SOFAStack

分布式事务
运维指南

产品版本：AntStack Plus 1.11.0

文档版本：20221026

蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，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。

商标声明

蚂蚁集团 ANT GROUP及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

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。蚂蚁集团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，并在蚂蚁集团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蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

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

目录

1.系统组件 05
2.系统日志 07
3.同城容灾 08
3.1. 产品架构 08
3.2. 容灾架构 09
3.3. 热切换方案 12
3.4. 容灾场景处理 12
4.日常运维 15
4.1. 监控和预警 15
4.1.1. 监控指标和预警项 15
4.1.2. 预警项运维动作 16
4.1.3. 日常巡检项 16
4.2. 服务巡检 17
4.3. 管理容器 18
4.3.1. 变更容器配置 18
4.3.2. 容器内存与 JVM 堆内存配置比 20
4.3.3. 重启容器 22
5.常见故障处理 24
5.1. 服务异常应急预案 24
5.2. 故障排查快速指南 24
5.3. JVM 配置问题排查应急方案 38
5.4. 其他问题 40

1.系统组件

组件角色说明

分布式事务产品包括以下 3 个组件：

- **分布式事务管理器**

  分布式事务应用 Server，负责接收事务消息，并协调事务发起方和参与方完成整个分布式事务。

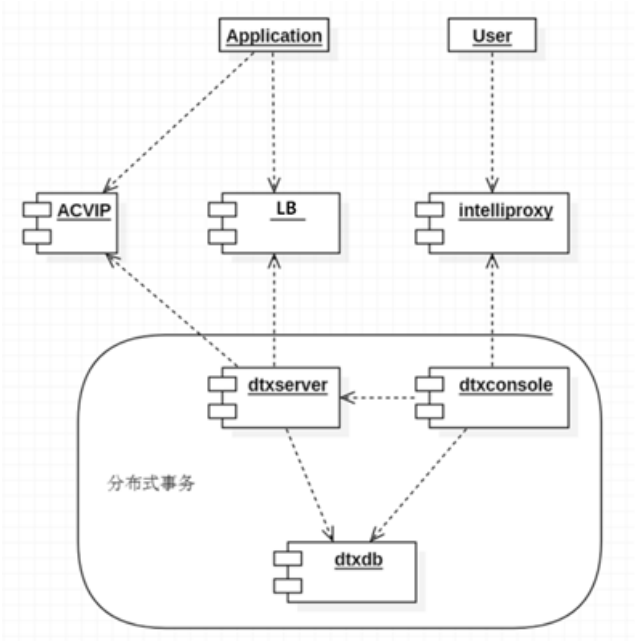
- **分布式事务控制台**

  Web 应用系统，负责提供分布式事务日志数据的查询和分析。

- **分布式事务数据存储**

  关系型数据库，存储分布式事务元数据、分布式事务事务日志数据。

部署拓扑说明



- **分布式事务管理器**：应用名 dtxserver。
- **分布式事务控制台**：应用名 dtxconsole。
- **分布式事务数据存储**：数据库实例名 dtxdb。
- **ACVIP**：SOFA 中间件云产品服务发现组件。
- **LB**：用于 VPC 环境下作为分布式事务管理器服务节点的网络暴露。
- **Intelliproxy**：SOFA 中间件云产品网关组件。
- **Application**：用户应用系统。
- **User**：用户浏览器。

资源清单说明

组件	资源依赖	服务端口
分布式事务管理器	2 * 服务器 (2 CPU、4 G 内存、100 G 磁盘) 需要根据具体业务量评估实例数。	14200
分布式事务控制台	2 * 服务器 (2 CPU、4 G 内存、100 G 磁盘)	80
分布式事务数据存储	1 * RDS（4 CPU、8 G 内存、200 G 磁盘）	3306

2.系统日志

分布式事务管理器系统日志文件

日志文件名称	日志文件路径	日志文件描述
common-default.log	/home/admin/logs/dtxserver/common-default.log	默认日志
common-error.log	/home/admin/logs/dtxserver/common-error.log	错误日志，记录系统的全部 ERROR 日志。
dtx-dal.log	/home/admin/logs/dtxserver/dtx-dal.log	数据库访问统计日志
dtx-schedule.log	/home/admin/logs/dtxserver/dtx-schedule.log	定时调度任务日志
dtx-recovery.log	/home/admin/logs/dtxserver/dtx-recovery.log	恢复任务日志
dtx-remote.log	/home/admin/logs/dtxserver/dtx-remote.log	事务协调器起日志

分布式事务控制台系统日志文件

日志文件名称	日志文件路径	日志文件描述
common-default.log	/home/admin/logs/dtxconsole/common-default.log	默认日志
common-error.log	/home/admin/logs/dtxconsole/common-error.log	错误日志，记录系统的全部 ERROR 日志。
dtx-rest.log	/home/admin/logs/dtxconsole/dtx-rest.log	REST 接口访问日志
dtx-dal.log	/home/admin/logs/dtxconsole/dtx-dal.log	数据库访问统计日志

