阿里云
大数据轻量专有云

运维指南

产品版本：V1.1.0

文档版本：20180327

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
	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。	 **禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	**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。	单击 **确定**。
courier字体	命令。	执行 cd /d C:/windows 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。	swich *{stand | slave}*

目录

1 MaxCompute

1.1 运维工具介绍

本章节主要介绍和MaxCompute产品相关的运维工具。目前在专有云MaxCompute运维场景中，着重使用的是新上线的大数据管家，通常都和专有云大数据产品一起输出。在一般状况下，所有的日常运维查询/变更都会在大数据管家中有相对应的入口和工作流来支持。

当然，在目前情况下，也会用其他工具来做补充，但是今后都会陆续整合到大数据管家中来。

1.1.1 大数据管家（BCC）

大数据管家（BCC）是为阿里各个大数据产品量身定做的运维管理平台，当前运维的大数据产品包括MaxCompute、DATAWORKS、AnalyticDB等。

大数据管家以服务组件的形式为产品提供运维功能，每个服务组件包含产品树结构、配置、自动化和手动服务自检、工作流、包管理、全局搜索、日志搜索、指标信息和Metrics信息，还包括各服务组件自定义的一些功能和轻量云中特有的功能。

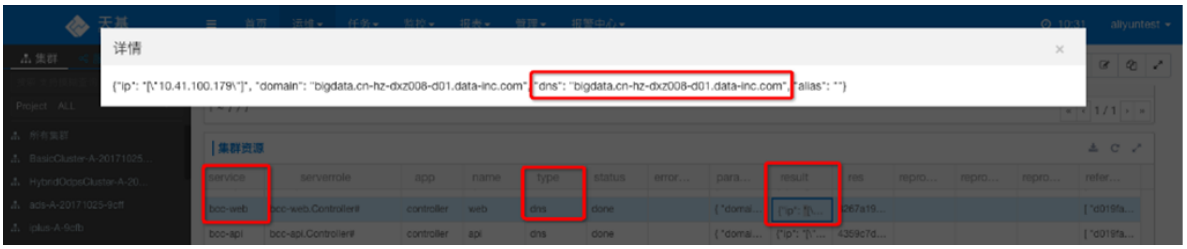
登录大数据管家的方式如下所示：

1. 登录天基，选择**运维** > **集群运维**，在**Proiect**区域查找**bcc**，进入BCC集群。

图 1-1: 查找BCC集群

2. 查看集群资源中**service**为**bcc-web**，**type**为**dns**的资源的**result**信息，获取到BCC的访问域名。

图 1-2: 获取BCC的访问域名。



3. 使用上一步骤查找到的域名，登录大数据管家。

登录完成之后，可以看到大数据管家首页展示，主要分为产品列表、任务、监控、自愈、运行中任务、管理、全局搜索。

图 1-3: 首页展示

大数据管家中各个功能的具体使用方式及说明，请参见系统中**帮助文档**内的《*BCC*用户手册》。

图 1-4: 帮助文档

1.1.2 Job Summary

作业结束后，输出日志中的一段信息与Log view的**Summary**信息是一样的，我们称之为job summary。

Job Summary对用户非常有用，从中可以看到很多有用的Job信息：

- Job run mode

  作业的运行模式通常有两种，即**fuxi job**和**service job**。分别表示普通的fuxi作业、准实时作业（使用SQL加速服务跑的作业）。如果希望通过准实时来提升作业速度却没有得到期望的结果，那么可以看看该作业的运行模式是不是对应的准实时模式。如果不是，那么很可能准实时模式没有开启，可以找运维人员通过管控平台开启对应 project 的SQL加速开关；如果开关已经打开，那么可能是开启的SQL加速服务可用实例数已经不足以运行当前作业。

- task的instance数

  可以知道该task有多少instance并发计算，如果逻辑上该task是可分片并发执行的，但该值很小，而用户的数据量又很大，那么说明并发度可能不够，可以通过配置来提升并发度。

- Instance的执行time

  该信息展示了同一任务各个instance执行所耗的最小时间（min）、最大时间（max）和平均时间（avg）。如果这三个值不均衡，大小相差很多，那么说明存在长尾instance，很可能是数据倾斜问题引起，个别instance由于处理的数据量远大于其他instance而成为长尾，需要用户找到长尾并关注数据的分布，从而优化数据及查询。

- input records和output records

这部分展示了同一任务各个instance的输入与输出的最小记录数（min）、最大记录数（max）和平均记录数（avg），如果输入或输出记录数的三个值不均衡，那么说明存在数据倾斜问题。需要关注该task的处理逻辑与数据分布。

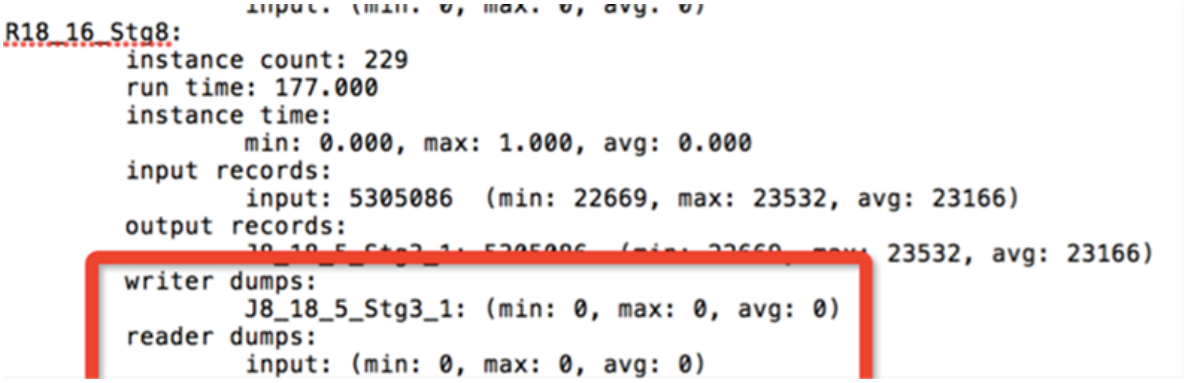
- Job的run time和task的run time

  该信息表明了job和task的实际运行时长（包含等待资源与调度等耗时），而每个task的run time是累加了其依赖task的run time的。

- writer dumps和reader dumps

  在某些job summary中，用户还会看到如下图所示的writer dumps和reader dumps。

  图 1-5: 显示结果



shuffle排序阶段，由于为instance分配的内存不足而导致的在外部排序中，数据从内存dump到磁盘的次数。

正常情况下应该是0，如果不为0，那么需要考虑增加分配给该task实例的内存。

1.2 常用运维命令

1.2.1 盘古组件常用运维命令

盘古的命令通常是pu和puadmin，请务必自行输入**各自命令 --help + 回车**，来查看完整的帮助。

- 类似linux的ls命令来查看指定文件夹的文件。

  `pu ls`

- 上传本地文件到盘古。

  `pu put`

- 查看文件的meta信息。

pu meta

- 显示所有的盘古master信息。

  puadmin gems

- 列出所有chunkserver的详细信息。

  puadmin lscs

- 查看版本信息。

  puadmin --buildinfo

- 对单台chunk server进行维护的时候，需要设置chunkserver的状态。具体操作如下：

  1. 查看当前状态。

     pyadmin cs -stat tcp://x.x.x.x:10260

  2. 设置chunkserver为shutdown状态来从集群脱离。

     pyadmin cs -stat tcp://x.x.x.x:10260 --set=shutdown

  3. 维护完毕后，重新加回到集群中。

     pyadmin cs -stat tcp://x.x.x.x:10260 --set=normal

1.2.2 伏羲常用运维命令

- 伏羲的运维命令是r，是rpc.sh的一个包装。

  alias r='sh /apsara/deploy/rpc_wrapper/rpc.sh'

- 查看所有服务job和service。

  r al

  通常在生产集群上，返回的list比较大。

- 查看单个任务状态。

  r wwl jobname

- 所有任务资源使用总览。

  r cru

- 停止指定作业。

  r jstop jobname

- 查看集群总资源。

r ttrl

- 查看集群空闲资源。

  r tfrl

  其他选项请直接运行命令获得。

1.2.3 MaxCompute常用命令

- MaxCompute通常使用odpscmd来进行运维操作，输入后，会得到一个MaxCompute的shell。

  odpscmd

- h命令可以得到完整的帮助文档。常用的基本操作有：

表 1-1: 常用命令

命令	说明
whoami;	查看使用者的云账号
show p;	查看历史执行过的instance
wait <instanceID>;	生成相应instanceID的logview
kill <instanceID>;	停止指定instance
tunnel upload/download;	数据上传下载工具
desc project <projectname> -extended;	查看project空间的使用情况
export <projectname> /local/file/path;	导出项目内所有表的DDL语句
create table tablename (...);	新建一张表
select count(*) from tablename;	查询表

2 大数据开发套件

2.1 登录服务器并查询信息

整套DataWorks是基于天基部署，应用信息和相关的数据库信息可以在相应的天基地址中查询到。

下文将介绍如何登录服务器并查询相应信息。

2.1.1 登录服务器

前提条件

查询到了相关的服务器地址，每个应用部署在两台机器上，应用包、配置信息都是相同的。

操作步骤

1. 确保网络环境可通，以及查询到跳板机的机器IP。

   说明：

   如果发现Ping不通，则说明网络环境不对。

   登录服务器可以使用putty或其他的软件使用。

2. 登录跳板机。
3. 执行ssh ip命令 ，这些IP应该和ag是免密登陆的，就是可以直接ssh上去。
4. 成功登录机器后，执行如下命令，切换到admin账户下。

   ```
   su - admin
   ```

5. 执行如下命令，切换至应用所在的目录。

   ```
   cd /home/admin/
   ```

   说明：

   需要到base-biz-alisa的目录下，执行如下命令：

   ```
   cd /home/admin/base-biz-alisa
   ```

2.1.2 查询应用信息

登录到各个应用所在的服务器上，以及找到相应应用所在的目录后，便可查询相关的应用信息。

查询应用包的配置信息

为方便运维管理，DataWorks的所有应用都是统一由*base-biz*开头，配置文件名称为*config.properties*。

除了gateway和cdp之外，其余应用的配置文件目录均为*/home/admin/APPNAME/target/APPNAME.war/WEB-INF/classes/config.properties*。

gateway的配置文件目录为*/home/admin/alisatasknode/target/alisatasknode/conf/config.properties*。

cdp的配置文件目录为*/home/admin/cdp_server/conf/config.properties*。

> **说明**：
> APPNAME表示各个具体组件的名称，如base-biz-alisa。

查看应用日志

为方便运维管理，DataWorks中除了gateway和cdp之外，其余应用的日志文件目录均为*/home/admin/APPNAME/logs/APPNAME.log*。

gateway的日志文件目录为*/home/admin/alisatasknode/logs/alisatasknode.log*。

cdp的日志文件目录为 */home/admin/cdp_server/logs/cdp_server.log*。

> **说明**：
> APPNAME表示各个具体组件的名称，如base-biz-alisa。

如何登录数据库

DataWorks中用到的数据库是mysql和postgresql（简称pg）两种，其中只有base-biz-phoenix这个应用使用了postgresql数据库。

