服务时，遇到Inventory check failed报错，应该如何处理？

- 问题场景

在购买实时计算Flink版服务时，遇到Inventory check failed报错。



- 问题原因
 - 您不小心删除了阿里云实时计算Flink版在RAM的角色AliyunStreamDefaultRole。
 - 您不小心修改了AliyunStreamDefaultRole的授权策略且无法恢复。

- 解决方案

请您参见[共享模式角色授权](#)恢复AliyunStreamDefaultRole角色。

注意

- 删除AliyunStreamDefaultRole角色后，导致实时计算Flink版正在运行的作业瞬间无法读写上下游外部存储，直到后续重新初始化AliyunStreamDefaultRole角色后才可恢复正常。请务必确认这项操作对于线上实时计算Flink版业务是否会造成影响。
- RAM删除角色和实时计算Flink版重新初始化操作，均需要您的主账号或者经主账号授权的RAM账号来完成。

控制台提示项目参数错误，项目不存在，应该如何处理？

- 问题描述

在阿里云实时计算Flink版控制台出现错误提示窗口。



- 原因分析

系统下线了已经到期但是没有续费的项目。

- 解决方案

您需要对项目进行续费，详情请参见[手动续费](#)。

作业无法启动，应该如何处理？

您在完成作业开发上线以后，在运维页面无法启动作业。原因有以下几点：

- 作业中存在Failover

- 排错指引

查看Failover报错信息，分析Job运行异常原因，通常Failover有以下报错。

```
Slot allocation request timed out
```

- 解决方案

这个问题通常是由资源不足引起的，请先检查资源是否充足，不足请先扩容。

- 资源不足

- 排错指引

在运行信息中，查看是否有如下的报错提示。

```
Submit blink job failed, name: test_copy, errcode:30011, errmsg:code:[30011], brief info:
[error occur while run app], context info:[details:[资源不足，请手动停止任务提交，并联系管理员
扩容.(yarn resource not available, so here try to kill it to avoid long time waiting)java.
lang.Exception: shell cmd execute failed
```

- 解决方案

在已使用CU接近已购买CU的时，如果您的作业所需CU较大，会导致作业无法启动，需要您重新[购买资源](#)。

- 获取资源配置失败

- 排错指引

在运行信息中，查看是否有如下的报错提示。

```
errcode:30011, errmsg:code:[30011], brief info:[error occur while run app], context info:
[details:[任务提交失败，请检查详细日志输出。(submit app failed)java.lang.UnsupportedOperation
Exception: Cannot set chaining strategy on Union Transformation.
```

运行信息	数据曲线	FailOver	CheckPoints	JobManager	TaskExecutor	血缘关系	告警设置	属性参数
1								
2								run instance failed, 操作错误: 系统处理BLINK作业异常。
3								错误原因: Submit blink job failed, name: icbu_buyer_category_change_handler, errcode:30011, errmsg:code:[30011], brief info:[error occur while run app], context info:[details:[任务提交失败，请检查详细日志输出。(submit app failed)java.lang.UnsupportedOperationException: Cannot set chaining strategy on Union Transformation.
4								at org.apache.flink.streaming.api.transformations.PartitionTransformation.setChainingStrategy(PartitionTransformation.java:105)
5								at com.alibaba.blink.launcher.autoconf.SqlJobConfigurer.configureTransformations(SqlJobConfigurer.java:260)
6								at org.apache.flink.streaming.api.graph.StreamGraphProperties.configureTransformations(StreamGraphProperties.java:200)
7								at org.apache.flink.streaming.api.graph.StreamGraphGenerator.generateInternal(StreamGraphGenerator.java:193)
8								at org.apache.flink.streaming.api.graph.StreamGraphGenerator.generate(StreamGraphGenerator.java:157)
9								at org.apache.flink.streaming.api.environment.StreamExecutionEnvironment.getStreamGraph(StreamExecutionEnvironment.java:1567)
10								at org.apache.flink.streaming.api.environment.StreamContextEnvironment.execute(StreamContextEnvironment.java:54)
11								at org.apache.flink.table.api.StreamTableEnvironment.execute(StreamTableEnvironment.scala:133)
12								at com.alibaba.blink.launcher.JobLauncher.runStream(JobLauncher.java:345)
13								at com.alibaba.blink.launcher.JobLauncher.main(JobLauncher.java:159)
14								at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
15								at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
16								at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
17								at java.lang.reflect.Method.invoke(Method.java:498)
18								at org.apache.flink.client.program.PackagedProgram.callMainMethod(PackagedProgram.java:579)
19								at org.apache.flink.client.program.PackagedProgram.invokeInteractiveModeForExecution(PackagedProgram.java:459)
20								at org.apache.flink.client.program.PackagedProgramUtils.executeProgram(PackagedProgramUtils.java:96)
21								at org.apache.flink.client.cli.CliFrontend.runProgram(CliFrontend.java:261)
22								at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:219)
23								at org.apache.flink.client.cli.CliFrontend.parseParameters(CliFrontend.java:1222)
24								at org.apache.flink.client.cli.CliFrontend.lambda\$createAndRunClient\$12(CliFrontend.java:1299)
25								at java.security.AccessController.doPrivileged(Native Method)
26								at javax.security.auth.Subject.doAs(Subject.java:422)
27								at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1804)
28								at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41)
29								at org.apache.flink.client.cli.CliFrontend.createAndRunClient(CliFrontend.java:1299)
30								at org.apache.flink.client.cli.CliFrontend.main(CliFrontend.java:1318)
31								at com.alibaba.blink.restserver.util.Tools.runShellCMD(Tools.java:211)
32								at com.alibaba.blink.restserver.util.Tools.runShellString(Tools.java:158)
33								at com.alibaba.blink.restserver.service.impl.AppRunTool.runApp(AppRunTool.java:272)
34								at com.alibaba.blink.restserver.service.impl.AppServiceImpl.runAppDTO(AppServiceImpl.java:74)
								at com.alibaba.blink.restserver.controller.AppController.runAppDTO(AppController.java:38)

- 解决方案

重新获取资源配置。BlinkSQL任务开发完成后，需要单击资源配置，通过获取自动生成JSON配置来生成一份默认的配置文件。

- 无作业指标

- 问题现象

作业运维曲线上没有任何指标，作业一直处于Created状态，也没有Failover报错。

- 原因分析

JM没运行起来。

- 解决方案

先检查资源是否足够，如果资源足够，请您手动增加CU后，重启作业。

作业上线后，作业的运维页面看不到拓扑图、瞬时值的指标或曲线指标，应该如何处理？

1. 查看是否有Failover错误信息

解决方案：如果有Failover错误信息，请参见[运行Failover](#)。

2. 运维页面没有拓扑图和瞬时值的数据

解决方案：请您[提交工单](#)进行咨询。工单格式如下：

- 项目名
- 作业名称
- 什么时间提交的作业
- 公司名称
- 报错信息截图

3. 运维页面的曲线指标，全部指标或者是部分指标都没有信息

解决方案：

- i. 首先确认是否使用了自定义的Source，如果是自定义的Source，Source的输入曲线为空是正常的，因为实时计算Flink版无法监控到自定义的Source。
- ii. 如果没有使用自定义的插件，但曲线指标不存在，请您[提交工单](#)进行咨询，工单格式如下：
 - 项目名
 - 作业名称
 - 什么时间提交的作业
 - 公司名称
 - 报错信息截图

新作业提交后，项目中现存作业出现异常的原因是什么？

独享模式采用的是CGroup（Control Group）非严格资源隔离模式，在集群资源存在空闲时，单个作业（作业A）资源可以超用（即实际使用资源可超过申请资源，运维界面对应显示资源健康分小于等于20分）。如果新提交作业（作业B）申请的资源大于实际剩余资源，将要求超用资源的作业A退还部分资源。现存作业（作业A）的资源减少后可能会出现业务延迟增大等异常。

② 说明

- 仅独享模式存在此问题。
- 新作业提交运行后，项目使用率过高（超过95%以上）可能会出现以上问题。

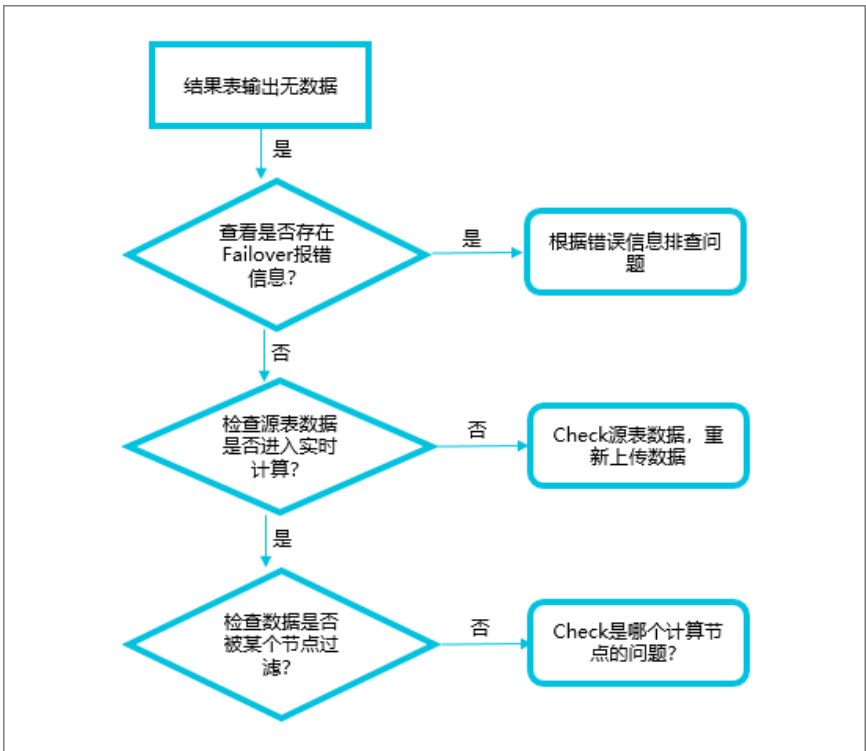
例如，1个项目的总资源为10CU，已经启动了两个作业，分别申请了2.5CU。由于资源有超用，其中一个作业A实际使用了3CU。总项目上来看，已使用CU（即实际申请CU）=5CU，但实际使用的资源为5.5CU。如果此时再提交运行1个申请5CU的作业，由于总资源为10CU，此时资源超用的作业A需要退还超用的0.5CU。因此，作业A实际使用的资源为2.5CU（与实际申请的一致），相比之前的实际使用的资源3CU有所减少，其处理能力也有所下降，可能会出现业务延迟增大的现象。

为什么作业没有输出？

• 场景描述

上线运行作业后，下游结果表中没有数据。

• 排错流程图



i. 检查作业中是否存在Failover

■ 排错指引

查看Failover报错信息，分析作业运行异常原因。

■ 解决方案

解决Failover问题，使作业正常运行。

ii. 检查源表数据是否进入实时计算Flink版

■ 排错指引

这种情况下没有Failover，但数据延时会很大，请查看数据曲线，检查各Source输入是否有数据。

- 解决方案

检查源表，保证上游有数据进入实时计算Flink版。

- iii. 检查数据是否被某个节点过滤

- 排错指引

- 在运行信息页面，查看每个节点的IN_Q（输入）和OUT_Q（输出）。如果输入有数据，输出为0，说明数据被这个节点过滤了。单击对应节点vertex顶部深蓝色区域后，再单击**Metrics**，查看数据输入和数据输出指标的详细信息。

- 在数据曲线页面，查看数据流入流出情况。

- a. 有数据流入：检查是否使用了数据存储的方式引入。

- b. 没有数据流出：可能是启动位点设置的有误。

- 解决方案

请参见[Vertex排查数据量问题](#)。

- Join过滤了数据。

- Window过滤了数据（检查Watermark的字段是否有效）。

- Where条件过滤了数据。

- iv. 检查下游是否由于默认缓存机制缓存了数据

解决方案：排除JOB的业务逻辑异常后，调整下游存储的batchsize的大小，设置为1。

 **说明** 该参数可能会造成下游数据库I/O压力过大、存在性能瓶颈的风险。

```
with(  
  type = 'DATAHUB',  
  endpoint = 'XXXX',  
  batchsize = '1'  
)
```

- v. 检查下游RDS，是否存在死锁

解决方案：请参见[写MySQL（TDDL/RDS）时，出现死锁（DeadLock）](#)。

另外，您可以使用调试模式（Debug），将计算结果打印到日志中，对日志进行分析，判断无输出结果的原因。详情请参见[如何通过调试模式（Debug）查看作业的输出信息？](#)

为什么使用滑动窗口没有数据输出？

- 问题描述：

使用[滑动窗口](#)没有数据输出。

- 解决方案

- 检查源表Watermark的时间字段是不是TIMESTAMP类型，如果不是，请您使用[计算列](#)进行转换。

 **说明**

- a. 请参见[概述](#)的时间类型中关于Watermark的说明了解更多。

- b. 请参见[数据源表概述](#)数据列章节了解如何进行转换。

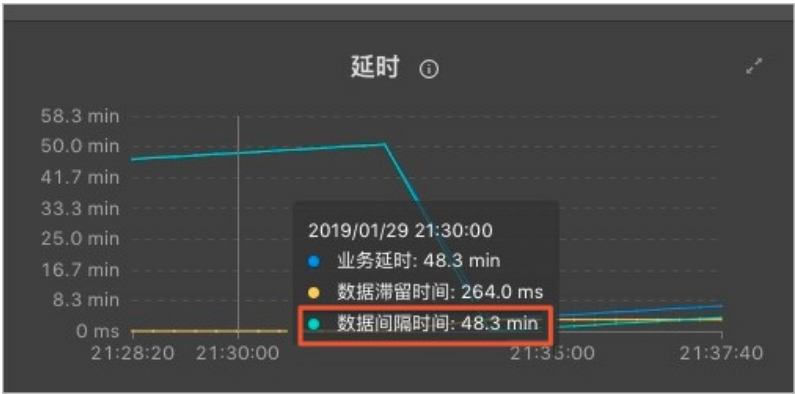
- 检查数据源表的并发是不是都有数据，只要有一个并发没有数据都不会触发Watermark形成窗口输出。如下图所示。

运行信息 数据曲线 FailOver CheckPoints JobManager TaskExecutor 血缘关系 属性参数								
返回运行拓扑 / Execution Vertex [0]								
Vertex Topology		SubTask List		Metrics Graph		Metrics Data		Accumulators
ID	操作	InQueue	OutQueue	RecvCnt	SendCnt	TPS	Retries	Status
2	跳转TaskExecutor日志	0 (0.00%)	0 (0.00%)	0	0	0	0	RUNNING
1	跳转TaskExecutor日志	0 (0.00%)	4 (100.00%)	0	78209	0.75	0	RUNNING
0	跳转TaskExecutor日志	0 (0.00%)	3 (50.00%)	0	79062	0.90	0	RUNNING

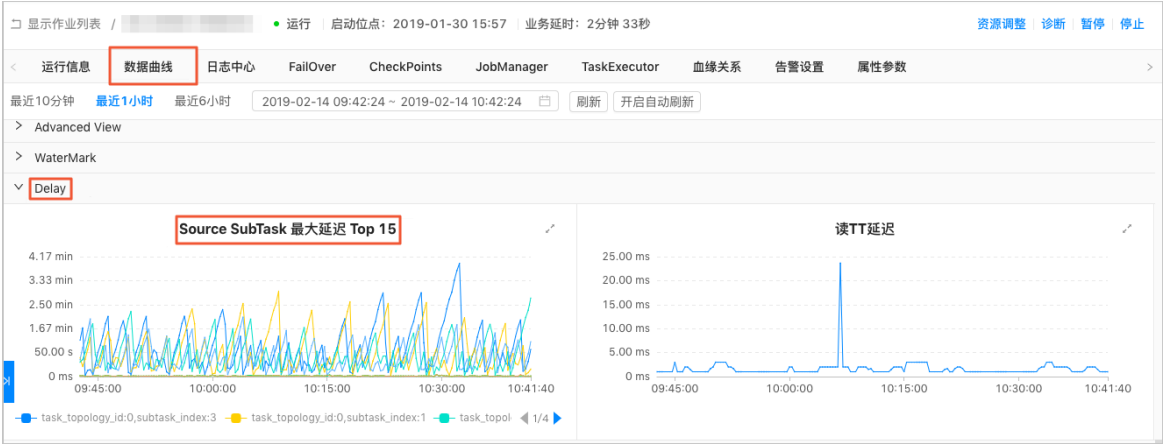
作业延时增加，应该如何处理？

- 排查指引
 - 数据间隔时间较大

数据间隔时间： $\text{no_data_delay} = \text{当前系统时间} - \text{最后一条数据到达实时计算Flink版的时间}$ ，即数据源中的数据到达实时计算Flink版的时间间隔。



数据间隔延时过大表明上游可能没有数据进入实时计算Flink版系统。实时计算Flink版数据间隔时间等于所有并发中的最大的延迟。您可以参见运维页面数据曲线 > Delay中的Source SubTask 最大延迟 Top 15曲线分析源表各并发的延迟。



- 作业性能不佳
 - SQL语法或者资源配置的不合理可能导致作业性能不佳，从而造成数据处理的滞后和作业延时的增加。

- 处理方法
 - 检查作业链路，确保上游所有Shard（Queue）都有最新的数据流入实时计算Flink版。
 - 如果在数据间隔时间较小的情况下，仍然存在作业延时增加的现象，则可能是作业性能不佳所导致。处理方法参见[高性能Flink SQL优化技巧](#)。SQL优化后，建议您先进行3~5次[自动配置调优](#)，如果调优效果不明显，再进行[手动配置调优](#)。

❓ 说明

- 第一次进行自动配置调优时，请不要指定CU数。
- 建议您先优化SQL，再进行资源配置，优化效率会更高。

作业延迟过大，应该如何处理？

线上作业延迟过大，表现为曲线图中**业务延迟**呈水平趋势，或逐步呈上升趋势，或在追历史数据（启动位点调整到当前时间之前）的过程中，曲线图中**业务延迟**下降非常缓慢。常见的原因有以下几种：

- 源数据中存在脏数据
 - 排错指引
 - 在曲线图中，查看各Source的脏数据，脏数据会在相应时间节点上显示数据量，没有则显示为0。
 - 查看Failover或Taskmanager.log日志中的报错信息，脏数据会在错误信息中标明。
 - 解决方案

更改SQL，过滤脏数据后，重新上线作业。
- 没有源表数据进入实时计算Flink版
 - 排错指引

这种情况下没有Failover，请查看[数据曲线](#)，检查各Source输入是否有数据。
 - 解决方案

检查源表，确保上游有数据进入实时计算Flink版。详情请参见[数据曲线](#)业务延迟3条曲线。
- RDS写入瓶颈
 - 排错指引

在RDS写入过慢时，建议您调大并发或增加资源，观察输出是否仍然为0或RPS极小。
 - 解决方案

新建一个Schema相同的结果表，重新注册存储引入。
- 源数据存在问题
 - 排错指引

请确认是否存在以下情况：

 - 是否存在历史数据。
 - 数据周期性输入为0。
 - 输入数据不稳定，表现为输入曲线有抖动。
 - 解决方案
 - 历史数据处理完成后（例如，追历史数据完成后），延迟会下降。
 - 确认数据生产机制是否异常。
 - 数据输入波动大，检查Source源。

- 部分节点存在反压
 - 排错指引

详情请参见[如何对实时计算Flink版3.0以上版本的作业进行反压检测？](#)
 - 解决方案
 - 优化SQL。详情请参见[高性能Flink SQL优化技巧](#)。
 - 优化上下游WITH参数。详情请参见[上下游存储文档的WITH参数介绍](#)。
 - 优化作业参数。详情请参见[高性能Flink SQL优化技巧](#)。
 - 资源调优。[AutoScale自动配置调优](#)后，再进行[手动调优](#)。

业务延迟显示作业无效，应该如何处理？

业务延迟显示无效的原因包括如下几种，您可以根据不同的原因进行对应的处理：

- 作业运行失败，无法获取稳定的埋点信息。

根据Failover报错信息提示修复问题，使作业正常运行。
- 您没有对自定义源表或Datastream作业进行埋点，或者埋点方式和实时计算Flink版数据存储不兼容。

请参见[Metric](#)文档编写埋点代码，确保和实时计算Flink版数据存储相互兼容。
- 用户数据的延迟和实时计算Flink版获取埋点数据的延迟重合。

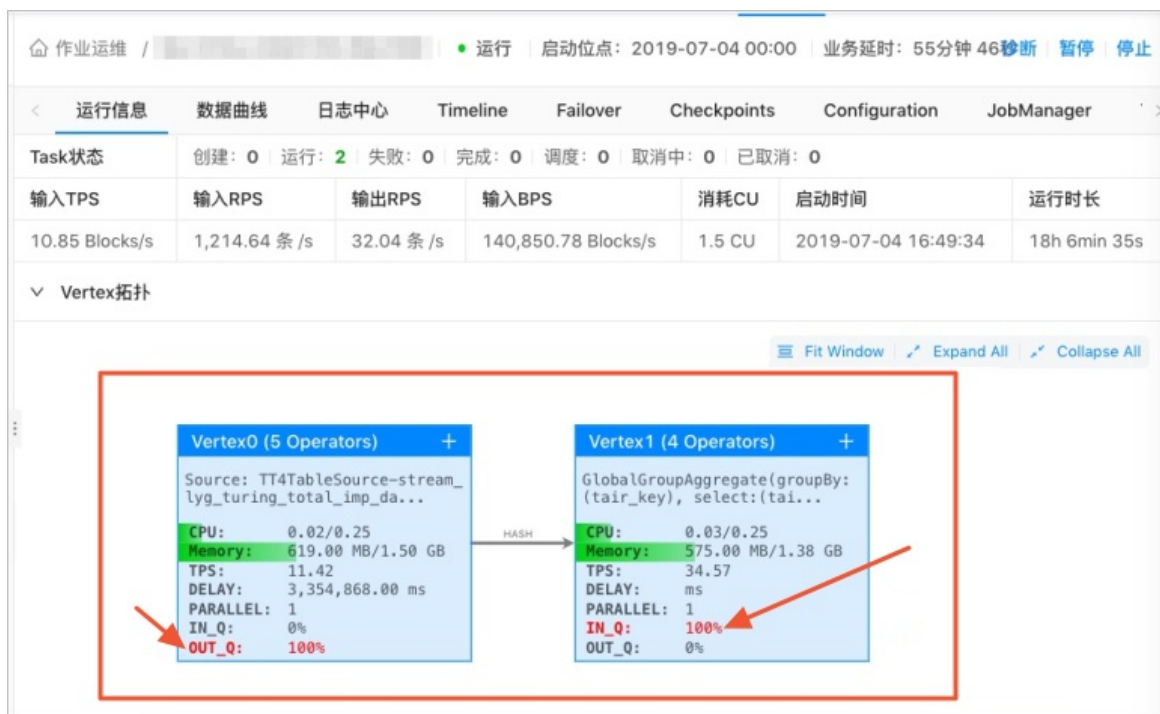
请您[提交工单](#)进行咨询。
- 多个作业显示无效。

可能是集群出现问题，请您[提交工单](#)进行咨询。

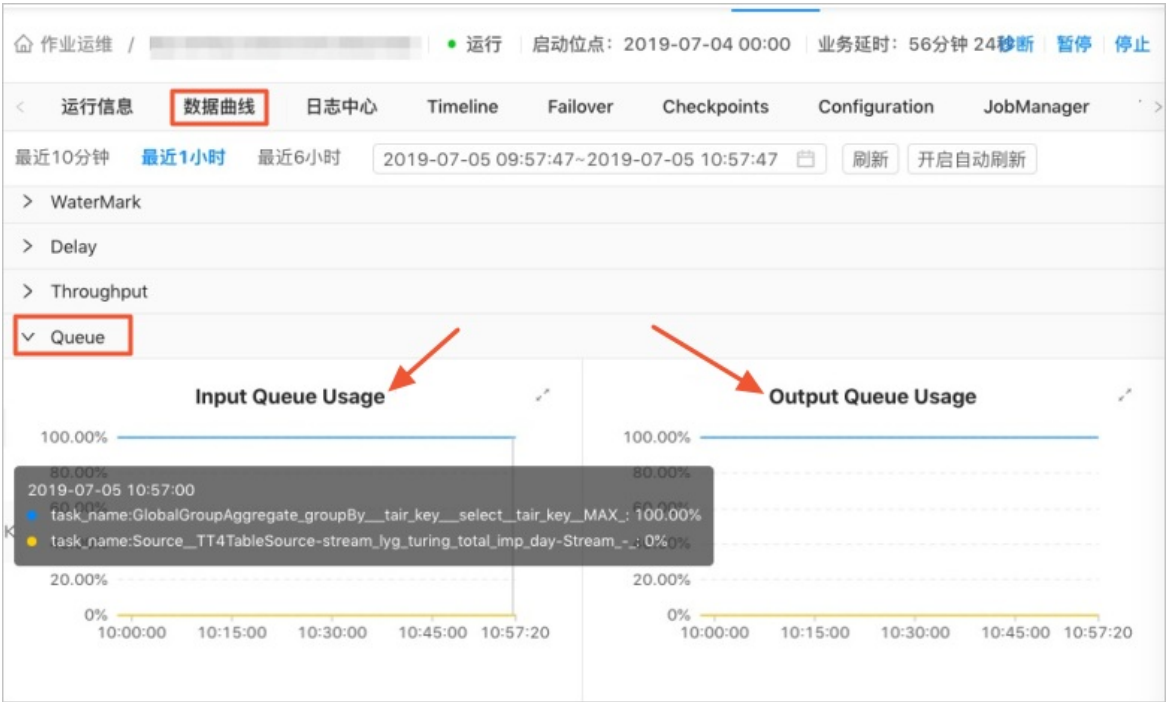
如何对实时计算Flink版3.0以上版本的作业进行反压检测？

反压检测是指当下游处理能力不足时，通知上游停止发送数据，从而避免数据丢失。实时计算Flink版3.0以上版本作业的反压检测步骤如下：

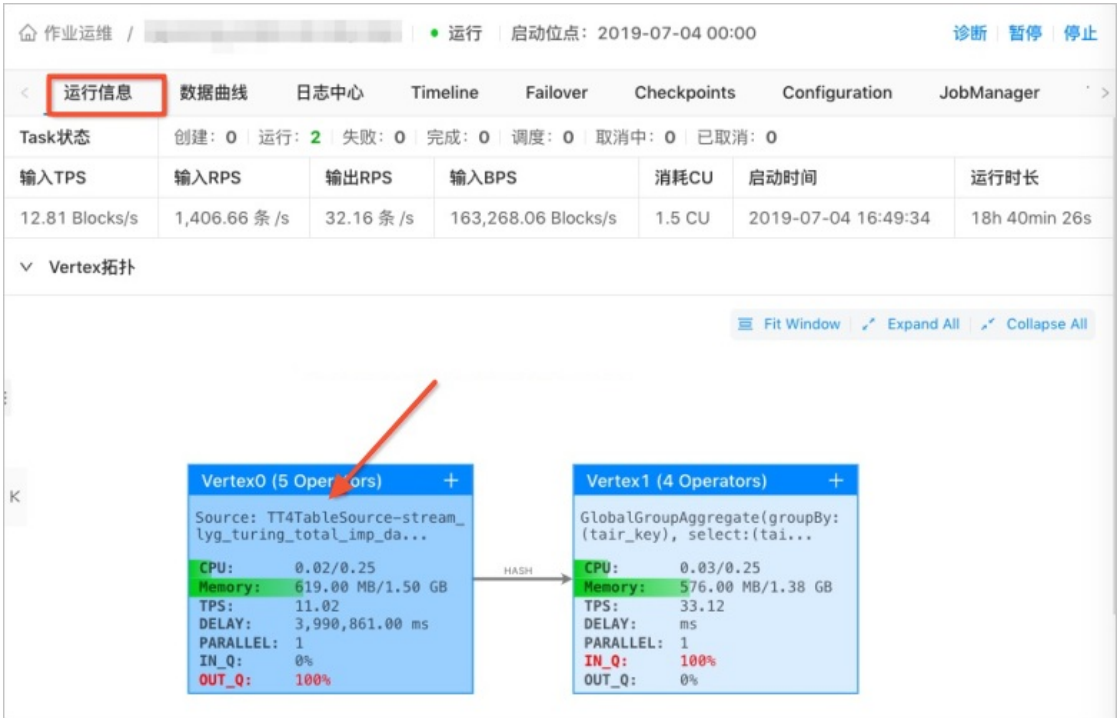
- 查看Vertex拓扑中的IN_Q与OUT_Q的参数值是否到达100%。



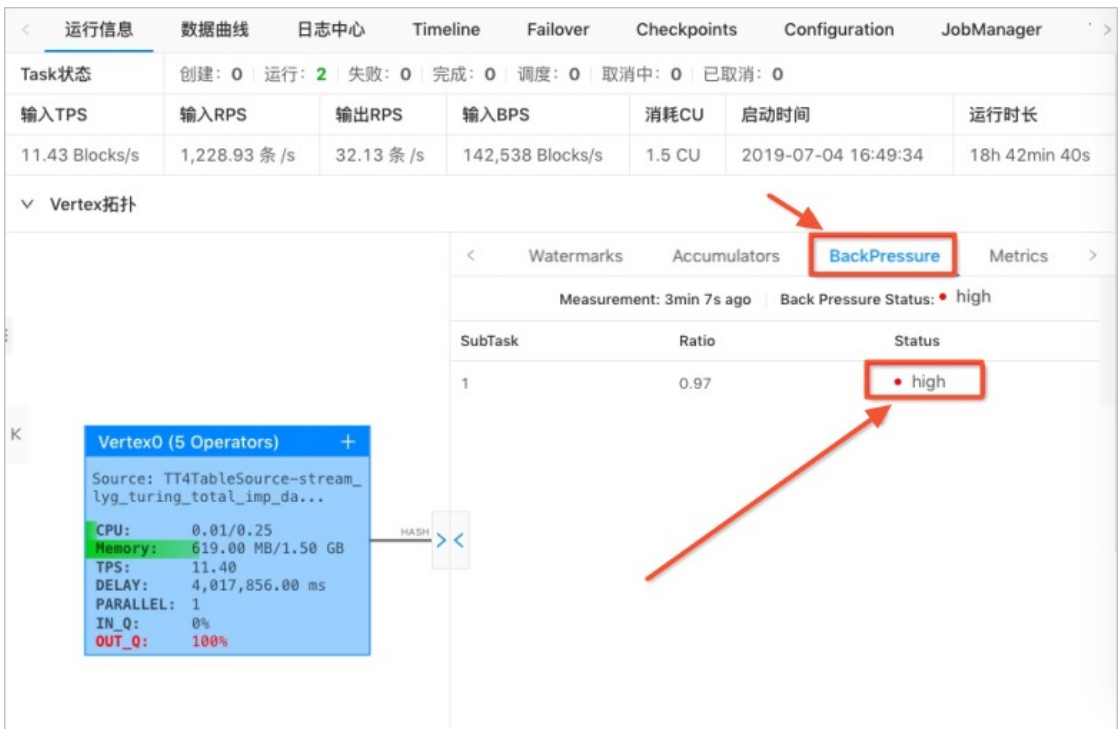
- IN_Q与OUT_Q值间歇性到达100%：作业不存在反压。
 - IN_Q与OUT_Q值长时间维持在100%：进行下一步检测。
2. 查看数据曲线 > Queue中的Input Queue Usage和Output Queue Usage是否达到100%。