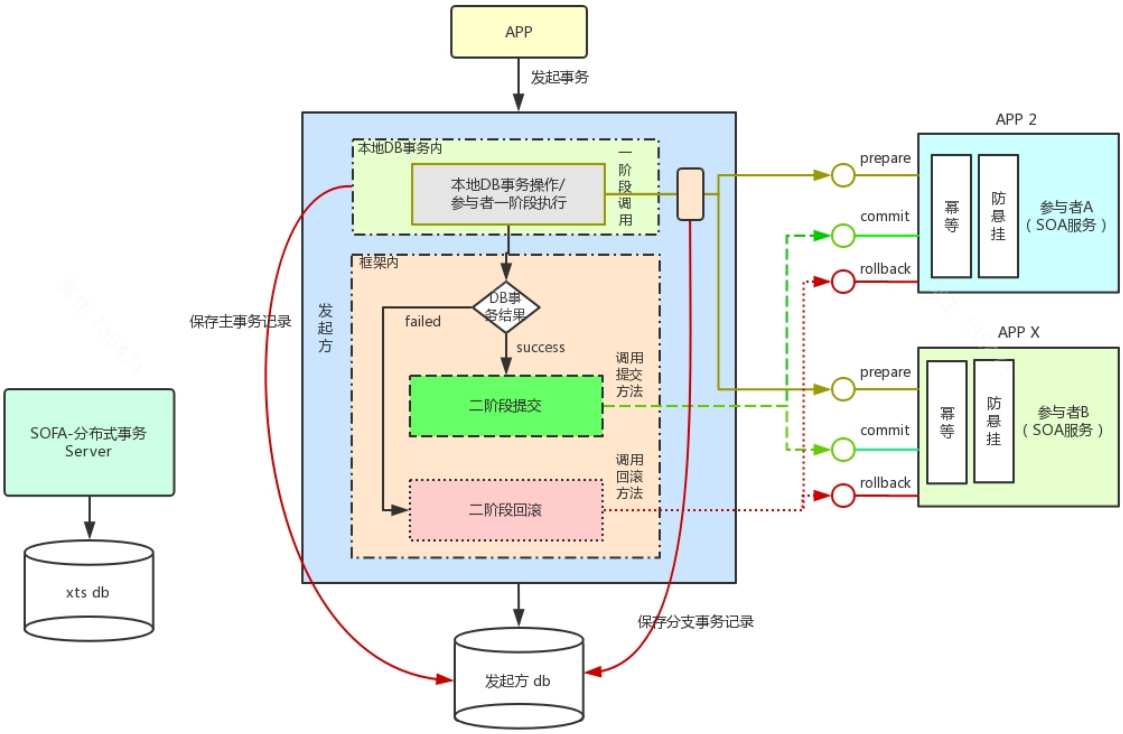
3.同城容灾

3.1. 产品架构

单机房架构

当前对外输出的只有同库和异库两种模式，dtxconsole 的访问逻辑同其他中间件，通过 intelliproxy 来访问。

同库发起方+Local 参与者



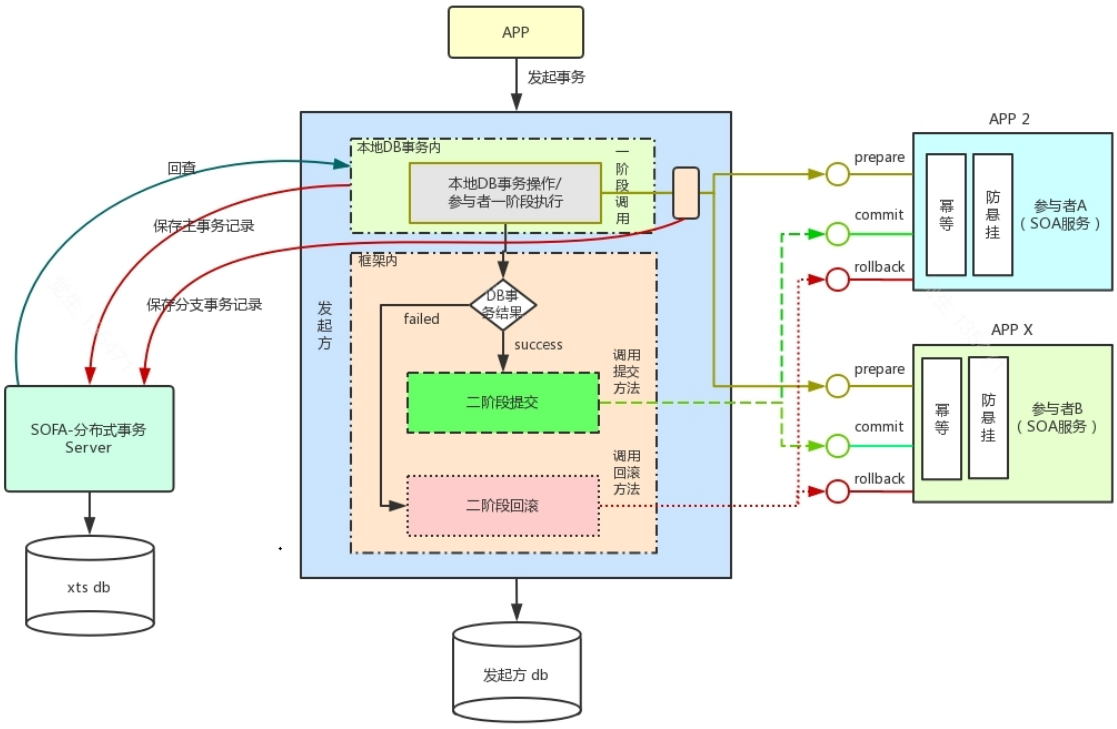
特点

- 该模式需要发起方维护 DB 来保存主事务记录和分支事务记录，对发起方来说成本略高。
- 发起方可以根据自己的业务量对 DB 进行扩容，访问本地 DB 性能较优。

适用对象

适用于吞吐量和性能要求比较高的业务。

异库发起方+Remote 参与者



特点

- 相比于同库发起方需要业务提供数据库来保存事务日志，异库发起方可以依赖 DTX Server 及其 DB，对一些规模较小的业务来说性价比更高。
- 事务日志由 DTX Server 负责存储和维护，但是发起方无法保证事务状态消息能成功无误发送至Server，故发起方需实现回查服务，DTX Server 会在事务状态未知的情况下，使用回查接口询问发起方事务状态。此模式发起方无需自己维护 DB，成本较低。

适用对象

适用于业务量较小的发起方，但由于发起方依赖了 DTX Server 及其 DB，事务发起方的业务量会受到 DTX Server 及其 DB 的瓶颈限制。

组件调用逻辑

- **dtx-server**

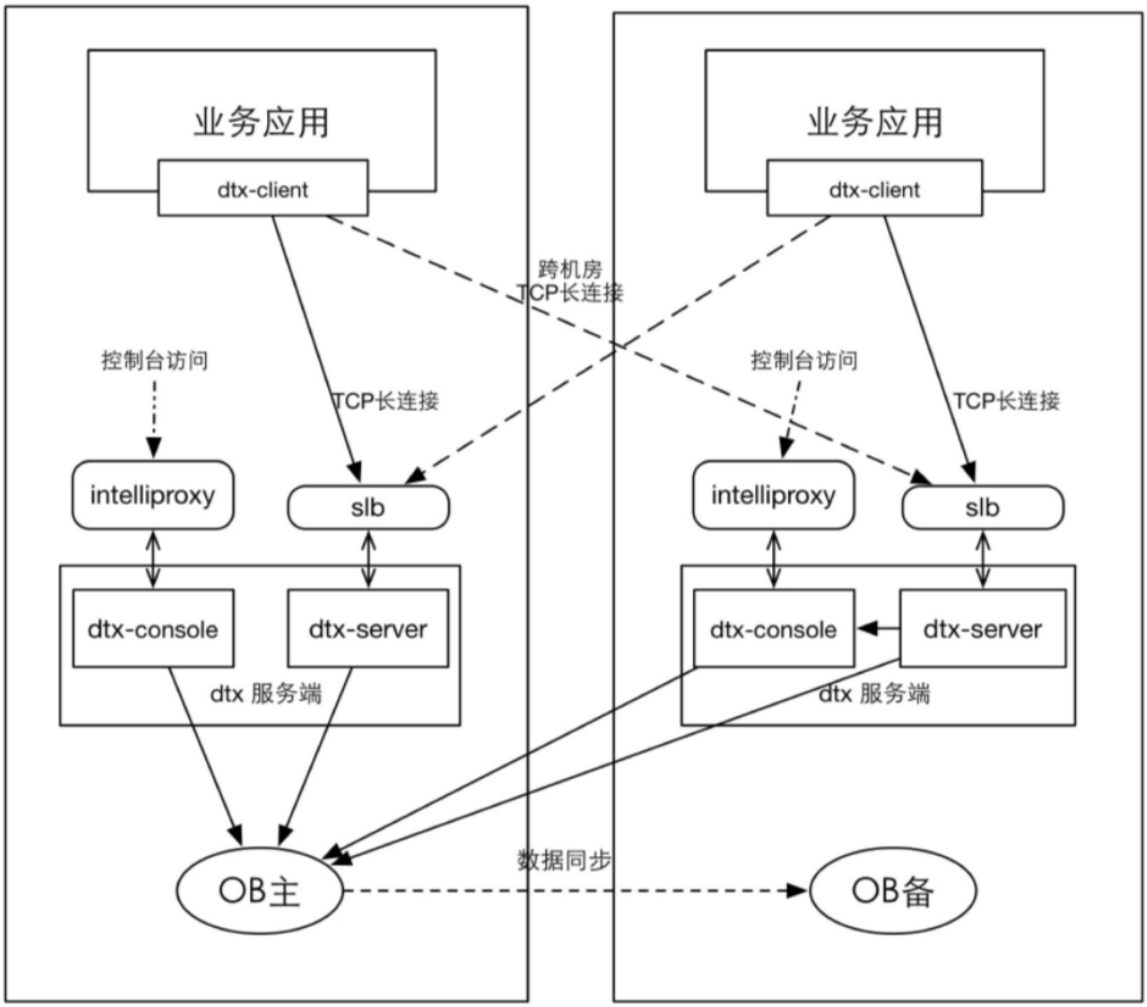
  负责与 dtx 客户端（dtx-client）通信，提供分布式事务发起、提交、回滚等功能。

- **dtx-console**

  提供分布式事务控制台功能，用户在控制台可访问分布式事务后台页面，查询历史事务数据等，dtx-console 后台流量来自 intelliproxy 的中转。

3.2. 容灾架构

架构图



容灾架构说明

- 部署时，主备机房需对等部署，数据库共库。
- DTX 客户端（dtx-client）随同业务系统部署，业务系统启动时，各机房的 DTX 客户端均会全连主机房和备机房所有 dtx-server 服务器，与其建立 TCP 长连接。
- DTX 客户端优先同本机房 dtx-server 通信。
- 如果本机房的 dtx-server 不可用，DTX 客户端会同其他机房的 dtx-server 通信；主机房和备机房的 dtx-server 只要有一个可用，dtx-client 就便能正常运行；如果主备机房的所有 dtx-server 均不可用，则 DTX 不可用。
- 事务恢复逻辑
  - 同库：事务恢复由业务方处理，DTX 提供了定时任务的代码 ，需要业务方配置。为了保证机房级容灾，业务数据库需要有容灾能力。
  - 异库：事务恢复由 dtx-server 处理，因为数据库共库，只要主备机房有可用的 dtx-server 就能执行事务恢复任务，容灾时无需额外切换操作。

> **说明**
> DTX 客户端优先同本机房 dtx-server 通信，本地机房不可用时，优先使用本地区其他机房。

产品依赖

外部产品依赖

- **ACVIP**：用于获取 SOFARegistry 地址。
- **SOFARegistry（注册中心）**：dtxconsole 注册 facade，通过 SOFARegistry 获取调用服务信息。
- **Intelliproxy**：通过 Intelliproxy 访问 dtxconsole 界面,依赖 intelliproxy 的容灾能力。
  - dtxconsole 通过 sdk sensor 向 SOFARegistry 注册自己的地址。
  - Intelliproxy 通过 ACVIP 获取 SOFARegistry 地址，再从 SOFARegistry 中获取 dtxconsole 地址。
- **OSP**：dtxconsole 调用 OSP facade 用于认证及获取 instanceid、workspace 等信息。