各个应用对应的数据库：

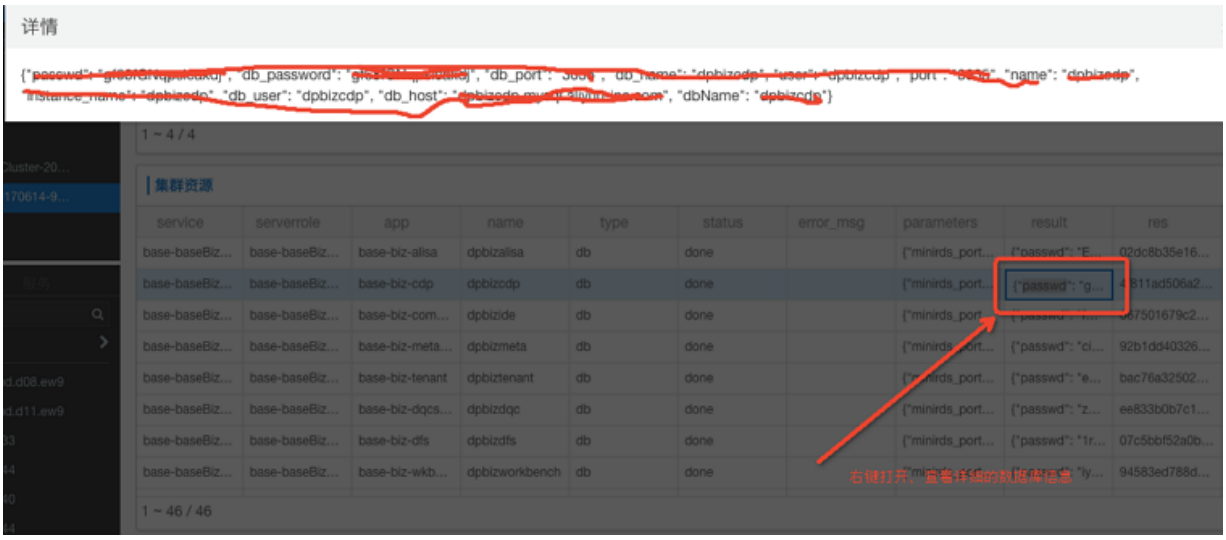
表 2-1: 数据库

应用	数据库	数据类型
base-biz-alisa	dpbizalisa	rds
base-biz-commonbase	dpbizide	rds
base-biz-dfs	dpbizdfs	rds

base-biz-dqcsupervisor	dpbizdqc	rds
base-biz-metaservice	dpbizmeta	rds
base-biz-tenant	dpbiztenant	rds
base-biz-wkbench	dpbizworkbench	rds
base-biz-cdp	dpbizcdp	rds
base-biz-phoenix	dpphoenix	postgre(pg)

查看各个RDS信息。

图 2-1: RDS数据库信息



登录到base-biz-phoenix的服务器上，可以查看到PG的链接信息。more /home/admin/base-biz-phoenix/target/base-biz-phoenix.war/WEB-INF/classes/config.properties |grep pg

- mysql

  登录到ag（admingateway机器，一般指中控机）上，执行如下命令：

  mysql –h db_host –P db_port –u db_user –D db_name –p db_password --default-character-set=utf8

- pg

  登录到ag（admingateway机器，一般指中控机）上，执行如下命令：

  /u01/pgsql/bin/psql -h ${db_host} -p${db_port} -U${db_user} -d${db_name}

说明：

可以在跳板机上执行history | grep mysql命令，查询历史命令信息，复制执行即可。

2.1.3 重启应用服务

某些情况下，重启应用是一种行之有效的解决方法。各个应用的启动、停止、重启都是通过应用自带的脚本实现的，但各个应用脚本名称和存放路径却略有差别。

2.1.3.1 一般应用重启

除base-biz-gateway和base-biz-cdp外，其他应用启动（start）、停止（stop）、重启（restart）方式均为：

`$/home/admin/APPNAME/bin/jbossctl start（stop/restart）`

说明：

应用重启前，需要切换为admin账号权限。

APPNAME换成需要操作的应用，例如：重启base-biz-alisa服务。

`$/home/admin/base-biz-alisa/bin/jbossctl restart`

监测应用启动（停止）是否成功，首先执行ps -xf命令查看进程是否存在，然后curl本地80端口checkpreload.htm文件，查看服务是否OK。

2.1.3.2 base-biz-cdp重启

base-biz-cdp的启动（start）、停止（stop）、重启（restart）方式为：

`$/home/admin/cdp_server/bin/appctl.sh start（stop/restart）`

base-biz-cdp启动的验证方式同上。

2.1.3.3 base-biz-gateway重启

base-biz-gateway的启动（start）、停止（stop）、重启（restart）方式为：

`$/home/admin/alisatasknode/target/alisatasknode/bin/serverctl start（stop/restart）`

监测base-biz-gateway正常启动的方式为：

`$ tail -f /home/admin/alisatasknode/logs/heartbeat.log`

心跳汇报正常则应用服务正常，否则服务不正常。

图 2-2: 心跳日志

2.2 应用运维

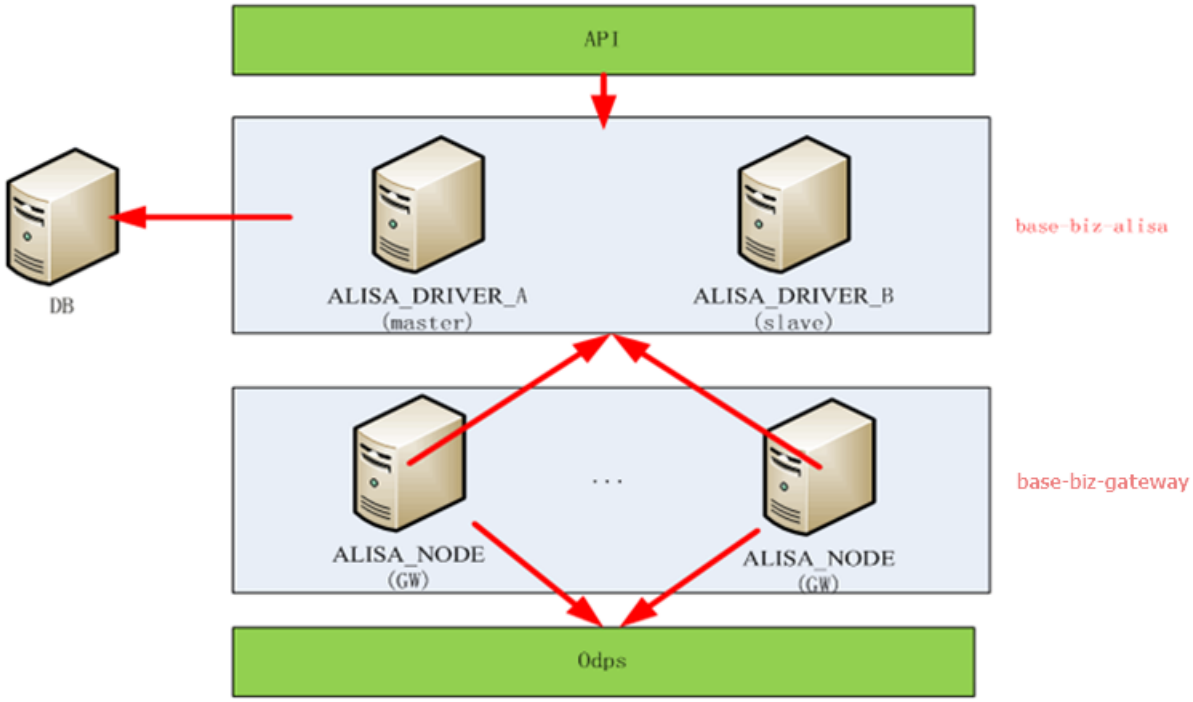
DataWorks由多个组件系统组成，DataWorks主要的任务操作运维请参见《用户指南》，这部分主要描述针对DataWorks平台中比较独立的两个组件的运维帮助：Alisa运维帮助和CDP运维帮助。

2.2.1 Alisa运维帮助

2.2.1.1 Alisa部署架构

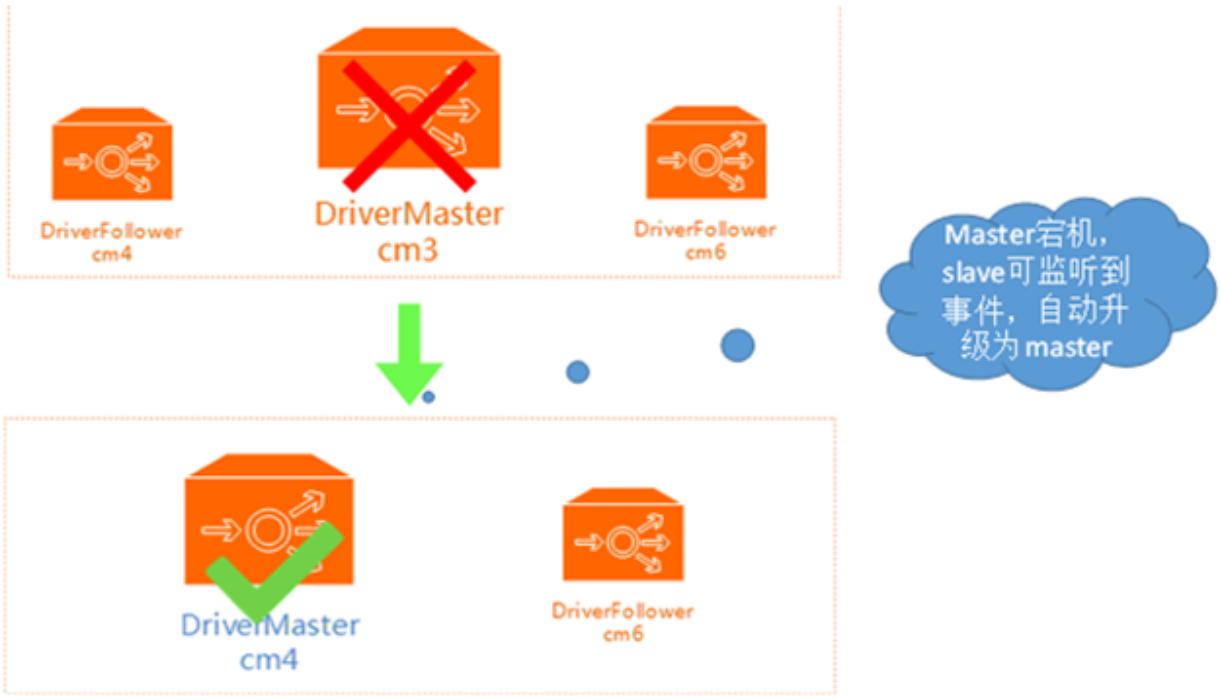
Alisa是一套支持多种任务类型、可水平扩展、高可用性的分布式执行引擎系统，整个系统主要分为两个模块：ALISA_DRIVER和ALISA_NODE，整体架构如图 *2-3: ALISA*架构所示。