- 未达到100%：作业不存在反压。
 - 达到100%：进行下一步检测。
3. 查看BackPressure的状态是否为high。
- i. 在运维 > 运行信息 > Vertex拓扑中，单击vertex顶部深蓝色区域。



ii. 在右侧Task信息区域，查看BackPressure参数。



- 红色high：该节点存在反压。
- 绿色ok：该节点不存在反压。

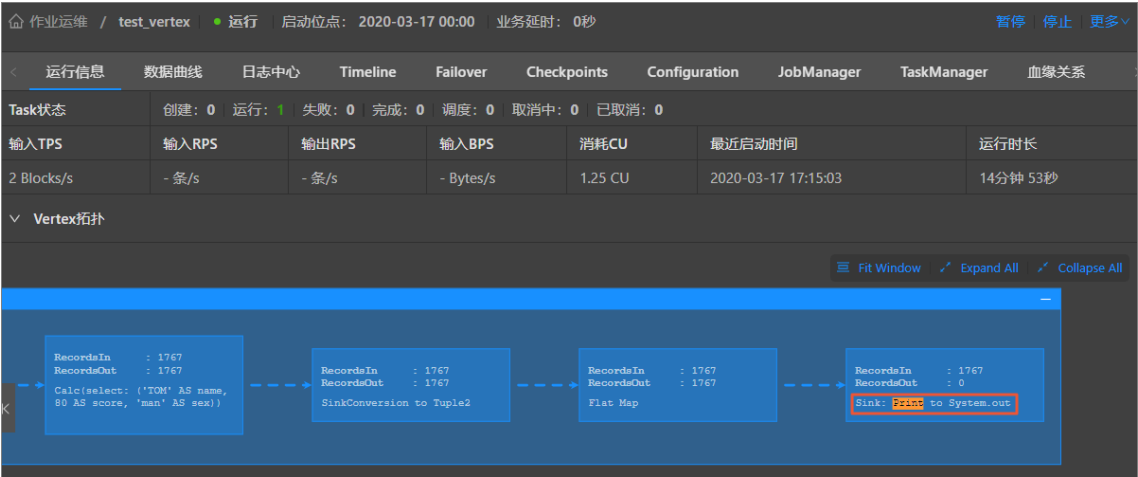
如何通过调试模式（Debug）查看作业的输出信息？

当结果表没有输出时，建议您使用调试模式（Debug），将计算结果打印到日志中，对日志进行分析，判断无输出结果的原因。例如存在脏数据等。根据作业是否包含UDX，日志查看的方法分为以下两种：

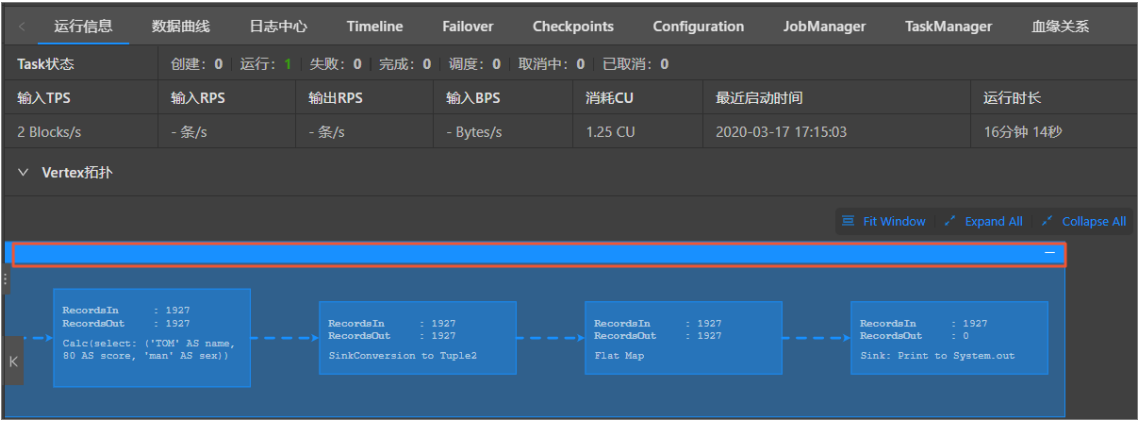
- 无UDX
 - i. 将结果表中的WITH参数改为 `type='print'` 。

```
CREATE table result_infor(  
  id bigint,  
  phoneNumber bigint,  
  name VARCHAR  
)with(  
  type='print'  
);
```

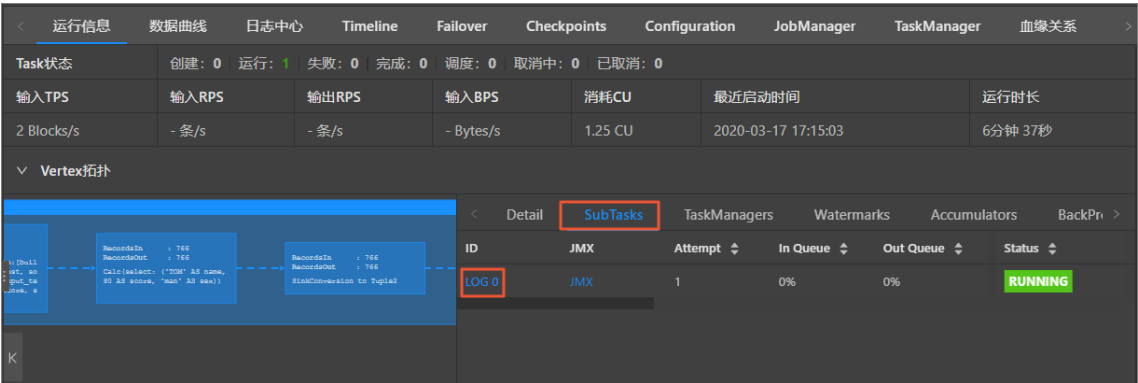
- ii. 重新上线并且启动作业。
- iii. 单击作业名称，进入运行信息页面。
- iv. 单击Expand all。
- v. 按住CTRL+F，输入print，找到Print节点所在的Vertex。



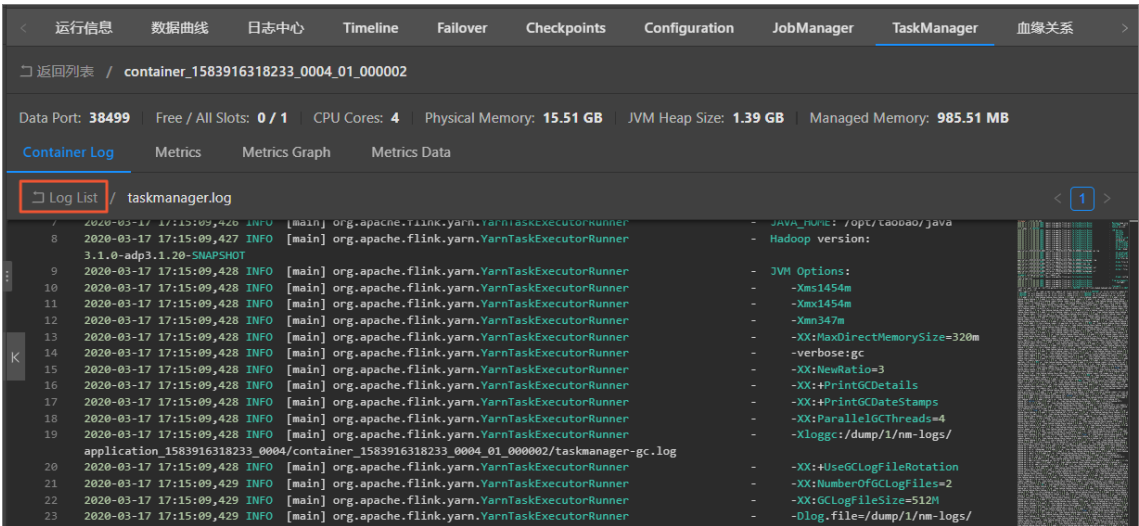
vi. 单击目标vertex顶部深蓝色区域。



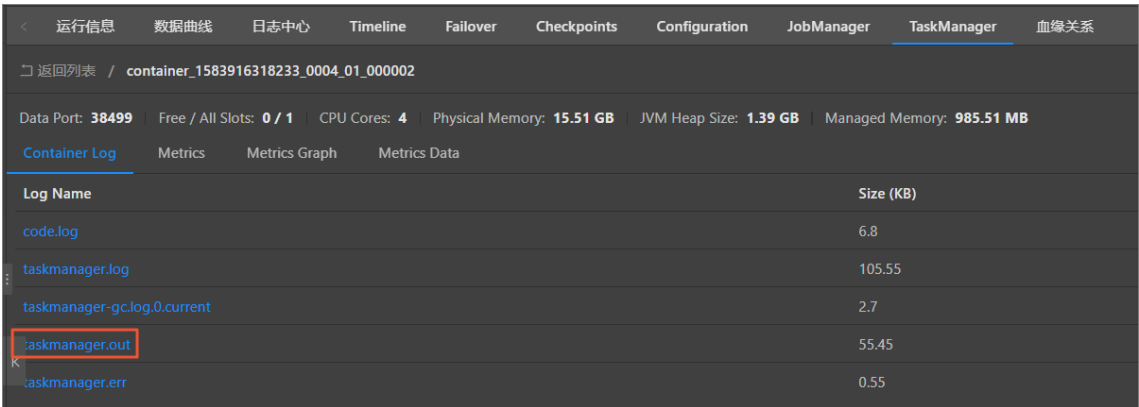
vii. 在SubTasks页签，单击LOG0，跳转至Container Log页面。



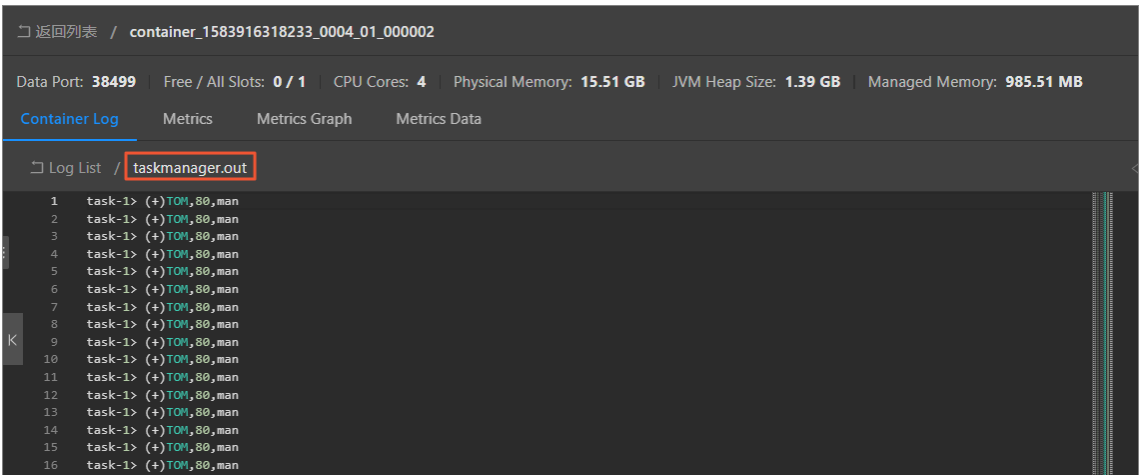
viii. 单击logList，返回日志列表。



ix. 单击taskmanager.out查看日志结果。



日志结果如下图所示。

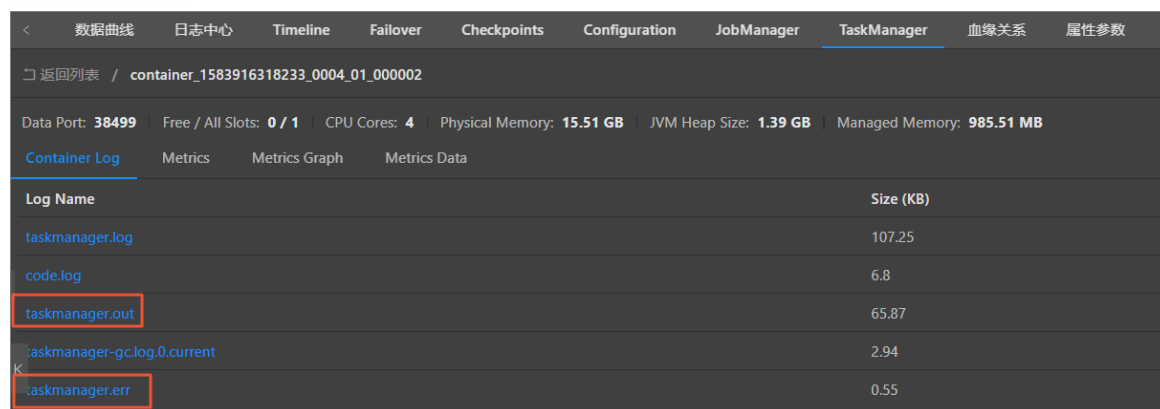


- 存在UDX

如果您使用UDX，在Java代码中又可以分为如下两种Debug方法：

- system out/err方法

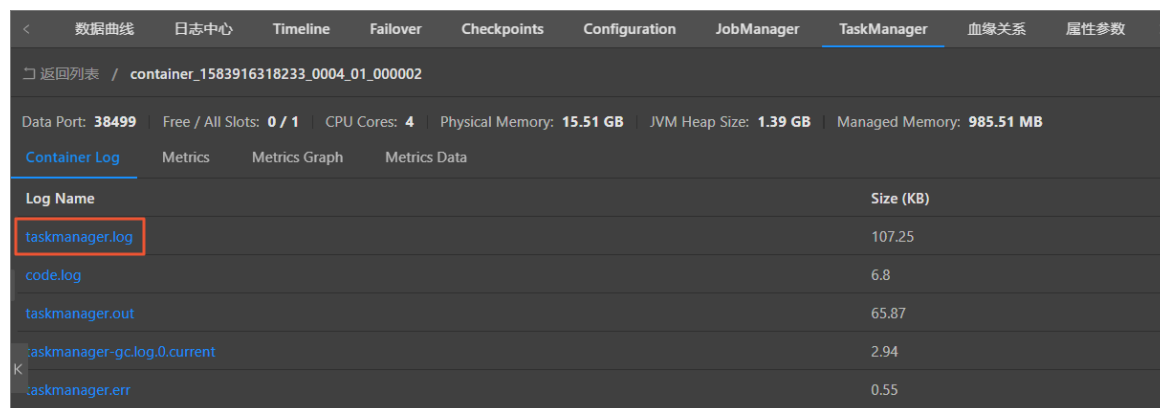
在Java代码中使用 `system.out` 或 `system.err` 方法，把调试的日志打印到 `taskmanager.out` 或 `taskmanager.err`。



Data Port: 38499 Free / All Slots: 0 / 1 CPU Cores: 4 Physical Memory: 15.51 GB JVM Heap Size: 1.39 GB Managed Memory: 985.51 MB	
Container Log Metrics Metrics Graph Metrics Data	
Log Name	Size (KB)
taskmanager.log	107.25
code.log	6.8
taskmanager.out	65.87
taskmanager-gc.log.0.current	2.94
taskmanager.err	0.55

- SLF4J的Logger方法

在Java代码中使用SLF4J的Logger功能，把调试的日志打印到 `taskmanager.log`。



Data Port: 38499 Free / All Slots: 0 / 1 CPU Cores: 4 Physical Memory: 15.51 GB JVM Heap Size: 1.39 GB Managed Memory: 985.51 MB	
Container Log Metrics Metrics Graph Metrics Data	
Log Name	Size (KB)
taskmanager.log	107.25
code.log	6.8
taskmanager.out	65.87
taskmanager-gc.log.0.current	2.94
taskmanager.err	0.55

如何处理调试过程、窗口或计算过程中的脏数据？

实际操作中您可能会遇到脏数据中断业务开展的情况，本文为您介绍在不同的场景中如何处理脏数据。

- 调试过程中的脏数据

- 错误详情

作业调试过程中出现报错信息：操作错误，所有的数据均被跳过，请检查DDL中的LENGTHCHECK项。

- 错误原因

单行字段条数检查策略LENGTHCHECK的默认值为NONE，即解析出的字段数小于定义字段数时，跳过这行数据。如果您的数据源存在脏数据，不符合单行字段条数检查的默认策略，则调试线上抽取数据时会报错，导致无法抽取到线上数据。

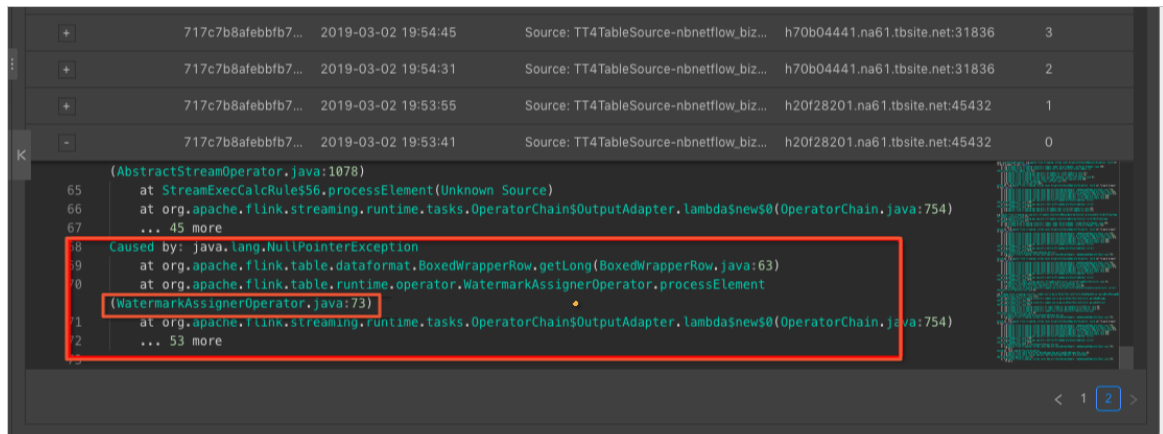
- 解决方案

建议在源表的WITH参数中增加lengthcheck='PAD'参数，即当解析出的字段数小于定义字段数时，系统会使用null在行尾填充缺少的字段。

- 窗口中的脏数据

◦ 错误详情

报错信息: Caused by: java.lang.NullPointerException



◦ 报错原因

WaterMark中存在NULL值导致NullPointerException报错。

◦ 解决方案

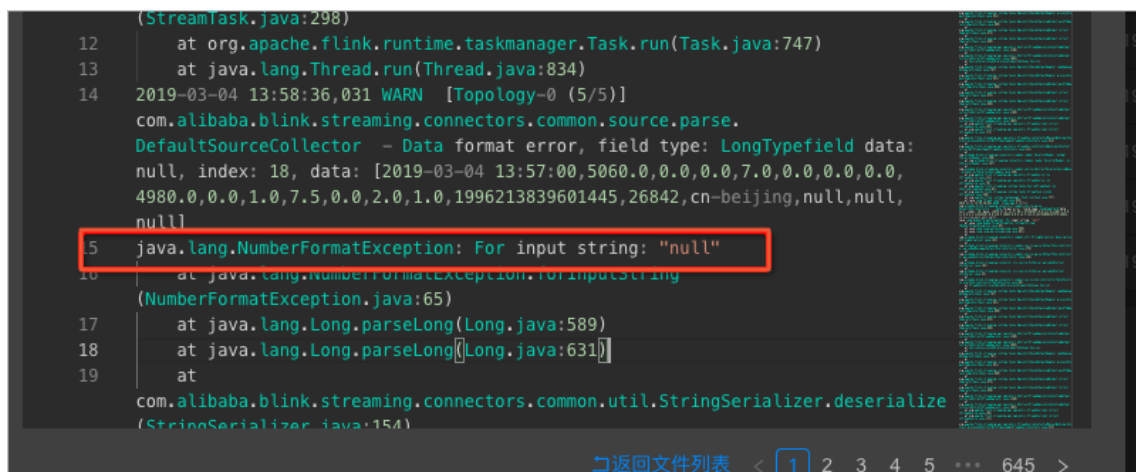
如果您使用Window函数，Event time的时间字段中存在NULL值，需要使用**计算列**进行过滤，示例如下。

```
ts as case when `datetime` is null
  then to_timestamp('1970-01-01 00:00:00')
  else `datetime` end
```

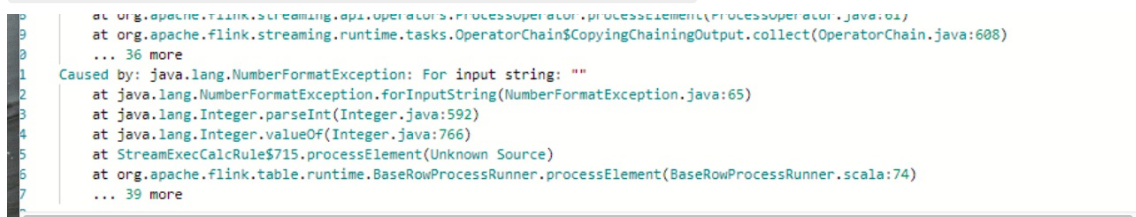
● 计算过程中的脏数据

◦ 错误详情

■ java.lang.NumberFormatException: For input string: "null"



■ java.lang.NumberFormatException: For input string: ""



- 报错原因

输入值是NULL字符串或空字符串，不能作为数字类型进行计算。

- 解决方案

确定脏数据后，通过WHERE或者CASE WHEN的方法对脏数据进行过滤。

 **说明** 实时计算Flink版2.2及以上版本支持在日志中查看脏数据，具体步骤请参见[如何从taskmanager.log中查找脏数据等报错信息？](#)

- 使用WHERE对脏数据进行过滤，示例如下。

```
select avg(test_id) from input
where test_id IS NOT NULL;
```

- 使用CASE WHEN对脏数据进行过滤，示例如下。

```
select avg(case when test_id IS Null
                then 0
                else test_id
              end as test_id)
from input;
```