底座产品依赖

- **元集群**：元集群主备机房独立部署，无容灾要求。
- **内网 DNS**：
  - DTX 通过 内网 DNS 提供的域名访问数据库，容灾时，需将数据库域名切换至备机房的数据库。故依赖内网 DNS 的容灾能力；公有云上直接使用数据库资源对外暴露的域名。
  - DTX 管控页面通过内网 DNS 提供的域名，双挂主备机房的内网负载均衡，对外提供服务。
- **内网负载均衡**：主备机房各自通过内网负载均衡做负载，并提供给业务应用访问。

  内网 DNS 和 内网负载均衡在不同底座下，有不同的选择。具体如下表所示：

底座	内网 DNS	内网负载均衡
AntStack	AntDNS	ALB
AntStack Plus 公有云	privatezone	SLB
AntStack Plus 物理机	AntDNS	-
AntStack Plus 物理机 + 飞天	AntDNS（forword opsdns）	-

> **重要**
> 物理机底座下，AntStack Plus 目前没有内网负载均衡，实际部署时用客户环境选用的负载均衡设备。

数据库依赖

OB/RDS：主备机房共库，依赖 OB/RDS 的容灾能力。

数据同步

主备机房共库，依赖数据库的容灾能力。

网络链路

- 业务应用通过内网负载均衡访问 dtx-server。
- 用户通过域名访问管控页面。

容灾能力

支持机房级/组件级容灾，容灾切换后无功能降级。

3.3. 热切换方案

容灾切换说明

- 故障恢复阶段可能出现调度任务到还未完全恢复的机房应用，所以需要在故障后进行显示切换，使得任务只会调度到健康机房，恢复后再取消切换。任务必须幂等，确保任务失败可以无损重试。
- 数据库容灾切换，依赖数据库容灾能力。

切换逻辑

1. 切换元数据库 DB、内网 DNS。
2. 切换 DTX 管控域名，摘除主机房流量。
3. 验证备机房 DTX 状态。

切换影响

数据库切换过程中，DTX 服务不可用。

切换原则

保证数据不丢失，对业务影响只是闪断。

3.4. 容灾场景处理

掉电场景

A 机房	B 机房	影响	容灾切换操作
掉电	正常	主机房正在执行中的事务受影响，dtx-server 服务恢复后这些事务自动回滚。 数据库主备切换期间，主备机房的 DTX 服务均不可用。	无需操作
正常	掉电	备机房正在执行中的事务受影响，dtx-server 服务恢复后事务自动回滚。 主机房 DTX 服务可用，备机房 DTX 服务不可用。	无需操作

掉电	掉电	全部服务不可用。	不考虑

断网场景

A 机房（外网入口）	B 机房（外网入口）	A 与 B 之间（机房间互联光纤）	影响	容灾切换操作
正常	正常	正常	-	-
断网	正常	正常	全部流量从 B 机房进入 。 （主机房正在执行中的事务受影响，dtx-server 服务恢复后这些事务自动回滚， DTX 备机房服务正常。）	无需操作
正常	断网	正常	全部流量从 A 机房进入 。 （DTX 主机房服务无影响，备机房正在执行中的事务受影响，dtx-server 服务恢复后这些事务自动回滚。）	无需操作
断网	断网	正常	全部服务不可用。	不考虑
正常	正常	断网	正在执行中的跨机房事务受影响， dtx-server服务端恢复后这些事务自动回滚。 主机房 DTX 服务可用，备机房 DTX 服务不可用，跨机房事务不可用。 数据库主备数据同步会受影响， 影响 DTX 主备数据同步。	切到 A 机房
断网	正常	断网	正在执行中的跨机房事务以及主机房正在执行中的事务受影响，dtx-server 服务端恢复后会自动回滚事务。 主机房 DTX 服务可用，但业务没有流量进来，备机房 DTX 服务不可用，跨机房事务不可用。 数据库主备数据同步会受影响， 影响 DTX 主备数据同步。	切到 B 机房

正常	断网	断网	正在执行中的跨机房事务以及备机房正在执行中的事务受影响，dtx-server 服务端恢复后会自动回滚事务。 主机房 DTX 服务可用，备机房 DTX 服务不可用；跨机房事务不可用。 数据库主备数据同步会受影响， 影响 DTX 主备数据同步。	切到 A 机房
断网	断网	断网	全部服务不可用。	不考虑

4.日常运维

4.1. 监控和预警

4.1.1. 监控指标和预警项

系统监控指标和预警项清单

监控指标	监控指标说明	预警触发条件
分布式事务管理器-磁盘空间监控	磁盘空间占用率监控	大于 80%
分布式事务管理器-端口监控	端口 14200 是否可以访问	端口 14200 无法访问
分布式事务管理器-内存监控	内存占用率	大于 80%
分布式事务管理器-CPU 利用率	CPU 利用率	大于 80%
分布式事务控制台-磁盘空间监控	磁盘空间占用率监控	大于 80%
分布式事务控制台-端口监控	端口 80 是否可以访问	端口 80 无法访问
分布式事务控制台-内存监控	内存占用率	大于 80%
分布式事务控制台-CPU 利用率	CPU 利用率	大于 80%

自定义监控指标和预警项清单

监控指标	监控指标说明	预警触发条件	报警级别
事务积压数量	未决事务积压数量，一般由于参与方系统异常导致积压。	>10	重大
		>6	次要
		>10	重大
		>6	次要

整体异常数量 监控指标	分布式事务系统整体每分 监控指标说明	预警触发条件	报警级别
事务二阶段失败数量	分布式事务每分钟二阶段执行失败的数量。	>5/min	重大
		>3/min	次要
数据库访问耗时监控	分布式事务系统每分钟访问数据库平均耗时。	>10ms	重大
		>7ms	次要
服务耗时监控	分布式事务提供的服务接口每分钟平均耗时。	一阶段接口 > 15ms 二阶段接口 > 200ms	重大
		一阶段接口 > 10ms 二阶段接口 > 150ms	次要

4.1.2. 预警项运维动作

分布式事务管理器端口预警项

预警项说明

分布式事务管理器的网络服务端口号是 14200，当此端口无法通过网络访问时，分布式事务发起方和分布式事务参与方应用系统都无法正常连接到分布式事务管理器。

运维动作说明

1. 通过预警项说明，确定端口异常的分布式事务管理器服务器地址，一般是服务器 IP 地址。
2. 登录异常分布式事务管理器服务器，并执行以下命令，确认端口服务是否存在。

```
netstat -ano | grep LISTEN | grep 14200
```

3. 如果服务端口 14200 确实未提供服务，说明预警准确，需要执行以下命令，进一步确定分布式事务管理器服务进程是否存在。如果进程不存在，则确认是服务实例宕机，需要重新启动服务实例。

```
ps -ef| grep java
```

4.1.3. 日常巡检项

系统监控指标巡检项目

巡检项说明

通过监控系统，查看分布式事务产品各项系统监控指标。

运维动作说明

如果发现预警指标异常，根据对应的预警项运维动作说明进行操作即可。

4.2. 服务巡检

系统组件监控检查

基础监控检查

登录监控系统，查看分布式事务（DTX）相关系统组件有无异常告警，需要关注的监控项内容如下表所示。

应用	分类	指标（关键字）	告警阈值
DTX-dtx DTX-dtx-console	基础监控-CPU 使用率监控	cpu_usage	>80%
	基础监控-内存使用率监控	mem_usage	>80%
	基础监控-磁盘使用率监控	disk_usage	>80%
	基础监控-load5	load5	>3
DTX-dtx	基础监控-端口监控	2022、14200	不通
DTX-dtx-console	基础监控-端口监控	80、2022	不通

自定义业务监控检查

通过监控系统查看分布式事务产品各项系统监控指标是否有异常告警产生。如果有异常告警产生，联系技术支持进行排查。

业务功能检查

1. dtx 服务验证。
   - 执行以下命令，判断业务端口是否存活。

     ```
     netstat -an |grep 14200
     ```

   - 执行以下命令，判断 RPC 端口是否存活。

     ```
     netstat -an |grep 12200
     ```

   - 执行以下命令，判断 acvip 连接。

     ```
     netstat -an |grep 9003
     ```

查看连接的 ACVIP 的地址是否与 sofa-portal 控制台中显示的 ACVIP 地址一致。不一致的话，需将连接的地址改成 sofa-portal 控制台中显示的 ACVIP 地址。