图 2-3: ALISA架构



ALISA_DRIVER：控制模块，主要负责任务的资源管理，内部依赖于数据库的方式实现了主备机制，一个环境只允许一个master。内部实现了一套内资源管理模型，负责将任务合理的分配到指定的节点上执行，可以水平扩展；主备机制保证了应用的可用性，图 *2-4:* 主备机制中三台ALISA_DRIVER，中间一台master，两台follower，位于不同的服务器cm3，cm4，cm5。如果cm3出现异常，则其余两台依旧可以正常工作。

图 2-4: 主备机制



ALISA_NODE：俗称gateway，系统的任务执行节点，部署在服务器上的一个agent，负责接收任务、执行任务、收集任务执行日志及查询日志等功能。对应的服务器可基于任务量扩容，一台服务器部署一个agent包。

2.2.1.2 Alisa的资源管理模型

Alisa是整个DataWorks平台的最底层， gateway成为任务真正执行的所在服务器，每个任务都会占用物理资源（cpu、内存、磁盘等）。如果一台机器上任务并发数过大，可能会导致机器无法正常使用，甚至有宕机的风险。Alisa实现了一套完整的资源管理模型：资源组-集群-gateway模型。

图 2-5: 资源组-集群-gateway管理模型

引入概念：

- 槽位（slot）：作为衡量一个任务所占资源大小的单位，设定每个sql任务占用1slot，同步任务占用10slot。
- Gateway：可以指一台部署了ALISA_NODE服务的服务器，一台服务器目前只允许部署一个agent包，任务真正执行所使用的服务器。使用槽位设置一台gateway运行并发执行的任务数。
- 集群（gateway）：Alisa将gateway使用集群管理模式，一台gateway只能从属于一个集群，利用集群隔离不同的gateway；一个集群下可允许有多个gateway，多个资源组。
- 资源组（group）：虚拟资源隔离概念，可以理解为调度资源。主要目的是做到项目之间的隔离，每个项目一个资源，不同的资源组从属于不同的集群，所以最后执行任务的gateway也不同，做到相应的隔离。使用槽位作为一个资源组允许执行的并发数。

通过资源模型，可以做到资源的管控和一些场景需求：

- 任务物理隔离：不同项目可以使用不同的资源组和不同的集群，项目A使用资源组A，对应的集群A，任务真正执行到gatewayA上；项目B则使用资源组B，这样做到了两个项目之间物理

上的完全隔离。实现该方式的前提是必须有两台服务器作为gateway，如图 2-5: 资源组-集群-gateway管理模型所示。

- 任务资源竞争：如果目前gateway无法物理扩容，所有项目都共用一个集群，则需要通过资源组槽位的控制来进行合理分配。例如：目前gateway总共就100个槽位可用，项目A有100个任务，项目B 有10个任务，但这10个任务必须跑起来。则可以将项目A对应的资源组槽位数设置为90，项目B对应的资源组槽位数设置为15，这样既保证A永远只能够占用90个槽位，又保证有空余的10个槽位给B使用。

2.2.1.3 如何扩容Gateway

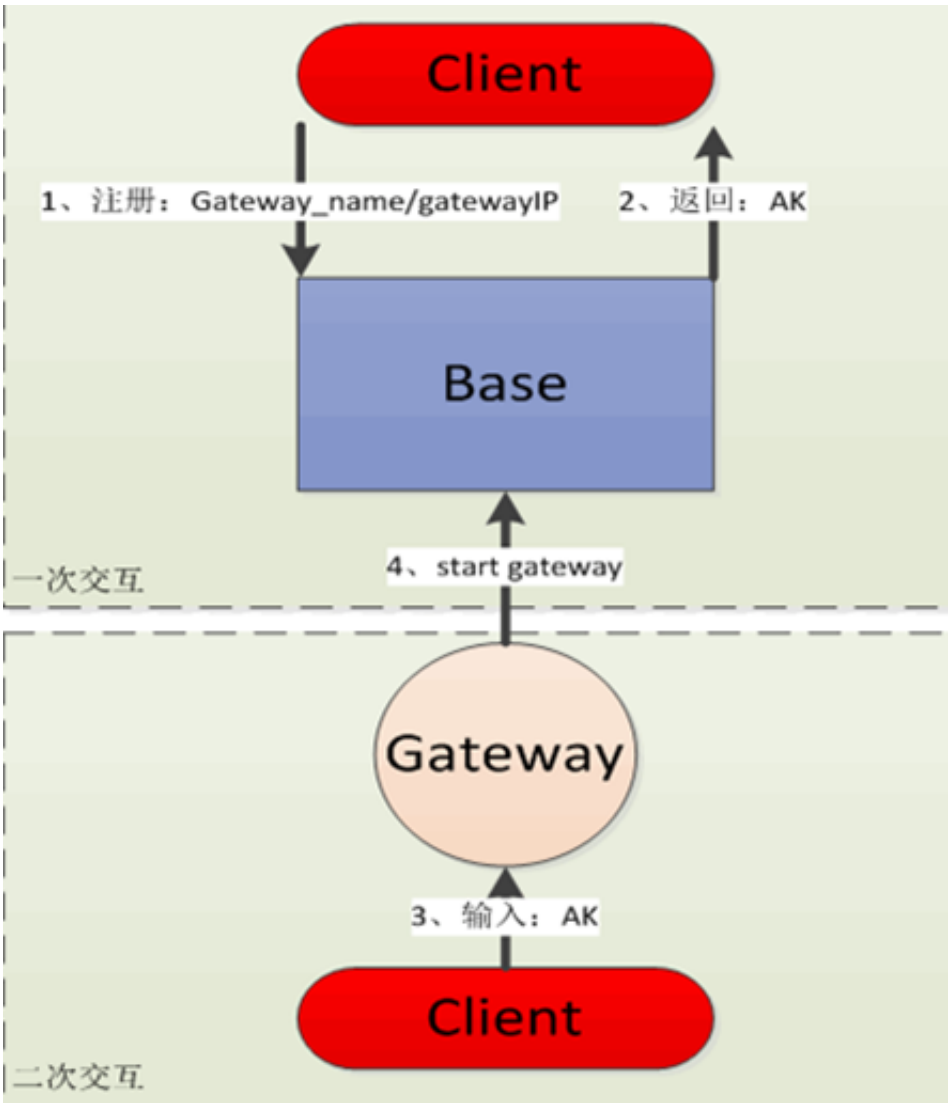
ALISA_NODE为整套DataWorks平台的最底层，也是任务真正启动执行的进程所在服务器，遇到以下情况时，需要扩容gateway。

- 任务量过大：目前标配的2台gateway已经不能满足当前的业务量，需要增加机器支撑。
- 需要到任务上的物理隔离：例如不同项目之间需要有不同gateway执行、有些特殊任务需要有资源随时执行，防止竞争导致任务无法执行等场景。

实现说明

将一台新的Gateway注册到DataWorks中，主要通过两次握手实现：先将gateway信息注册到平台中心，然后基于平台返回的AK启动gateway的服务，若两者信息一致，则可注册成功。如图 2-6: 注册流程图所示。

图 2-6: 注册流程图



一次交互：主动将gateway信息（hostname、hostIP）注册到平台中控系统，平台中控返回一对AK。

二次交互：将AK注入到gateway指定配置中，启动gateway，gateway主动访问平台中控服务，注册完成。

操作步骤

1. 登录DataWorks运维平台的租户管理界面，添加调度资源后，输入资源名称，单击**确定**。

> **说明**：
>
> 如果已经有调度资源了，此步骤省略。

图 2-7: 新增调度资源

2. 选择图 2-7: 新增调度资源中对应的调度资源，进入到**配置服务器**，单击**添加服务器**，输入要添加的服务器的主机名（hostname）和ip（hostname -i），确定后进入图 2-8: 配置服务器界面。

图 2-8: 配置服务器

3. 经过上述步骤后，已经将服务器信息注册到了DataWorks中，但是还不能服务，呈现如图中的**已停止**。 此时拿到上述页面中的唯一标识和密码（单击**显示AK密码**），获取到用户名密码，然后登录到服务器上部署ALISA_NODE安装包。
4. 登录到服务器中，使用刚获取的用户名和密码开始执行命令。

> **说明**：
> 如果唯一标识不是zz_开头，请在执行命令输入username时加上。下面步骤可查看界面中的**执行初始化**相应命令。

5. 登录到需要扩容的gateway机器上，安装部署alisatasknode、datax安装包，或者按照gateway的部署方式部署gateway镜像。
6. 部署完毕后，修改配置项。

   修改/home/admin/alisatasknode/target/alisatasknode/conf/config.properties下的两个参数：

   `alisa.driver.access.username=zz_用户名`

   `alisa.driver.access.password=密码`

其中，用户名为上述添加调度资源时使用的唯一标识，密码为对应的调度资源密码。

7. 重新启动gateway。

   sudo su admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart。

2.2.1.4 如何修改资源组、gateway槽位信息

在DataWorks的使用过程中，随着业务量的增长，最初的资源部署模式会有一定的限制，如很多任务开始处于**等待资源**状态，gateway的并发数设置不合理，需要调整等。

前提说明

由于目前没有界面化的操作底层数据，需要运维人员登录到base-biz-alisa的服务器上，执行不同功能点的命令，如执行下文查询槽位信息和修改最大槽位信息的命令，参数说明如下：

- groupname：需要修改的资源组的名称，唯一标识。
- clustername：添加的资源组对应的集群名称（可以从base-biz-alisa的数据库中alisa_group和alisa_node表中查询）。
- nodename：gateway的名称。

执行命令时，需要有一个用户名密码，可以在base-biz-alisa的数据库中alisa_access_account表中查询到。任意一个username/password都可以。

查询槽位使用情况

1. 查询当前环境中所有资源组槽位使用情况。

   curl -u username:password --digest -H "Accept: application/json" -H "Content-type: application/json" -X GET “http://localhost:7001/alisa/4.0/resource/group”

2. 查询当前环境中所有gateway的槽位使用情况。

   curl -u username:password --digest -H "Accept: application/json" -H "Content-type: application/json" -X GET “http://localhost:7001/alisa/4.0/resource/node”

   返回结果：

   - useslot：当前已使用槽位。
   - maxSlot：最大槽位数。

   如果这两个值相等，这说明槽位被打满了，无法继续执行新的任务。

修改最大槽位信息

1. 修改某个资源组的最大槽位（下面是修改为200，可以按需设置）。

```
curl -u username:password --digest -H "Accept: application/json" -H "Content-type: application/json" -X PUT –d ‘200’ “http://localhost:7001/alisa/4.0/resource/cluster/{clustername}/group/{groupname}/maxslot”
```

2. 修改某个gateway的最大槽位（下面是修改为200，可以按需设置）。

```
curl -u username:password --digest -H "Accept: application/json" -H "Content-type: application/json" -X PUT –d ‘200’ “http://localhost:7001/alisa/4.0/resource/cluster/{clustername}/group/{nodename}/maxslot”
```