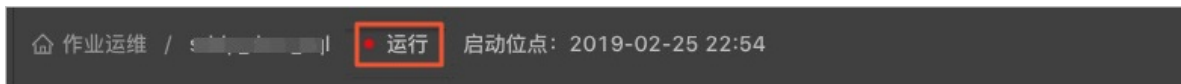
如何利用报错信息快速定位作业问题？

当作业状态运行前的标点显示为红色或者黄色时，表明作业的运行状态不正常。

- 黄色：作业正在调度。（可能因为资源不足导致作业运行失败。）

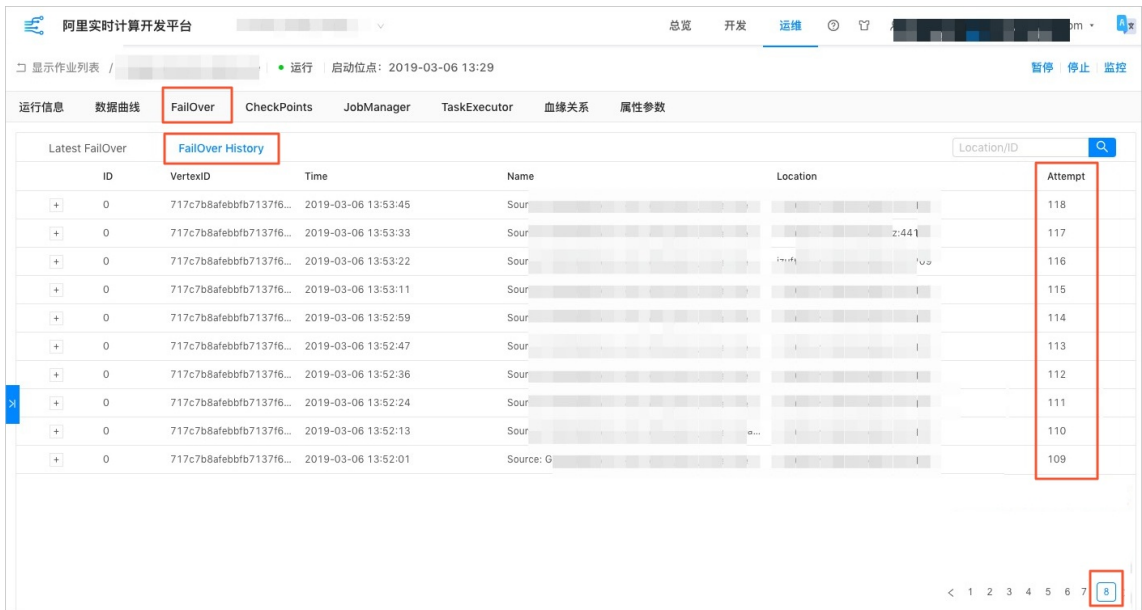


- 红色：作业存在错误。

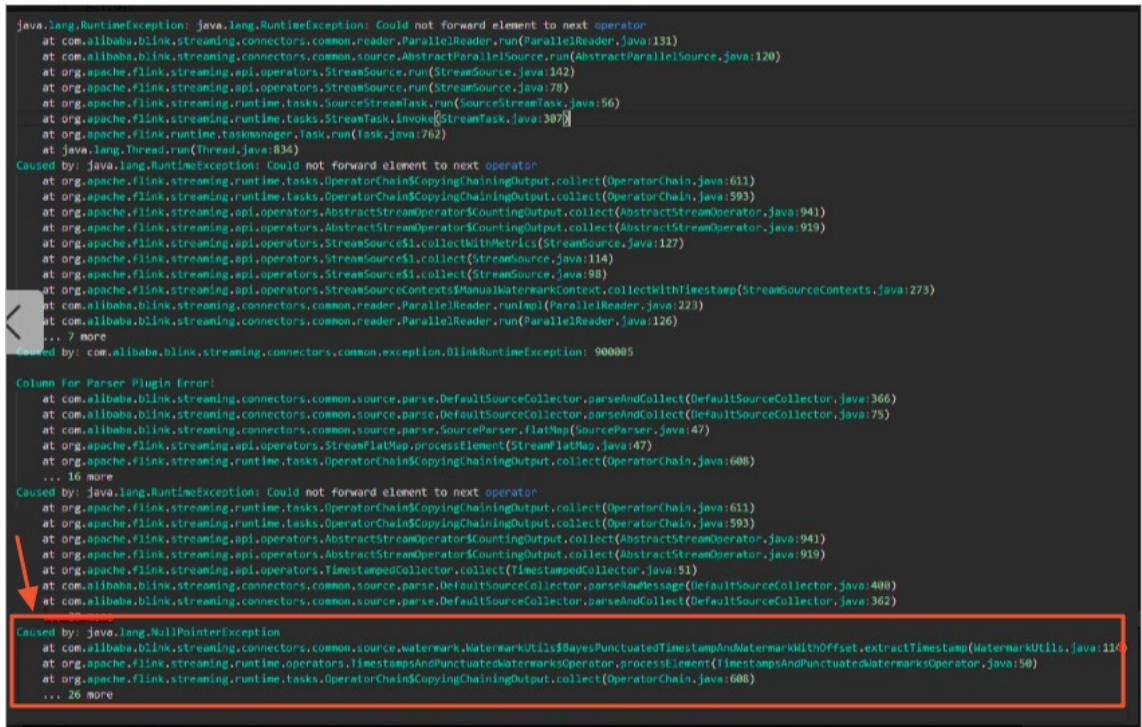


通过以下两种方法快速定位作业问题：

- 查看Attempt为0的Failover历史信息。
 - i. 在Failover > Failover History页面，查看Attempt为0的Failover历史信息。



ii. 查看Failover当前页面最后一个Caused by信息。找到最开始的Failover，并将页面拉至最下端，由下至上，找到Cause by信息。



② 说明 第1个Failover中的Cause by信息，往往是导致作业异常的根因，根据该根因的提示信息，可以快速定位作业异常的原因。

● 无法查看Attempt为0的Failover信息：

i. 在数据曲线页面，通过最近6小时、最近1天或最近1周维度，查看Failover曲线中第一次Failover的时间点。

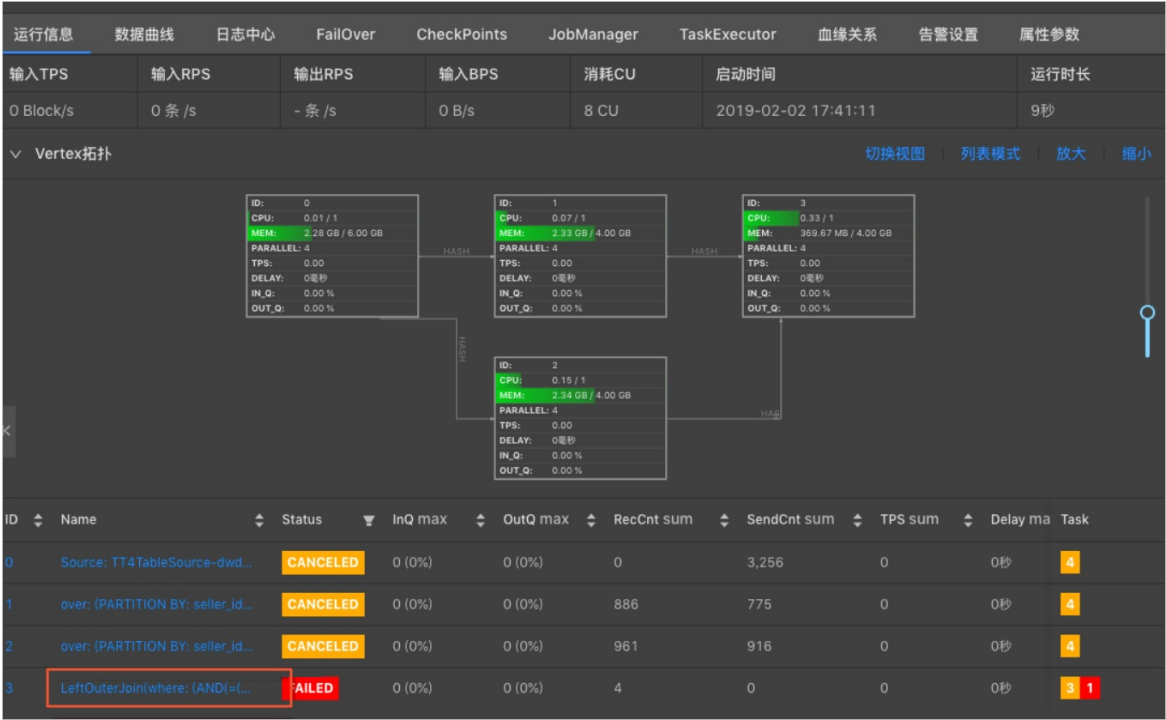


- ii. 在taskmanager.log日志中查看报错的原因。报错原因查询方法参见[如何从taskmanager.log中查找脏数据等报错信息？](#)

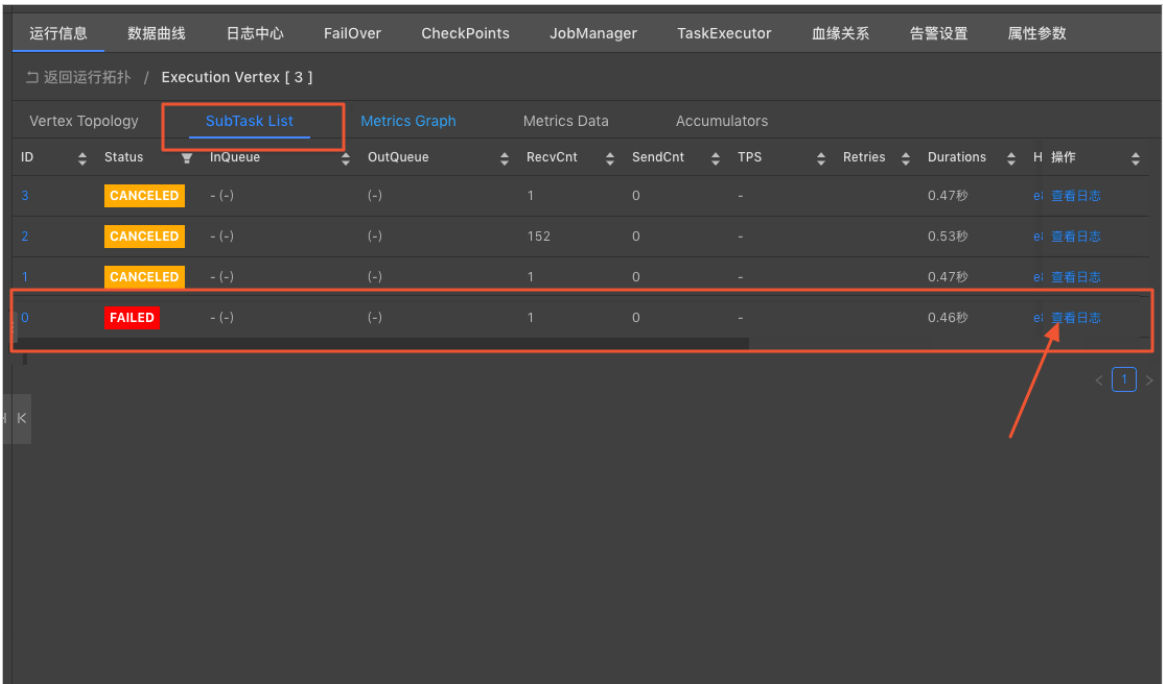
如何从taskmanager.log中查找脏数据等报错信息？

作业运行所产生的信息会记录到 taskmanager.log 日志文件中，如果您的作业出现运行失败的状态，您可以在 taskmanager.log 日志文件中查看可能的报错原因，例如脏数据。在 taskmanager.log 日志文件中查看报错信息的步骤如下：

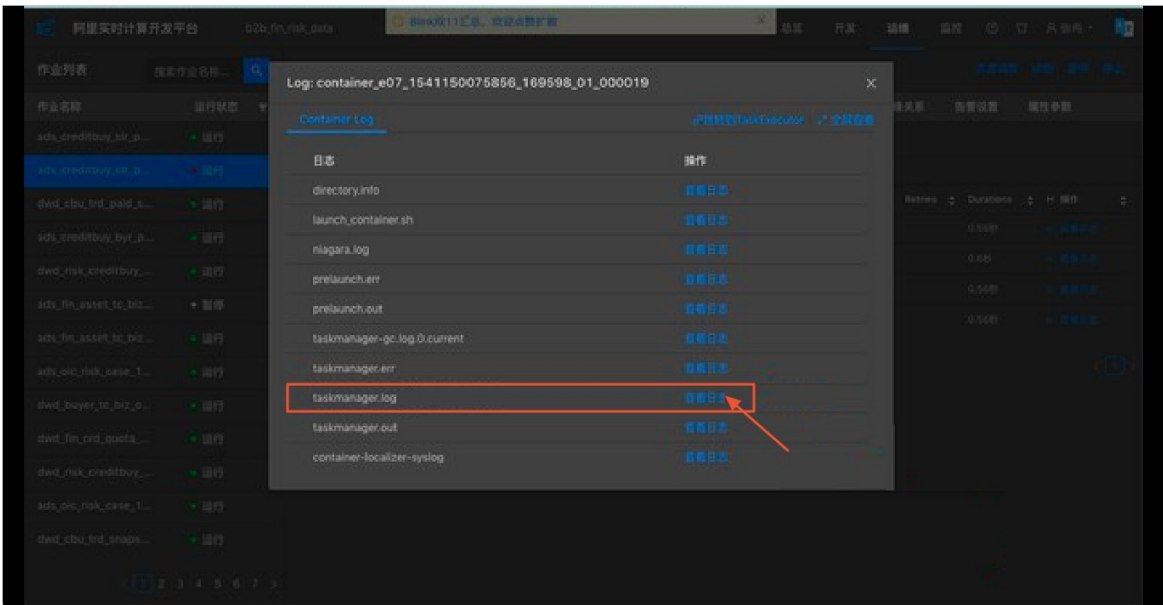
1. 单击Status为FAILED的节点或者Source节点。



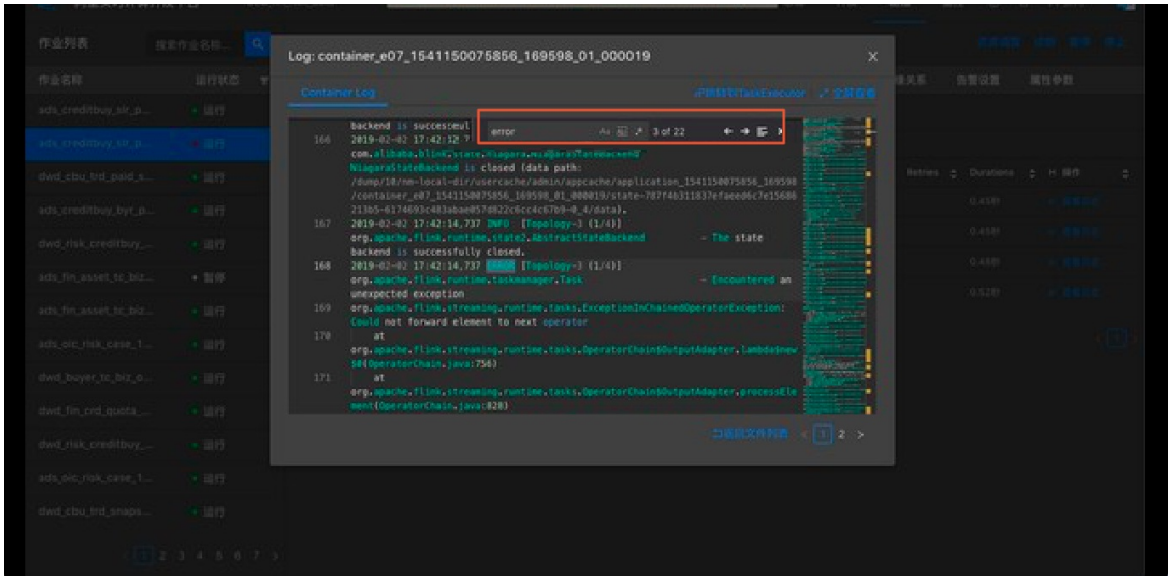
2. 在运行信息 > SubTask List 中，单击Status为FAILED的Subtask操作列下的查看日志。



3. 单击taskmanager.log日志操作列下的查看日志。



4. 按下 `ctrl+F` 搜索报错关键字（例如error），并查看报错信息。



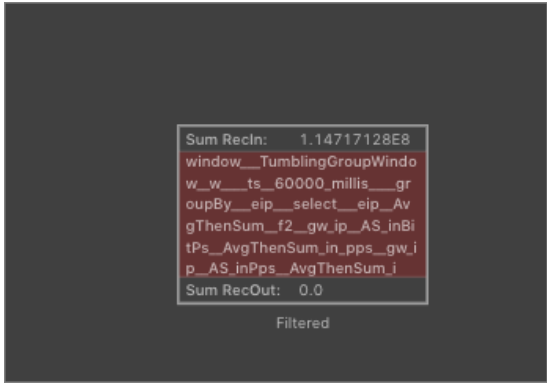
说明 建议从最近的时间开始查看，最近的有效报错信息通常是导致作业运行失败的根本原因。

Vertex排查数据量问题

● 现象分析
数据经过JOIN、WHERE或WINDOW等节点，数据量变少为正常现象，因为数据可能因为条件限制被过滤或JOIN不上。当某个Operator为红色时，代表数据被该节点过滤。



- 过滤场景
 - Window节点过滤

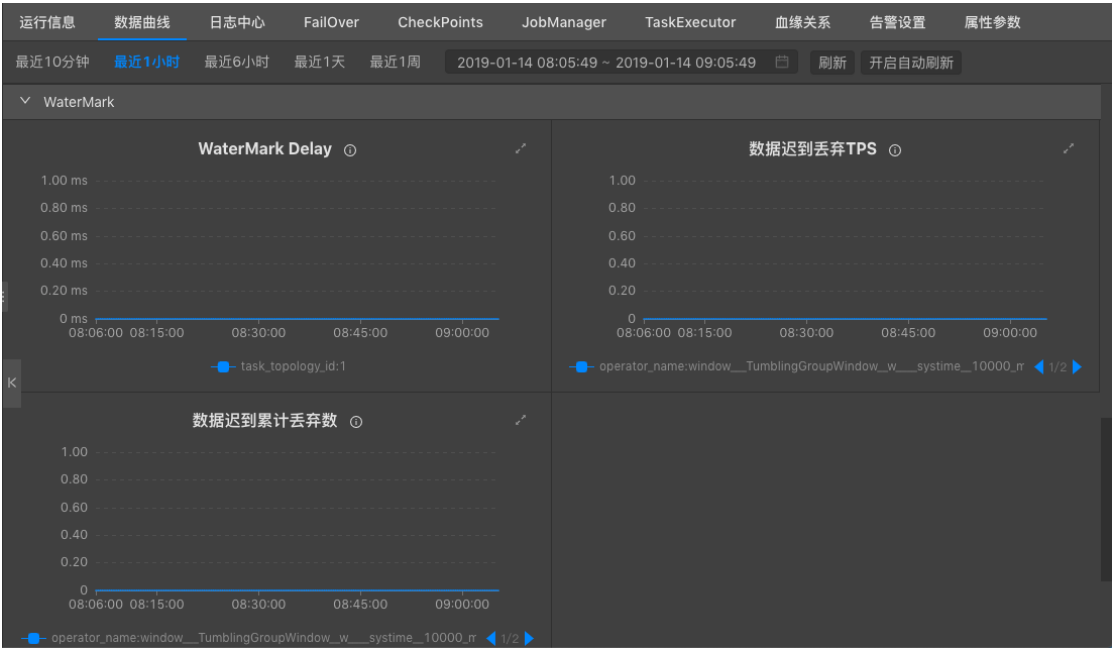


排查方案：

- a. 检查开窗的字段ts是否正常：
- 如果您的开窗时间是1970年（时间戳为10位）、15XXX年（时间戳为16位）等非13位时间戳的开窗时间，则需要使用[计算列](#)转换为13位时间戳后，再进行开窗操作。
 - 如果您不确定您的ts字段是否正常，建议您写一个虚拟的Sink，将ts字段打印到日志中，详情请参见[如何通过调试模式（Debug）查看作业的输出信息？](#)

```
CREATE TABLE test_output (  
    ts                timestamp  
) WITH (  
    type = 'print'  
);  
INSERT into test_output  
SELECT  
    ts  
FROM 源表;
```

b. 查看数据曲线中有关WaterMark的三个曲线。详情请参见数据曲线。



2.x以下的版本：查看数据源表的并发是否都有数据，只要一个并发没有数据，就不会触发WaterMark形成窗口输出。

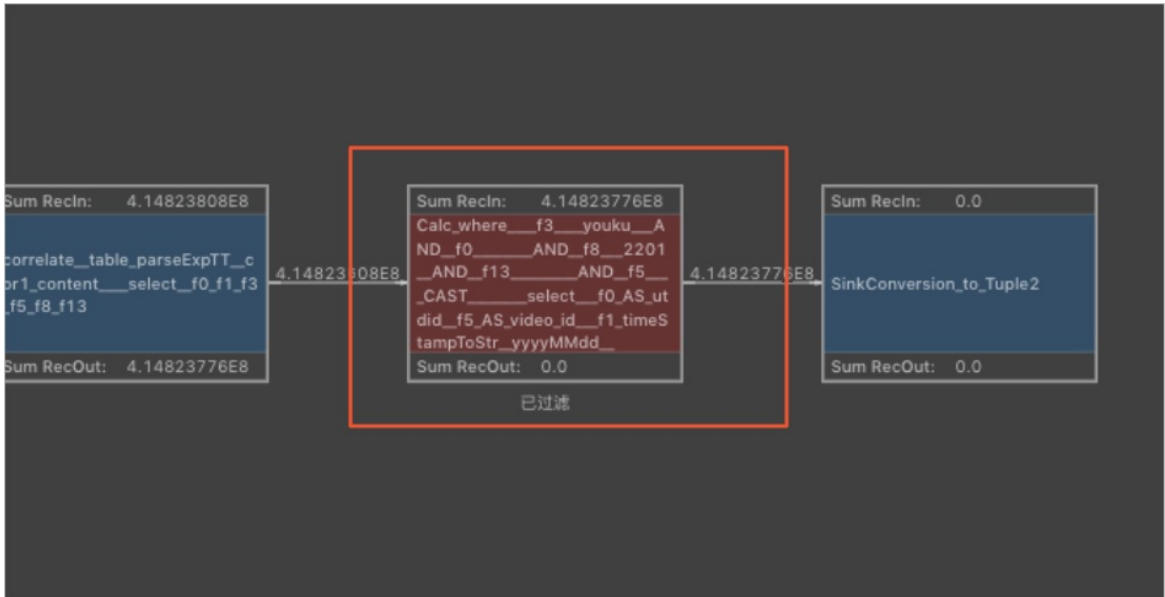
运行信息	数据曲线	FailOver	CheckPoints	JobManager	TaskExecutor	血缘关系	告警设置	属性参数
最近10分钟 最近1小时 最近6小时 最近1天 最近1周 2019-01-14 08:05:49 ~ 2019-01-14 09:05:49 刷新 开启自动刷新								
WaterMark								
WaterMark Delay								
数据迟到丢弃TPS								
数据迟到累计丢弃数								

返回运行拓扑 / Execution Vertex [0]								
Vertex Topology		SubTask List	Metrics Graph		Metrics Data		Accumulators	
ID	操作	InQueue	OutQueue	RecvCnt	SendCnt	TPS	Retries	Status
2	跳转TaskExecutor日志	0 (0.00%)	0 (0.00%)	0	0	0	0	RUNNING
1	跳转TaskExecutor日志	0 (0.00%)	4 (100.00%)	0	78209	0.75	0	RUNNING
0	跳转TaskExecutor日志	0 (0.00%)	3 (50.00%)	0	79062	0.90	0	RUNNING

说明 假设，窗口设置为1分钟的滚动窗口，数据被窗口过滤。检查下一个窗口是否有数据到来，如果下个窗口没有数据到来，则不会触发上一个窗口的结束，因此窗口没有输出。

如果排查后，发现没有问题，请您提交工单进行咨询。

◦ Where节点过滤



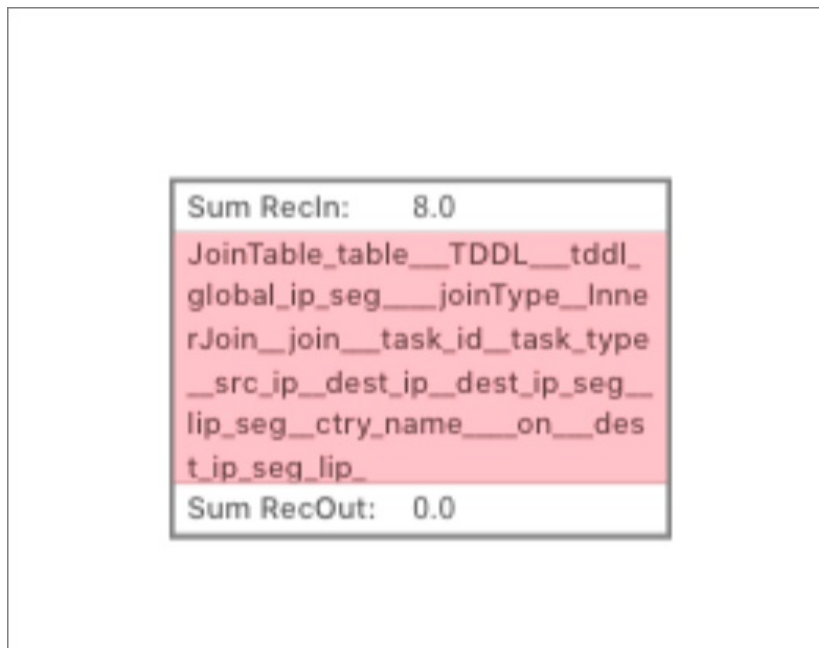
排查方案

- a. 将Where条件注释掉后，再将没有Where条件的数据打印到日志中，检查是否有数据满足Where条件。详情请参见[如何通过调试模式（Debug）查看作业的输出信息？](#)
- b. 当有符合Where条件的数据时，请检查Where条件的写法。例如，在Where条件中使用了to_timestamp、date_format等函数时，您可以将Where条件中使用的函数打印到虚拟Sink中，检查函数使用的方法是否正确。

```
CREATE TABLE test_output (  
    t1          varchar  
) WITH (  
    type = 'print'  
);  
INSERT into test_output  
SELECT  
    DATE_FORMAT(datetime1, 'yyMMdd')  
FROM 源表;
```

如果您排查后，发现没有问题，请您[提交工单](#)进行咨询。

- JOIN节点过滤

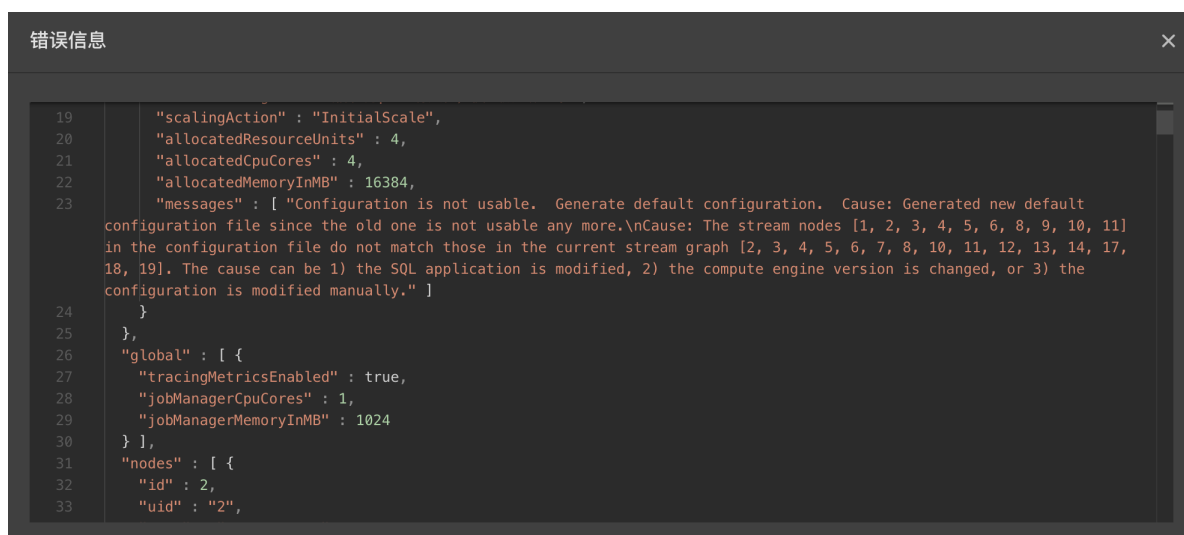


排查方案：检查ON条件，如果您不确定ON条件是否有问题，您可以将左右两个表的数据打印到日志中检查。详情请参见[如何通过调试模式（Debug）查看作业的输出信息？](#)