- 执行以下命令，判断与注册中心是否连接。

```
netstat -an |grep 9600
```

2. dtxconsole 服务验证
   - 执行以下命令，判断 Nginx 端口是否存活。

```
netstat -an |grep 8080
```

   - 执行以下命令，判断业务端口是否存活。

```
netstat -an |grep 80
```

   - 访问 `sofa-dtx.{domain}` 控制台页面，检查各项页面功能是否正常无报错。

事务积压数检查

通过 DTX 的自定义监控项，重点关注事务积压数量。

监控指标	监控指标说明	监控阈值	报警级别
事务积压数量	未决事务积压数量，一般由于参与方系统异常导致积压。	>10	重大
		>6	次要
整体异常数量	分布式事务系统整体每分钟异常数量。	>10	重大
		>6	次要
事务二阶段失败数量	分布式事务每分钟二阶段执行失败的数量。	>5/min	重大
		>3/min	次要

4.3. 管理容器

4.3.1. 变更容器配置

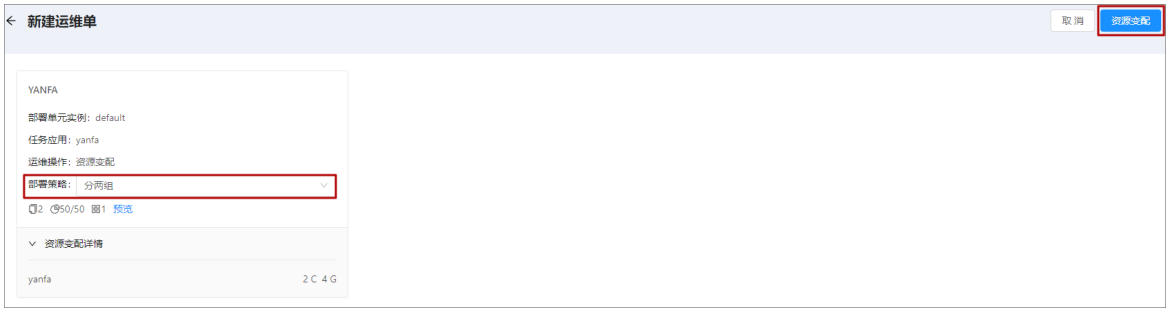
您可以通过云游 Local 控制台对应用的容器规格配置进行变更。

操作步骤

1. 登录云游 Local 控制台。
2. 在左侧导航栏，选择 **产品运维 > 产品基线**。

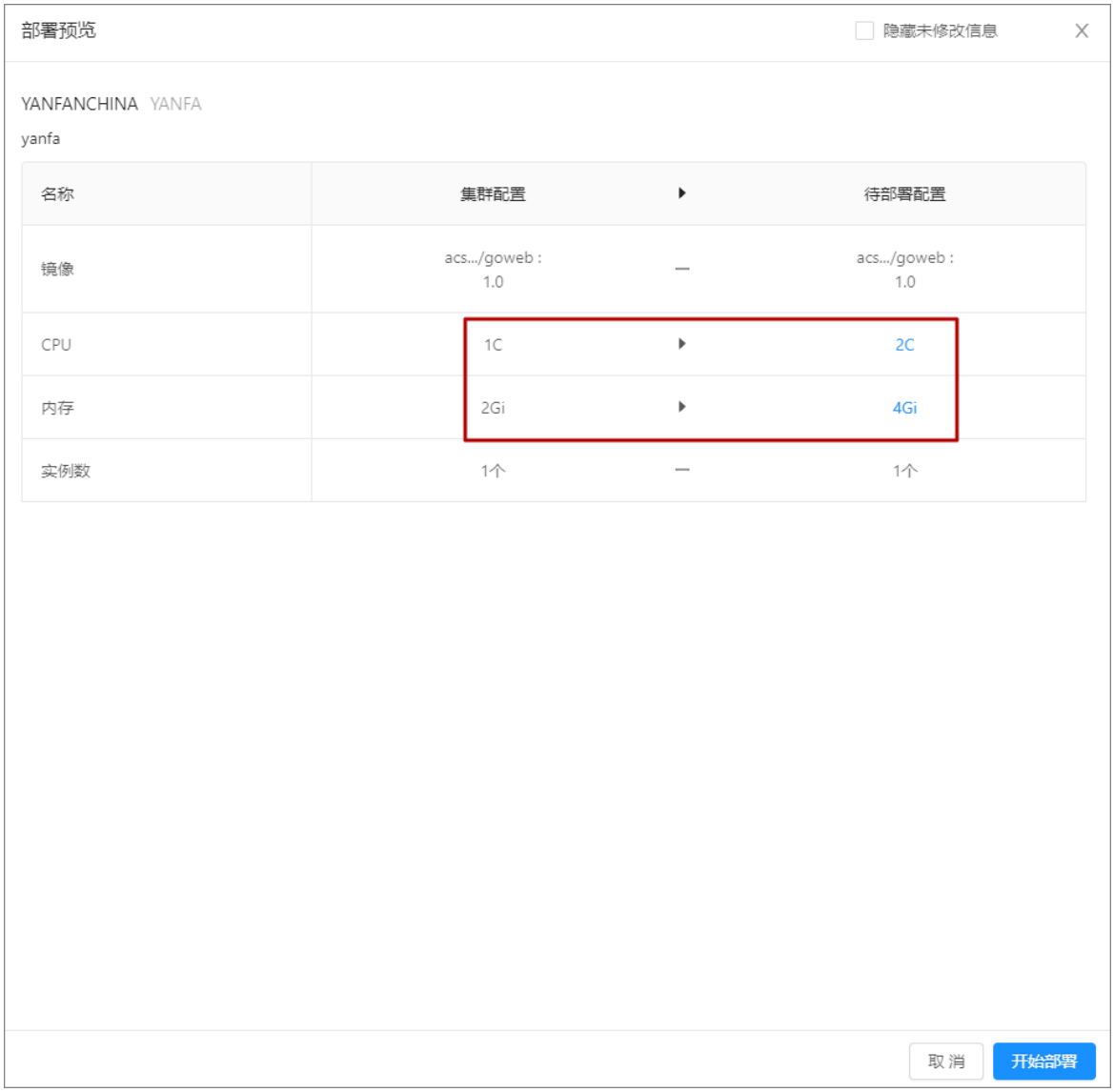
3. 在产品列表中，单击目标产品。
4. 单击 **应用** 页签，之后单击目标应用。
5. 单击 **容器** 页签，之后单击 **容器变配**。
6. 在 **应用变配规格** 对话框设置 **目标 CPU** 和 **目标内存** 参数，之后单击 **确定**。

7. 在 **新建运维单** 页面，将 **部署策略** 改为 **分两组**。



8. 单击 **资源变配**，在**部署预览**面板确认部署参数无误后单击 开始部署，之后单击 **确定**。

   这里主要确认当前实例的 CPU、内存和期望变更的 CPU、内存。



重要

变配之后，AntStack Plus 底座环境，容器会自动进行滚动重启；AntStack 1X 和 2X 底座环境，您需要分批次手动重启容器。详情请参见 重启容器。

4.3.2. 容器内存与 JVM 堆内存配置比

中间件各产品容器内存与 JVM 堆内存的配置比如下：

产品名称	JVM 堆内存建议配置

产品名称	JVM 堆内存建议配置
动态配置 DRM	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
注册中心 SOFARegistry	3/4 系统内存
DsrConsole	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
任务调度 TS	根据环境自动适配，适配方式如下： • 系统内存 = 2 GB；JVM = 1000 MB • 系统内存 = 4 GB；JVM = 3000 MB • 系统内存 = 8 GB；JVM = 6000 MB • 系统内存 = 10 GB；JVM = 9000 MB • 系统内存 = 14 GB；JVM = 12000 MB
消息队列 MQ	系统自动调整，无需设置。
消息队列 DMS	Console：系统内存 = 4 GB；JVM = 3 GB Broker：系统内存 = 8 GB；JVM = 4 GB
分布式事务 DTX	JVM 堆内存小于系统内存。 例如系统内存为 8 GB，JVM < 8 GB。

产品名称	JVM 堆内存建议配置
数据访问代理 ODP	根据环境自动适配，适配方式如下： • 系统内存 < 2 GB；JVM 使用系统默认值（根据不同版本进行调整） • 2 GB < 系统内存 < 6 GB；JVM = 2048 MB • 6 GB < 系统内存 < 10 GB；JVM = 4800 MB • 10 GB < 系统内存 < 14 GB；JVM = 5600 MB • 系统内存 > 14 GB；JVM = 9600 MB
Mesh	不涉及
API 网关	不涉及
消息队列 SOFAKafka	系统自动调整，无需设置。