返回结果：

- useslot：当前已使用槽位。
- maxSlot：最大槽位数。

如果这两个值相等，这说明槽位被打满了，无法继续执行新的任务。

2.2.2 CDP运维帮助

DataWorks中的数据同步使用了阿里云产品数据集成（Cloud Data Pipeline，简称CDP），CDP是阿里集团对外提供的稳定高效、弹性伸缩的数据集成平台，为阿里云大数据计算引擎（包括MaxCompute、ADS、OSPS）提供离线（批量）的数据进出通道。

目前CDP支持数据通道包括（但不限于）：

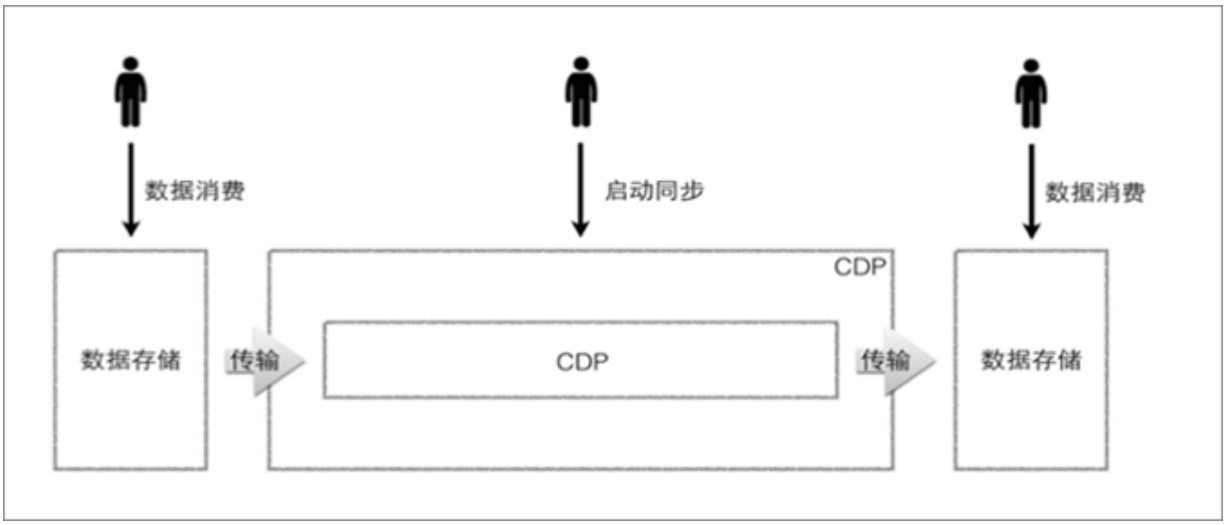
- 关系型数据库：RDS（MYSQL、SqlServer、PostgreSQL）、DRDS、Oracle。
- NoSQL数据存储：OCS。
- 数据仓库：MaxCompute、ADS。
- 非结构化存储：OSS、FTP。

2.2.2.1 原理和概念

目前CDP支持批量数据同步模式，离线数据同步指的是数据周期性（例如每天、每周、每月等）、成批量地从源端系统传输到目标端系统。对于离线数据同步系统，数据以读取Snapshot（快照）的方式从源端传输到目的端，离线同步存在生命周期，一个离线同步的任务有起始同样也有结束状态。

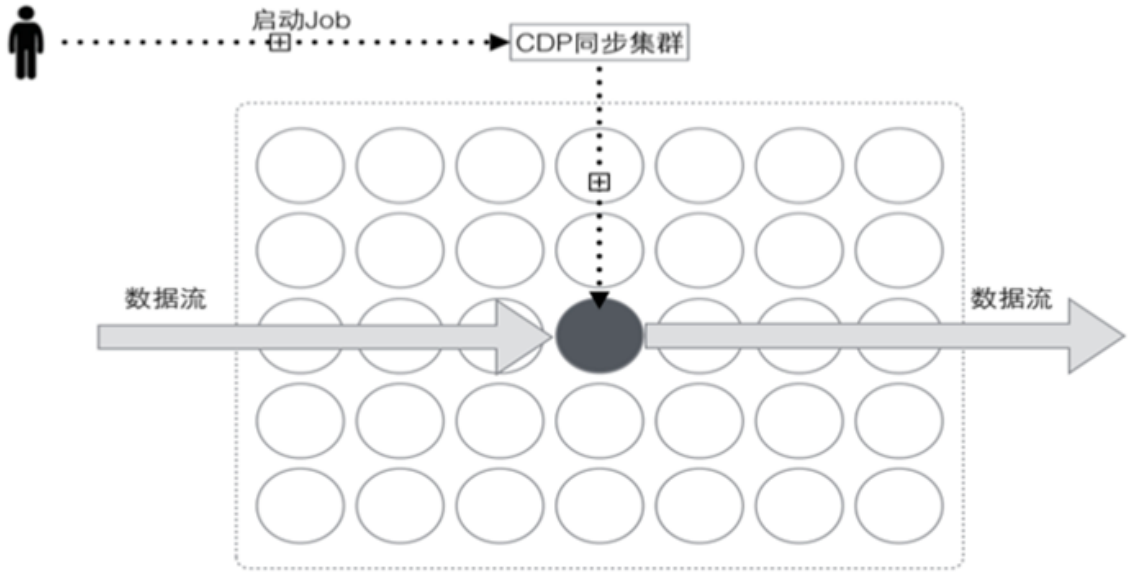
CDP只完成数据同步/传输过程，并且整体数据传输过程完全控制于CDP同步集群模型下，同步的通道以及同步数据流对用户完全隔离。同时，CDP本身不提供传输同步数据流的消费功能，即您不能直接通过CDP的API消费数据流，所有针对数据操作，您必须在同步数据流两端存储端操作，如图 *2-9: CDP*同步模型所示。

图 2-9: CDP同步模型



CDP提供一套抽象化的数据抽取插件（Reader）、数据写入插件（Writer），并基于此框架设计一套简化版的中间数据传输格式，从而达到任意结构化、半结构化数据源之间数据传输之目的。从用户角度来看，一个CDP运行Job如图 *2-10: CDP Job模型*所示。

图 2-10: CDP Job模型



图中的虚线代表API调用控制，灰色方向箭头代表数据流向。圆形节点代表底层一台数据同步执行节点，黑色节点代表正在执行当前数据同步业务的执行节点。

下面简要描述您使用CDP Job API触发调用CDP的Job同步过程：

1. 使用CDP Job启动API，向CDP服务端发起调用，提交一个离线数据同步Job。

2. CDP收到Job API请求后，将负责做必要的安全和权限校验。
3. 校验通过后，CDP下发相应的Job到执行集群节点启动离线数据同步任务。
4. Job启动后，根据您提供源端（Reader）、目的端（Writer）的配置信息，加载并初始化相关插件，连接两端数据源，开始数据同步工作。
5. Job运行过程中，将随心跳向CDP汇报当前进度、速度、数据量等关键运行指标，您根据Job的状态API实时获取该Job运行状态，直至Job运行结束（成功或者失败）。

CDP底层使用了离线数据同步工具DataX。可以简单的理解为CDP是DataX的云化、服务化。CDP接受到任务请求时，会通过集群资源调度服务（现在是Alisa）下发到执行机器，在执行机器上启动DataX同步进程。实际的数据流动为：**数据源头** > **DataX进程内存** > **数据目的段**。

2.2.2.2 CDP-Console

CDP-Console是CDP提供的，基于命令行交互方式的操作CDP管理工具，在DataWorks的执行节点即alisa gateway上面安装有可用的CDP-Console。

目录与文件结构

CDP Console的目录文件树如下所示：

```
cdp-console/
|-- README
|-- bin
|   `-- cdp
|-- conf
|   `-- cdp.properties
|-- lib
|   |-- cdp-console-1.1.0-SNAPSHOT.jar
|   |-- cdp-sdk-java-1.1.0-20150123.024540-90.jar
|   |-- ...
`-- template
    |-- odps2rds_job.json
    |-- pipeline.json
    |-- rds2odps_job.json
    `-- stream2stream_job.json
```

其中：

- /bin 目录下存放Console执行启动命令。
- /conf 目录下存放Console配置信息。
- /lib 目录下存放Console执行需要的jar包。
- /template 目录下存放Console提供的样例配置。

配置文件

Console配置文件会存放CDP Console相关的配置信息。

Console在启动前会根据CDP_HOME寻找相关的配置文件，并自动加载使用该配置。

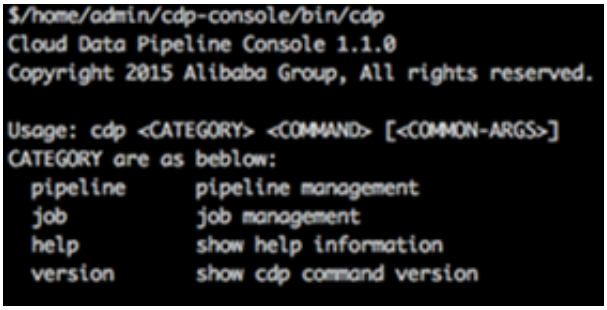
配置文件路径为 *${CDP_HOME}/conf/cdp.properties*，配置文件以**.properties**格式提供，具体说明如下：

```
# CDP console版本
client.version=201602262222
# CDP服务访问点
service.url=http://cdp.aliyun.com/api
# connection超时设置，可选，默认5000
service.timeout.connection=5000
# socket超时设置，可选，默认120000
service.timeout.socket=120000
# 用户的AK信息
auth.security.id=
auth.security.key=
# 默认的pipeline
user.pipeline=
# 是否为verbose打印，取值[false|true]，默认为false
client.verbose=false
# 输出格式设置，取值[text|json]，默认为text
client.output=text
# 客户端连接服务器重试次数和重试间隔
client.retry.time=5
client.retry.interval=1000
```

命令行使用

在bash命令行执行${CDP_HOME}/bin/cdp命令，当出现如下类似帮助的字样，表示安装和启动成功。

图 2-11: CDP Console命令启动



Console提供两种输入方式方便用户进行CDP的认证和鉴权，包括使用cdp.properties的配置文件，或者直接在命令启动作为参数传入。

命令行参数优先级高于配置，即两者同时存在时，优先选择命令行参数。

默认情况下，用户不需要指定AUTH信息，此时Console使用了配置文件中的**auth.security.id**和**auth.security.key**值作为鉴权参数，这两个参数需要分别为合法的Access Key ID 和 Access Key Secret。

例如执行命令：`cdp pipeline -list`。

由于没有指定AUTH信息，Console默认使用cdp. properties中的配置作为访问CDP服务的鉴权配置。而在一些情况下，一个CDP Console可能被多个客户使用，以完成数据同步功能。因此一套id+key的cdp. properties配置无法满足需求，需要将鉴权信息作为命令行参数传入，具体由命令行的调用者给出，具体格式为：

`cdp <CATEGORY> <COMMAND> -security id:key`

其中–security 参数后需要跟阿里云账号的id和key，中间使用冒号分隔，具体id和key的值和上面cdp.properties一致。当您指定security参数时，Console默认直接使用您命令行指定的鉴权信息。即同样的参数命令行传递优先级高于cdp.properties配置文件。