如果ON条件没有问题，请您[提交工单](#)进行咨询。

重新获取配置资源报错，该如何处理？

- 报错信息

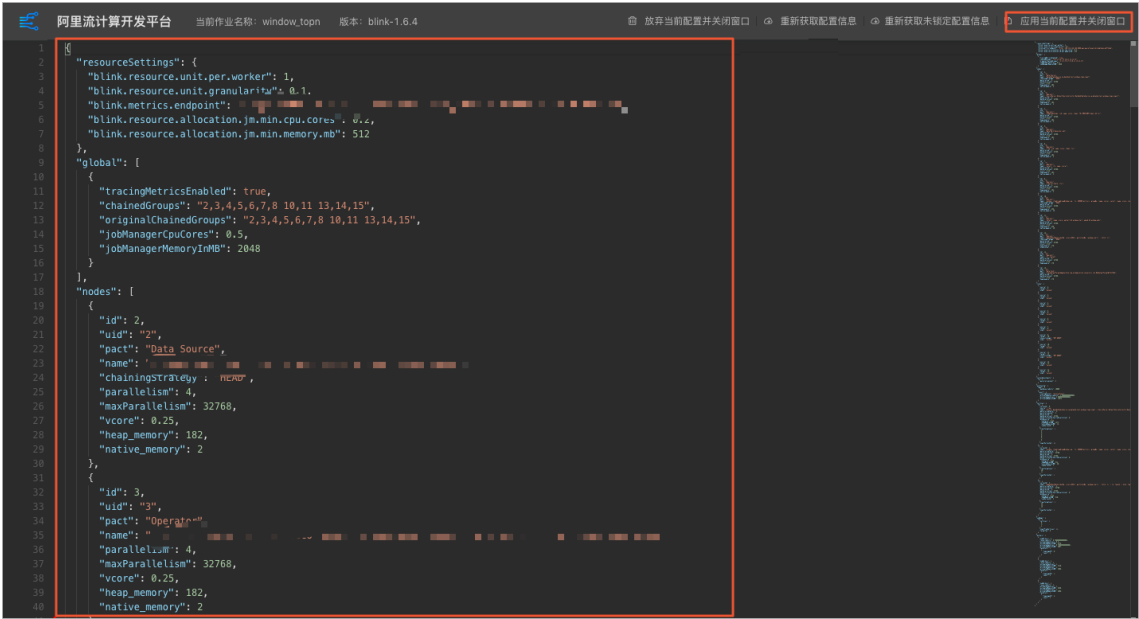


- 报错原因

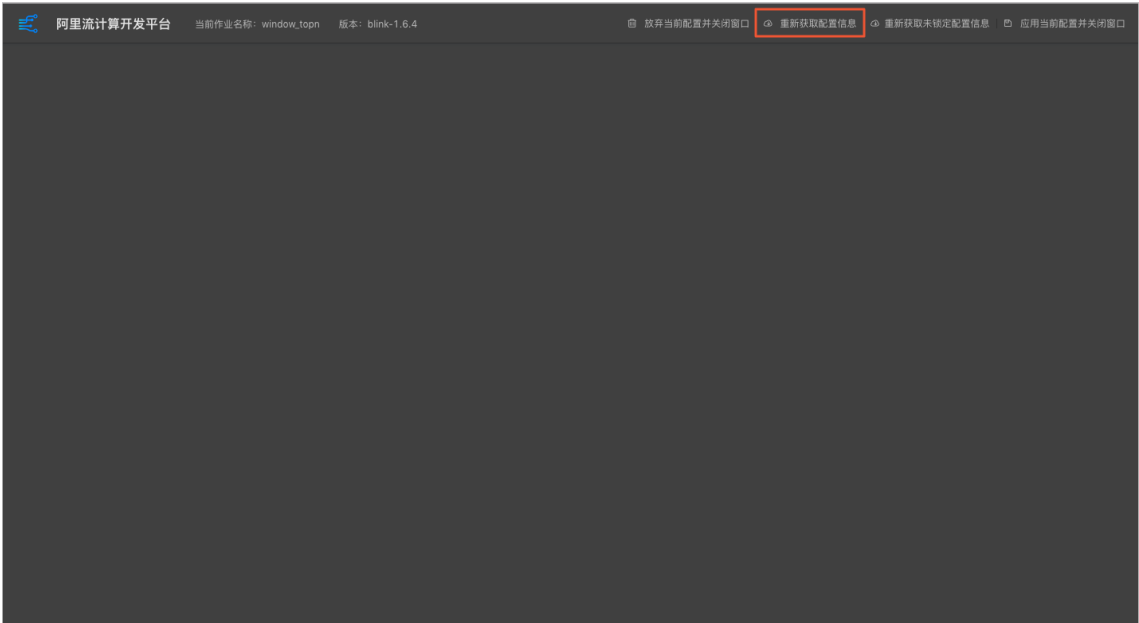
无法正常获取当前资源配置信息。

- 解决方案

- 在资源配置页面，按住CTRL+E将资源配置信息删除后，应用当前配置并关闭窗口。



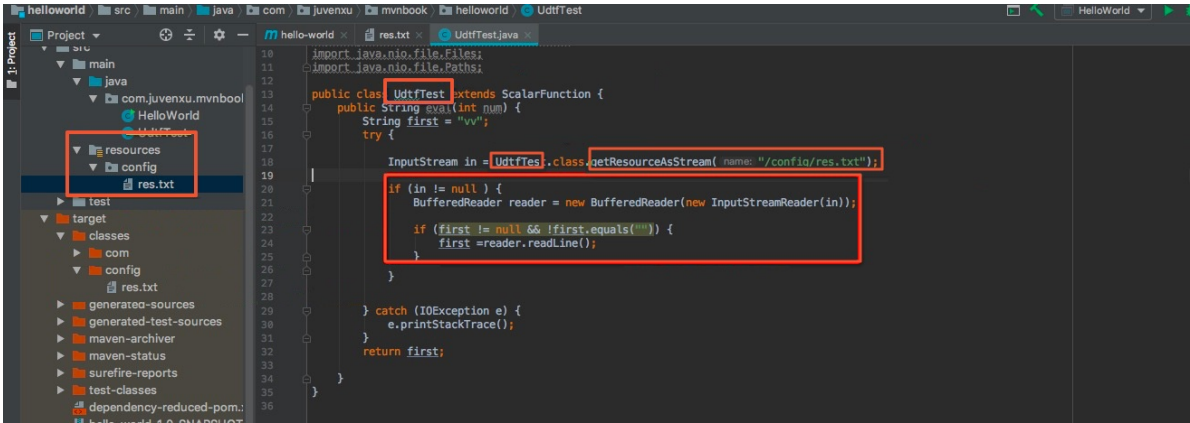
ii. 重新获取配置资源，再应用当前配置并关闭窗口。



在使用UDX函数时，引用JAR包内的本地文件报错

如果引用的JAR包内的本地文档报错信息为空指针，您可以从以下4个方面进行确认：

- 本地文件是否存放在resources文件夹下。
- 是否使用当前加载路径（UdtfTest）。
- 获得相对路径下资源的方法是否为 `getResourceAsStream(name:'/config/res.txt')` 。
- 是否正确读取文件内的内容。



GROUP BY数据出现热点、数据倾斜

- 问题描述
在作业处理的速度很慢时，增加了资源后，作业处理速度并没有明显提升。
- 问题原因
数据出现热点、数据倾斜。
- 解决方案
 - 优化前

```
SELECT
ID,
COUNT(distinct NAME)
FROM AA
GROUP BY ID;
```

- 优化后
- ```
CREATE VIEW SSS AS
SELECT
ID,
NAME
FROM AA
GROUP BY ID,NAME;
INSERT INTO SS
SELECT
ID,
COUNT (NAME)
FROM SSS
GROUP BY ID;
```

❓ 说明 把SQL拆分成两次进行GROUP BY，第一次去重，第二次聚合。

OpenAPI DEMO报错

- 报错信息



```

 "inputVertices": [],\n "subbags": {\n "0": {\n "vertices": [
 "inputFromVertices": [],\n "targetTps": 0.0\n },\n "2": {\n "vertices": [
 "inputFromVertices": [],\n "targetTps": 0.0\n },\n "4": {\n "vertices": [
 "inputFromVertices": [],\n "targetTps": 0.0\n }\n }\n },\n "estimatedMemoryInMB": 2868.0,\n "allocatedCpuCores": 0.7,\n "allocatedMemoryInMB": 0\n }\n },\n "engineVersion": "bliss-2.2.7",\n "engineJobHandler": "application_1510543056345_10028|1ee6912f3a849\nServiceUnavailable: The request has failed due to a temporary failure of the server.\nRequestId: DD304876-E624-4CCE-8748-72EA81EFBA06\nnull\nDisconnected from the target VM, address: '127.0.0.1:5001', transport: 'socket'\n{"errorMessage": "C6444C0", "runSummary": {"id": "304876", "actualState": "TER\njobName": "blissmall_dim_clean"}}

```

- 报错原因  
您使用了4.3.0以下版本的 `aliyun-java-sdk-core` 。
- 解决方案  
请使用4.3.0及以上版本的 `aliyun-java-sdk-core` 。

云监控控制台页面配置监控告警时，显示无数据，应该如何处理？

- 问题描述

实时计算Flink版作业运维页面有数据曲线，且有数据传给云监控消费的日志信息，但在云监控控制台配置监控告警时，显示无数据。
- 问题原因

子账号没有授予相关权限。
- 解决方案

添加AliyunCloudMonitorFullAccess和AliyunCloudMonitorReadOnlyAccess权限。

\* 选择权限

☒ 系统策略

☐ 自定义策略

新建权限策略

云监控



| 权限策略名称                           | 备注                       |
|----------------------------------|--------------------------|
| AliyunCloudMonitorFullAccess     | 管理云监控（CloudMonitor）的权限   |
| AliyunCloudMonitorReadOnlyAccess | 只读访问云监控（CloudMonitor）的权限 |

 **说明** 如果添加AliyunCloudMonitorFullAccess和AliyunCloudMonitorReadOnlyAccess权限后，云监控控制台仍然没有数据，请您[提交工单](#)，产品名称选择云监控。

## 5.上下游存储

本文汇总了实时计算上下游存储常见报错与解决方案。

- [如何避免结果表写入不同步问题？](#)
- [存放的数据产品不在实时计算支持的存储列表，应该如何处理？](#)
- [DataHub作为结果表存在大量重复数据，应该如何处理？](#)
- [DataHub中Shard在只读状态下可以被读取数据吗？](#)
- [实时计算每处理一条记录都会去新增或者更新RDS数据库表吗？将实时计算结果写入数据库，是否会对RDS产生性能影响？](#)
- [实时计算上游数据源是否支持RDS数据库？](#)
- [如何使用Java连接实时计算与消息队列？](#)
- [如何把结果数据输出到一个存在主键限制的外部存储？](#)
- [使用SINK WITH PRIMARY KEY有什么限制？](#)
- [哪些类型的结果表支持在DDL中定义PRIMARY KEY，并能够按照PRIMARY KEY进行INSERT、UPDATE和DELETE？](#)
- [在Sink节点的并发度不为1，多个并发中存在相同PRIMARY KEY的数据时，Sink节点如何保证多个并发间对相同PRIMARY KEY的数据更新顺序一致？](#)
- [为什么相同PRIMARY KEY的数据写到外部存储的顺序和Sink的输入流相同PRIMARY KEY的数据顺序不一致？](#)

### 如何避免结果表写入不同步问题？

- 问题描述

把相同的数据保存到属于不同数据库但表结构相同的两个结果表里，两张表数据的写入速率不同，但最终的结果一致。
- 问题原因

由于结果表的并发量不同，数据分发出现数据乱序。
- 解决方案
  - 对作业进行自动配置调优，详情请参见[自动配置调优](#)。
  - 为两个并发不同的结果表设置相同的节点，让两个节点串联（Chain）在一起。详情请参见[手动配置调优](#)。

### 存放的数据产品不在实时计算支持的存储列表，应该如何处理？

- 如果您所选择的阿里云存储产品不在实时计算系统支持范围之内，请您[提交工单](#)进行咨询。
- 如果您使用了自建开源存储，需要将您的数据迁移到实时计算支持的列表存储中。数据迁移的过程中如果遇到问题，您可以[提交工单](#)进行咨询。

### DataHub作为结果表存在大量重复数据，应该如何处理？

- 问题描述

从DataHub的Topic A中读取数据，通过实时计算处理后写入到DataHub中的Topic B，发现有大量重复数据。
- 原因分析

上述问题是由DataHub存储机制导致的。DataHub是一种消息队列服务，写入一条数据，就存储一条数据，不会对数据的重复性进行判断，而会将每一条输入的数据当成新数据进行储存。在此场景中，运算过程中的计算数据同样被DataHub储存，导致产生重复数据。

- 解决方案

请您将结果表换成RDS。RDS会根据主键更新结果，不会储存计算过程数据。

## DataHub中Shard在只读状态下可以被读取数据吗？

- 问题描述

DataHub中Shard在只读状态下可以被读取数据吗？

|         |                     |  |  |
|---------|---------------------|--|--|
| 创建时间    | 2018-04-03 17:37:46 |  |  |
| 修改时间    | 2018-04-03 17:37:46 |  |  |
| Shard数量 | 12                  |  |  |
| 类型      | TUPLE               |  |  |
| 注释      | stg_product_logs    |  |  |

| Shards | DataConnector       | Metric | Schema              | Subscription |
|--------|---------------------|--------|---------------------|--------------|
| 状态     | 最早数据时间              |        | 最新数据时间              |              |
| CLOSED | 2018-05-08 15:28:45 |        | 2018-05-15 14:21:47 |              |
| CLOSED | 2018-05-08 15:29:02 |        | 2018-05-15 14:22:48 |              |

- 问题解析

Shard在可读写状态下是可以被读取数据的。上图中Shard状态为CLOSED，而CLOSED是只读状态，并非可读写状态，因此无法被读取。关于Shard状态说明，请参见[Shard状态说明](#)。

## 实时计算每处理一条记录都会去新增或者更新RDS数据库表吗？将实时计算结果写入数据库，是否会对RDS产生性能影响？

实时计算的每条结果数据会触发对RDS的写入。如果存在KEY，重复的值会更新（UPDATE），不重复的值会新增（INSERT）。您可以通过设置Batchsize参数来降低对RDS性能的影响。

## 实时计算上游数据源是否支持RDS数据库？

不支持，实时计算支持以下两类上游数据源来驱动计算：

- 消息队列类型。例如，消息队列Kafka、消息队列RocketMQ版和DataHub等。
- 非消息队列类型。例如，大数据计算服务（MaxCompute，原名ODPS）和表格存储Tablestore等。

## 如何使用Java连接实时计算与消息队列？

您可以使用Flink Datastream的方式实现实时计算读取消息队列RocketMQ版中的数据。详情请参见[概述](#)。

## 如何把结果数据输出到一个存在主键限制的外部存储？

A：例如，输出结果数据到一张存在PRIMARY KEY限制的MySQL表，MySQL建表语句如下。

```
CREATE TABLE words (
 word VARCHAR(30) NOT NULL,
 cnt BIGINT,
 CONSTRAINT word_pk PRIMARY KEY(word)
);
```

Q: Blink统计每个word的出现频次后，把结果输出到真实的物理表words中。Blink定义的结果表rds\_output的DDL中声明了PRIMARY KEY为word，Sink处理每条输入数据的逻辑为：

- 如果words表中不存在和rds\_output主键一致的数据(w,c1)，则在words表插入数据(w,c1)。
- 如果words表中存在和rds\_output主键一致的数据(w,c1)，则在words表中更新该条记录，即替换(w,c)为(w,c1)。

#### ④ 说明

- Blink结果表rds\_output定义中的PRIMARY KEY必须和MySQL真实的物理表的PRIMARY KEY一致。
- 如果rds\_output表DDL里未声明PRIMARY KEY，在作业运行阶段，向words表中插入一条word已经存在的记录，则会因为MySQL的主键限制引发运行异常。

---Split用于把句子切分成单词。

```
CREATE FUNCTION split AS 'com.hjc.test.blink.sql.udx.SplitUdtf';
CREATE TABLE tt_input(
 sentences VARCHAR
) with (
 type='datahub',
 endPoint='http://dh-cn-hangzhou.aliyun-inc.com',
 project='<yourProjectName>',
 topic='<yourTopic>',
 accessId='<yourAccessID>',
 accessKey='<yourAccessSecret>'
);
CREATE TABLE rds_output(
 word VARCHAR,
 cnt BIGINT,
 PRIMARY KEY(word)
) with (
 type='rds',
 url='<yourDatabaseURL>',
 tableName='words',
 userName='<yourDatabaseUserName>',
 password='<yourDatabasePassword>'
);
INSERT INTO rds_output
SELECT word, COUNT(1) AS cnt FROM
(
 SELECT word FROM tt_input, LATERAL TABLE(split(sentences)) as T(word)
) GROUP BY word;
```

## 使用SINK WITH PRIMARY KEY有什么限制？

SINK WITH PRIMARY KEY用于把数据输出到一个有主键限制的外部存储中，例如MySQL。使用SINK WITH PRIMARY KEY时，必须同时满足以下两个条件：

- Sink的外部存储必须具有PRIMARY KEY限制，且Blink结果表定义里声明的PRIMARY KEY必须和物理存储的PRIMARY KEY一致。
- Sink必须具有按主键进行INSERT、UPDATE和DELETE的能力。

## 哪些类型的结果表支持在DDL中定义PRIMARY KEY，并能够按照PRIMARY KEY进行INSERT、UPDATE和DELETE？

首先，结果表对应的外部存储必须具有PRIMARY KEY限制。其次，Connector Sink支持单个并发间按照PRIMARY KEY进行INSERT、UPDATE和DELETE。同时满足以上两个条件，才可以在Sink对应的DDL中声明PRIMARY KEY。

Blink内置的Connectors中，仅以下Connectors支持按照PRIMARY KEY进行INSERT、UPDATE和DELETE：

- 云数据库HBase版
- 交互式分析Hologres
- 分析型数据库MySQL3.0
- 分析型数据库MySQL
- 表格存储TableStore
- 云数据库RDS SQL Server版
- 云数据HybridDB for MySQL
- 云数据库Redis版
- 云数据库RDS PostgreSQL版
- 云数据库RDS版
- Oracle
- Elasticsearch
- Phoenix5

## 在Sink节点的并发度不为1，多个并发中存在相同PRIMARY KEY的数据时，Sink节点如何保证多个并发间对相同PRIMARY KEY的数据更新顺序一致？

不能保证多并发间对相同PRIMARY KEY的数据更新顺序一致。因为多个并发间如果存在相同PRIMARY KEY的数据，最终写入到外部存储的数据取决于各个并发间完成写动作的前后顺序，这将导致结果的不确定性。

为了避免因为Sink节点多并发导致的数据不确定性，需要将相同PRIMARY KEY的数据落在同一个Sink并发节点上，以下Pattern都可以让Sink的输入数据满足要求：

- {Aggregate -> Sink}或{Aggregate -> [非状态节点，例如calculate] -> Sink}
- {双流Join -> Sink}或{双流Join -> [非状态节点，例如calculate] -> Sink}

其中Aggregate的GroupKey或双流Join的JoinKey需要满足以下条件：

- SINK PRIMARY KEY需要和JoinKey或GroupKey任何一个相同。
- SINK PRIMARY KEY必须是JoinKey或GroupKey任何一个的子集。
- 如果SINK PRIMARY KEY不和JoinKey或GroupKey任何一个相同，且不是它们任何一个的子集，需要显式的从JoinKey或GroupKey转换到SINK PRIMARY KEY，最直接的方式就是加一层Aggregate，从而以PRIMARY KEY作为Aggregate的GroupKey。

## 为什么相同PRIMARY KEY的数据写到外部存储的顺序和Sink的输入流相同PRIMARY KEY的数据顺序不一致？

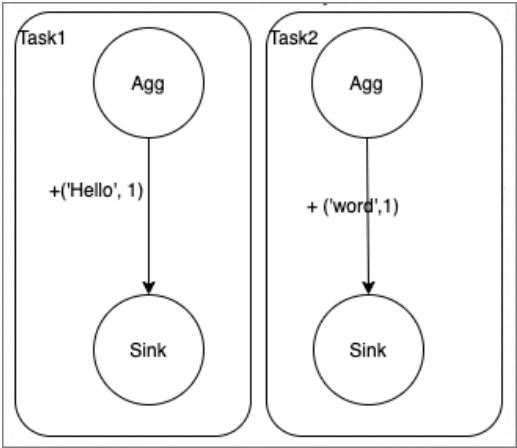
- 作业SQL有问题

例如，由于Sink的PRIMARY KEY和上游的Aggregate的GroupKey并无关联，因此，相同PRIMARY KEY的数据落在不同的并发，最终导致外部存储中的数据顺序和Sink的输入流的不一致，因而数据结果也不正确。

错误示例1

```
---Split用于把句子切分成单词。
CREATE FUNCTION split AS 'com.hjc.test.blink.sql.udx.SplitUdtf';
CREATE TABLE tt_input(
 sentences VARCHAR
) with (
 type='datahub',
 endPoint='http://dh-cn-hangzhou.aliyun-inc.com',
 project='<yourProjectName>',
 topic='<yourTopic>',
 accessId='<yourAccessID>',
 accessKey='<yourAccessSecret>'
);
---错误示例：Sink的PRIMARY KEY和Aggregate的GroupKey不一致。
CREATE TABLE rds_output(
 word VARCHAR,
 cnt BIGINT,
 PRIMARY KEY(cnt)
) with (
 type='rds',
 url='<yourDatabaseURL>',
 tableName='words'
 userName='<yourDatabaseUserName>',
 password='<yourDatabasePassword>'
);
INSERT INTO rds_output
SELECT word, COUNT(1) AS cnt FROM
(
 SELECT word FROM tt_input, LATERAL TABLE(split(sentences)) as T(word)
) GROUP BY word
```

❓ 说明 该示例中，Sink的PRIMARY KEY和Aggregate的GroupKey不一致，导致相同cnt的数据落在不同的并发上，最终写到Sink里的数据不确定。PRIMARY KEY为1的记录可能为('Hello',1)或('word',1)，这取决于哪条记录最后写到RDS。



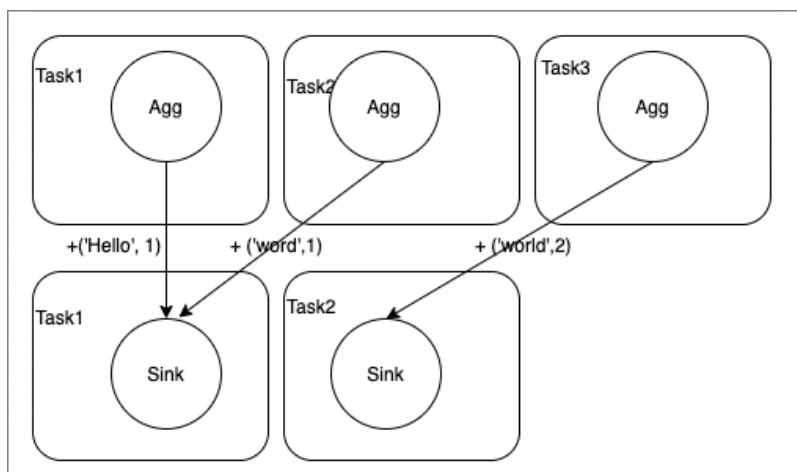
错误示例2

```

---Split用于把句子切分成单词。
CREATE FUNCTION split AS 'com.hjc.test.blink.sql.udx.SplitUdtf';
CREATE TABLE tt_input(
 sentences VARCHAR
) with (
 type='datahub',
 endPoint='http://dh-cn-hangzhou.aliyun-inc.com',
 project='<yourProjectName>',
 topic='<yourTopic>',
 accessId='<yourAccessID>',
 accessKey='<yourAccessSecret>'
);
--- 错误示例：Sink的Primary和Aggregate的GroupKey不一致。
CREATE TABLE rds_output(
 word VARCHAR,
 cnt BIGINT,
 PRIMARY KEY(cnt)
) with (
 type='rds',
 url='<yourDatabaseURL>',
 tableName='words'
 userName='<yourDatabaseUserName>',
 password='<yourDatabasePassword>'
 -- 声明了partitionBy是cnt，Sink前强制按照partitionBy进行一次Shuffle。
 partitionBy = 'cnt'
);
INSERT INTO rds_output
SELECT word, COUNT(1) AS cnt FROM
(
 SELECT word FROM tt_input, LATERAL TABLE(split(sentences)) as T(word)
) GROUP BY word
;

```

❓ 说明 该示例中，Sink的Primary和Aggregate的GroupKey仍然不一致，和错误示范1唯一的区别为Sink声明了PartitionedKey，导致写入Sink前会按照partitionKey进行一次Shuffle。但是这样的结果仍然具有不确定性，PRIMARY KEY为1的记录可能为('Hello',1)或('word',1)，这取决于哪条记录最后写到RDS。



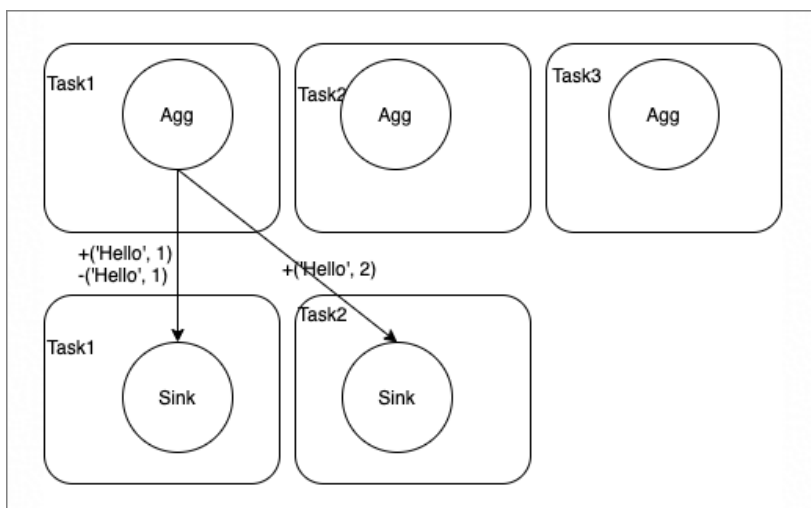


- 作业Planjson被破坏

作业的Planjson被手动修改了Sink或Aggregate并发度，导致Aggregate和下游的Sink不能Chain起来，破坏了Sink输入数据的Distribution属性。

```
CREATE TABLE tt_input(
 word VARCHAR
) with (
 type='datahub',
 endPoint='http://dh-cn-hangzhou.aliyun-inc.com',
 project='<yourProjectName>',
 topic='<yourTopic>',
 accessId='<yourAccessID>',
 accessKey='<yourAccessSecret>'
);
CREATE TABLE rds_output(
 word BIGINT,
 cnt BIGINT,
 PRIMARY KEY(word)
) with (
 type='rds',
 url='<yourDatabaseURL>',
 tableName='words',
 userName='<yourDatabaseUserName>',
 password='<yourDatabasePassword>'
);
INSERT INTO rds_output
 SELECT word, COUNT(1) AS cnt FROM tt_input GROUP BY word;
```

❓ 说明 该示例中，SQL本身没有问题，但在Planjson上手动调整Sink节点的并发度，导致原来的Forward的边变成了Rebalance的边。最终PRIMARY KEY为Hello对应的记录可能为('Hello',2)，也可能不存在该PRIMARY KEY，这取决于哪条记录最后写到RDS。



## 6.SQL常见问题

### 6.1. 语法检查报错

本文汇总了实时计算语法检查常见报错与解决方案。

SQL编写完成后，请您先进行语法检查。如果出现报错信息，请更改SQL后，再上线运行您的作业。常见的语法检查报错信息如下：

- 报错：column '<columnName>' is ambiguous
- 报错：Failed to encode 'xxxx' in character set 'ISO-8859-1'
- 报错：'Rds' is not supported as a source table
- 报错：Query result and target table 'table\_name' field length not match
- 报错：Interval field value 100 exceeds precision of MINUTE(2) field
- 报错：ODPS \*\*\* Part it ion null table count ODPS tables row count exceeds maxRowCount limit {2}
- 报错：SQL validation failed: index (55) must be less than size (55)

#### 报错：column '<columnName>' is ambiguous

- 报错信息

```
column '<columnName>' is ambiguous 。
```

- 问题原因

关联的两张表存在同名字段，在引用时未在字段前指定表名。

- 解决方案

- 修改前

```
select
column_name
.....
join table_A.column_name = table_B.column_name
```

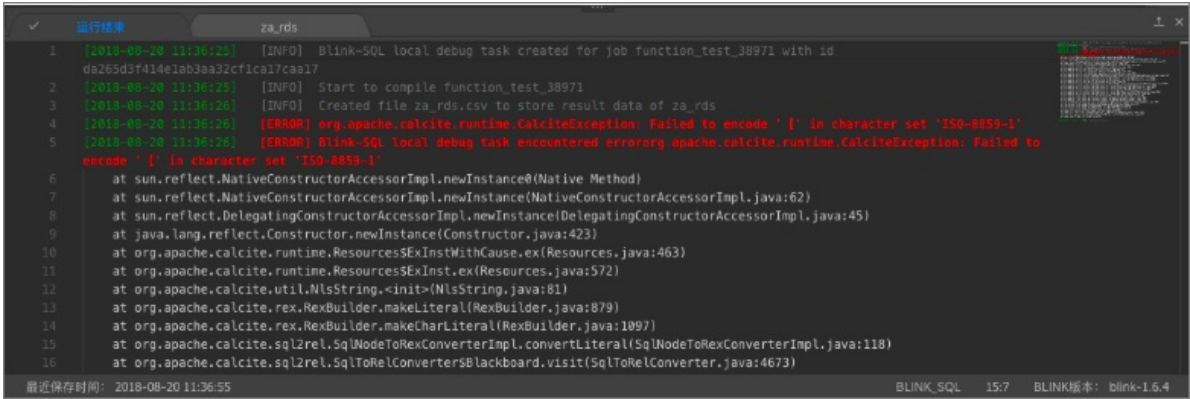
- 修改后

```
select
table_A.column_name
.....
join table_A.column_name =table_B.column_name
```

#### 报错：Failed to encode 'xxxx' in character set 'ISO-8859-1'

- 报错信息

```
[ERROR] org.apache.calcite.runtime.CalciteException: Failed to encode '\[' in character set 'ISO-8859-1' [ERROR] Blink-SQL local debug task encountered errororg.apache.calcite.runtime.CalciteException: Failed to encode '\[' in character set 'ISO-8859-1'
```



- 报错原因  
Flink SQL中不支持直接使用中文字符，包括中文文字和中文标点符号。
- 解决方案  
在中文字符前面添加 `_UTF16`。

`_UTF16'【'`  
`_UTF16'您好'`

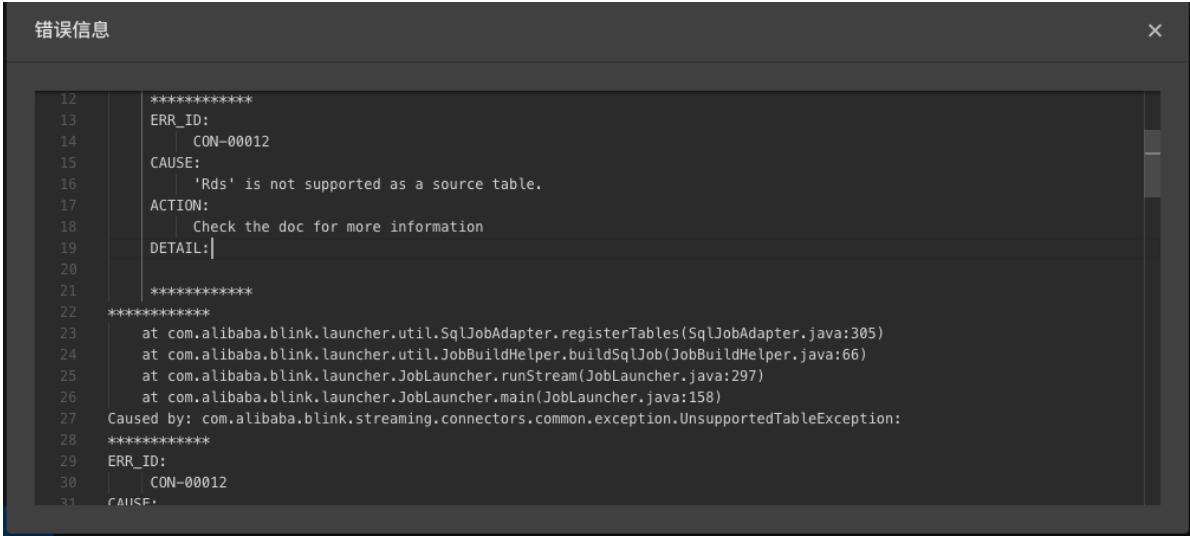
说明 常量都需要添加 `_UTF16`。

报错：'Rds' is not supported as a source table

- 报错信息
- ERR\_ID:  
CON-00012

CAUSE:  
'Rds' is not supported as a source table.