4.3.3. 重启容器

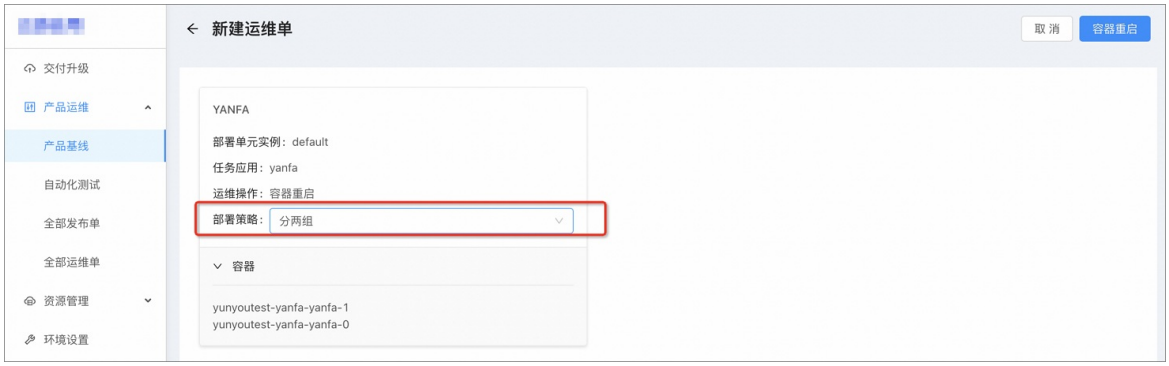
重启容器是指将指定容器（Pod）销毁后重新按期望调度出相同数量的新容器。

操作步骤

1. 登录云游 Local 控制台。
2. 在左侧导航栏，选择 **产品运维** > **产品基线**。
3. 在产品列表中，单击目标产品。
4. 单击 **应用** 页签，然后单击目标应用。
5. 单击 **容器** 页签，然后单击 **批量操作** 的 **重启** 按钮。

6. 选中一个或多个目标容器后单击 **确定**。
7. 在 **新建运维单** 页面单击 **容器重启**，然后单击 **确定**。

如果您选择了多个容器，建议将 **部署策略** 修改为 **分两组**，以免所有运行容器同时销毁，导致业务中断。



5.常见故障处理

5.1. 服务异常应急预案

问题描述

- 分布式事务控制台不可访问。
- 分布式事务功能异常。

环境检查

1. 登录共享中间件控制台，确认控制台可以正常访问。
2. 登录监控系统，观察 dtx-console/dtap-console、dtx/dtap 告警。

实施步骤

- 如果 dtx-console/dtx 容器为退出状态，尝试启动该故障容器。
- 如果 dtx-console/dtx 容器状态正常，webshell 进入容器，执行以下命令，检查 dtx-console/dtx 服务状态。

```
#ps -fe | grep java
```

- 如果 dtx-console/dtx 进程不在，尝试重启容器，会自动拉起业务容器。

结果验证

1. 登录告警界面，无业务告警。
2. 登录业务容器，确认 dtx-console/dtap-console、dtx/dtap 服务正常运行。

5.2. 故障排查快速指南

紧急情况

在遇到以下情况时，建议您立即联系技术支持团队。

- 机房掉电。
- 服务器宕机且未自动重启，或重启后依然有问题。
- 数据丢失。
- 误操作且无法回滚，可能会造成业务损失。

一般故障排查

针对一般故障，排查步骤如下：

1. 查看监控平台。

   在监控平台查看各容器（应用）的健康状况。如果有报错，则根据报错信息进行排查。
2. 查看日志。
3. 重启容器。

> **警告**
>
> 重启容器，请谨慎操作！

在不影响数据存储、业务正常运行的前提下，可以重启相应容器并检查其运行状态。

故障排查错误码

SDK 集成相关

异常码	异常详情	错误说明	解决方案
DTX-000	dtx component scanner init error.	SDK 启动失败，分布式事务不可用	/
DTX-001	Bean[%s] can not have @DtxTransaction and @TccBusinessAction both.	同一个 bean 不能既有 @DtxTransaction 又有 @TccBusinessAction 注解。	修改代码去掉 bean 的 @DtxTransaction 注解或者 @TccBusinessAction 注解，只保留一个。
DTX-002	the length of bizType[%s] can not be more than 30.	@DtxTransaction 注解上的 bizType 长度不能超过30 个字符。	将bizType长度控制在 30 字符以内。
DTX-003	init dtx configuration error.	初始化分布式事务配置项出现异常	/
DTX-003	Class [%s] can not have @TccBusinessAction and @TwoPhaseBusinessAction both.	同一个 bean 不能既有 @TccBusinessAction 又有 @TwoPhaseBusinessAction 注解。	修改代码去掉 bean 的 @TccBusinessAction 注解或者 @TwoPhaseBusinessAction 注解，只保留一个。
DTX-005	Unknown remoting service bean type, dtx will ignore this bean.	未知的远程服务类型，将忽略此 bean。	/
DTX-006	tcc 2.0 prepare error, txId: [%s].	TCC 2.0 一阶段方法执行失败。	排查业务一阶段方法执行失败原因。
DTX-007	tcc 2.0 prepare control, idempotency failed, txId: [%s].	TCC 2.0 一阶段方法幂等控制执行失败。	/

DTX-008	tcc 2.0 prepare method @ShardingKey param type must be String, className: [%s].	@ShardingKey 标记参数的必选是 String 类型。	修改 @ShardingKey 标记参数的类型为 String。
DTX-009	tcc 2.0 the java.sql.DataSource bean [%s] named by @TccBusinessAction.dataSourceBeanId() on class [%s] is not exist in spring applicationContext.	@TccBusinessAction 的 dataSourceBeanId 属性指定的 DataSource bean 不存在。	检查 @TccBusinessAction的 dataSourceBeanId 属性指定的 DataSource bean 是否存在。
DTX-010	tcc 2.0 the interface [%s] has no @TccBusinessAction method.	TCC 2.0 的参与者接口无 @TccBusinessAction 注解。	检查确保 TCC 2.0 的参与者接口上有@TccBusinessAction 注解。
DTX-011	tcc 2.0 action record is empty,txId:[%s],actionId: [%s].	TCC 2.0 的参与者记录为空，可能是参与者的一阶段未执行，或者一阶段方法未执行成功，此时执行二阶段提交出现此异常。	检查一阶段方法是否成功执行。
DTX-012	tcc 2.0 action commit failed,txId:[%s],actionId: [%s].	TCC 2.0 的参与者执行二阶段提交方法时出现异常。	排除二阶段提交方法执行异常的原因。
DTX-013	tcc 2.0 commit/rolback failed, txId or actionId is empty.	TCC 2.0 的参与者执行二阶段方法出现异常，原因是事务 ID 为空。	/
DTX-014	tcc 2.0 action rollback failed,txId:[%s],actionId: [%s].	TCC 2.0 的参与者执行二阶段回滚方法出现异常。	排除二阶段回滚方法执行异常的原因。
DTX-015	tcc anti-suspend, the 'TccAntiSuspendDAO' must not be null.	开启了 TCC 防悬挂，但是为配置TccAntiSuspendDAO。	配置 TccAntiSuspendDAO 详见用户指南防悬挂接入部分。
DTX-016	tcc anti-suspend(prepare method) intercept, txId: [%s].	TCC 一阶段方法重复执行，防悬挂拦截器被触发。	检查 TCC 一阶段方法被重复执行的原因。

DTX-017	tcc anti-suspend(prepare method) , the DataSource can't be null.	配置了防悬挂 TccAntiSuspendDAO，必须设置其 dataSource 属性。	配置 TccAntiSuspendDAO 详见用户指南防悬挂接入部分。
DTX-018	tcc anti-suspend, ITccAntiSuspendDAO didn't init.	TCC 防悬挂 TccAntiSuspendDAO 初始化未完成。	/
DTX-019	tcc anti-suspend failed, txId:[%s].	TCC 防悬挂控制出现未知异常	/
DTX-020	@TwoPhaseBusinessAction method , has no serviceId.	TCC 参与者接口的 @TwoPhaseBusinessAction 主键的 name 属性为空。	TCC 参与者接口的 @TwoPhaseBusinessAction 主键的 name 属性不得为空，且需要全局唯一。
DTX-021	unknown actionType:[%s].	TCC 参与者接口的 @TwoPhaseBusinessAction 主键的 actionType 属性配置了未知值。	TCC 参与者接口的 @TwoPhaseBusinessAction 主键的 actionType 属性只能是 local 或 remote。
DTX-022	report tcc branch failed, txId:[%s],actionId:[%s], serviceId:[%s].	创建 TCC 参与者的分支事务记录出现异常	检查 SDK 与分布式事务服务器的网络连接是否正常、检查服务器返回创建分支事务记录事务的原因。
DTX-023	report tcc branch error, txId:[%s],actionId:[%s], serviceId:[%s].	创建 TCC 参与者的分支事务记录出现未知异常	检查 SDK 与分布式事务服务器的网络连接是否正常。
DTX-024	Dtx transaction timeout error, total time cost :%d (s), more than %d (s).	分布式事务执行超时，导致事务超时回滚	检查事务执行超时原因，如需自定义事务超时时间，可以设置 @DtxTransaction 注解的 timeout 属性值（单位秒）。
DTX-025	dtx transaction rollbacked error,txId[%s].	分布式事务回滚出现异常	检查 SDK 和服务器之间的网络连接是否正常。
DTX-026	dtx transaction start failed, invoke dtx-server failed, request:[%s],response:[%s].	分布式事务开启创建主事务记录，分布式事务服务器返回创建失败。	检查分布式事务服务器创建主事务记录失败原因。