公共参数

公共参数是指一部分可在多个命令参数搭配使用的参数，例如指定命令行输出格式等。公共参数大部分在客户端配置文件中也可配置，当两者同时提供时，以命令行参数指定值为准。另一般可被多个用户共享使用的参数，多为配置文件配置。

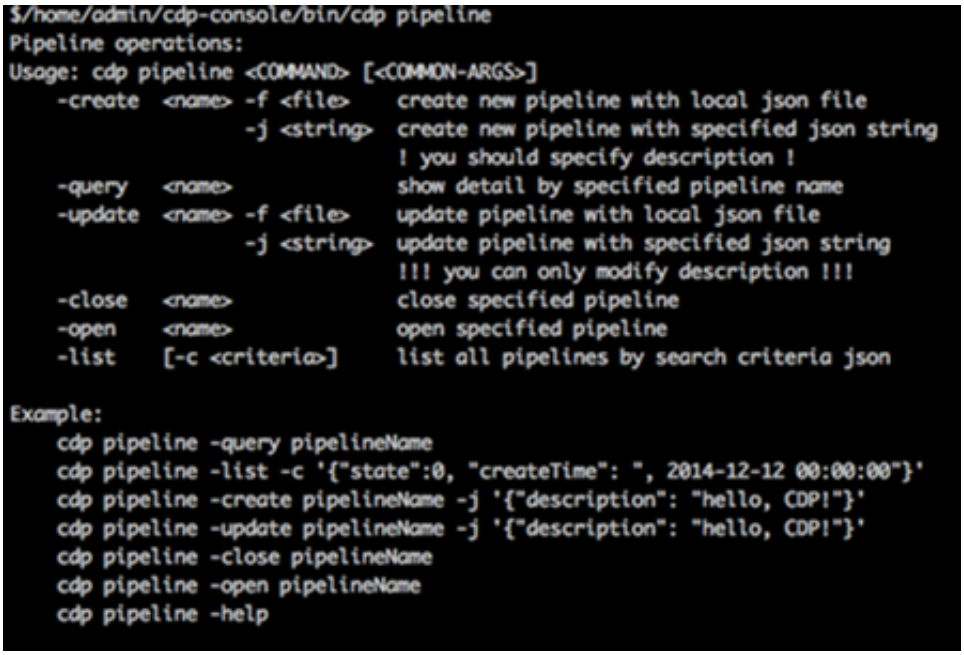
> **说明**：
> 鉴权参数实际上也属于公共参数一部分。

- -security：阿里云账号的id和key，中间使用冒号分隔。
- -url：指定CDP服务访问点。cdp.properties配置项service.url和此命令行功能一致，一般情况下仅连接一套CDP服务，此配置项在配置文件中设置即可。该参数适配Console所有命令。
- -p：指定操作作业对象属于的管道pipeline。对于离线作业Job的增删改查需要在一个管道Pipeline内完成，此参数设置了作业对应管道。cdp.properties配置项user.pipeline和此命令行功能一致具体如在指定的管道启动一个离线同步作业：`cdp job -start -p ${pipeline}`。

Pipeline管理

CDP Console针对管道Pipeline提供了create（创建）、query（查找）、close（关闭）、open（打开）、list（检索）等操作。CDP Console执行Pipeline类目命令的示例如图 *2-12: CDP Pipeline*相关命令所示。

图 2-12: CDP Pipeline相关命令



作业管理

CDP Console针对作业提供了start（启动）、query（查找）、list（检索）、log（日志）、stop（停止）、status（状态）等操作。

此类命令需要给出作业编号<id>以及管道名字<pipeline>，作业编号在启动一个作业时由CDP Server返回给客户端。您可以选择在命令行终端通过-p指定Pipeline，或者通过cdp.properties配置文件指定，优先级为命令行>配置文件。CDP Console执行Job类目命令的示例如图 *2-13: CDP Job相关命令*所示。

图 2-13: CDP Job相关命令

2.2.2.3 如何创建Shell DataX任务

在DataWorks中创建Shell类型同步任务，可以直接启动DataX命令行进行数据同步，其他具体的命令行工具部署在Alisa Gateway上，在DataWorks中创建Shell任务进行调用，原理都是类似的。您可通过以下步骤创建Shell任务。

1. 登录DataWorks控制台。
2. 进入**数据开发**页面，右键单击**任务开发**选择**新建任务**。
3. 填写新建任务弹出框各 配置项，选择SHELL节点类型。

图 2-14: 新建任务

4. 在shell节点中填写如下代码。

```
#!/bin/bash
shell_datax_home='/home/admin/shell_datax'
mkdir -p ${shell_datax_home}
shell_datax_config=${shell_datax_home}/${ALISA_TASK_ID}
echo '"
{
    "job": {
        "setting": {
            "speed": {
                "byte": 10485760
```

```
            },
            "errorLimit": {
                "record": 0,
                "percentage": 0.02
            }
        },
        "content": [
            {
                "reader": {
                    "name": "streamreader",
                    "parameter": {
                        "column": [
                            {
                                "value": "${bizdate}",
                                "type": "string"
                            },
                            {
                                "value": "${hour}",
                                "type": "string"
                            },
                            {
                                "value": 19890427,
                                "type": "long"
                            },
                            {
                                "value": "1989-06-04 00:00:00",
                                "type": "date"
                            },
                            {
                                "value": true,
                                "type": "bool"
                            },
                            {
                                "value": "test",
                                "type": "bytes"
                            }
                        ],
                        "sliceRecordCount": 100000
                    }
                },
                "writer": {
                    "name": "streamwriter",
                    "parameter": {
                        "print": false,
                        "encoding": "UTF-8"
                    }
                }
            }
        ]
    }
}
"' > ${shell_datax_config}

bizdate=$1
hour=$2
datax_params='-p "-Dbizdate=${bizdate} -Dhour=${cyctime}"'

echo "`date '+%Y-%m-%d %T'` shell datax config: ${shell_datax_config}"
echo "`date '+%Y-%m-%d %T'` shell datax params: -p \"-Dbizdate=${bizdate} -Dhour=${hour
}\""
/home/admin/datax3/bin/datax.py ${shell_datax_config} -p "-Dbizdate=${bizdate} -Dhour=${
hour}"
```

```
shell_datax_run_result=$?

rm ${shell_datax_config}

if [${shell_datax_run_result} -ne 0]
then
    echo "`date '+%Y-%m-%d %T'` shell datax ended failed :("
    exit -1
fi
echo "`date '+%Y-%m-%d %T'` shell datax ended success ~ "
```

说明：

- JSON模板由基本配置Setting、reader和writer组成，根据需要修改相应的配置，即可完成相应通道的数据同步工作。本示例是一个完整的全流程的shell任务示例，包含了常见的所有考虑点，比如临时文件重名、删除、任务状态、变量替换等。
- shell_datax_config=${shell_datax_home}/${ALISA_TASK_ID}表示当前shell任务一次执行临时代码文件在磁盘上的路径，不要修改，这里已经保证了多个任务文件名不会相同了，依赖ALISA的环境变量。
- echo命令将datax任务的配置文件重定向到上面的文件中，后续datax命令执行时，会读取配置文件的内容。注意这里使用了datax的变量占位符功能，json中有${bizdate}和${hour}模板。
- bizdate=$1和hour=$2是获取调度参数，本质上是将调度参数和datax里面定义的占位符关联起来，shell任务定义的多个调度参数通过空格分隔。
- /home/admin/datax3/bin/datax.py ${shell_datax_config} -p "-Dbizdate=${bizdate} -Dhour=${hour}" 执行datax同步进程，注意datax的命令行格式，**${shell_datax_config}**是配置文件地址，**-p**系列参数是用于参数变量替换。
- 任务执行后，根据datax同步进程成功或失败，shell程序成功和失败。这里也进行了相关帮助信息的打印和临时文件的清理。请不要混淆datax同步进程的成功和其他清理临时文件成功直接的关系。
- 如果需要创建其他shell任务，原理都是类似的，比如cdp console的任务。关键是配置文件JSON部分的不同。

2.2.2.4 如何调优任务运行速度

CDP数据同步并发原理

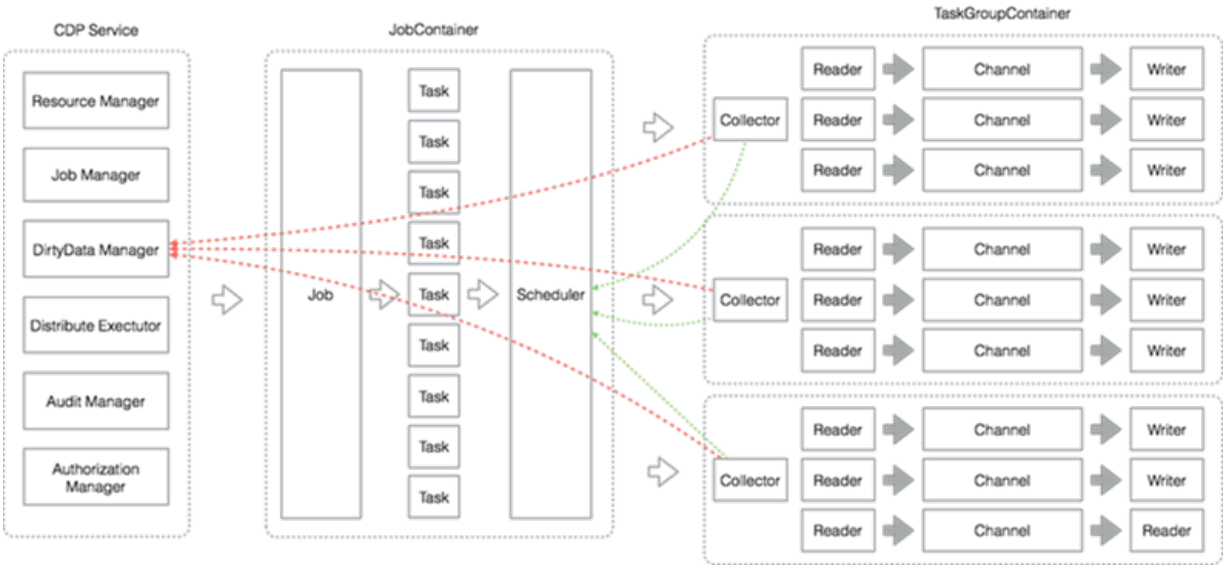
一个同步作业Job为您提供的视角，在运行时，会根据切分逻辑切分为多个Task，多个Task组成一个TaskGroup进行管理。对于每一个Task都有对应的Reader和Writer，Reader读取源头数据存

入中央缓冲区Channel，Writer从中央缓冲区Channel获取数据写入目的端。具体切分方式主要控制在Reader端，Reader端完成切分后，Writer端根据Reader端切分的份数切分成指定份数，保证Reader和Writer的1:1关系。

- 关系数据库RDBMS：根据切分列最小值最大值获得区间，将区间划分多个子区间，形成SQL Where字句得到各个Task的读取SQL。
- MaxCompute：根据分区MaxCompute数据表文件大小切分。
- OSS：根据文件粒度切分。