ACTION:  
Check the doc for more information

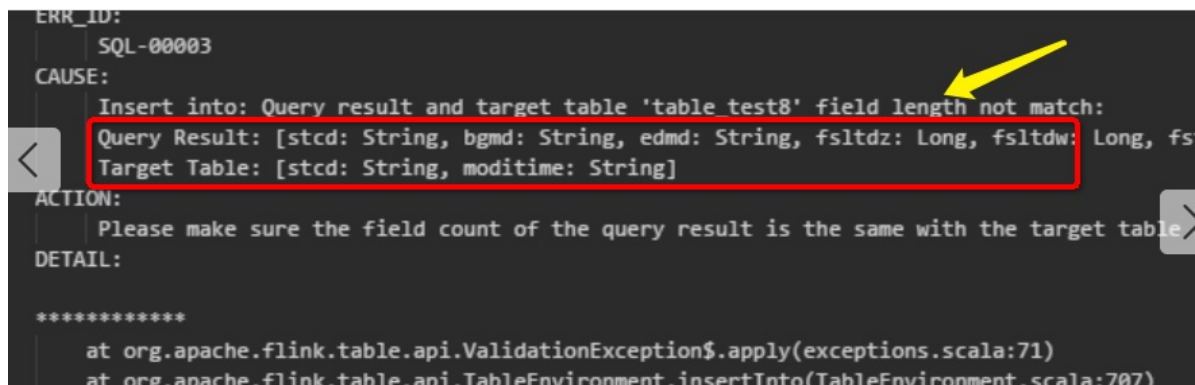


- 报错原因
  - 作业中声明了RDS的表，但是没有INSERT该表。

- RDS作为维表，但是没有声明维表PERIOD FOR SYSTEM\_TIME。
- 解决方案
  - INSERT作业中声明的RDS表。
  - 在作业中的维表中写入声明PERIOD FOR SYSTEM\_TIME。

## 报错：Query result and target table 'table\_name' field length not match

- 报错信息



```
ERR_ID:
SQL-00003
CAUSE:
Insert into: Query result and target table 'table test8' field length not match:
Query Result: [stcd: String, bgmd: String, edmd: String, fsltdz: Long, fsltdw: Long, fs
Target Table: [stcd: String, moditime: String]
ACTION:
Please make sure the field count of the query result is the same with the target table
DETAIL:

at org.apache.flink.table.api.ValidationException$.apply(exceptions.scala:71)
at org.apache.flink.table.api.TableEnvironment.insertInto(TableEnvironment.scala:707)
```

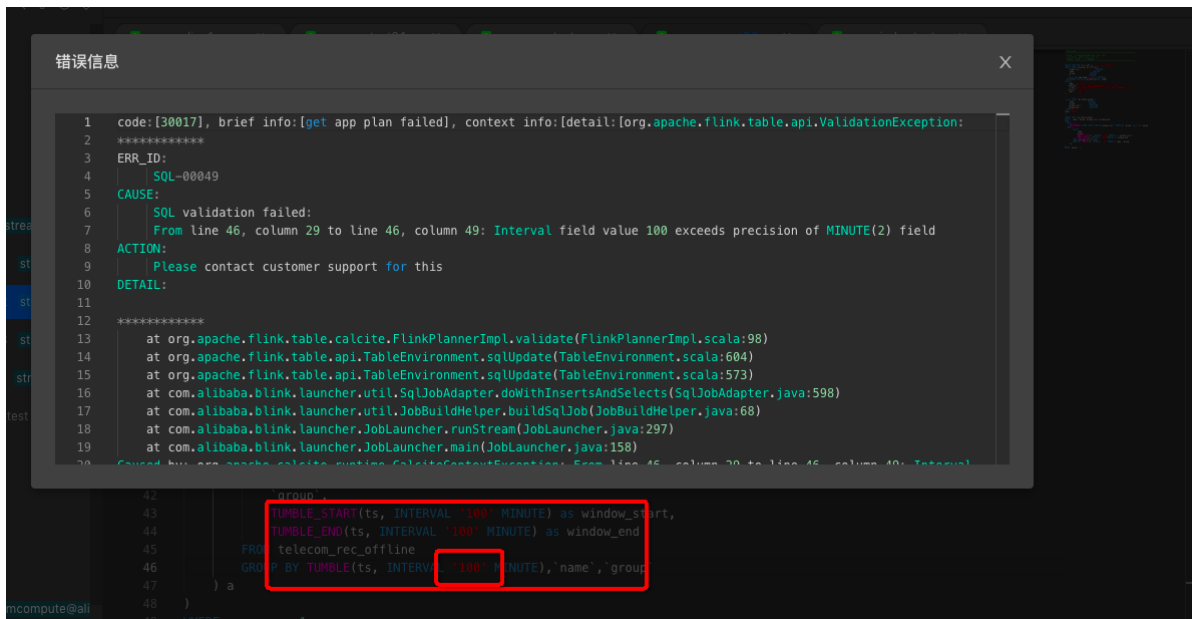
? 说明 报错提示中的table\_name指代具体实例中的表名。

- 报错原因  
写入的字段与结果表中声明字段的数量、类型、顺序不一致。
- 解决方案  
修改语句使写入语句的字段数量、类型、顺序都与结果表保持一致。

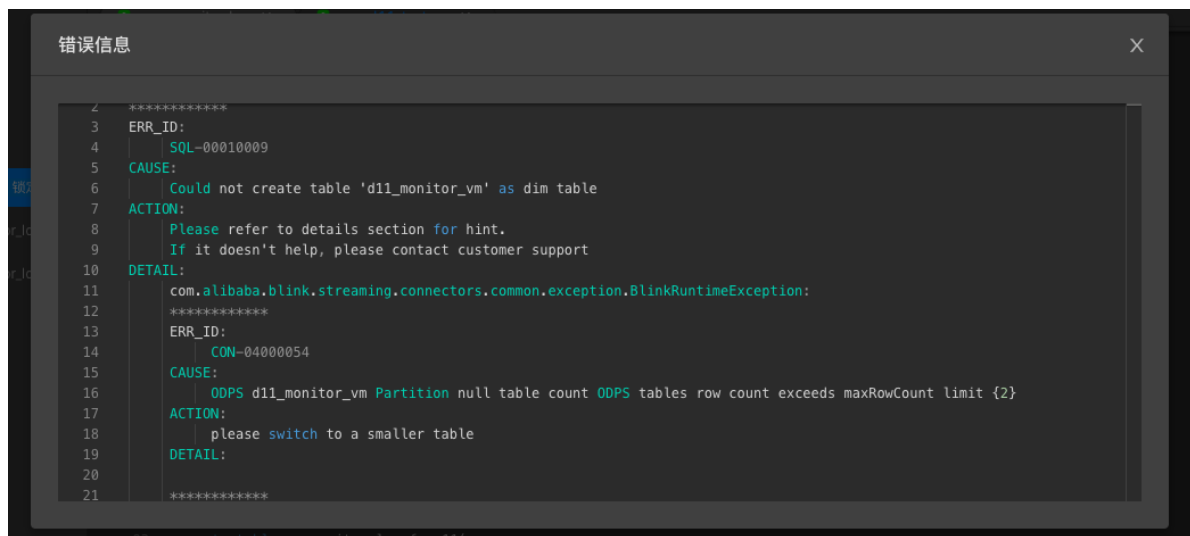
## 报错：Interval field value 100 exceeds precision of MINUTE(2) field

- 报错信息

在窗口里的年、月、日、时、分、秒设置超过两位数时，语法检查报错。



在MaxCompute（ODPS）作为维表时，语法检查报错。

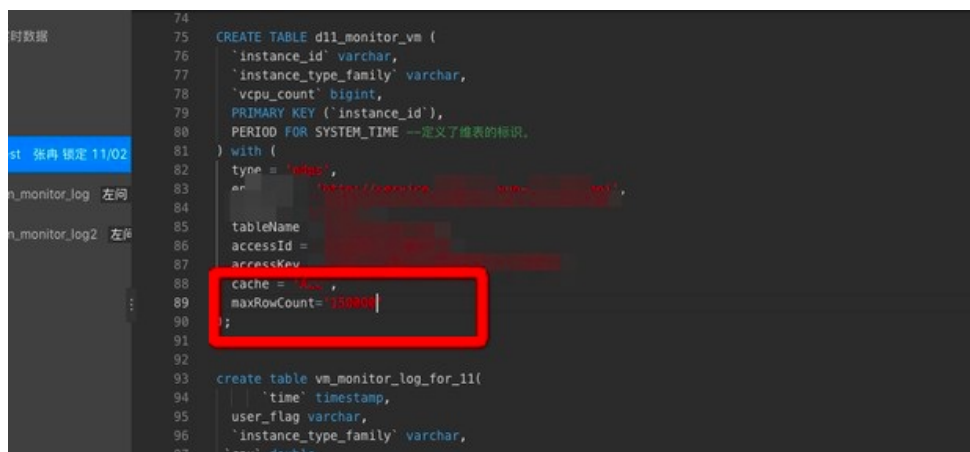


- 报错原因

MaxCompute（ODPS）维表数据记录数超过了maxRowCount默认值 100000 。

- 解决方案

修改maxRowCount参数值，如下图所示。



报错：SQL validation failed: index (55) must be less than size (55)

- 报错信息

在使用MaxCompute（ODPS）结果表的动态分区功能时，语法检查报错。

```
code:[30017], brief info:[get app plan failed], context info:[detail:
>>>>
org.apache.flink.table.api.ValidationException:

ERR_ID:
 SQL-00120001
CAUSE:
 SQL validation failed:
 index (55) must be less than size (55)
ACTION:
 Please see descriptions above. If it doesn't help, please contact customer support for
 this.
DETAIL:

<<<<<]
```

- 报错原因

MaxCompute（ODPS）结果表DDL语句中没有声明分区列。

- 解决方案

在DDL语句中声明MaxCompute（ODPS）的分区列。例如示例中的`partition` = 'ds=2018\*\*\*\*'。

```
create table odps_output(
 id INT,
 user_name VARCHAR,
 content VARCHAR
) with (
 type = 'odps',
 endPoint = 'http://service.cn.maxcompute.aliyun-inc.com/api',
 project = 'projectName',
 tableName = 'tableName',
 accessId = 'yourAccessKeyId',
 accessKey = 'yourAccessKeySecret',
 `partition` = 'ds=2018****'
);
```

MaxCompute（ODPS）结果表DDL语句详情，请参见[创建MaxCompute（ODPS）结果表](#)。

## 6.2. 调试报错

本文汇总了实时计算调试常见报错与解决方案。

- 报错：Query result field length not match TableSink analysis\_result field length
- 报错：Cannot apply 'EXTRACT' to arguments of type 'EXTRACT(<INTERVAL HOUR> FROM <CHAR(19)>)'
- 报错：originated by DoubleParser:NUMERIC\_VALUE\_FORMAT\_ERROD
- 报错：Query result field type(s) not match TableSink T2 field type(s)
- 报错：The type 'B' is not supported for the CSV input format.
- 报错：检测到调试数据与schema字段数量不符
- 报错：操作错误，所有数据均被跳过，请检查DDL中的LENGTHCHECK项

**报错：Query result field length not match TableSink analysis\_result field length**



● 报错信息

```
[2018-07-11 17:01:17] [INFO] Blink-SQL local debug task created for job warn_url_leak_31880 with id f0f911f1dcbe52d0e004027c0ac64ad
[2018-07-11 17:01:17] [INFO] Start to compile warn url leak 31880
[2018-07-11 17:01:17] [INFO] Created file analysis_result.csv to store result data of analysis_result
[2018-07-11 17:01:17] [ERROR] Blink-SQL local debug task encountered errororg.apache.flink.table.api.ValidationException: Query result field length
not match TableSink analysis_result field length.
Query fields: [id: String, EXPR$1: String, EXPR$2: String, EXPR$3: String, EXPR$4: String, EXPR$5: String, EXPR$6: String, EXPR$7: String, EXPR$8:
String, EXPR$9: String, EXPR$10: String, EXPR$11: String, EXPR$12: String, EXPR$13: String, EXPR$14: String, EXPR$15: String, EXPR$16: String,
EXPR$17: String, EXPR$18: String, EXPR$19: String, EXPR$20: String, EXPR$21: String, EXPR$22: String, EXPR$23: String, EXPR$24: String, EXPR$25:
String, EXPR$26: String, EXPR$27: String, user_name: String, user_id: String, area_code: String]
TableSink fields: [id: String, risk_event_name: String, rule_id: String, rule_name: String, trigger_time: String, analysis_type: Integer,
status_enum: Integer, process_type_enum: Integer, know_ids: String, know_names: String, process_relate_id: String, weight: Integer, related_ips:
String, security_domain_guids: String, security_domain_names: String, logs_info: String, dst_ips: String, dst_asset_guids: String,
dst_security_domain_guids: String, dst_security_domain_names: String, marks: String, risk_event_code: String, priority_level: String, asset_guids:
String, event_type: String, event_type_level: String, device_type: String, device_type_level: String, result_guid: String, rule_code: String,
username: String, user_id: String, area_code: String]
at org.apache.flink.table.api.ValidationException$.apply(exceptions.scala:71)
at org.apache.flink.table.api.TableEnvironment.insertInto(TableEnvironment.scala:696)
at org.apache.flink.table.api.TableEnvironment.sqlUpdate(TableEnvironment.scala:611)
at org.apache.flink.table.api.TableEnvironment.sqlUpdate(TableEnvironment.scala:569)
at com.alibaba.blink.launcher.util.SqlJobAdapter.dealInserts(SqlJobAdapter.java:482)
at com.aliyun.blink.debug.BlinkSqlLocalDebug.run(BlinkSqlLocalDebug.java:115)
at com.aliyun.blink.debug.BlinkDebugTask.call(BlinkDebugTask.java:66)
at com.aliyun.blink.debug.BlinkDebugTask.call(BlinkDebugTask.java:12)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1152)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:627)
at java.lang.Thread.run(Thread.java:852)
```

● 报错原因

查询结果字段长度与结果表字段长度不匹配。

● 解决方法

- 调整INSERT语句中的字段数量，与结果表一致。
- 调整INSERT语句中插入的字段数据类型和字段顺序，与结果表一致。
- 如果使用前两种方法排查后，仍然报错，请使用二分法定位问题。

报错：Cannot apply 'EXTRACT' to arguments of type 'EXTRACT(<INTERVAL HOUR> FROM <CHAR(19)>)'

● 报错信息

```
运行结果 T2
1 [2018-07-09 11:45:13] [INFO] Blink-SQL local debug task created for job test_bxy_34673 with id 248c93855576a52359c5ec214411dac8
2 [2018-07-09 11:45:13] [INFO] Start to compile test_bxy_34673
3 [2018-07-09 11:45:13] [INFO] Created file T2.csv to store result data of T2
4 [2018-07-09 11:45:13] [ERROR] org.apache.calcite.sql.validate.SqlValidatorException: Cannot apply 'EXTRACT' to arguments of type
'EXTRACT(<INTERVAL HOUR> FROM <CHAR(19)>)', Supported form(s): 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
5 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
6 [2018-07-09 11:45:13] [ERROR] org.apache.calcite.runtime.CalciteContextException: From line 20, column 3 to line 20, column 29:
Cannot apply 'EXTRACT' to arguments of type 'EXTRACT(<INTERVAL HOUR> FROM <CHAR(19)>)', Supported form(s): 'EXTRACT
(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
7 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
8 [2018-07-09 11:45:13] [ERROR] Blink-SQL local debug task encountered errororg.apache.flink.table.api.ValidationException: SQL
validation failed. From line 20, column 3 to line 20, column 29: Cannot apply 'EXTRACT' to arguments of type 'EXTRACT(<INTERVAL
HOUR> FROM <CHAR(19)>)', Supported form(s): 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
9 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
10 Caused by: org.apache.calcite.runtime.CalciteContextException: From line 20, column 3 to line 20, column 29: Cannot apply
'EXTRACT' to arguments of type 'EXTRACT(<INTERVAL HOUR> FROM <CHAR(19)>)', Supported form(s): 'EXTRACT(<DATETIME_INTERVAL> FROM
<DATETIME_INTERVAL>)'
11 'EXTRACT(<DATETIME_INTERVAL> FROM <DATETIME_INTERVAL>)'
最近保存时间: 2018-07-09 11:45:13 BLINK SQL 20-28 BLINK版本: blink-1.6.4
```

● 报错原因

EXTRACT函数的参数类型使用不正确。

● 解决方案

使用正确的EXTRACT函数参数类型，详情请参见EXTRACT。

报错：originated by DoubleParser:NUMERIC\_VALUE\_FORMAT\_ERROD

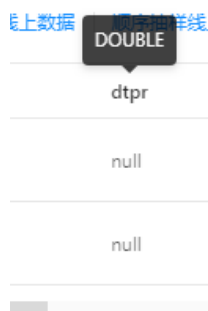
● 报错信息

在DataHub中存在NULL值时，如果从DataHub里抽取数据，报错如下。

```
161 [2018-05-18 17:14:29] [INFO] Stopping Akka RPC service.
162 [2018-05-18 17:14:29] [INFO] Flink Mini Cluster is shut down
163 [2018-05-18 17:14:29] [ERROR] Blink-SQL local debug task encountered error org.apache.flink.runtime.client.JobExecutionException: Job execution failed.
164 org.apache.flink.api.common.io.ParseException: Parsing error for column 10 of row '
null,null,null,null,null,48145-05-30 07:13:20,null,null' originated by DoubleParser: NUMERIC_VALUE_FORMAT_ERROR.
165 at org.apache.flink.runtime.jobmaster.JobMaster.jobStatusChanged(JobMaster.java:1304)
166 at org.apache.flink.runtime.jobmaster.JobMaster.access$3600(JobMaster.java:143)
167 at org.apache.flink.runtime.jobmaster.JobMaster$JobManagerJobStatusListener$1.run(JobMaster.java:1512)
168 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleRunAsync(AkkaRpcActor.java:273)
169 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleMessage(AkkaRpcActor.java:131)
170 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.access$000(AkkaRpcActor.java:72)
171 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor$1.apply(AkkaRpcActor.java:110)
172 at akka.actor.ActorCell$$anonfun$become$1.applyOrElse(ActorCell.scala:534)
173 at akka.actor.Actor$class.aroundReceive(Actor.scala:467)
174 at akka.actor.UntypedActor.aroundReceive(UntypedActor.scala:97)
```

#### ● 报错原因

CSV文件会把NULL值转化为字符串NULL，系统在处理NULL时，发现记录类型与字符类型不一致，就会提示格式转换错误。在抽取DataHub数据作为样本调试时，通常会出现此错误，如果项目正式上线，则不会出现类似于下图的错误。



#### ● 解决方案

- 过滤DataHub中NULL值。
- 处理NULL为您期望的值。例如，使用CASE WHEN函数处理NULL值。

### 报错：Query result field type(s) not match TableSink T2 field type(s)

#### ● 报错信息

```
运行结果 T2
1 [2018-07-09 11:52:40] [INFO] Blink-SQL local debug task created for job test_bxy_34674 with id 339125b7b419203952b7e9659a0acf7b
2 [2018-07-09 11:52:40] [INFO] Start to compile test_bxy_34674
3 [2018-07-09 11:52:40] [INFO] Created file T2.csv to store result data of T2
4 [2018-07-09 11:52:40] [ERROR] Blink-SQL local debug task encountered error org.apache.flink.table.api.ValidationException: Query result field type(s) not match TableSink T2 field type(s).
5 Query mismatched fields: [EXPR$0: String]
6 TableSink mismatched fields: [var1: Long]
7 at org.apache.flink.table.api.ValidationExceptions.apply(exceptions.scala:71)
8 at org.apache.flink.table.api.TableEnvironment.insertInto(TableEnvironment.scala:716)
9 at org.apache.flink.table.api.TableEnvironment.sqlUpdate(TableEnvironment.scala:611)
10 at org.apache.flink.table.api.TableEnvironment.sqlUpdate(TableEnvironment.scala:569)
11 at com.alibaba.blink.launcher.util.SqlJobAdapter.dealInserts(SqlJobAdapter.java:482)
12 at com.aliyun.blink.debug.BlinkSqlLocalDebug.run(BlinkSqlLocalDebug.java:115)
13 at com.aliyun.blink.debug.BlinkDebugTask.call(BlinkDebugTask.java:66)
14 at com.aliyun.blink.debug.BlinkDebugTask.call(BlinkDebugTask.java:12)
15 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
16 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1152)
17 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:627)
```

#### ● 报错原因

类型转换异常。

#### ● 解决方案

查看INSERT语句中的数据类型和结果表声明的数据类型是否一致。

## 报错：The type '[B' is not supported for the CSV input format.

- 报错信息

```
110 [2021-01-12 14:22:28] [INFO] Successfully got table 'kafka_stream_input_src' from catalog 'builtin'
111 [2021-01-12 14:22:28] [ERROR] Blink-SQL local debug task encountered error, org.apache.flink.table.api.ValidationException:
112 *****
113 ERR_ID:
114 | SQL-00120001
115 CAUSE:
116 | SQL validation failed:
117 | The type '[B' is not supported for the CSV input format.
118 ACTION:
119 | Please see descriptions above. If it doesn't help, please contact customer support for this.
120 DETAIL:
121 |
122 *****
123 Caused by: java.lang.IllegalArgumentException: The type '[B' is not supported for the CSV input format.
124 | at org.apache.flink.table.calcite.FlinkPlannerImpl.validate(FlinkPlannerImpl.scala:114)
125 | at org.apache.flink.table.util.TableEnvironmentUtil$.queryToRel(TableEnvironmentUtil.scala:38)
126 | at org.apache.flink.table.api.TableEnvironment.sqlQuery(TableEnvironment.scala:1392)
127 | at com.alibaba.blink.launcher.util.SqlJobAdapter.registerViews(SqlJobAdapter.java:482)
128 | at com.alibaba.blink.debug.Blink3xSqlDbg.doRun(Blink3xSqlDbg.java:161)
129 | at com.alibaba.blink.debug.common.BlinkSqlDebug.run(BlinkSqlDebug.java:95)
130 | at com.alibaba.blink.debug.common.BlinkDebugTask.call(BlinkDebugTask.java:26)
131 | at com.alibaba.blink.debug.common.BlinkDebugTask.call(BlinkDebugTask.java:7)
132 | at java.util.concurrent.FutureTask.run(FutureTask.java:266)
133 | at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1147)
134 | at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)
135 | at java.lang.Thread.run(Thread.java:834)
136
137 [2021-01-12 14:22:28] [INFO] dbg process is finish!
138 [2021-01-12 14:22:28] [INFO] process is finish!
139 [2021-01-12 14:22:28] [ERROR] requestId: 202103121422-7uK5EVMA25bayes channel handler exception happened. java.lang.NumberFormatException: nu
140
```

- 报错原因

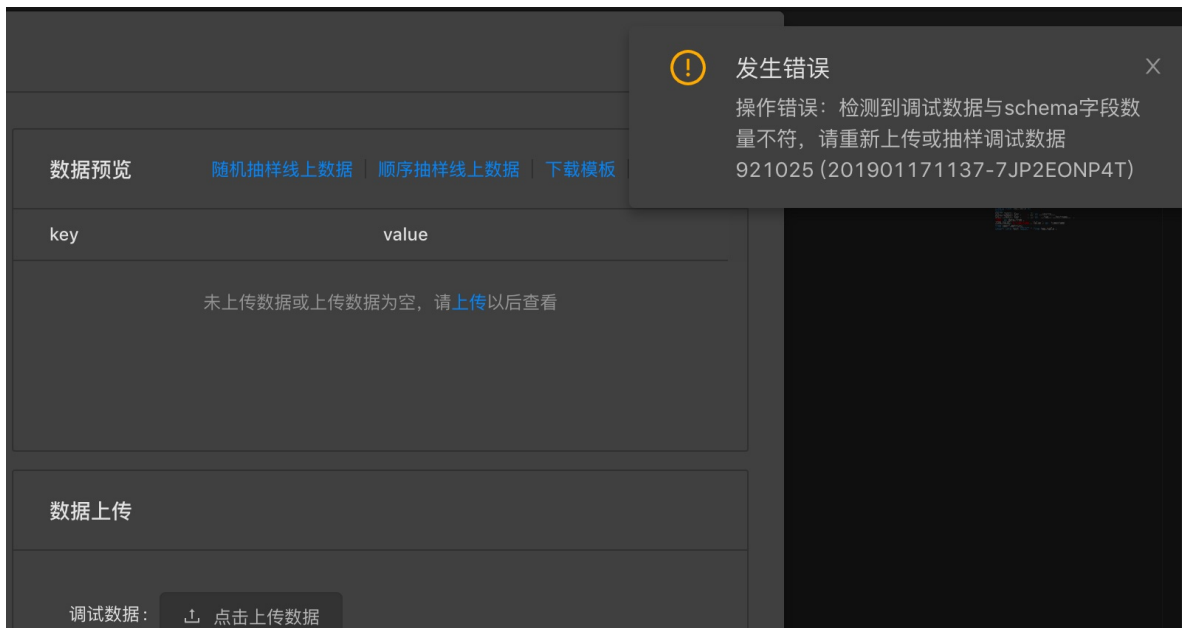
二进制数据类型不支持本地调试。

- 解决方案

语法检查没有问题请进行线上调试，详情请参见[线上调试](#)