DTX-027	dtx transaction commit failed, request:[%s],response:[%s].	分布式事务提交，服务器返回提交失败。	检查分布式事务服务器提交分布式事务失败原因。
DTX-028	dtx transaction commit error, request:[%s].	分布式事务提交出现未知异常	检查 SDK 和服务器之间的网络连接是否正常。
DTX-029	dtx transaction rollback failed, request:[%s],response:[%s].	分布式事务回滚，服务器返回回滚失败。	检查服务器回滚分布式事务失败原因。
DTX-030	dtx transaction rollback error, request:[%s].	分布式事务回滚出现未知异常	检查 SDK 和服务器之间的网络连接是否正常。
DTX-031	Dtx transaction timeout error, total time cost：%d (s), more than %d (s).	分布式事务执行超时，事务回滚。	检查事务执行超时原因，如需自定义事务超时时间，可以设置 @DtxTransaction注解的 timeout 属性值（单位秒）。

框架相关

异常码	异常详情	错误说明	解决方案
DTX-200	There must be only one framework, but found two, [%s] and [%s].	运行框架只能有一个，但是系统识别到多个运行框架。	/
DTX-201	There are more than one remoting protocols on one bean, beanName: [%s] with type [%s].	当前运行环境下，未找到任何远程通信协议。	/
DTX-202	There must be only one configuration, but found two, [%s] and [%s].	配置项文件只能有一个，但是系统识别到至少 2 个配置项文件。	确保配置项文件只有一个。
DTX-203	There must be one IConfiguration implement.	无任何配置项文件。	确保配置项文件存在。
DTX-204	No framework found.	未识别到任何运行框架。	确保运行框架是分布式事务能支持的。

SDK 与云端服务通信相关

异常码	异常详情	错误说明	解决方案
DTX-301	There must be one address registry, but found none.	未找到事务云服务的地址。	提交工单，检查事务云服务是否健康运行。
DTX-302	init dtx-servers' tcp long-conections failed.	客户端与 DTX 服务端建立长连接失败。	提交工单，检查事务云服务是否健康运行。
DTX-303	unavailable dtx-server connections,serverIp:[%s]	客户端无法建立到 serverIp 的 TCP 长连接。	提交工单，检查事务云服务是否健康运行。
DTX-304	rpc (to dtx-server) invoke error.	向 DTX 服务端发送消息出现异常。	提交工单，检查事务云服务是否健康运行。

SQL 解析

异常码	异常详情	错误说明	解决方案
DTX-400	unknown dbType:[%s]	不支持的数据库类型。	目前支持 Oceanbase、MySQL、Oracle，其他数据库暂不支持。
DTX-402	parser sql error,sql:[%s]	解析 SQL 语句出现异常。	检查分布式事务是否支持 SQL 语法。
DTX-403	value type of param [%s]: [%s] is not support.	不支持的 SQL 语句表达式。	检查分布式事务是否支持 SQL 语法。
DTX-403	unknown dbType for in select subquery.	嵌套子查询不支持。	嵌套子查询不支持，请修改 SQL 语法。
DTX-404	insert's duplicateKeyUpdateClauses is not expression.	insert on duplicate key update 语句语法有错误。	检查 SQL 语句语法是符合标准 SQL 语法。
DTX-405	the insert sql's columns or valurClauses is null.	insert 语句语法有错误。	检查 SQL 语句语法是符合标准 SQL 语法。

DTX-406	aggregate method [%s] is not support, only support aggregate methods: SUM,COUNT,AVG,MIN,MAX.	集合函数目前只支持SUM、COUNT、AVG、MIN、MAX，其它集合函数不支持。	集合函数目前只支持SUM、COUNT、AVG、MIN、MAX，修改 SQL 语句。
DTX-450	There must be primaryKey in table[%s] of db[%s].	业务表必须有主键。	给所有业务表添加主键。

数据源相关（FMT 模式下）

异常码	异常详情	错误说明	解决方案
DTX-500	transaction [%s]:[%s] on db [%s] fulsh branchInfo to db failed.	保存 undo log 至 dtx_branch_info 表失败。	检查业务数据库是否正常。
DTX-501	transaction [%s]:[%s] on db [%s] report branchInfo to dtx-server failed.	向 dtx-sserver 发送创建分支事务记录消息失败。	检查客户端与 DTX 服务端网络连接是否正常、检查事务云服务服务器是否正常。
DTX-502	transaction [%s]:[%s] on db [%s] commit error	业务本地事务提交失败。	检查业务数据库是否正常
DTX-503	transaction [%s]:[%s] on db [%s] rollback error	业务本地事务回滚失败。	检查业务数据库是否正常
DTX-504	sql parser result is empty, sql:[%s]	SQL 语句解析结果为空。	检查 SQL 语法 DTX 是否支持。
DTX-505	'INSERT INGORE' and 'INSERT ON DUPLICATE UPDATE' statement must contain primaryKey column(s), sql:[%s]	'INSERT INGORE' and 'INSERT ON DUPLICATE UPDATE' 中不得使用自增主键句式。	'INSERT INGORE' and 'INSERT ON DUPLICATE UPDATE' 不支持使用自增主键，请在 SQL 语句的列与值中添加主键字段。
DTX-506	'UPDATE' and 'DELETE' statement must have where-statement, sql:[%s]	update 和 delete 语句必须有 where 条件。	修改 SQL 语句，添加 where 条件。
DTX-507	'UPDATE' statement can not update primaryKey column value(s), sql:[%s]	update 语句不得更新主键值。	修改 SQL 语句。

DTX-508	unknown sqlType, sql:[%s]	不支持的数据库类型。	目前支持 Oceanbase、MySQL、Oracle，其他数据库暂不支持。
DTX-509	In READ_COMMITTED mode, complex query is not supported, you can use READ_UNCOMMITTED mode, sql:[%s]	读已提交模式不支持复杂查询语句。	修改 SQL 语句为简单查询语句，或者使用度未提交模式。
DTX-510	Insert-select statement is not supported, sql:[%s]	Insert-select 语法不支持。	修改 SQL 语句。
DTX-511	branchLogItems is empty, nothing to undo, dbId:[%s]	业务库无 undo log，undo 操作返回。	/
DTX-512	check dirty failed, branchLogItem:[%s].	脏写校验失败，数据回滚异常。	人工介入订正数据。
DTX-513	insert branchInfo error, branchInfo:[%s]	保存 undo log 至业务数据库出现异常。	检查业务数据库是否正常。
DTX-514	lock table row failed, data may be locked by another transaction. db:[%s],tableName:[%s] ,primaryKey:[%s]	添加行锁失败，数据行锁正在被其他事务占用。	检查数据行锁占用事务未结束原因。
DTX-515	insert rowlock failed, txId = %s, had tried count: %d	保存行锁数据至业务数据库失败。	检查业务数据库是否正常、检查数据行锁是否被其他事务占用。
DTX-516	transaction:[%s] lock table row failed;row:[%s:%s] was locked by transaction:[%s].	行锁添加失败，行锁被其他事务占用。	检查数据行锁占用事务未结束原因。
DTX-517	insert statement primary-key's value must be simple value, sql:%s.	insert 语句中的主键字段必选是简单数据类型：字符串或数字。	修改主键数据类型。