切分一般根据数据源不同会有区别，不限于此处讨论的内容。Writer端切分一般需要和Reader端协调一致，保证Reader-Writer一一对应关系。具体流程图如图 *2-15: Job*切分流程图所示：

图 2-15: Job切分流程图



具体任务切分的份数是由指定的作业速度和数据源类型决定的。每个Task的Channel对应的速度是固定的，目前CDP公共云生产环境为1MB/S，指定每个同步作业Job的执行速度，比如10MB/S，则预计切分数量为：**作业速度 / 每个Channel速度。这里切分数量ChannelNumber为10个Channel，即会有10个并发Task执行线程。**

DataX进程根据ChannelNumber要求完成指定切分，并发读取数据源数据写入目的端。实际切分时，切分份数可能超过ChannelNumber，目的是保障尽量切分的块比较小而均匀，但是同时运行的Task执行线程数量仍然为ChannelNumber，切分数也可能小于ChannelNumber。具体由于数据源限制，如OSS、FTP实际是根据读区文件数量切分的。

Reader->Channel->Writer中：Channel为中间缓冲池子，可以完成流量控制，具体为每隔指定时间如20毫秒检查Channel中数据变化状态，协调读写线程，保证Channel中数据流量的变化维持在预定值范围内，如1MB/S（Bytes流量速度）或者1M Record Per Second（记录流量速度）。

总结：CDP或DataX并发同步的并发执行线程数由作业运行速度、数据源类型影响和决定。需要合理调整作业速度，指导切分算法逻辑进行任务调优。

MaxCompute

在底层实现上，MaxCompute根据您配置的源头项目/表/分区/表字段等信息，通过Tunnel从MaxCompute系统中读写数据。

对于读取：支持读取分区表、非分区表，不支持读取虚拟视图。当要读取分区表时，需要指定出具体的分区配置，比如读取t0表，其分区为pt=1，ds=hangzhou，那么您需要在配置中配置该值。当要读取非分区表时，不能提供分区配置。表字段可以依序指定全部列，也可以指定部分列，或者调整列顺序，或者指定常量字段，也可以指定分区列。

说明：

由于是通过Tunnel读取底层文件，所以同步时不支持MaxCompute SQL对数据做裁剪和变换。

对于写入：当要写入分区表时，需要指定出具体的分区配置，如果分区不存在同步进程会创建指定分区；如果为非分区表，则不配置分区信息。

在数据同步时，并发切分是在Reader端进行，MaxCompute则是在MaxCompute Reader插件中完成切分split操作。目前MaxCompute支持2种切分模式：

- 根据分区partition切分，比如配置分区为pt=*，则找出正则匹配的所有分区，每个分区一个并发task。
- 根据Record切分，这种模式是在分区切分的基础上再做记录的切分，根据一个分区Record总数和切分份数计算具体数据读区偏移量，然后各个并发task读区指定偏移初的数据。

切分完成后任务根据作业速度A、单线程通道速度B（一般为1MB/S）可以计算出任务的并发读写线程数（C=A / B）。注意任务实际切分数为D，可能大于C（为保障切分均匀切分较细），等于C，或者小于C（比如按照分区切分，如果仅1个分区，则1个切分）。

关系型数据库MYSQL、Oracle、PostgreSQL等

关系型数据库的读写统一采用各个数据库提供的JDBC接口完成。以关系数据库MYSQL的一个表demo为例：

假设demo表有id，name两列。其中id的取值范围为[1, 100]，假设计算需要切分为5份，则切分的区间为：[1 <= id < 20]，[20 <= id < 40]，[40 <= id < 60]，[60 <= id < 80]，[80 <= id < 100]，[id is null] 实际切分份数可能有出入，出于精细化切分，切分均匀的考虑。区间切分完成后，可以通过多个线程或进程并发读取各个区间的数据。

这里一个完整的同步作业即将demo表数据同步为一个Job，切分后每个区间的同步为一个Task，多个Task（一般为5个）方便进行管理，分布式模式下减少资源消耗，抽象为一个TaskGroup，如5个Task组成一个进程，避免每个Task一个进程的消耗。

说明：

上面的切分寻找了一列id，实现时使用了数据库聚合函数min、max找到区间范围，切分列的选取建议选择主键、或者有索引的整型列。请注意具体切分算法的原理，对于任务性能调优较为重要，在非数据库性能、网络传输带宽非瓶颈时，一般调整切分能有较为明显的改善。

OSS、Ftp等

以OSS为例：若一个bucket中有多个object，可以配置一个object名称前缀表示需要拖取的object范围，如果有10个object需要同步，则简单的切分为10个Task，并发读取多个object。对于本地文件或Ftp文件还可考虑进一步的文件内部切分，类似MaxCompute（原ODPS）。可以简单的等价为一个目录下多个文件，并发读取每个文件。

常见的调优方法

根据如上的原理分析，CDP或DataX的并发切分是根据数据源头进行的，切分份数受作业运行速度影响，对于数据同步速率常见的调优方式有：

- **MaxCompute（原ODPS）**：默认按照Record方式进行切分，对于政企输出中的Data IDE，可保证按照作业配置运行速度 / 每个Channel的速度（部署时确定一般为1MB/S）。如果速度不理想，请排查：作业配置速度是否过小、任务执行机器网卡、CPU、内存、负载情况，可使用tsar --traffic命令。
- **关系数据库**：对于这类情况，建议检查切分列是否配置就绪，作业速度配置是否符合预期，机器网卡负载情况等。对于Oracle需要特殊注意，Oracle对于数字类型全部按照Number类型存储（整数和浮点数），Data IDE中切分列检查难以探测到合理的切分列，任务配置时请留意此处配置。切分列如果选择非主键或无索引，min、max聚合检查效率问题，可能导致数据读取SQL执行计划慢。对于关系数据库Dump等，后续会提供非JDBC接口的导出工具，便于批量的数据初始化上云。

调整示例

以一个TB级别的Oracle同步为例，目标Data IDE环境系统情况为：

- 单个作业最大速度（20mbps）：job.max.speed=20971520。
- 整个应用空间总速度（200mbps）：pipeline.job.bandwidth=200。
- 每个通道速度（5mbps），计算切分个数：channel.speed.byte=5242880。
- 一个应用空间资源槽位个数为：400，每个切分通道使用资源数目为：2。
- 目标环境中的Oracle为超大表，每日增量在TB级别，网卡带宽不是瓶颈。

按照一般切分原理执行SQL聚合查询类似为：

`select min(usernum),max(usernum) from tabal_name where begintime > time1 and begintime < time2`

由于历史原因此表没有数字型主键、没有符合要求的索引，导致上面Oracle SQL长时间不返回，无法应用CDP自带的切分任务逻辑。该切分逻辑的逻辑为：找到区间范围内的指定切分列的最小值，最大值，根据需要切分的份数将区间范围[min, max]切分为多个小区间，然后多线程并发执行多个sql查询，获取各个小区间的数据，并发通过MaxCompute tunnel写入MaxCompute 。按照常规的流程难以满足数据同步速率的要求，故推荐以下方案：

- 方案1：如果将作业速度设置为5mbps（5mbps/ channel.speed.byte 为1，一个切分数目，占用2个资源），则可同时运行200个CDP任务。如果200个CDP作业同时运行，每个作业实际提供3mbps速度，则可达到600mbps的流量。这个时候Oracle能够提供的并发连接数，每个sql查询速度是瓶颈了。将任务分散在多个库多个表上面可缓解次问题。
- 方案2：如果将作业速度设置为100mbps，实际上Oracle sql查询提供不了这么快的速度，但是占用资源数目为：切分数目*每个切分资源数目=作业速度100mbps/ (channel.speed.byte 5mbps)* 每个切分槽位2=40。总资源数目为400，则可同时运行10个同步作业。由于sql连接数目减少，每个sql提供的查询速度性能更高一些，大约为8mbps，所以最大速度为80mbps左右。但是此时却将整个DataWorks的Gateway资源占用完了，目前资源的申请占用暂时是静态的。
- 方案3：这里的Oracle表有begintime字段，依照此字段做分区，每30分钟一个分区，一天48个分区。可配置一个小时调度任务，每小时执行一次，同步该小时区间范围内的数据（通过where条件过滤），调度会每天触发24次该任务，完成该天的数据同步。
- 最后建议提供的配置作业方式为：配置多个同步作业，每个作业使用小带宽（占用资源槽位少），并发执行，如此可完成较大速度的同步传输。目前使用这种方式，针对单个Oracle数据库，测试任务运行速度可达到50-100mbps。

其他说明

CDP或者DataX对于关系型数据库使用了较为一致的JDBC模型进行数据同步，带来了数据源的丰富性，能够识别非法数据（比如把字符串abc写入int中），JDBC作为一个高层协议牺牲了一部分效率，速度是难以达到数据库Dump底层文件这类效率的。问题的焦点一般在任务合理的切分上，以Oracle为例：目前Oracle主键是一种在用模式，rowid（纯静态数据）、随机抽样、文件dump等模式后续会逐渐提供到。第三个常见点在机器资源上，如果您的环境中涉及到较大规模的数据量，同步任务较多，请考虑DataWorks执行集群扩容，以增加实际的并行数。

2.2.3 常见故障处理

本部分主要介绍一些DataWorks的常见问题、故障的排查处理方法。

2.2.3.1 调度任务日志解析

DataWorks中执行任务的所有日志都有统一的格式输出，主要分为三个部分：任务调度信息、任务内部执行日志、结束信息，使用了“**==========**”作为分隔符。

调度信息

任务执行时，会将各个信息打印到日志中，方便后期排查问题，执行服务器信息，任务所在项目等。

任务内部执行日志

这个部分日志基于任务具体逻辑实现，shell任务即为任务打印的日志信息，MaxCompute（原ODPS）任务即为odps服务端打印的日志信息，mr任务用户自定义的日志不在此处显示，需要到logview中查询。

结束信息

任务结束后，会将任务执行的最终状态打印到日志中，会显示的说明任务是成功还是失败。在DataWorks中，成功的唯一返回码是0，其余的返回码均表示失败，日志的最后一行打印的是任务执行的日志存储目录。

说明：

由于DataWorks判断任务成功与否是基于任务进程的退出码决定的，所以在配置shell任务的时候，需要注意退出码，0表示成功，-1表示kill，其余的码均表示失败。

2.2.3.2 gateway异常解决方案

背景信息

在使用DataWorks的过程中，可能会遇到任务突然大量处于**等待资源**状态，或者某台gateway服务器硬件异常等情形，可能是因为base-biz-gateway处于异常状态。

操作步骤

1. 执行如下命令，判断gateway目前是否处于存活状态。

   ```
   curl -u username:password --digest -H "Accept: application/json" -H "Content-type: application/json" -X GET “http://localhost:7001/alisa/4.0/resource/node”
   ```

   返回的结果中对应的gateway中的live字段是否为true。

2. 如果gateway已经处于终止态（live=false），则检查当前gateway的各项指标。

   - 查看磁盘：`df -h`， 查看*/home/admin*目录下的磁盘状态。
   - 查看内存使用：`free –g`。
   - 查看句柄数：`ulimit -n`，如果过小，则设置为“131072”。
   - 检查*/home/admin*目录以及所有子目录的权限。