## 报错：检测到调试数据与schema字段数量不符

- 报错信息



- 报错原因

在调试时，默认使用逗号分隔字段，如果您的数据中存在JSON数据，则数据会被解析成多个字段。

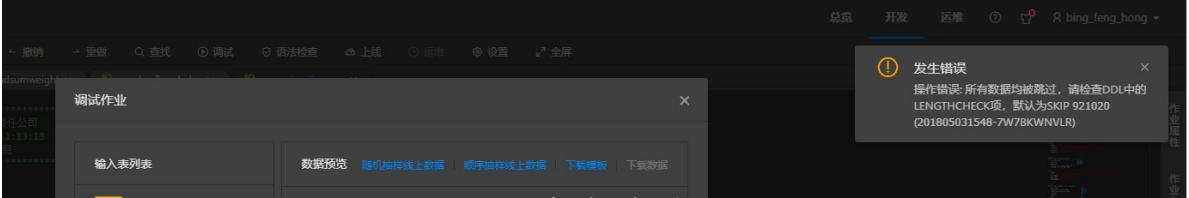
- 解决方案

在作业参数中设置分隔符为非逗号。

```
debug.input.delimiter= |
```

**报错：操作错误，所有数据均被跳过，请检查DDL中的LENGTHCHECK项**

- 报错信息



- 错误原因

源表中存在脏数据，导致无法抽取到任何线上数据。

- 解决方法

在源表的WITH参数中增加 `lengthcheck='PAD'` 语句。

## 6.3. 上线报错

本文汇总了实时计算上线常见报错与解决方案。

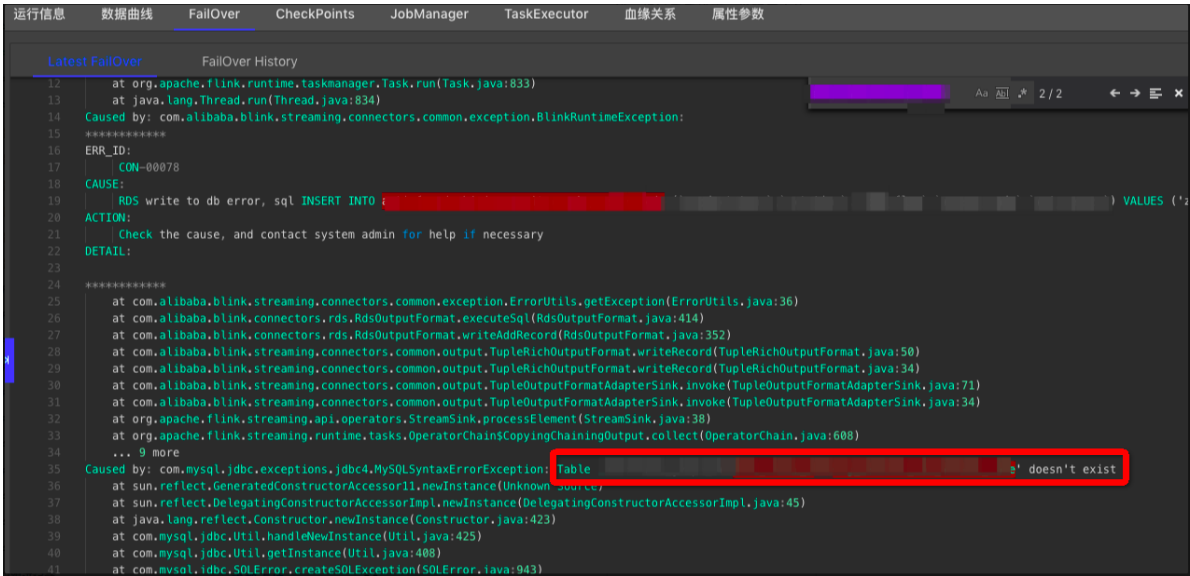
- 报错：Table 'database.table' doesn't exist
- 报错：Could not forward element to next operator,null pointer Exception
- 报错：ERROR in next method The response is not valid json
- 报错：ERROR in next method AccessKeyId ST5.NK1p3aqjkkagcUHITpc\*\*\*\* is inactive
- 报错：open catalog connection failed
- 报错：获取数据超时，请重试或尝试手动编辑
- 警告：WARNING（不影响当前流程）

**报错：Table 'database.table' doesn't exist**

- 报错信息

说明

报错中 `database.table` 指代具体数据库名和表名。



```
12 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:833)
13 at java.lang.Thread.run(Thread.java:834)
14 Caused by: com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException:
15 *****
16 ERR_ID:
17 CON-00078
18 CAUSE:
19 RDS write to db error, sql INSERT INTO ` ` VALUES (':
20 ACTION:
21 Check the cause, and contact system admin for help if necessary
22 DETAIL:
23 *****
24
25 at com.alibaba.blink.streaming.connectors.common.exception.ErrorUtils.getException(ErrorUtils.java:36)
26 at com.alibaba.blink.connectors.rds.RdsOutputFormat.executeSql(RdsOutputFormat.java:414)
27 at com.alibaba.blink.connectors.rds.RdsOutputFormat.writeAddRecord(RdsOutputFormat.java:352)
28 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:50)
29 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:34)
30 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:71)
31 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:34)
32 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:38)
33 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
34 ... 9 more
35 Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Table ' ' doesn't exist
36 at sun.reflect.GeneratedConstructorAccessor11.newInstance(Unknown Source)
37 at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
38 at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
39 at com.mysql.jdbc.Util.handleNewInstance(Util.java:425)
40 at com.mysql.jdbc.Util.getInstance(Util.java:408)
41 at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:943)
```

● 报错原因

数据库中不存在作业中声明的表。

● 解决方案

- 检查数据存储中的表是否引用成功。如果不成功，请再次尝试或手动输入。
- 检查您的数据库中是否存在报错信息中的表，如果没有，请自行创建。

❓ 说明 阿里云实时计算不具有数据存储功能，所有涉及表创建的DDL操作，都是对外部数据表、存储的引用声明。

## 报错：Could not forward element to next operator,null pointer Exception

● 报错信息

```
java.lang.RuntimeException: java.lang.RuntimeException: Could not forward element to next operator
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:131)
 at com.alibaba.blink.streaming.connectors.common.source.AbstractParallelSource.run(AbstractParallelSource.java:120)
 at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:142)
 at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:78)
 at org.apache.flink.streaming.runtime.tasks.SourceStreamTask.run(SourceStreamTask.java:56)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:387)
 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:762)
 at java.lang.Thread.run(Thread.java:834)
Caused by: java.lang.RuntimeException: Could not forward element to next operator
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:611)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:593)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:941)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:919)
 at org.apache.flink.streaming.api.operators.StreamSource$1.collectWithMetrics(StreamSource.java:127)
 at org.apache.flink.streaming.api.operators.StreamSource$1.collect(StreamSource.java:114)
 at org.apache.flink.streaming.api.operators.StreamSource$1.collect(StreamSource.java:98)
 at org.apache.flink.streaming.api.operators.StreamSourceContexts$ManualWatermarkContext.collectWithTimestamp(StreamSourceContexts.java:273)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.runImpl(ParallelReader.java:223)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:126)
 ... 7 more
Caused by: com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException: 900005

Column For Parser Plugin Error!
 at com.alibaba.blink.streaming.connectors.common.source.parse.DefaultSourceCollector.parseAndCollect(DefaultSourceCollector.java:366)
 at com.alibaba.blink.streaming.connectors.common.source.parse.DefaultSourceCollector.parseAndCollect(DefaultSourceCollector.java:75)
 at com.alibaba.blink.streaming.connectors.common.source.parse.SourceParser.flatMap(SourceParser.java:47)
 at org.apache.flink.streaming.api.operators.StreamFlatMap.processElement(StreamFlatMap.java:47)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
 ... 16 more
Caused by: java.lang.RuntimeException: Could not forward element to next operator
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:611)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:593)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:941)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:919)
 at org.apache.flink.streaming.api.operators.TimestampedCollector.collect(TimestampedCollector.java:51)
 at com.alibaba.blink.streaming.connectors.common.source.parse.DefaultSourceCollector.parseAndCollect(DefaultSourceCollector.java:400)
 at com.alibaba.blink.streaming.connectors.common.source.parse.DefaultSourceCollector.parseAndCollect(DefaultSourceCollector.java:362)
 ... 20 more
Caused by: java.lang.NullPointerException
 at com.alibaba.blink.streaming.connectors.common.source.watermark.WatermarkUtils$BayesPunctuatedTimestampAndWatermarkWithOffset.extractTimestamp(WatermarkUtils.java:114)
 at org.apache.flink.streaming.runtime.operators.timestampsAndPunctuatedWatermarksOperator.processElement(TimestampsAndPunctuatedWatermarksOperator.java:50)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
 ... 26 more
```

- 报错原因

数据源端存在NULL值。

- 解决方案

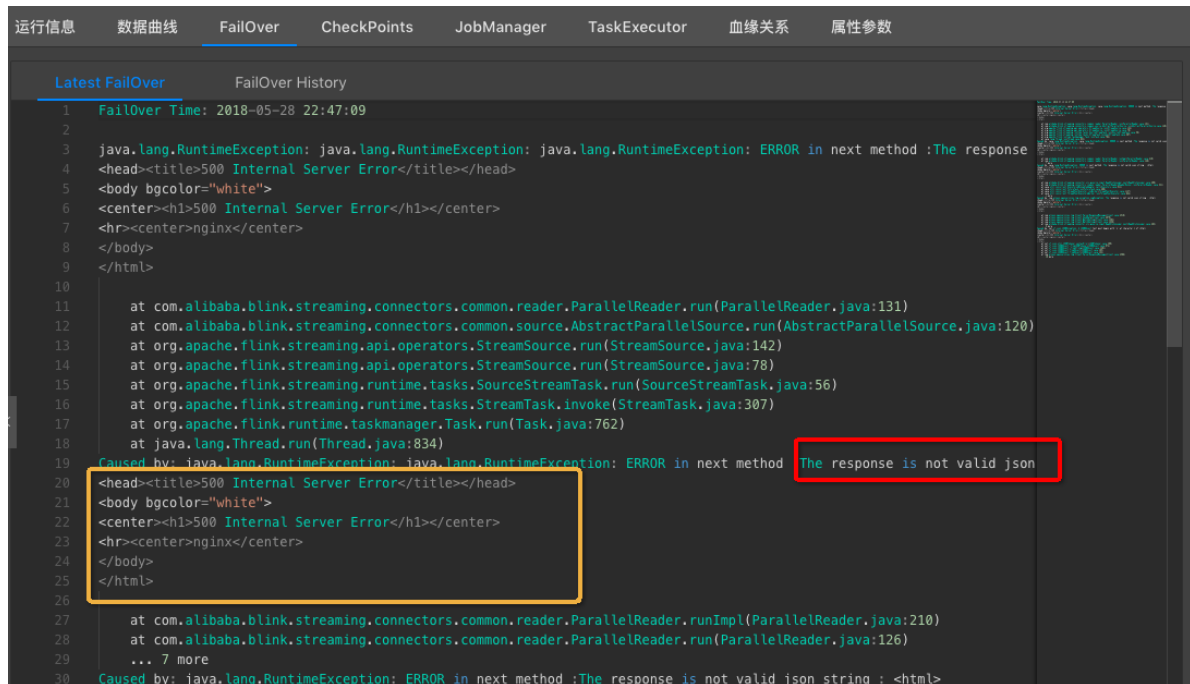
使用计算列过滤NULL值。通常Watermark里的Event time时间字段里存在NULL值。

```
ts as case when `datetime` is null
then timestamp '1970-01-01 00:00:00'
else `datetime` end
```

## 报错：ERROR in next method The response is not valid json

- 报错信息





- 报错原因

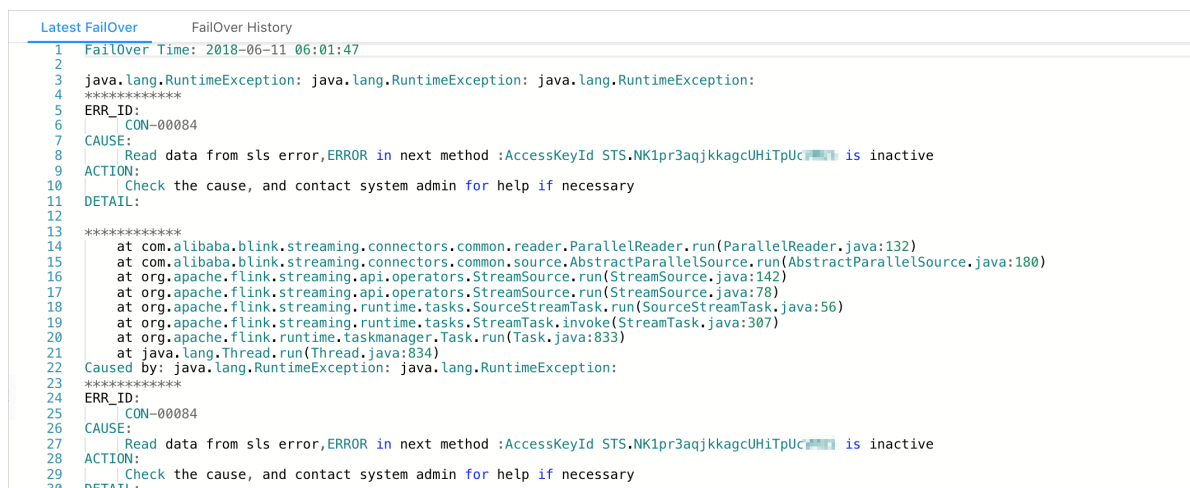
SLS发生了网络震荡，导致连接实时计算失败。

- 解决方案

重新启动作业。如果还是报错，请您[提交工单](#)。

## 报错：ERROR in next method AccessKeyId STS.NK1p3aqjkkagcUHiTpC\*\*\*\* is inactive

- 报错信息



- 报错原因

SLS的Token过期。

- 解决方案

设置参数stsUpdateSeconds='3600'。

## 报错：open catalog connection failed

- 报错信息

```
code:[100004], brief info:[open catalog connection failed], context info:[details:
[properties:{"endpoint":"???", "name":"blink", "type":"default"}.exception:Failed to create H
ive Metastore client]]
```

- 报错原因

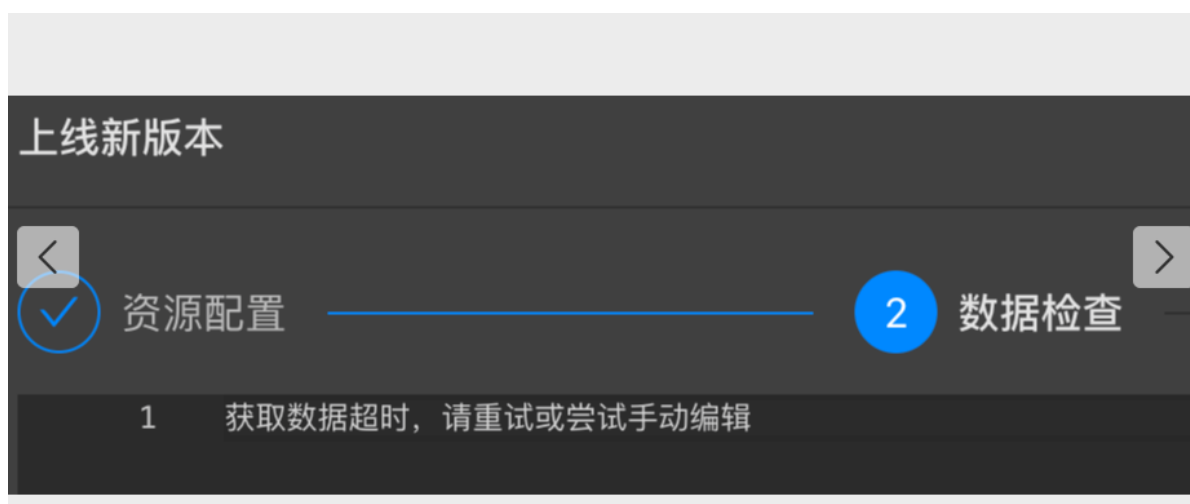
DDL或DML语句中，表名或字段大小写不一致。

- 解决方案

建议您仔细检查并修改表名或字段的大小写。

## 报错：获取数据超时，请重试或尝试手动编辑

- 报错信息



- 报错原因

- 自定义表的参数传入失败。
- 由于Case When判断过多，需要加长编译时间。

- 解决方案

- 检查自定义表的参数是否传入成功。
- 在作业参数中配置blink.job.submit.timeoutInSeconds参数（默认为30s，可以适当增加，但不能超过90s）。

## 警告：WARNING（不影响当前流程）

- 警告详情

- 详情1

```
1 WARNING(不影响当前流程):
2 Autoconf cannot scale up parallelism of source (vertexId: 0) to 30, because it is
 constrained to the number of source partitions 29.
3 com.alibaba.blink.launcher.metrics.MetricsException: Error fetching metrics for job
 cdnlog.alicacheolog with time between Wed Nov 21 10:35:18 CST 2018 and Wed Nov 21
 10:38:18 CST 2018. Previous job statistics and configuration will be used.
```



- 详情2

```
1 WARNING:
2 Configuration is not usable. Generate default configuration. Cause:
The stream nodes in the configuration [1, 2, 3, 4, 5, 6, 7, 8, 10, 11,
14, 15, 16, 17, 18, 19, 21, 22, 25, 26, 27, 28], do not match those in
stream graph [1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 14, 15, 16, 17, 18, 19].
The could be caused by updating the application without updating the
configuration, or incorrect modification to the configuration. Please
check the configuration and try again
```

- 警告原因

- 详情1

- 因为上游Shard的限制，所以并发不能扩展到30。
    - 因为取不到上一次对应时间点的指标曲线，所以使用了上一次的资源配置，即本次调优实际并没有生效。

- 详情2

可能是在不更新配置的情况下更新了应用程序，或者对配置的修改不正确，导致与作业资源配置不匹配。

- 解决方案

- 详情1

不影响作业运行，可以直接上线运行该作业。

- 详情2

重新获取配置信息后，反复执行上线 > 调优。

## 6.4. 启动报错

本文汇总了实时计算启动常见报错与解决方案。

- 报错：No such edge in stream graph:1 -&g
- 报错：Slot allocation request timed out for
- 报错：Submit blink job failed
- 报错：CLI-00000001
- 报错：NoClassDefFoundError或NoSuchMethodError
- 报错：cp path NOT exist or is empty!

### 报错：No such edge in stream graph:1 -&g

- 报错信息

从Blink2.x版本升级到Blink3.x版本时，启动报错。

```
The program finished with the following exception:

org.apache.flink.client.program.ProgramInvocationException: The main method caused an error.
 at org.apache.flink.client.program.PackagedProgram.callMainMethod(PackagedProgram.java:607)
 at org.apache.flink.client.program.PackagedProgram.invokeInteractiveModeForExecution(PackagedProgram.java:466)
 at org.apache.flink.client.program.OptimizerPlanEnvironment.getOptimizedPlan(OptimizerPlanEnvironment.java:94)
 at org.apache.flink.client.program.PackagedProgramUtils.createJobGraph(PackagedProgramUtils.java:76)
 at org.apache.flink.client.cli.CliFrontend.runProgram(CliFrontend.java:239)
 at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:216)
 at org.apache.flink.client.cli.CliFrontend.parseParameters(CliFrontend.java:1055)
 at org.apache.flink.client.cli.CliFrontend.lambda$main$9(CliFrontend.java:1131)
 at java.security.AccessController.doPrivileged(Native Method)
 at javax.security.auth.Subject.doAs(Subject.java:422)
 at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1682)
 at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41)
 at org.apache.flink.client.cli.CliFrontend.main(CliFrontend.java:1131)
Caused by: java.lang.RuntimeException: No such edge in stream graph: 1 -> 2
 at org.apache.flink.streaming.api.graph.StreamGraph.getStreamEdges(StreamGraph.java:731)
 at com.alibaba.blink.launcher.autoconfig.StreamGraphConfigurer.configure(StreamGraphConfigurer.java:59)
 at com.alibaba.blink.launcher.JobLauncher.execute(JobLauncher.java:535)
 at com.alibaba.blink.launcher.JobLauncher.runStream(JobLauncher.java:469)
 at com.alibaba.blink.launcher.JobLauncher.main(JobLauncher.java:242)
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.lang.reflect.Method.invoke(Method.java:498)
 at org.apache.flink.client.program.PackagedProgram.callMainMethod(PackagedProgram.java:590)
 ... 12 more
```

- 报错原因

Blink2.x和Blink3.x的PLAN JSON不兼容。

- 解决方案

您需要重新获取配置资源后，重新上线并启动作业。详情请参见[上线](#)和[启动](#)。

## 报错：Slot allocation request timed out for

- 报错信息

```
Message

java.util.concurrent.TimeoutException: Slot allocation request timed out for ad466a9c8f1a82b48e9d81df3476cb7e
 at org.apache.flink.runtime.instance.SlotPool.checkTimeoutSlotAllocation(SlotPool.java:460)
 at org.apache.flink.runtime.instance.SlotPool.access$500(SlotPool.java:91)
 at org.apache.flink.runtime.instance.SlotPool$3.run(SlotPool.java:442)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleRunAsync(AkkaRpcActor.java:273)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleMessage(AkkaRpcActor.java:131)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.access$000(AkkaRpcActor.java:72)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor$1.apply(AkkaRpcActor.java:110)
 at akka.actor.ActorCell$$anonfun$become$1.applyOrElse(ActorCell.scala:534)
 at akka.actor.Actor$class.aroundReceive(Actor.scala:467)
 at akka.actor.UntypedActor.aroundReceive(UntypedActor.scala:97)
 at akka.actor.ActorCell.receiveMessage(ActorCell.scala:516)
 at akka.actor.ActorCell.invoke(ActorCell.scala:487)
 at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:238)
 at akka.dispatch.Mailbox.run(Mailbox.scala:220)
 at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:397)
 at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)
 at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)
 at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)
 at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)
```

- 报错原因

在作业CU不足时，无法分配资源，导致请求超时。

- 解决方案

重新设置作业的CU值。如果您项目中的CU不足，请在项目管理中对项目进行扩容，详情请参见[变配指导](#)和[续费指导](#)。

## 报错：Submit blink job failed

- 报错信息

数据曲线 FailOver CheckPoints JobManager TaskExecutor Configuration 血缘关系 代码配置

```
run instance failed, 操作错误: 系统处理BLINK作业异常。
错误原因: Submit blink job failed, name: user_day_activity_by_namespace, code:30010, msg:code:[30010], brief info:[error occur while
app], context info:[details:[任务提交超时, 请检查详细日志输出. (submit app exceed time, maybe consider to tune up submit timeout)
java.lang.Exception:
2018-01-04 17:09:04,241 INFO org.apache.flink.yarn.cli.FlinkYarnCLI - No path for the flink jar passed. Us
the location of class org.apache.flink.yarn.YarnClusterDescriptorV2 to locate the jar
2018-01-04 17:09:04,294 INFO org.apache.hadoop.yarn.client.RMPProxy - Connecting to ResourceManager at
b21d11599.cloud.et2.tbsite.net/11.219.228.171:8050
Cluster started: Yarn cluster with application id null
Using address localhost:0 to connect to JobManager.
JobManager web interface address http://
Using the parallelism provided by the remote cluster (0). To use another parallelism, set it at the ./bin/flink client.
Starting execution of program
2018-01-04 17:09:04,422 INFO org.apache.flink.yarn.YarnClusterClientV2 - Starting program in interactive mode
[streamEnv] update k=blink.checkpoint.external.path,
v=hdfs://et2cloudprod/blink_rest_server/root.syncstore/user_day_activity_by_namespace/external_cp
[streamEnv] update k=state.backend.rocksdbs.ttl.ms, v=129600000
[streamEnv] update k=state.backend.rocksdbs.checkpointpath,
v=hdfs://et2cloudprod/blink_rest_server/root.syncstore/user_day_activity_by_namespace/rocksdbs_cp
[streamEnv] update k=stsAccesskey, v=v91V042KRCL7mFEYfLP02INwSv71m0Jz18RUsNbRzFM=
[streamEnv] update k=stsRegionId, v=cn-shanghai
[streamEnv] update k=__inner__rg_sample_data_conf_accessKey__, v=prc1bmIam4hRyPQ9Q66oPZd9gidXDY
[streamEnv] update k=stsAccessId, v=9juusi0kMF6pRs8c
[streamEnv] update k=__inner__projectName__, v=syncstore
[streamEnv] update k=blink.checkpoint.mode, v=EXACTLY_ONCE
[streamEnv] update k=blink.checkpoint.external.cleanup, v=RETAIN_ON_CANCELLATION
```

- 报错原因

修改了SQL代码，但未重新生成JSON资源配置。

- 解决方案

修改SQL代码后，需要重新生成JSON配置。详情请参见[上线](#)。

 说明 如果只修改了资源参数，则不需要重新生成JSON配置。

## 报错：CLI-00000001

- 报错信息

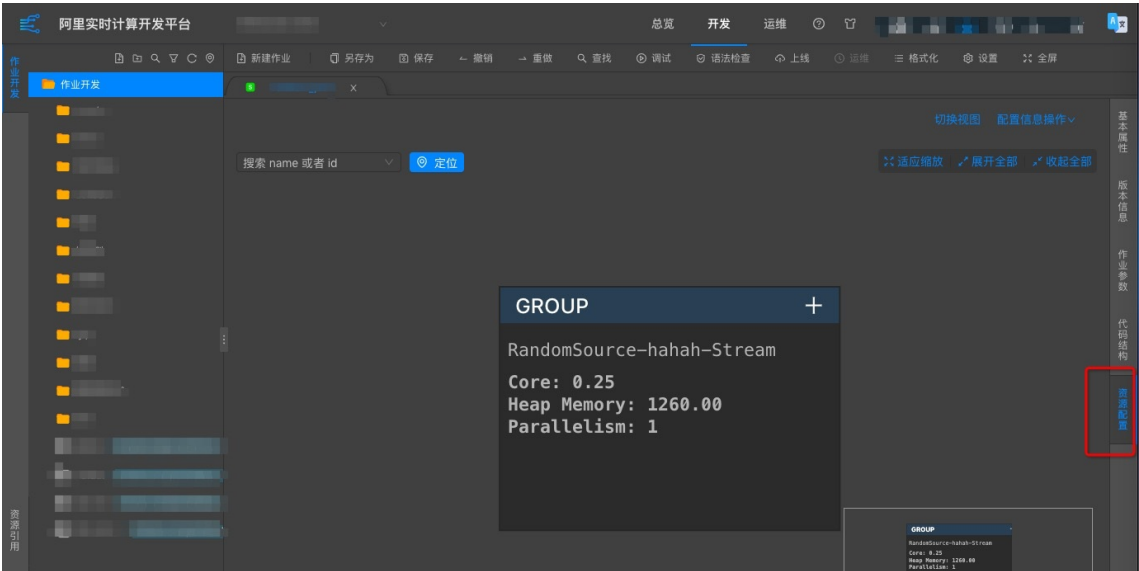
```
ERR_ID
CLI-00000001
CAUSE:
AutoConf:an error occurred while applying the resource configuration file.
Node or edge number in stream graph and json file not matched,please generate a new jso
n.
ACTION:
The reason may be that the content of the SQL has been modified or the blink version nu
mber has been changed.
Please regenerate the resource configuration file.
DETAIL:
```

```
Caused by: com.alibaba.blink.launcher.autoconfig.UnexpectedConfigurationException:

ERP_ID:
 CLI-00000001
CAUSE:
 AutoConf: an error occurred while applying the resource configuration file.
 Node or edge number in stream graph and json file not matched, please generate a new json.
ACTION:
 The reason may be that the content of the SQL has been modified or the blink version number has been changed.
 Please regenerate the resource configuration file.
DETAIL:

 at com.alibaba.blink.launcher.autoconfig.StreamGraphConfigurer.configure(StreamGraphConfigurer.java:44)
 at com.alibaba.blink.launcher.JobLauncher.execute(JobLauncher.java:547)
 at com.alibaba.blink.launcher.JobLauncher.runStream(JobLauncher.java:481)
 at com.alibaba.blink.launcher.JobLauncher.main(JobLauncher.java:254)
```