DTX-518	Oracle insert statement primary-key's value can not be Sequence.nextVal/Sequence .currVal, sql:%s.	Oracle 数据库 insert 语句，主键不得使用sequence。	修改 insert 语句语法。
DTX-519	Insert statement primary-key's value can not be Expression, sql:%s.	insert 语句主键值不得事表达式。	修改 insert 语句语法。
DTX-520	Unsupport dbType:%s.	不支持的数据库类型。	目前支持 Oceanbase、MySQL、Oracle，其他数据库暂不支持。
DTX-521	unsupport sub query in where condition.	不支持 where 条件中的子查询。	修改 SQL 语句语法。
DTX-522	join select sql can not get one tableName.	join 语句获取多个表名，会产生歧义，不支持 join 语句。	修改 SQL 语句语法或者改成读未提交模式。
DTX-523	select from more than on tableName or sub select statement, can not get a tableName.	join 语句获取多个表名，会产生歧义，不支持 join 语句。	修改 SQL 语句语法，或者改成读未提交模式。
DTX-524	select from nest sub select statement, can not get a tableName.	不支持嵌套查询。	修改 SQL 语句语法，或者改成读未提交模式。
DTX-525	select sql has no tableName.	查询语句无表名称。	修改 SQL 语句语法。
DTX-526	unknown parameter, %s:%s.	不支持的数据类型。	修改 SQL 语句语法。
DTX-527	executeBatch error, statement:%s.	executeBatch 执行出现异常。	根据异常类型判断出错原因。
DTX-528	execute Statement, but sql is empty.	SQL 语句为空。	/

DTX-529	execute statement error, statement:%s.	执行 SQL 语句出现异常。	根据异常类型判断出错原因。
DTX-530	Statements is empty, no sql need execute.	SQL 语句为空。	/
DTX-531	there is not sql to executed.	SQL 语句为空。	/
DTX-532	while getAutoCommit() is true can not invoke executeBatch.	executeBatch 方法必选在 db 事务内执行。	请先开启 db 事务。
DTX-533	AddBatch count is not equal to parameters row count.	AddBatch 参数个数不匹配。	检查业务代码，确保正确使用了 AddBatch。
DTX-534	All parameters of batch prepareStatement must have addBatch() follow.	executeBatch 的参数设置之后必选调用 addBatch 方法。	检查业务代码，确保正确使用了 AddBatch。
DTX-535	Invoking executeBatch() method need all sql statements are addBatch(), but sql[%s] is not.	executeBatch 用法不正确。	检查业务代码，确保正确使用了 AddBatch。
DTX-536	Invoking executeBatch() method need all sql statements are addBatch(sql),but sql[%s] is not.	executeBatch 用法不正确。	检查业务代码，确保正确使用了 AddBatch。
DTX-537	Invoking executeQuery/executeUpdate/execute method need all sql statements are non-addBatch sql, but sql[%s] is addBatch.	addBatch 的 SQL 调用了 executeQuery/executeUpdate/execute 方法执行。	检查业务代码，确保正确使用了 jdbc。
DTX-538	PreparedStatement with addBatch sql cann't use limit-statement, sql:[%s].	PreparedStatement 的 addBatch 用法，SQL 语句中不支持带 limit 分页句式。	修改业务代码及 SQL 语句。

DTX-539	parameter count ambiguous, where-statement expect the count more than %d, but parameter count is %d.	PreparedStatement 的 SQL 语句的参数个数与实际参数个数不匹配。	检查代码，确保参数个数与 SQL 语句中参数个数相等。
DTX-540	parameter index ambiguous, where-statement expect the index is %d, but parameter count is %d.	PreparedStatement 的 SQL 语句的参数个数与实际参数个数不匹配。	检查代码，确保参数个数与 SQL 语句中参数个数相等。
DTX-541	parameter data ambiguous, where-statement expect the parameter row count is %d, but actual is %s.	PreparedStatement 的批量 SQL 语句的参数行数与实际参数行数不匹配。	检查代码，确保参数个数与 SQL 语句中参数个数相等。
DTX-542	query log error, querySql: [%s],prameters: [%s],isPrepare:%s."	查询 undo log 出现异常。	检验业务数据库是否可用。
DTX-543	check dirty data error,tableName: [%s],oldRows: [%s],newRows:[%s].	脏写校验失败,业务数据出现脏写。	人工介入订正数据。
DTX-544	auto-generate primary-key must be only one column,sql: [%s].	自增主键不能是复合主键。	修改主键为单列主键。
DTX-545	addBatch() must include primary-key(s) in sql, auto-generate primary-key don't support,sql: [%s].	addBatch 的 SQL 语句中必选包含主键，不支持自增主键。	修改 SQL 语句。
DTX-546	insert without primary-key(s), do not support batch insert ,sql: %s.	批量插入时，SQL 语句中必选有主键。	修改 SQL 语句，添加主键列至 SQL 语句中。
DTX-547	preparedstatement addBatch, insert sql's values must be single,the sql:[%s].	preparedstatement 的 addBatch 用法时，insert 语句不支持批量。	修改 SQL 语句。

DTX-548	parameter index ambiguous, where-statement expect the index is %d, but parameter count is %d.	SQL 语句需要的参数个数与实际的参数个数不匹配。	检查业务代码和 SQL 语句的参数是否匹配。
DTX-550	parameter data ambiguous, where-statement expect the parameter row count is %d, but actual is %s.	SQL 语句需要的参数个数与实际的参数个数不匹配。	检查业务代码和 SQL 语句的参数是否匹配。
DTX-551	parameter index ambiguous, where-statement expect the index is %d, but parameter count is %d.	SQL 语句需要的参数个数与实际的参数个数不匹配。	检查业务代码和 SQL 语句的参数是否匹配。
DTX-552	parameter data ambiguous, where-statement expect the parameter row count is %d, but actual is %s.	SQL 语句需要的参数个数与实际的参数行数不匹配。	检查业务代码和 SQL 语句的参数是否匹配。
DTX-553	the type of primary key must be 'String' or 'Number'.	主键字段的类型必选是字符串或者数字。	修改主键、或者修改主键数据类型。
DTX-554	can not get the primarykey of insert statement: [%s],parameters:[%s].	自增主键的 insert 语句，无法获取 insert 数据之后的主键值。	修改 insert 语句添加主键。
DTX-556	the columns count of sql's values is not equal to the columns count of the table, table's column count: [%d],clomuns values:[%d]	SQL 语句中的字段类与与业务表的字段列个数不匹配。	检查 SQL 语句是否正确，如果数据库表结构变更过，请重启机器刷新 DTX 缓存。
DTX-557	check undo datas result is true, undo do nothing, branchLogItem:[%s].	undo log 数据为空，undo 操作返回。	/
DTX-558	unknown sqlType, sqlType: %s, branchLogItem:[%s].	不支持的 SQL 语句类型。	查看 DTX 用户指南，修改 SQL 语句为 DTX 支持的 SQL 类型。

DTX-559	check dirty datas failed, the data is dirty, tableName: [%s],postRows: [%s],dbRows: [%s],branchLogItem:[%s].	数据脏写校验失败，业务数据出现脏写。	人工介入动作数据。
DTX-560	insert branch info into oracle db failed, txId:%s, actionId: %s.	向 oracle 数据库保存undo log 出现异常。	检查业务数据库是否正常运行。
DTX-561	connection must setAutoCommit(false), conn: %s, rowLocks: %s.	保存行锁记录时，未开启 db 事务。	/
DTX-562	preparedStatement only support single sql,but sql[%s] has more one sql.	preparedStatement 中不支持多个 SQL 语句。	修改 SQL 语句语法。
DTX-563	transaction [%s] on db [%s] executeQuery(%s) error.	executeQuery 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-564	transaction [%s] on db [%s] executeUpdate([%s]) error.	executeUpdate 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-565	transaction [%s] on db [%s] executeUpdate([%s], int) error.	executeUpdate 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-566	transaction [%s] on db [%s] executeUpdate([%s], int[]) error.	executeUpdate 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-567	transaction [%s] on db [%s] executeUpdate([%s], String[]) error.	executeUpdate 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-568	transaction [%s] on db [%s] execute([%s]) error.	execute 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-569	transaction [%s] on db [%s] execute([%s], int) error.	execute 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。

DTX-570	transaction [%s] on db [%s] execute([%s], int[]) error.	execute 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-571	transaction [%s] on db [%s] execute([%s],String[]) error.	execute 方法执行出现未知异常。	根据抛出的异常类型找出异常原因。
DTX-572	addBatch(String) is not allowed while getAutoCommit() is true.	addBatch(String) 方法必选在开启数据库事务之后使用。	在 addBatch(String) 方法前先开启 db 事务。
DTX-573	executeBatch() is not allowed while getAutoCommit() is true.	executeBatch 方法必选在开启数据库事务之后使用。	在 executeBatch 方法前先开启 db 事务。
DTX-574	don't support closeOnCompletion() ; need explicit invoke close() method to close the statement.	closeOnCompletion 方法不支持。	手动调用 close 方法关闭 Statement。
DTX-576	WrappedDtxDataSource.getUniqueDbId() is empty.	WrappedDtxDataSource 的 uniqueDbId 属性不得为空。	设置 WrappedDtxDataSource 的 uniqueDbId 属性，需要保证全局唯一。
DTX-577	WrappedDtxDataSource.targetDataSource is null.	WrappedDtxDataSource 的 targetDataSource 属性不得为空。	设置 WrappedDtxDataSource 的 targetDataSource 属性。
DTX-578	WrappedDtxDataSource.targetDataSource can not be type of IDtxDataSource.	WrappedDtxDataSource 的 targetDataSource 属性所指的 DataSource 对象不能是 IDtxDataSource 类型。	修改 WrappedDtxDataSource 的 targetDataSource 属性为非 IDtxDataSource 类型。
DTX-579	DruidDtxDataSource.jdbcUrl is empty.	DruidDtxDataSource 的 dbType 属性不得为空	设置 DruidDtxDataSource的 dbType 属性。
DTX-580	DruidDtxDataSource.userName is empty.	DruidDtxDataSource 的 userName 属性不得为空。	设置 DruidDtxDataSource的 userName 属性。
DTX-581	DruidDtxDataSource.password is empty.	DruidDtxDataSource 的 password 属性不得为空。	设置 DruidDtxDataSource的 password 属性。