3. **重启base-biz-gateway**。

   说明：

   - 设置ulimit的方法：

     打开sudo vim /etc/secrity/limit.conf，添加下面两行：

     ```
     soft    nofile    131072
      hard    nofile    131072
     ```

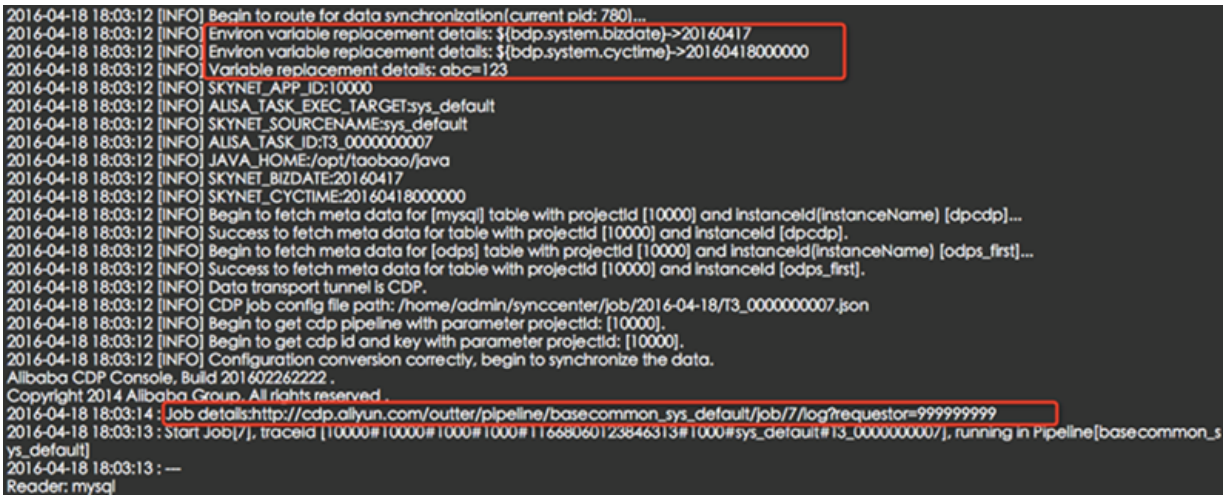
   - 每次变更修改都需要重启base-biz-gateway的应用agent服务，重启命令：

     ```
     sudo  su  admin /home/admin/alisatasknode/target/alisatasknode/bin/serverctl   restart
     ```

2.2.3.3 CDP任务日志关键点

数据同步日志中两个关键信息如图 *2-16:* 日志中的红框所示，分别是调度参数替换信息（如果您对于任务变量有疑问先确认此处）、底层datax日志链接（目前只有在任务执行完成后，CDP Console才能获取完整DataX日志，如果您期望在任务运行时就看到底层DataX日志，可以复制此链接在浏览器中打开，能够看到即时的最新底层细节日志）。

图 2-16: 日志



上图第二个红框的下一行，Start Job字符串后的数字即是CDP的作业ID，如果您使用CDP自己的Web界面，可以使用此ID进行作业检索和定位。

数据同步的日志中**All Task WaitWriterTime 100.000s | All Task WaitReaderTime 50.000s** 这类信息是任务读写数据源的等待时间，时间越长对应的端相对性能越低。

2.2.3.4 CDP任务探测数据源性能

说明：

请一定查看此命令行探测工具的帮助信息，有完整的说明实例。探测工具本质上是形成streamreader->待测数据源、待测数据源->streamwriter的DataX同步任务。streamreader或streamwriter属于纯内存相关无数据源耦合，便于检查另一端的性能。

测试写端数据源性能时，会向目的数据源进行写数据操作，请留意测试表写入测试数据是否会影响生产。建议使用测试表验证写性能。

使用streamreader测试写端性能时，stream支持的数据类型有string|long|date|double|bool|bytes同时支持产生随机数据示例如：

```
[{"type":"string","value":"abc"},{"type":"string","random":"10,20"}]
```

支持随机函数， 示例如表 *2-17: 随机函数示例*所示。

表 2-2: 随机函数示例

类型	示例	说明
LONG	random 0,10	0到10之间的随机数字。
STRING	random 0,10	0到10长度之间的随机字符串。
BOOL	random 0,10	false和true出现的比率。
DOUBLE	random 0,10	0到10之间的随机浮点数。
DATE	random 2014-07-07 00:00:00, 2016-07-07 00:00:00	开始时间->结束时间之间的随机时间，日期格式默认（不支持逗号）yyyy-MM-dd HH:mm:ss。
BYTES	random 0,10	0到10长度之间的随机字符串获取其UTF-8编码的二进制串配置了混淆函数后，可不配置value。

在base-biz-gateway分组的机器上部署有同步引擎DataX，DataX既能进行数据同步又能进行一定的数据源能力探测，便于出现速度异常时的问题排查。性能探测工具的入口是：*/home/admin/datax3/bin/perftrace.py* ，您可以直接运行即可以看到相应的帮助信息。

- datasourceType：数据源类型，支持mysql|drds|oracle|ads|sqlserver|postgresql|db2等。
- jdbcUrl：数据源访问地址，不同数据库格式有区别，可以看帮助信息中的demo。
- username：数据源访问用户。
- password：数据源访问密码。
- table：测试性能读写的表。
- column：读写的列。
- splitPk：测试读端时的切分列。
- where：测试读端时待读取的数据限制条件。
- fetchSize：测试读端时数据读取批量条数。
- reader-sliceRecordCount：测试写端时，读端随机生成记录条数。
- reader-column：测试写端时，读端随机生成记录列。
- batchSize：测试写端时，写出每次批量条数。
- preSql：测试写端时，前置sql。
- postSql：测试写端时，后置sql。

- url：测试写ADS时，ADS的访问地址，格式为IP：Port。
- schme：测试写ADS时，ADS的数据库名。
- writer-print：测试读端时，读出的数据是否打印到控制台。
- -c --channel：测试时DataX线程数。
- -f --file：使用已有的dataX配置文件进行测试，支持本地文件和Http地址。
- -t --type： 测试读端数据源还是写端数据源。
- -h --help：命令行帮助 。

详细的DataX文件配置格式请参见：*https://github.com/alibaba/DataX*。

2.2.3.5 查看CDP服务运行状态

背景信息

CDP服务是否正常：目前各类环境输出中，都配置有监控任务，会监控CDP服务是否正常启动，在服务中断时会有相关报警提示。CDP服务本身也会有定时检查，发现服务进程不存在会自启动。如果您想手动确认，可通过linux的curl命令，具体为（请注意您所在环境CDP服务访问域名）：`curl http://CDP域名/api/inner/status.taobao`，正常时返回I'm OK!字符串。您也可以尝试浏览器打开http://CDP域名/api/inner/status.taobao这个网址，确认是否下载内容为I'm OK!的txt文件。

CDP管理控制台：CDP服务本身提供的有运维管理控制台，可进行一些统计展示、资源调整。直接访问http://CDP域名可以登录CDP的Web界面，访问http://CDP域名/web/view/admin/admin.html登录CDP的系统运维界面（需要CDP的系统管理员权限）。

调整CDP管道资源限制。

如果您需要调整CDP本身的资源限制，主要涉及到的配置文件项有：

- 一个应用空间最大速度pipeline.job.bandwidth。
- 切分时一个子任务的最大速度channel.speed.byte。
- 一个作业的最大速度job.max.speed。

未来这些修改都可在DataWorks统一系统运维中完成，如果手工修改，操作步骤如下所示：

操作步骤

1. 登录跳板机。
2. CDP配置修改，CDP服务部署在base-biz-cdp分组下。配置文件的地址为：*/home/admin/cdp_server/conf/config.properties*。

3. 修改完成后重启服务（一般2台互为冗余），具体为切换到admin账户重启，`sudo su admin /home/admin/cdp_server/bin/appctl.sh restart`。修改完成后，新创建的管道资源运行的作业速度都会有新配置的参数值约束。如果您需要修改原有管道资源限制，请看下面Alisa协调调整进一步的说明。

> **说明**：
> 上面第二步的配置文件中db.mysql.url、db.mysql.username、db.mysql.password为CDP服务使用的数据库访问信息。其中t_job 表为运行作业流水表，t_project表为应用管道创建流水表。
>
> 示例如下：
>
> - 查询当前作业运行状态的SQL语句：`select * from t_job where state = 3\G;`
> - 查看应用速度的SQL语句：`select * from t_project;`

2.2.3.6 同步引擎DataX配置项的修改

操作步骤

1. 同步引擎DataX部署在base-biz-gateway分组，配置文件路径为*/home/admin/datax3/conf/core.json*。
2. 一般调整配置为单通道速度限制，将单个切分的速度调整为5Mbps，即修改**byte**的值为5242880。

图 2-17: 修改配置项

byte：限制每秒多少Byte数据。默认是一个并发速度1048576， 即1MB。

2.2.3.7 CDP部署规模测算

不同的环境对于同步速率、性能有不同的要求，这里给出一份部署规模测算关系，可供运维部署参考。

> **说明**：

此种测算方式并非严格意义上的数学测算，是在网络环境处于非瓶颈期（不考虑机器万兆网卡但机器连接的路由器为10M bytes这类情况），且已考虑了系统冗余情况的前提下进行的。请根据实际需要因地制宜。

表 2-3: 部署测算分步计算公式

步骤	分布计算公式	公式注释
1	真实执行速率 = 每同步数据量/同步总时长	此数据为真实统计到的流量速度情况，这里的单位为Byte。
2	CDP执行速率 = 真实执行速率 * 2	考虑实际情况，机器资源使用率不会达到100%，故按照资源使用50%计算。
3	CDP通道数 = CDP执行速率	默认政企输出单通道限速为1MB/s。
4	总内存数 = CDP通道数 * 200MB * 1.5	1.单task所耗内存为200MB左右 2.DATAX内存 = task所占内存 * 1.5。
5	槽位数 = CDP通道数 * 2	CDP中单通道占用Alisa的槽位数为2

表 2-4: 部署测算完整计算公式

完整计算公式
总内存数 = (每日同步数据量/同步总时长) * 2 * 200MB * 1.5
总槽位数 = (每日同步数据量/同步总时长) * 2 * 2
虚拟机器数 = (每日同步数据量/同步总时长) * 2 * 0.2G * 1.5 / 6G
物理机器数量 = (每日同步数据量/同步总时长) * 2 * 0.2G * 1.5 / 40G

表 2-5: 根据同步量推算部署机器详细测算表

每日同步数据量	同步总时长(小时)	真实执行速率	CDP执行速率	CDP通道数	总内存数	槽位数	部署机器
100G	8	12.5GB/h 3.5MB/s	25GB/h 7MB/s	7	3G	14	1台虚拟机
1T	8	125GB/h 35MB/s	250GB/h 70MB/s	70	21G	140	4台虚拟机
1T	24	42GB/h 12MB/s	84GB/h 24MB/s	24	8G	48	2台虚拟机