- 报错原因  
更改SQL版本后，导致新SQL的PLAN JSON和资源配置不兼容。
- 解决方案
  - i. 在实时计算控制台的开发页面，单击右侧的资源配置。



- ii. 单击资源配置右上角切换视图，进入配置信息页面。
  - iii. 删除所有JSON信息。

```
1 {
2 "parameters": {
3 "blink.target_resource_units": 5,
4 "blink.metrics.job.name": " ",
5 "blink.metrics.start.time.ts": 1566814528851,
6 "blink.metrics.end.time.ts": 1566818128851,
7 "blink.metrics.start.time": "2019-08-26T18:15:28+08:00",
8 "blink.metrics.end.time": "2019-08-26T19:15:28+08:00"
9 },
10 "resourceSettings": {
11 "blink.resource.unit.per.worker": 1,
12 "blink.resource.unit.granularity": 0.1,
13 "blink.metrics.endpoint": "http://100.118.128.166:9000 ",
14 "blink.resource.allocation.jm.min.cpu.cores": 0.2,
15 "blink.resource.allocation.jm.min.memory.mb": 512
16 },
17 "autoConfig": {
18 "goal": {
19 "maxResourceUnits": 10000,
20 "targetResourceUnits": 5
21 },
22 "result": {
23 "attemptId": 4,
24 "attempts": 4,
25 "attemptsCurrentPlan": 4,
26 "statusMessage": "Unable to get running metrics, continue to use last configuration.",
27 "scalingAction": "ScaleToTargetResource",
28 "allocatedResourceUnits": 0.7000000000000001,
29 "allocatedCpuCores": 0.7000000000000001,
30 "allocatedMemoryInMB": 2866,
31 "messages": [
32 "com.alibaba.blink.launcher.metrics.MetricsException: No metric result for . - 7:40:00 CST 2019-Mon Aug 26 19:15:20 CST 2019/parserTps.rate/AVG/ container_type=0/false}. Previous job statistics and configuration will be used."
33],
34 "history": {
35 "1": {
36 "attemptId": 1,
37 "statusMessage": "New job, using default configuration."
38 },
39 "2": {
40 "attemptId": 2,
41 "statusMessage": "Unable to get running metrics, continue to use last configuration."
42 }
43 }
44 }
45 }
46 }
```

- iv. 在资源配置右上角，单击配置新增操作 > 重新获取配置信息。
- v. 在资源配置右上角，单击配置新增操作 > 应用当前配置。
- vi. 重新上线作业。



② 说明 在重新上线时，请选择手动资源配置。

报错：NoClassDefFoundError或NoSuchMethodError

• 报错信息

```
java.lang.NoSuchMethodError
java.lang.NoClassDefFoundError
```

• 报错原因

依赖包冲突。

- 解决方案

请参见[概述](#)和[Datastream开发](#)中的POM文件，配置依赖。

② 说明

- POM依赖的版本需要和开发页面右下角的版本保持一致。
- Blink相关的依赖请使用 `<scope>provided</scope>` 的方式进行打包。
- 自定义的依赖请使用Shade的方式进行打包，避免与Blink底层依赖发生冲突。

## 报错：cp path NOT exist or is empty!

- 报错信息

作业暂停恢复时，报错如下。

```
Submit job for the [1] times,last error occurred at [Tue Nov 10 00:00:00 CST 2020],lastErrorMessage is as follows:
run instance failed, BLINK error: bayes submit job failed.job name:[random_print], brs errorCode:[30026], brs errorMessage:
code:[30026], brief info:[hdfs://a****/]ification failed, context info:[details:cp path NOT exist or is empty!]
```

- 报错原因

在作业Checkpoint未成功时，进行暂停恢复操作，导致作业恢复时，无法查询到对应的HDFS路径。

- 解决方案

您需要停止再启动作业，详情请参见[停止](#)和[启动](#)。

## 6.5. 运行Failover

本文汇总了实时计算Flink版运行Failover常见报错与解决方案。

- 报错：java.lang.ArrayIndexOutOfBoundsException -1
- 报错：java.util.concurrent.TimeoutException
- 报错：the message body size over max value
- 报错：Caused by: org.codehaus.janino.JaninoRuntimeException: Code of method...64KB
- 报错：Caused by: java.net.SocketException: Connection reset
- 报错：Seek out of range
- 报错：pool.GetConnectionTimeoutException
- 报错：the last packet sent successfully to the server was milliseconds ago
- 报错：Incorrect String value:'\xF0\x9F...'for column
- 报错：Some executions status are not reported and failing the job now
- 报错：The heartbeat of TaskManager with id container...timed out
- 报错：cause by:java.lang.IndexOutOfBoundsException
- 报错：No pooled slot available and request to ResourceManager for new slot failed
- 报错：Ots attribute key type error, table ots\_sink column key uv
- 报错：OutOfMemory
- 报错：OutOfMemoryError: Java heap space



- 报错：java.lang.StackOverflowError
- 报错：Suspend the old job to update it
- 报错：MySQLSyntaxErrorException: Unknown column
- 报错：TaskManager lost/killed
- 报错：The sts token has been expired
- 报错：OTSParameterInvalid, Message: Invalid update row request: missing cells in request
- 报错：There is an invalid character in tagvalue
- 报错：Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLTransactionRollbackException: Deadlock found when trying to get lock; try restarting transaction
- 报错：Value '0000-00-00 00:00:00' can not be represented as java.sql.Timestamp
- 报错：Failover:offset (0) + length (8) exceed the capacity of the array: 1
- 作业启动一段时间后报错停止，Failover里出现报错信息，应该如何处理？

## 报错：java.lang.ArrayIndexOutOfBoundsException -1

- 报错信息

```

39 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:593)
40 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:593)
41 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:593)
42 at org.apache.flink.streaming.api.operators.TimestampedCollector.collect(TimestampedCollector.java:51)
43 at org.apache.flink.table.runtime.collector.HeaderCollector.collect(HeaderCollector.scala:41)
44 at org.apache.flink.table.runtime.collector.HeaderCollector.collect(HeaderCollector.scala:28)
45 at DataStreamSourceConversion$42.processElement(Unknown Source)
46 at org.apache.flink.table.runtime.conversion.ToBaseRowProcessRunner.processElement(ToBaseRowProcessRunner.scala:71)
47 at org.apache.flink.streaming.api.operators.ProcessOperator.processElement(ProcessOperator.java:61)
48 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
49 ... 26 more
50 Caused by: java.lang.RuntimeException: Could not forward element to next operator
51 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:611)
52 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:593)
53 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:593)
54 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:593)
55 at org.apache.flink.streaming.api.operators.StreamMap.processElement(StreamMap.java:38)
56 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
57 ... 36 more
58 Caused by: java.lang.ArrayIndexOutOfBoundsException: -1
59 at org.apache.flink.types.Row.getField(Row.java:76)
60 at com.alibaba.blink.connectors.ads.AdsOutputFormat.constructPKFilterKey(AdsOutputFormat.java:494)
61 at com.alibaba.blink.connectors.ads.AdsOutputFormat.writeAddRecord(AdsOutputFormat.java:506)
62 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:506)
63 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:506)
64 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:38)
65 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:38)
66 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:38)
67 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
68 ... 41 more

```

- 报错原因

实时计算Flink版引用的外部数据源主键字段大小写敏感。

- 解决方案

修改结果表的主键字段大小写，和外部数据源表保持一致。例如，如果分析型数据库MySQL版结果表的字段是大写的，则在实时计算Flink版的DDL语句中，结果表的主键字段也必须是大写的。

## 报错：java.util.concurrent.TimeoutException

- 报错信息

AutoScale启动时，报错如下。



```
java.util.concurrent.TimeoutException
 at org.apache.flink.runtime.concurrent.FutureUtils$Timeout.run(FutureUtils.java:834)
 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
 at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(ScheduledThreadPoolExecutor.java:186)
 at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:299)
 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1147)
 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)
 at java.lang.Thread.run(Thread.java:834)
```

- 报错原因

作业上线时指定的CU数，不足以支持AutoScale进行迭代。

- 解决方案

在作业开启上线时，如果开启AutoScale，则提高到最大CU数。详情请参见[AutoScale自动配置调试](#)。

## 报错：the message body size over max value

- 报错信息

```
ERR_ID:
| CON-00066
CAUSE:
| Can not write to metaq/mq, retry attempts 3, errMsg Retrying cost [4002] ms, exceed timeout threshold [3] *
| [1000] ms. Error Message: [the message body size over max value, MAX: 4194304]
ACTION:
| Check the cause, and contact system admin for help if necessary
DETAIL:

| at com.alibaba.blink.streaming.connectors.common.exception.ErrorUtils.getException(ErrorUtils.java:39)
| at com.alibaba.blink.streaming.connectors.common.exception.ErrorUtils.throwException(ErrorUtils.java:53)
| at com.alibaba.blink.streaming.connector.metaq.MetaQOutputFormat.writeAddRecord(MetaQOutputFormat.java:193)
| at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord
| (TupleRichOutputFormat.java:50)
| at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord
| (TupleRichOutputFormat.java:34)
| at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke
| (TupleOutputFormatAdapterSink.java:71)
| at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke
| (TupleOutputFormatAdapterSink.java:34)
```

- 错误原因

在数据写入MQ结果表时，MQ消息超过了Body的长度限制。详情请参见[消息队列RocketMQ版消息Body的长度限制是多少？](#)。

- 解决方案

调整MQ消息写入Body的长度，限制在Body长度范围内。

## 报错：Caused by: org.codehaus.janino.JaninoRuntimeException: Code of method...64KB

- 报错信息

```
Caused by: java.lang.RuntimeException: Compiling "StreamExecCalcRule$6109": Code of method
"calcProjectApply_1$(LStreamExecCalcRule$6109;Lorg/apache/flink/table/dataformat/BaseRow;)V
" of class "StreamExecCalcRule$6109" grows beyond 64 KB
 at org.codehaus.janino.UnitCompiler.compileUnit(UnitCompiler.java:361)
 at org.codehaus.janino.SimpleCompiler.cook(SimpleCompiler.java:234)
 at org.codehaus.janino.SimpleCompiler.compileToClassLoader(SimpleCompiler.java:446)
 at org.codehaus.janino.SimpleCompiler.cook(SimpleCompiler.java:213)
 at org.codehaus.janino.SimpleCompiler.cook(SimpleCompiler.java:204)
 at org.codehaus.commons.compiler.Cookable.cook(Cookable.java:80)
 at org.codehaus.commons.compiler.Cookable.cook(Cookable.java:75)
 at org.apache.flink.table.codegen.CodeGenUtils$.compile(CodeGenUtils.scala:997)
 ... 18 more
Caused by: org.codehaus.janino.JaninoRuntimeException: Code of method "calcProjectApply_1$(
LStreamExecCalcRule$6109;Lorg/apache/flink/table/dataformat/BaseRow;)V" of class "StreamExe
cCalcRule$6109" grows beyond 64 KB
 at org.codehaus.janino.CodeContext.makeSpace(CodeContext.java:974)
 at org.codehaus.janino.CodeContext.write(CodeContext.java:884)
 ...
```

- 报错原因

JVM对某个方法的字节编码长度限制为64KB。而此作业中，单个字段的字节编码长度就已超过了64KB。

- 解决方案

使用[自定义标量函数（UDF）](#)方法修改此作业。

## 报错：Caused by: java.net.SocketException: Connection reset

- 报错信息

```
The last packet successfully received from the server was 3,991 milliseconds ago. The last
packet sent successfully to the server was 3,992 milliseconds ago.
 at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
 at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.ja
va:62)
 at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccesso
rImpl.java:45)
 at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
 at com.mysql.jdbc.Util.handleNewInstance(Util.java:425)
 at com.mysql.jdbc.SQLException.createCommunicationsException(SQLException.java:989)
 at com.mysql.jdbc.MysqlIO.send(MysqlIO.java:3743)
 at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:2506)
 at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2677)
 at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2545)
 at com.mysql.jdbc.ConnectionImpl.setAutoCommit(ConnectionImpl.java:4842)
 at com.alibaba.druid.pool.DruidPooledConnection.setAutoCommit(DruidPooledConnection.java:
709)
 ... 6 more
Caused by: java.net.SocketException: Connection reset
 at java.net.SocketOutputStream.socketWrite(SocketOutputStream.java:113)
 at java.net.SocketOutputStream.write(SocketOutputStream.java:153)
 at java.io.BufferedOutputStream.flushBuffer(BufferedOutputStream.java:82)
 at java.io.BufferedOutputStream.flush(BufferedOutputStream.java:140)
 at com.mysql.jdbc.MysqlIO.send(MysqlIO.java:3725)
 ... 11 more
```

- 报错原因

链接超时导致了Failover。

- 解决方案

- i. 检查RDS是否为高可用模式，如果不是，请切换至高可用模式。
- ii. With参数中的 URL 中设置 `autoReconnect=true`，如下图所示。

```
'jdbc:mysql://localhost:3306/test?useUnicode=true&characterEncoding=gbk&autoReconnect=true'
```

如果有多个参数，请用and（&）符号进行连接。

```
type='rds',
url='jdbc:mysql://localhost:3306/test?useUnicode=true&characterEncoding=gbk&autoReconnect=true'
```

❓ 说明 如果按照以上方案操作后，还是报错，请您[提交工单](#)。

## 报错：Seek out of range

- 报错信息

```
page (2017-12-22 09:53:23)
lang.RuntimeException: java.lang.RuntimeException: com.aliyun.datahub.exception.InvalidParameterException: wifiana_test/group_rssi_data_part_tenminutes/0: Seek Out of Range (Status
at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:131)
at com.alibaba.blink.streaming.connectors.common.source.AbstractParallelSource.run(AbstractParallelSource.java:107)
at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:142)
at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:78)
at org.apache.flink.streaming.runtime.tasks.SourceStreamTask.run(SourceStreamTask.java:56)
at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:307)
at org.apache.flink.runtime.taskmanager.Task.run(Task.java:762)
at java.lang.Thread.run(Thread.java:834)
ed by: java.lang.RuntimeException: com.aliyun.datahub.exception.InvalidParameterException: wifiana_test/group_rssi_data_part_tenminutes/0: Seek Out of Range (Status Code: 400; Error
at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.runImpl(ParallelReader.java:210)
at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:126)
... 7 more
ed by: com.aliyun.datahub.exception.InvalidParameterException: wifiana_test/group_rssi_data_part_tenminutes/0: Seek Out of Range Status Code: 400; Error Code: InvalidParameter; Re
at com.aliyun.datahub.model.serialize.JsonErrorParser.parse(JsonErrorParser.java:31)
at com.aliyun.datahub.model.serialize.GetCursorResultJsonDeser.deserialize(GetCursorResultJsonDeser.java:19)
at com.aliyun.datahub.model.serialize.GetCursorResultJsonDeser.deserialize(GetCursorResultJsonDeser.java:14)
at com.aliyun.datahub.DatahubClient.getCursor(DatahubClient.java:627)
at com.aliyun.datahub.DatahubClient.getCursor(DatahubClient.java:699)
at com.alibaba.blink.streaming.connectors.datahub.input.RawDatahubRecordReader.seek(RawDatahubRecordReader.java:332)
at com.alibaba.blink.streaming.connectors.datahub.input.RawDatahubRecordReader.seek(RawDatahubRecordReader.java:51)
at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader$ReaderRunner.run(ParallelReader.java:336)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1147)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)
1 more
```

- 原因分析

- 实时计算Flink版处理数据的速率小于数据进入DataHub的速率，会造成数据堆积，延时增加。当延时大于DataHub生命周期时，实时计算Flink版读取的数据已经过期，最终导致系统报错。
- 您的本地设备系统时间大于DataHub所在机器的时间。实时计算Flink版作业启动位点默认时间为您本地设备的系统时间，如果启动位点在DataHub机器时间之后，则会导致 `Seek out of Range` 的报错。

- 解决方案

- 对实时计算Flink版的性能进行调优，使实时计算Flink版处理数据的速率大于等于数据进入DataHub的速率。调优方法请参见：
  - [高性能Flink SQL优化技巧](#)
  - [AutoConf自动配置调优](#)
  - [AutoScale自动配置调优](#)
  - [手动配置调优](#)

- 调整用户本地设备系统时间，使其和DataHub所在机器时间保持一致。

## 报错：pool.GetConnectionTimeoutException

### • 报错信息

```
Message (2018-03-16 11:54:19)

java.lang.RuntimeException: Cannot create connection to RDS.
 at com.alibaba.blink.connectors.rds.dim.RdsRowFetcher.connectToTable(RdsRowFetcher.java:67)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.reconnectToTable(MySqlRowFetcherBase.java:147)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.open(MySqlRowFetcherBase.java:133)
 at org.apache.flink.api.common.functions.util.FunctionUtils.openFunction(FunctionUtils.java:36)
 at com.alibaba.blink.table.runtime.JoinTableProcessRunner.open(JoinTableProcessRunner.scala:57)
 at com.alibaba.blink.table.runtime.JoinTableWithCalcProcessRunner.open(JoinTableWithCalcProcessRunner.scala:46)
 at org.apache.flink.api.common.functions.util.FunctionUtils.openFunction(FunctionUtils.java:36)
 at org.apache.flink.streaming.api.operators.AbstractUdfStreamOperator.open(AbstractUdfStreamOperator.java:156)
 at org.apache.flink.streaming.api.operators.ProcessOperator.open(ProcessOperator.java:51)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.openAllOperators(StreamTask.java:445)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:297)
 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:762)
 at java.lang.Thread.run(Thread.java:834)
Caused by: com.alibaba.druid.pool.GetConnectionTimeoutException: wait millis 16429, active 30, maxActive 30, creating 0
 at com.alibaba.druid.pool.DruidDataSource.getConnectionInternal(DruidDataSource.java:1329)
 at com.alibaba.druid.pool.DruidDataSource.getConnectionDirect(DruidDataSource.java:1148)
 at com.alibaba.druid.pool.DruidDataSource.getConnection(DruidDataSource.java:1128)
 at com.alibaba.druid.pool.DruidDataSource.getConnection(DruidDataSource.java:1118)
 at com.alibaba.blink.connectors.rds.dim.RdsRowFetcher.connectToTable(RdsRowFetcher.java:63)
 ... 12 more
```

### • 报错原因

Pool的链接池最大值为30个，而您的链接数大于30，导致数据写入失败。

### • 解决方案

调大维表或流表JOIN节点的Core参数。例如，Core的默认值是0.1，可以调成0.25或1。详情请参见[手动配置调优](#)。

## 报错：the last packet sent successfully to the server was milliseconds ago

### • 报错信息

- the last packet successfully received from the server was milliseconds ago
- the last packet sent successfully to the server was milliseconds ago

### • 报错原因

如果RDS的 `net_read_timeout` 和 `net_write_timeout` 参数设置过小，在网络条件较差或客户端处理Block耗时较长时，会导致RDS连接实时计算Flink版提示连接超时的报错。

| 参数                                                             | 作用                                                               |
|----------------------------------------------------------------|------------------------------------------------------------------|
| <code>net_read_timeout</code> 和 <code>net_write_timeout</code> | 服务器端等待客户端发送的网络包和发送给客户端网络包的超时时间，该参数仅对TCP/IP链接并且是Activity状态下的线程有效。 |

### • 解决方案

调整 `net_read_timeout` 参数和 `net_write_timeout` 参数值，操作步骤如下：

- 登录RDS管理控制台。
- 在页面左上角，选择实例所在区域。
- 单击目标实例ID。
- 在左侧导航栏中单击参数设置。
- 在可修改参数页签，查找 `net_read_timeout` 和 `net_write_timeout` 。

 说明 可以使用ctrl+f或command+f的组合键在页面中搜索您需要修改的参数。

- 单击 `net_read_timeout` 和 `net_write_timeout` 后的。
- 输入目标值，并单击确定。
- 单击可修改参数页签右上角的提交参数。
- 在弹出的对话框中，单击确定，完成参数修改。

## 报错：Incorrect String value: '\xF0\x9F...' for column

### • 报错信息

```
系统错误：内部错误：在批量写入时发生错误 sql template sql:INSERT INTO
at com.alibaba.blink.connectors.rds.RdsExceptionUtil.getBlinkException(RdsExceptionUtil.java:509)
at com.alibaba.blink.connectors.rds.RdsOutputFormat.execBatchAdd(RdsOutputFormat.java:509)
at com.alibaba.blink.connectors.rds.RdsOutputFormat.batchWrite(RdsOutputFormat.java:447)
at com.alibaba.blink.connectors.rds.RdsOutputFormat.sync(RdsOutputFormat.java:399)
at com.alibaba.blink.connectors.rds.RdsOutputFormat$.run(RdsOutputFormat.java:204)
at java.util.TimerThread.mainLoop(Timer.java:555)
at java.util.TimerThread.run(Timer.java:505)
Caused by: java.sql.BatchUpdateException: Incorrect string value: '\xF0\x9F\x92\xA4' for column 'c' at row 42
at sun.reflect.GeneratedConstructorAccessor14.newInstance(Unknown Source)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
at com.mysql.jdbc.Util.handleNewInstance(Util.java:425)
at com.mysql.jdbc.Util.getInstance(Util.java:408)
at com.mysql.jdbc.SQLException.createBatchUpdateException(SQLException.java:1162)
at com.mysql.jdbc.PreparedStatement.executeBatchInternal(PreparedStatement.java:1582)
at com.mysql.jdbc.PreparedStatement.executeBatch(PreparedStatement.java:1248)
at com.mysql.jdbc.StatementImpl.executeBatch(StatementImpl.java:958)
at com.alibaba.druid.pool.DruidPooledPreparedStatement.executeBatch(DruidPooledPreparedStatement.java:559)
at com.alibaba.blink.connectors.rds.RdsOutputFormat.execBatchAdd(RdsOutputFormat.java:482)
... 5 more
Caused by: java.sql.SQLException: Incorrect string value: '\xF0\x9F\x92\xA4' for column 'c' at row 42
at com.mysql.jdbc.SQLException.createBatchUpdateException(SQLException.java:964)
at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:3970)
at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:3906)
at com.mysql.jdbc.MySQLIO.sendCommand(MySQLIO.java:2524)
at com.mysql.jdbc.MySQLIO.sqlQueryDirect(MySQLIO.java:2677)
at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2549)
at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1861)
at com.mysql.jdbc.PreparedStatement.executeUpdateInternal(PreparedStatement.java:2073)
at com.mysql.jdbc.PreparedStatement.executeUpdateInternal(PreparedStatement.java:2009)
at com.mysql.jdbc.PreparedStatement.executeLargeUpdate(PreparedStatement.java:5098)
at com.mysql.jdbc.PreparedStatement.executeBatchInserts(PreparedStatement.java:1543)
... 9 more
```

### • 错误原因

数据中存在Emoji表情，数据库编码不能解析。

### • 解决方案

您需要在RDS控制台对应实例的参数设置中，修改character\_set\_server参数值为utf8mb4。此外，数据库和表的字符集也要同样设置为utf8mb4。

 说明

- character\_set\_server参数修改后，需要重启实例才能生效。
- 如果需要将\F0\x9F...字符串存入数据库，则将反斜线（\）替换为双反斜线（\\）。

## 报错：Some executions status are not reported and failing the job now

### • 报错信息



Some executions status are not reported and failing the job now.