5.3. JVM 配置问题排查应急方案

故障现象

- 应用所在 ECS 出现内存溢出。
- 应用 JVM 出现内存溢出的错误信息，且应用无法正常提供服务、事务无法开启和进行事务恢复。

排查步骤

1. 登录云游 Local，进入目标应用，打开 **容器** 页签。
2. 单击 **应用变配规格**，查看当前应用容器规格。

3. 登录报错容器，执行 `top` 命令，查看内存使用情况。

```
top - 11:11:11 up 4 days, 22:58,  0 users,  load average: 22.14, 22.11, 22.14
Tasks:  14 total,   1 running,  13 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 83.8 id,  0.0 wa,  0.0 hi,  0.0 si, 16.2 st
KiB Mem :  4194304 total,  2253924 free, 18446744+used, 38511104 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 31268236 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   956 admin     20   0   35.9g   1.6g  14076 S   1.0 40.5 103:00.15 java
     1 root      20   0  116112  13252   4036 S   0.0  0.3   1:56.24 supervisord
     9 root      20   0  112388   4104   3100 S   0.0  0.1   0:04.08 sshd
    11 root      20   0  204184   2492   1896 S   0.0  0.1   0:09.35 rsyslogd
    13 root      20   0   24792   1628    956 S   0.0  0.0   0:04.18 crond
    16 admin     20   0   13756   1632   1320 S   0.0  0.0   0:00.00 bash
 61897 root      20   0   18216   2116   1676 S   0.0  0.1   0:00.00 bash
 61915 root      20   0   62524   2064   1512 R   0.0  0.0   0:00.01 top
 61917 root      20   0   74632   2316   1380 S   0.0  0.1   0:00.00 crond
 61918 admin     20   0   11624   1192   1004 S   0.0  0.0   0:00.00 sh
 61920 admin     20   0   11756   1664   1276 S   0.0  0.0   0:00.02 bash
 66650 admin     20   0   11756    900    508 S   0.0  0.0   0:00.00 bash
 66653 admin     20   0    8488    380    292 S   0.0  0.0   0:00.00 sleep
113266 root      20   0   18216   2124   1676 S   0.0  0.1   0:00.01 bash
```

4. 执行 `ps -ef | grep dtx` 命令，查看应用 JVM 内存配置情况。

```
admin       956      16  1 6月22 ?        01:43:05 /opt/software/java/bin/java -Dzdal_password_encrypt=true -Dserialize.blacklist.file=blacklist -server -verbose
:gc -Xms1800m -Xmx1800m -Xmn680m -Xss256k -Xloggc:/home/admin/logs/gc.log -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/admin/logs -XX:ErrorFile=/home
/admin/logs/hs_err_pid%p.log -Dcom.alipay.sofa.middleware.config.log.level=INFO -Dcom.alipay.ldc.datacenter=default -Dcom.alipay.ldc.zone=default -Dzmode=false
 -Dspring.config.additional-location=/home/admin/app-run/dtx-server-server-1.5.0-executable/BOOT-INF/classes/config/application-prod.properties -Drpc_bind_netw
ork_interface=eth0 -jar /home/admin/app-run/dtx-server-server-1.5.0-executable.jar
```

根据 JVM 内存配置不同，可能有以下情况：

- JVM 内存配置不合理。

a. 检查内存配置是否按照应用配置的自适应脚本进行配置。

   例如容器规格是 2C4G，应用 JVM 内存配置是否是 1.8 GB，是否小于容器配置的内存配置，详参容器内存与 JVM 堆内存配置比。

b. 若 JVM 内存配置不合理，则内存溢出原因是自适应脚本未执行。

- JVM 内存配置正常，自适应脚本生效，仍出现内存溢出。

  a. 下载内存溢出时 Java 自动生成的 dump 文件，分析是否有程序逻辑错误，导致内存溢出。

  b. 查看 RMS 监控的应用内存使用情况。

     若应用在一段时间内内存使用平稳，出现问题时内存陡增，则排除应用内存泄漏，需要根据日志排查是否有流量陡增、事务恢复数过多，或其他异常原因。

  c. 若 Java 未生成 dump 文件，则排查 DTX 日志文件。

     查看 `dtx-recovery.log` 日志中进行事务恢复时的异常事务数，事务恢复数是否超出正常时的事务恢复数。

```
2021-08-04 11:54:06.654 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 11:54:51.713 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 11:55:06.716 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 12:04:07.706 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:27, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 13:02:26.486 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 13:00:48.062 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:27, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 13:09:04.295 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:487, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 14:06:23.593 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 14:06:37.635 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:343, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 14:09:49.843 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:782, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 14:46:36.036 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 14:46:40.966 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:27, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 14:47:35.998 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 14:47:41.000 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 15:01:36.269 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 15:01:41.264 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 15:02:36.284 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 15:02:41.282 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 15:04:41.309 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 15:04:46.303 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:27, taskId:default#prod#default#dtx_remote#business_activity#business_action
2021-08-04 15:06:46.325 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:0, taskId:default#prod#default#dtx_remote_compatible#business_activity#business_action
2021-08-04 15:06:51.327 [RECOVERY-PROD-DEFAULT-DEFAULT] INFO  DTX-RECOVERY - the num of record will be recoveried:28, taskId:default#prod#default#dtx_remote#business_activity#business_action
```

> **重要**
>
> 问题发生后，若异常事务数超出正常时的数量，需要业务方进一步排查出现大量异常事务的原因。

应急预案

针对上述两种故障现象，分别有以下两种应急预案：

- **机器内存溢出**

  i. 重启机器，并升级机器规格。

  ii. 机器重启后检查应用 JVM 配置是否合理。

  iii. 若不合理，则重启对应应用，使应用内存自适应脚本执行生效。

     可在云游通过强制重发应用将应用重启，并建议对应用容器进行扩容。

- **应用 JVM 内存溢出**
  i. 重启应用，释放内存。
  ii. 重启完成后对容器进行规格变配，提升容器规格。
  iii. 对应用进行扩容，扩容后再次重启应用，使内存自适应脚本正常执行。
  iv. 执行完成后，检查应用 JVM 的内存，查看 JVM 内存配置是否合理。

5.4. 其他问题

部署 SOFA 应用时出现报错“DTX-301”，且 InstanceId 配置正确

```
Caused by: com.alipay.dtx.common.exception.DtxRpcException: DTX-301:There must be one address com.alipay.dtx.registry,
        at com.alipay.dtx.registry.server.AddressSubscribe.subscribeServerAddress(AddressSubscribe.java:69)
        at com.alipay.dtx.rpc.client.address.impl.ServerAddressManager.init(ServerAddressManager.java:58)
        ... 29 common frames omitted
```

排查方法

1. 查看 `/home/admin/conf` 日志下是否有对应 InstanceId 的 dtxserver 信息。
   - 若有对应的 dtxserver 信息，查看 DTX 目录下的 `dtx-remote.log` 报错日志是否有错误信息，若有相关错误信息，根据错误信息进行排查。
   - 若没有对应的 dtxserver 信息，检查 InstanceId 和 ACVIP 的 endpoint 是否正确，若不正确则修改为正确的。
2. 登录 dtxserver 机器，查看 `common-error.log` 具体错误信息，根据错误信息排查具体原因。

当主机房宕机后，备机房 dtx-server 如何找到备机房对等的参与者做事务恢复？

主备机房的 dtx-client 会与主备机房的 dtx-server 保持长连接，当主机房宕机后，备机房的 dtx-server 会通过备机房会从保持长连接且匹配的 dtx-client 中随机选一个执行提交或回滚操作。如果有一个分支事务提交回滚失败，会不断重试，直至最终成功。