- 错误原因

因为作业没有启动成功，所以没有记录启动时间。默认启动时间为UNIX起始时间 1970-01-01 08:00:00。

- 解决方案

请重新上线并启动作业。

## 报错：The heartbeat of TaskManager with id container...timed out

- 报错信息

```

Latest FailOver FailOver History
1 FailOver Time: 2019-01-23 16:31:03
2
3 java.util.concurrent.TimeoutException: Heartbeat of TaskManager with id container_e08_1548199152109_0107_01_000018
 timed out.
4 at org.apache.flink.runtime.jobmaster.JobMaster$TaskManagerHeartbeatListener.notifyHeartbeatTimeout
 (JobMaster.java:1545)
5 at org.apache.flink.runtime.heartbeat.HeartbeatManagerImpl$HeartbeatMonitor.run(HeartbeatManagerImpl.java:358)
6 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
7 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
8 at org.apache.flink.runtime.concurrent.akka.ActorSystemScheduledExecutorAdapter$ScheduledFutureTask.run
 (ActorSystemScheduledExecutorAdapter.java:154)
9 at akka.dispatch.TaskInvocation.run(AbstractDispatcher.scala:39)
10 at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:415)
11 at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)
12 at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)
13 at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)
14 at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)
15

```

- 报错原因

- 分布式物理机网络失联。
- Failover的节点对应Taskmanager的内存设置太小，垃圾回收机制会导致心跳超时。

- 解决方案

- 如果此问题出现的不频繁，则不用关注，通常Failover后作业能正常恢复。
- 调大Failover的节点对应Taskmanager的内存值。

## 报错：cause by:java.lang.IndexOutOfBoundsException

- 报错信息

引用MaxCompute维表时，报错如下。

```

22 at org.apache.flink.streaming.api.operators.StreamMap.processElement(StreamMap.java:38)
23 at org.apache.flink.streaming.runtime.tasks.OperatorChainCopyingChainingOutput.collect(OperatorChain.java:608)
24 ... 12 more
25 Caused by: java.lang.RuntimeException: com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException: 821109
26
27 系统错误-内部错误：在批量写入时发生错误 sql template sql:INSERT INTO `rds_dim` (`d`,`e`,`f`) values (?,?,?) ON DUPLICATE KEY UPDATE e= values(e),f= values(f),preparedStatement
 at com.alibaba.blink.connectors.rds.RdsOutputFormat.writeAddRecord(RdsOutputFormat.java:278)
28 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:36)
29 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:34)
30 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:70)
31 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:33)
32 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:38)
33 at org.apache.flink.streaming.runtime.tasks.OperatorChainCopyingChainingOutput.collect(OperatorChain.java:608)
34 ... 17 more
35 Caused by: com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException: 821109
36
37 系统错误-内部错误：在批量写入时发生错误 sql template sql:INSERT INTO `rds_dim` (`d`,`e`,`f`) values (?,?,?) ON DUPLICATE KEY UPDATE e= values(e),f= values(f),preparedStatement
 at com.alibaba.blink.connectors.rds.RdsExceptionUtil.getBlinkException(RdsExceptionUtil.java:33)
38 at com.alibaba.blink.connectors.rds.RdsOutputFormat.batchWrite(RdsOutputFormat.java:447)
39 at com.alibaba.blink.connectors.rds.RdsOutputFormat.batchWrite(RdsOutputFormat.java:447)
40 at com.alibaba.blink.connectors.rds.RdsOutputFormat.sync(RdsOutputFormat.java:399)
41 at com.alibaba.blink.connectors.rds.RdsOutputFormat.run(RdsOutputFormat.java:284)
42 at java.util.TimerThread.mainLoop(Timer.java:555)
43 at java.util.TimerThread.run(Timer.java:505)
44 Caused by: java.sql.BatchUpdateException: Unknown column 'd' in 'field list'
45 at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
46

```

- 报错原因

不支持Schema定义中写入分区列。

- 解决方案

将分区列从DDL中删除，详情请参见[创建MaxCompute维表](#)。

## 报错：No pooled slot available and request to ResourceManager for new slot failed

- 报错信息

```
FailOver Time: 1970-01-01 08:00:00

org.apache.flink.runtime.jobmanager.scheduler.NoResourceAvailableException: No pooled slot available and request to ResourceManager for new slot failed
 at org.apache.flink.runtime.instance.SlotPool.slotRequestToResourceManagerFailed(SlotPool.java:450)
 at org.apache.flink.runtime.instance.SlotPool.access$400(SlotPool.java:91)
 at org.apache.flink.runtime.instance.SlotPool$2.apply(SlotPool.java:430)
 at org.apache.flink.runtime.instance.SlotPool$2.apply(SlotPool.java:426)
 at org.apache.flink.runtime.concurrent.impl.FlinkFuture$3.recover(FlinkFuture.java:192)
 at akka.dispatch.Recover.internal(Future.scala:268)
 at akka.dispatch.japi$RecoverBridge.apply(Future.scala:184)
 at akka.dispatch.japi$RecoverBridge.apply(Future.scala:182)
 at scala.util.Failure$$anonfun$recover$1.apply(Try.scala:216)
 at scala.util.Try$.apply(Try.scala:192)
 at scala.util.Failure.recover(Try.scala:216)
 at scala.concurrent.Future$$anonfun$recover$1.apply(Future.scala:326)
 at scala.concurrent.Future$$anonfun$recover$1.apply(Future.scala:326)
 at scala.concurrent.impl.CallbackRunnable.run(Promise.scala:36)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleRunAsync(AkkaRpcActor.java:273)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleMessage(AkkaRpcActor.java:131)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.access$000(AkkaRpcActor.java:72)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor$1.apply(AkkaRpcActor.java:110)
 at akka.actor.ActorCell$$anonfun$become$1.applyOrElse(ActorCell.scala:534)
 at akka.actor.Actor$class.aroundReceive(Actor.scala:467)
 at akka.actor.UntypedActor.aroundReceive(UntypedActor.scala:97)
 at akka.actor.ActorCell.receiveMessage(ActorCell.scala:516)
 at akka.actor.ActorCell.invoke(ActorCell.scala:487)
 at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:238)
 at akka.dispatch.Mailbox.run(Mailbox.scala:220)
 at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:397)
 at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)
 at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)
 at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)
 at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)
Caused by: java.lang.IllegalArgumentException: Priority 2 is already assigned to TaskManagerResourceProfile (TaskResourceProfile=ResourceProfile {cpuCores=0, 16666666666666666})
```

- 错误原因

作业指定的CU不足。

- 解决方案

您需要进行[手动配置调优](#)。

## 报错：Ots attribute key type error, table ots\_sink column key uv

- 报错信息

```
com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException:

ERR_ID:
 CON-04000038
CAUSE:
 Ots attribute key type error, table ots_sink column key uv
ACTION:
 contact ots admin for help.
DETAIL:

 at com.alibaba.blink.streaming.connectors.common.exception.ErrorUtils.getException(ErrorUtils.java:39)
 at com.alibaba.blink.streaming.connectors.common.exception.ErrorUtils.throwException(ErrorUtils.java:53)
 at com.alibaba.blink.connectors.ots.OtsOutputFormat.createPut(OtsOutputFormat.java:416)
 at com.alibaba.blink.connectors.ots.OtsOutputFormat.writeAddRecord(OtsOutputFormat.java:257)
 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:107)
```



```

rd(TupleRichOutputFormat.java:50)
 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:34)
 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:71)
 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:34)
 at org.apache.flink.streaming.api.functions.sink.SinkFunction.invoke(SinkFunction.java:52)
 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:56)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.pushToOperator(OperatorChain.java:635)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.collect(OperatorChain.java:612)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.collect(OperatorChain.java:575)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:763)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:741)
 at DataStreamSinkConversion$50.processElement(Unknown Source)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.pushToOperator(OperatorChain.java:635)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.collect(OperatorChain.java:612)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingWithOneInputStreamOperatorOutput.collect(OperatorChain.java:575)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:763)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:741)
 at org.apache.flink.table.runtime.util.StreamRecordCollector.collect(StreamRecordCollector.java:44)
 at org.apache.flink.table.runtime.aggregate.GroupAggFunction.processElement(GroupAggFunction.scala:152)
 at org.apache.flink.table.runtime.aggregate.GroupAggFunction.processElement(GroupAggFunction.scala:44)
 at org.apache.flink.table.runtime.KeyedProcessOperator.processElement(KeyedProcessOperator.java:167)
 at org.apache.flink.streaming.runtime.io.StreamInputProcessor.processRecordOrMark(StreamInputProcessor.java:286)
 at org.apache.flink.streaming.runtime.io.StreamInputProcessor.processInput(StreamInputProcessor.java:231)
 at org.apache.flink.streaming.runtime.tasks.OneInputStreamTask.run(OneInputStreamTask.java:112)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:348)
 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:732)
 at java.lang.Thread.run(Thread.java:834)

```

- 报错原因
  - valueColumns参数中包含主键。
  - valueColumns参数中各字段之间存在空格。
- 解决方案

- 去除设置在valueColumns参数中的主键。
- 删除valueColumns参数中各字段之间的空格。

## 报错：OutOfMemory

- 报错信息

在线上阶段和调试阶段出现内存溢出（OutOfMemory）的报错。

- 报错原因

- 调试阶段
  - 调试分配的内存不足。
  - 测试引用的源表过多。
  - 数据量过大（每个上传文件里面的测试数据均为1000条以上）。

- 线上阶段

作业的内存分配不足导致内存溢出。

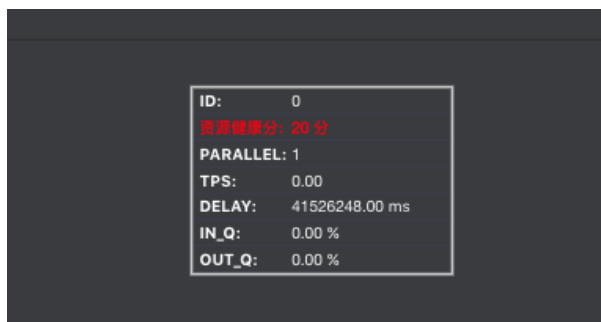
- 解决方案

- 调试阶段

在不影响测试结果情况下，建议您缩减测试数据为每个文件20条左右。

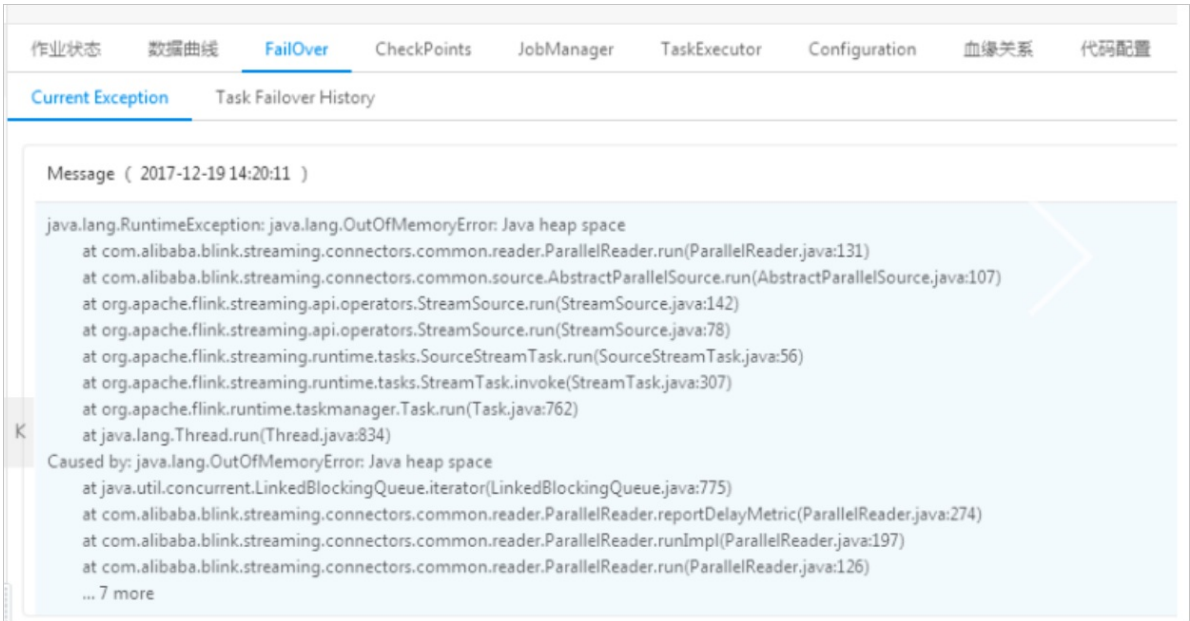
- 线上阶段

在运行拓扑图中查看资源健康分在60分以下的节点，上调该节点的CU数。具体步骤请参看[手动配置调优](#)。



## 报错：OutOfMemoryError: Java heap space

- 报错信息



- 报错原因
- 解决方案

内存不足。

在运行拓扑图中查看资源健康分在60分以下的节点，上调该节点的CU数。具体步骤请参看[手动配置调优](#)。



报错：java.lang.StackOverflowError

- 报错信息



此报错不影响作业，AutoScale功能完成后，作业即可恢复运行。

## 报错：MySQLSyntaxErrorException: Unknown column

### • 报错信息

```
Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Unknown column 'd' in 'field list'
```

### • 报错原因

Flink SQL中字段的定义和您上传的表字段的定义不一致。

### • 解决方案

请您根据Failover界面的报错提示，修改Flink SQL字段的定义，确保和您上传的表字段的定义一致。

## 报错：TaskManager lost/killed

### • 报错信息

```
java.lang.Exception: The assigned slot: SimpleSlot (1) - container_e05_1505041177764_25978_01_000033 @ hdpet2mainse011132138130.et2.tbsite.net (dataPort=59317) - ALLOCATED/ALIVE is asked to release from TaskManager: container_e05_1505041177764_25978_01_000033 @ hdpet2mainse011132138130.et2.tbsite.net (dataPort=59317), probably due to TaskManager lost/killed
 at org.apache.flink.runtime.instance.SimpleSlot.releaseSlot(SimpleSlot.java:217)
 at org.apache.flink.runtime.instance.SlotPool.releaseTaskManager(SlotPool.java:699)
 at sun.reflect.GeneratedMethodAccessor44.invoke(Unknown Source)
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.lang.reflect.Method.invoke(Method.java:498)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleRpcInvocation(AkkaRpcActor.java:183)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.handleMessage(AkkaRpcActor.java:135)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor.access$000(AkkaRpcActor.java:72)
 at org.apache.flink.runtime.rpc.akka.AkkaRpcActor$1.apply(AkkaRpcActor.java:110)
 at akka.actor.ActorCell$$anonfun$become$1.applyOrElse(ActorCell.scala:534)
 at akka.actor.Actor$class.aroundReceive(Actor.scala:467)
 at akka.actor.UntypedActor.aroundReceive(UntypedActor.scala:97)
 at akka.actor.ActorCell.receiveMessage(ActorCell.scala:516)
 at akka.actor.ActorCell.invoke(ActorCell.scala:487)
 at akka.dispatch.Mailbox.processMailbox(Mailbox.scala:238)
 at akka.dispatch.Mailbox.run(Mailbox.scala:220)
 at akka.dispatch.ForkJoinExecutorConfigurator$AkkaForkJoinTask.exec(AbstractDispatcher.scala:397)
 at scala.concurrent.forkjoin.ForkJoinTask.doExec(ForkJoinTask.java:260)
 at scala.concurrent.forkjoin.ForkJoinPool$WorkQueue.runTask(ForkJoinPool.java:1339)
 at scala.concurrent.forkjoin.ForkJoinPool.runWorker(ForkJoinPool.java:1979)
 at scala.concurrent.forkjoin.ForkJoinWorkerThread.run(ForkJoinWorkerThread.java:107)
```

### • 报错原因

由于GROUP BY、JOIN等节点的state\_size默认为0，不会申请Native Memory。当节点的Native Memory没有申请或太小时，就被YARN断开了。

### • 解决方案

- 使用[手动配置调优](#)进行资源调优。
- 配置GROUP BY、JOIN等节点的state\_size为1。每个Group，建议Native Memory配置为至少20M。

## 报错：The sts token has been expired

- 报错信息

```
java.lang.RuntimeException: java.lang.RuntimeException: com.aliyun.datahub.exception.NoPermissionException: The sts token has been expired. (Status Code: 403; Error Code: NoPermission; Request ID: 201802230613304db99a0a76d8a13b)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:131)
 at com.alibaba.blink.streaming.connectors.common.source.AbstractParallelSource.run(AbstractParallelSource.java:107)
 at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:142)
 at org.apache.flink.streaming.api.operators.StreamSource.run(StreamSource.java:78)
 at org.apache.flink.streaming.runtime.tasks.SourceStreamTask.run(SourceStreamTask.java:56)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:307)
 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:762)
 at java.lang.Thread.run(Thread.java:834)
Caused by: java.lang.RuntimeException: com.aliyun.datahub.exception.NoPermissionException: The sts token has been expired. (Status Code: 403; Error Code: NoPermission; Request ID: 201802230613304db99a0a76d8a13b)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.runImpl(ParallelReader.java:210)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader.run(ParallelReader.java:126)
 ... 7 more
Caused by: com.aliyun.datahub.exception.NoPermissionException: The sts token has been expired. (Status Code: 403; Error Code: NoPermission; Request ID: 201802230613304db99a0a76d8a13b)
 at com.aliyun.datahub.model.serialize.JsonErrorParser.parse(JsonErrorParser.java:44)
 at com.aliyun.datahub.model.serialize.GetCursorResultJsonDeser.deserialize(GetCursorResultJsonDeser.java:19)
 at com.aliyun.datahub.model.serialize.GetCursorResultJsonDeser.deserialize(GetCursorResultJsonDeser.java:14)
 at com.aliyun.datahub.DatahubClient.getCursor(DatahubClient.java:627)
 at com.aliyun.datahub.DatahubClient.getCursor(DatahubClient.java:699)
 at com.alibaba.blink.streaming.connectors.datahub.input.RawDatahubRecordReader.seek(RawDatahubRecordReader.java:397)
 at com.alibaba.blink.streaming.connectors.datahub.input.RawDatahubRecordReader.next(RawDatahubRecordReader.java:278)
 at com.alibaba.blink.streaming.connectors.common.reader.ParallelReader$ReaderRunner.run(ParallelReader.java:341)
 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1147)
 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:622)
 ... 1 more
```

- 报错原因

为了保证用户的数据安全，STS Token会有一定的生命周期，系统会自动重新获取Token。

- 解决方案

建议您在With参数中设置stsUpdateSeconds='3600'。

```
WITH (
 type='datahub',
 url='',
 username='',
 password='',
 tableName='',
 stsUpdateSeconds='3600'
)
```

- stsUpdateSeconds主动更新roleArn的时间，时间单位为秒。
- stsUpdateSeconds设置的值，需比各存储中，登录Session的过期时间值小。例如，如果登录Session过期时间设置为6小时，则stsUpdateSeconds参数建议设置为stsUpdateSeconds='10800'。

☒ 允许自主管理多因素设备

登录Session过期时间： 小时（有效值：6 - 24）

登录掩码设置：

网络掩码决定哪些IP地址会受到登录控制台的影响，包括密码登录和SSO登录，但使用AccessKey发起的API访问并不受影响。如果指定掩码，子用户必须只能从指定的IP地址进行登录。如果不指定任何掩码，登录控制台功能将适用于整个网络。当需要配置多个掩码时，请使用分号来分隔掩码，例如：  
42.120.66.0/24; 42.120.74.98

## 报错：OTSParameterInvalid, Message: Invalid update row request: missing cells in request

- 报错信息

数据写入OTS结果表时，报错如下。

```
Code: OTSParameterInvalid, Message: Invalid update row request: missing cells in request.
```

- 报错原因

通常是因为valueColumns整行数据都为NULL。

- 排查思路

设置结果表Type为Print，查找valueColumns整行全部都是NULL的数据。



**说明** 如果数据量大时，您可以添加WHERE条件快速排查valueColumns整行全部都是NULL的数据。

## 报错：There is an invalid character in tagvalue

- 报错信息



```
java.lang.IllegalArgumentException: There is an invalid character in tagvalue. the tag is +
<method:getFishpoolFollower$401> , the char is '$'
 at com.alibaba.hitsdb.client.value.request.Point.checkTag(Point.java:355)
 at com.alibaba.hitsdb.client.value.request.Point.checkPoint(Point.java:313)
 at com.alibaba.hitsdb.client.value.request.Point$MetricBuilder.build(Point.java:278)
 at com.alibaba.blink.connectors.hitsdb.DefaultRowPointConverter.convert(DefaultRowPointCo
nverter.java:63)
 at com.alibaba.blink.connectors.hitsdb.HiTSDbOutputFormat.writeAddRecord(HiTSDbOutputForm
at.java:130)
 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord
(TupleRichOutputFormat.java:50)
 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord
(TupleRichOutputFormat.java:34)
 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invo
ke(TupleOutputFormatAdapterSink.java:71)
 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invo
ke(TupleOutputFormatAdapterSink.java:34)
 at org.apache.flink.streaming.api.functions.sink.SinkFunction.invoke(SinkFunction.java:52
)
 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:56)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$OutputAdapter.lambda$new$0(Oper
atorChain.java:767)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$OutputAdapter.processElement(Oper
atorChain.java:841)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingOutput.pushToOperator(Oper
atorChain.java:689)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingOutput.collect(Operator
Chain.java:673)
 at org.apache.flink.streaming.runtime.tasks.OperatorChain$ChainingOutput.collect(Operator
Chain.java:639)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect
(AbstractStreamOperator.java:1106)
 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect
(AbstractStreamOperator.java:1084)
 at DataStreamSinkConversion$373.processElement(Unknown Source)
 at org.apache.flink.streaming.runtime.io.StreamInputProcessor.processInput(StreamInputPro
cessor.java:255)
 at org.apache.flink.streaming.runtime.tasks.OneInputStreamTask.run(OneInputStreamTask.jav
a:89)
 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:298)
 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:747)
 at java.lang.Thread.run(Thread.java:834)
```

- 报错原因

TSDB输入数据中存在非法字符 `$` 。

- 解决方案

TSDB输入数据只可包含大小写英文字母、中文、数字，以及特殊字符 `-`、`_`、`.`、`/`、`()`、`:`、`,`、`[]`、`=`、```、`#`。详情请参见[TSDB请求参数输入限制](#)。

 **说明** 建议您在写入数据前进行字符替换。

## 报错：Caused by: com.mysql.jdbc.exceptions.jdbc4.MySQLTransactionRollbackException: Deadlock found when trying to get lock; try restarting transaction

- 报错信息

写MySQL（TDDL/RDS）时，出现死锁（DeadLock）。

 说明 在实时计算Flink中，下游数据库使用MySQL等关系数据库（对应的Connector为TDDL/RDS），当实时计算Flink版频繁写某个表或者资源时，存在死锁风险。

- 死锁形成的示例

假设完成一次INSERT需要依次抢占(A,B)2个锁。A是一个范围锁，有2个事务(T1,T2)，表的Schema为(id(自增主键),nid(唯一键))。T1包含2条insert(null,2),(null,1)，T2包含1条insert(null,2)。

- i. t时刻，T1第一条INSERT插入，此时T1持有(A,B)2个锁。
- ii. t+1时刻T2开始插入，需要等待锁A来锁住(-inf,2]，此时A被T1拥有，且锁住了(-inf,2]，区间存在包含关系，所以T2依赖T1释放A。
- iii. t+2时刻T1第二条INSERT执行，需要A锁住(-inf,1]，该区间属于(-inf,2]，所以需要排队等T2释放锁，所以T1依赖T2释放A。

当T1和T2相互依赖且相互等待时死锁形成。

- RDS/TDDL、OTS数据库引擎锁的区别

- RDS/TDDL：InnoDB的行锁是针对索引加的锁，不是针对单条记录加的锁。所以虽然是访问不同行的记录，但是如果使用相同的索引键，会出现锁冲突，造成了整个区域的数据都无法更新。
- OTS：单行锁，不影响其他数据更新。

- 死锁的解决方案

高QPS/TPS或高并发写入情况场景，建议使用OTS作为结果表，可以解决死锁的问题。通常，不建议使用TDDL或者RDS作为Flink Job的结果表。

如果必须要使用MySQL等关系数据库作为Sink节点，有以下建议：

- 确保没有其他读写业务方的干扰。
- 如果Job的数据量不大可以尝试单并发写入。但是在高QPS/TPS、高并发情况下，写入性能会降低。
- 尽可能不使用UniqueKey（唯一主键），带UniqueKey表的写入可能会导致死锁。如果业务要求表必须包含UniqueKey，请按照字段区分能力从大到小排列来定义UniqueKey，可以大幅降低死锁出现概率。例如，您可以把MD5函数放在day\_time(20171010)前面，就可以使得字段区分能力从大到小排列来定义UniqueKey，从而解决死锁问题。
- 根据业务特点做分库分表，尽可能避免单表写入，实施细节请联系对应的数据库管理员。

## 报错：Value '0000-00-00 00:00:00' can not be represented as java.sql.Timestamp

- 报错信息

```
Caused by: java.sql.SQLException: Value '0000-00-00 00:00:00' can not be represented as java.sql.Timestamp
 at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:964)
 at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:897)
 at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:886)
 at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:860)
 at com.mysql.jdbc.ResultSetRow.getTimestampFast(ResultSetRow.java:937)
 at com.mysql.jdbc.ByteArrayRow.getTimestampFast(ByteArrayRow.java:130)
 at com.mysql.jdbc.ResultSetImpl.getTimestampInternal(ResultSetImpl.java:5918)
 at com.mysql.jdbc.ResultSetImpl.getTimestamp(ResultSetImpl.java:5588)
 at com.alibaba.druid.pool.DruidPooledResultSet.getTimestamp(DruidPooledResultSet.java:220)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.deserializeToInnerObject(MySqlRowFetcherBase.java:316)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.toResultRow(MySqlRowFetcherBase.java:251)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.flatMap(MySqlRowFetcherBase.java:218)
 at com.alibaba.blink.connectors.mysql.dim.MySqlRowFetcherBase.flatMap(MySqlRowFetcherBase.java:58)
 at com.alibaba.blink.streaming.connectors.common.LookupFunctionWrapper.eval(LookupFunctionWrapper.java:56)
 at LookupFunction$146.flatMap(Unknown Source)
 at org.apache.flink.table.runtime.join.TemporalTableJoinProcessRunner.processElement(TemporalTableJoinProcessRunner.scala:85)
 at org.apache.flink.table.runtime.join.TemporalTableJoinProcessRunner.processElement(TemporalTableJoinProcessRunner.scala:33)
 at org.apache.flink.streaming.api.operators.ProcessOperator.processElement(ProcessOperator.java:66)
```

- 问题原因

RDS维表中TIMESTAMP类型的字段中存在非法格式的数据0000-00-00 00:00:00，维表JOIN非法格式的数据就会报错。

- 解决方案

在RDS维表WITH参数 `URL` 后添加 `?zeroDateTimeBehavior=convertToNull`，输出为null。

## 报错：Failover:offset (0) + length (8) exceed the capacity of the array: 1

- 报错信息

```
Caused by: java.lang.IllegalArgumentException: offset (0) + length (8) exceed the capacity of the array: 1
 at org.apache.hadoop.hbase.util.Bytes.explainWrongLengthOrOffset(Bytes.java:548)
 at org.apache.hadoop.hbase.util.Bytes.toLong(Bytes.java:522)
 at org.apache.hadoop.hbase.util.Bytes.toLong(Bytes.java:495)
```

- 问题原因

云数据库HBase版的数据类型定义错误。

- 解决方案

修改云数据库HBase版的数据类型和物理表一致。

## 作业启动一段时间后报错停止，Failover里出现报错信息，应该如何处理？

- STS Token过期

```
1 FailOver Time: 2018-04-23 23:04:01
2
3 java.lang.RuntimeException: Could not forward element to next operator
4 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:611)
5 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:593)
6 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:941)
7 at org.apache.flink.streaming.api.operators.AbstractStreamOperator$CountingOutput.collect(AbstractStreamOperator.java:919)
8 at org.apache.flink.streaming.api.operators.StreamMap.processElement(StreamMap.java:38)
9 at org.apache.flink.streaming.runtime.io.StreamInputProcessor.processInput(StreamInputProcessor.java:282)
10 at org.apache.flink.streaming.runtime.tasks.OneInputStreamTask.run(OneInputStreamTask.java:66)
11 at org.apache.flink.streaming.runtime.tasks.StreamTask.invoke(StreamTask.java:307)
12 at org.apache.flink.runtime.taskmanager.Task.run(Task.java:762)
13 at java.lang.Thread.run(Thread.java:834)
14 Caused by: com.alibaba.blink.streaming.connectors.common.exception.BlinkRuntimeException: 821304
15
16 用户错误-连接错误: 连续4次往Datahub写入数据失败, 请联系Datahub系统管理员进行排查。
17 错误信息: The sts token has been expired. (Status Code: 403; Error Code: NoPermission; Request ID: 2018042323035229b99a0a8b030437)
18 at com.alibaba.blink.streaming.connectors.datahub.DatahubExceptionUtil.getBlinkException(DatahubExceptionUtil.java:28)
19 at com.alibaba.blink.streaming.connectors.datahub.sink.DatahubOutputFormat.writeDataHubPack(DatahubOutputFormat.java:386)
20 at com.alibaba.blink.streaming.connectors.datahub.sink.DatahubOutputFormat.batchWrite(DatahubOutputFormat.java:406)
21 at com.alibaba.blink.streaming.connectors.datahub.sink.DatahubOutputFormat.sync(DatahubOutputFormat.java:238)
22 at com.alibaba.blink.streaming.connectors.datahub.sink.DatahubOutputFormat.writeAddRecord(DatahubOutputFormat.java:460)
23 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:50)
24 at com.alibaba.blink.streaming.connectors.common.output.TupleRichOutputFormat.writeRecord(TupleRichOutputFormat.java:34)
25 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:70)
26 at com.alibaba.blink.streaming.connectors.common.output.TupleOutputFormatAdapterSink.invoke(TupleOutputFormatAdapterSink.java:33)
27 at org.apache.flink.streaming.api.operators.StreamSink.processElement(StreamSink.java:38)
28 at org.apache.flink.streaming.runtime.tasks.OperatorChain$CopyingChainingOutput.collect(OperatorChain.java:608)
29 ... 9 more
30
```

STS Token过期导致的报错，不影响线上作业。详情请参见[报错：The sts token has been expired](#)。

- 脏数据处理失败

- 排错指引

在Failover中查看脏数据具体内容。

- 解决方案

修改SQL，过滤脏数据后，重新[上线](#)作业。

- 写入MySQL类型数据库死锁

- 排错指引

- 手动调优后，检查Sink节点拆分后的部分节点与同时写入的其他节点之间，是否出现了Rebalance。
- 检查Sink节点是否多并发写入数据库。
- 检查出现死锁的表中PK与数据库表定义的UK是否一致，包括顺序。
- 检查定义的PK是否按照区分度高低进行排列。

- 解决方案

详情请参见[写MySQL（TDDL/RDS）时，出现死锁（DeadLock）](#)。