每日同步数据量	同步总时长(小时)	真实执行速率	CDP执行速率	CDP通道数	总内存数	槽位数	部署机器
10T	8	1.25TGB/h 350MB/s	2.5TB/h 700MB/s	700	210G	1400	4台物理机
100T	8	12.5TGB/h 3.5GB/s	25T 7GB/s	7000	2100G	14000	40台物理机

表 2-6: 根据部署机器推算同步最大数据量详细测算表

部署机器	槽位数	总内存数	CDP通道数	CDP执行速率	真实执行速率	同步总时长(小时)	每日同步数据量
1台虚拟机	30	6G	20	72GB/h 20MB/s	36GB/h 10MB/s	8	288G
1台虚拟机	30	6G	20	72GB/h 20MB/s	36GB/h 10MB/s	24	864G
1台物理机	200	40G	133	478GB/h 133MB/s	240GB/h 66MB/s	8	1.91T
1台物理机	200	40G	133	478GB/h 133MB/s	240GB/h 66MB/s	24	5.76T

表 2-7: 政企输出通出GATEWAY规格

型号	网卡	CPU	Mem(GB)	最大传输速率	槽位数
物理机	双千兆	24core	48G	200MB/s	200
虚拟机	使用物理机卡	4core	8G	30MB/s	40

3 分析型数据库

3.1 配置管理

3.1.1 Zookeeper配置节点总览

Analytic DB 根节点的 zk 路径为：*/app/garuda/clusterName*，根节点下的主要子节点的路径及主要作用在以下表格中做简要陈述，以下表格省略全路径，默认全路径为*/app/garuda/clusterName/* + 子节点路径。

节点路径	作用
global	ADS各模块全局配置。
dbmg	ResourceManager模块节点注册地址。
lnmg	ComputeNode模块节点注册地址。
mnmg	FrontNode节点注册地址。
odpsbuild	Builder节点注册地址。
taskmanager	TaskManager模块节点注册地址。
unmg	BufferNode模块节点注册地址。

3.1.2 各模块全局配置

各模块的全局配置都在*/app/garuda/clusterName/global* 路径下，本节主要详细讲解此路径下主要配置的作用。

以下章节中默认省略全路径*/app/garuda/clusterName*。

3.1.2.1 Meta全局配置

/global/cfs 下保存了 Analytic DB（原ADS）所有模块的公用配置，列表如下：

配置路径	配置项	配置作用	默认配置
apsara	wrapperApsaraVersion	ads各所有组件所使用飞天的版本	飞天版本号
meta	url	ads元数据库的路径	ads:mysql://mysql_address/ads_meta

配置路径	配置项	配置作用	默认配置
message	url	ads message库的路径	ads:mysql://mysql_address/message
odps	project_name	ads使用的odps project名称	adsmr
odps	end_point	ads使用的odps api endporint地址	
odps	tunnel_end_point	ads使用的odps tunnel endpoint地址	
odps	odps_build_project	ads build使用的odps project，默认与odps project相同	adsmr
pangu	username	ads飞天集群提交飞天任务的用户名	ads
pangu	host	ads飞天集群pangu路径	pangu://xxx
pangu	replication	pangu上文件存储的副本数	3
pangu	capability	ADS使用pangu的秘钥	xxx
pangu	ioBuffer	ADS使用pangu的iobuffer大小	2000000
pangu	type	ADS使用的存储媒介默认是pangu	pangu
pangu	password	ADS使用pangu的password	已废弃，但不能删
quota	quotaEnabled	是否开启配置管理，专有云中默认不待	false
resource	resourceDisableEnlarge	开关，禁止DB扩容，一般在发布时使用	false
resource	resourceDisableService	开关，禁止申请资源，一般在发布时使用	false
resource	resourceDisableCreateDB	开关，禁止创建DB，一般在发布时使用	false
sysdb	url	ADS sysdb的链接地址，各个模块读取这个地址来写instance_profile和query_profile这两张元数据表	sysdb.xxx

3.1.2.2 各模块全局配置

*/global/config*下保存了各个模块的公共配置，介绍如下。

对应模块	配置路径	配置项	配置作用	默认配置
all	clusterName	ADS集群名	clusterName	-
ResourceManager	cms	cms下所有的配置项	是云监控所需的配置项	专有云中目前尚未使用
ComputeNode	localnode	cloudRootPath	odps上ads数据的存储路径	/garuda/adsmr
ComputeNode	localnode	taskThreadCount	-	10
ComputeNode	localnode	queryMaxTime	-	60000
ComputeNode	localnode	version	执行build任务所用的ComputeNode版本	xxx.jar
ResoureManager	quota	quotaDisabled	ads配额管理功能，专有云中不待，默认关闭	true
ResourceManager	disableResourceOperationDisabled	禁止资源操作，如申请/释放资源。在配额管理功能打开时，此配置项生效	false	-
ResourceManager	disableDataOperationDisabled	禁止数据操作，如上下线，数据导入等在配额管理功能打开时，此配置生效	false	-
ResourceManager	repository	rootPath	pangu上ADS文件的存储根目录	/garuda
ResourceManager	repository	packageRepository	pangu上ADS各模块启动程序存放路径	/repository-[clusterName]
ResourceManager	repository	configRepository	pangu上ADS配置文件存放路径	/repository-[clusterName]
ResourceManager	repository	scriptRepository	pangu上ADS启动脚本存放路径	/repository-[clusterName]
ResourceManager	resourcemanager	gallardoUIUrl	FuxiService UI的http地址	-
ResourceManager	resourcemanager	isGallardoService	是否使用FuxiService，即采用0.8版本飞天拉起各模块还是采用0.7版本，手动	true

对应模块	配置路径	配置项	配置作用	默认配置
			启动各模块，默认都是采用飞天拉起各模块	
ResourceManager	resourcemanager	gallardoServerUrl	FuxiService server的地址	-
ResourceManager	resourcemanager	pushMessageDisable	?	-
ResourceManager	resourcemanager	useDataAgent	是否开启下载进程	true
BufferNode	updatenode	baselineTimeRange	BufferNode每天merge baseline时间段，在此时间段内，buffernode会将所有实时表都做一遍merge baseline操作	[20 TO 6]（晚上8点到早上6点）
BufferNode	updatenode	stopMergeBaseline	是否暂停自动merge baseline功能	false

3.1.2.3 FrontNode全局配置

FrontNode 作为 Analytic DB 的前端节点，负责 sql 解析，下发，数据聚合等功能。

/global/config/master 和 */global/config/query* 下的配置都是 FrontNode 的配置，详情如下。

配置路径	配置项	配置作用	默认值
master	aliyunBaseUrl	umm api地址，在专有云中，ads需要从umm处获取用户aliyun ID进行鉴权	-
master	aliyunIDUrl	-	-
master	useAuth	开关，是否进行身份认证	true
master	useAcl	开关，是否进行鉴权	true
master	fakeAuth	开关，是否使用伪认证	false
master	withDNSEnv	开关，是否使用DNS	true
master	withSLBEnv	开关，是否使用负载均衡	true
master	dbReservedName	ADS保留了一些保留字，用户不能建与这些保留字相同的DB	admin,sysdb, agentdb
master	commandCorePoolSize		500

配置路径	配置项	配置作用	默认值
master	connectionsOverloadThreshold		2000
master	userDBLimit	单个用户可创建的最大DB数	10
master	ddlFactTableGroupTablesLimit	单事实表组可创建的最大表数	256
master	ddlDimTableGroupTablesLimit	维度表组可创建的最大表组	256
master	ddlTableColumnsLimit	单表最大列数	990
master	ddlMinFactTablePartitions	实时表最小分区数	1
master	ddlMaxFactTablePartitions	事实表最大分区书	256
master	ddlMinSubPartitions	最小二级分区数	1
master	ddlMaxSubPartitions	最大二级分区数	1095
master	detailedLog	是否打印详细日志，只在排查问题时开	false
master	auditLog	是否打印审计日志，默认不打印	false
master	tailToleranceEnabled	是否开启长尾配置	false
master	additionalColumn		true
master	spliceColumns		true

3.1.2.4 Builder配置

*/global/mrbuild*下是 Builer 模块的一些配置，详情如下。

配置路径	配置项	配置作用	默认值
common	jobExpiredInHour	-	-
common	cleanJobIntervalInHour	-	-
fuxi	serviceDisabled	-	-
fuxi	maxWorkers	-	-
odps	serviceDisabled	-	-
odps	maxWorkers	-	-

3.1.2.5 其他重要配置

*/global/taskmanager/config/concurrent*保存了 build 任务的并发度，目前实时表和非实时表公用一个并发配置，默认 20，即同时可有 20 个表 build 数据。目前 0.8 版本下，专有云和一体机都使用脚本进行部署。部署按顺序分为三个大部分：apsara，fuxiservice，ads，依次依赖。下文从 Analytic DB 角色，机型，部署结构等部分阐述了 Analytic DB 的部署框架。

3.2 运维基础

3.2.1 进程启停

3.2.1.1 飞天进程启停

飞天集群启停都在 Admin Gateway 上以 admin 账号操作，以下命令会重启整个飞天集群，请慎重使用，具体命令如下。

- 启动：

  `/home/admin/dayu/bin/allapsara start`

- 停止：

  `/home/admin/dayu/bin/allapsara stop`

- 状态检查：

  `/home/admin/dayu/bin/allapsara status`

需要对单机飞天进程操作，请登录单台机器，进入 admin 账号，执行以下命令。

- 启动：

  `/home/admin/dayu/bin/apsarad start`

- 停止：

  `/home/admin/dayu/bin/apsarad stop`

- 查看状态：

  `/home/admin/dayu/bin/apsarad status`

3.2.1.2 FuxiService进程启停

- FuxiService 进程为无状态程序，启停方式十分简单，分别登录进程所在机器，执行以下命令。
  - 停止

分别等待显示出每个进程的 pid：

```
$jps|grep GallardoServer
1032 GallardoServer
```

```
$jps | grep GallardoUI
1512 GallardoUI
```

```
$jps | grep RmUI
2283 RmUI
```

停止：

```
kill -9 $pid
```

- 启动

  分别对应机器，执行以下命令。

  启动GallardoServer：

  ```
  /home/admin/install/gallardo-server/bin/startGallardoServer.sh
  ```

  启动 GallardoUI：

  ```
  /home/admin/install/gallardo-ui/bin/startGallardoUI.sh
  ```

  启动 RMUI：

  ```
  /home/admin/install/rmui/bin/startRmUI.sh
  ```

- 检查状态执行jps 查看进程是否启动。

  ```
  $jps
  1032 GallardoServer
  8982 Jps
  2283 RmUI
  1512 GallardoUI
  ```

3.2.1.3 Analytic DB进程启停

- ResourceManager 和 Builer 作为 Analytic DB 的控制节点，目前是在单机手动启动，启停方法如下：得到 RM/Builder 地址，然后登录AG，切换到 admin账号下，执行 search ads_rm/search ads_bu，即可得到控制节点的部署地址。

- 停止：

```
/home/admin/garuda/bin/garuda.sh stop
```

- 启动：

```
/home/admin/garuda/bin/garuda.sh start
```

- 查看状态：

```
ps -ef|grep virgo
```

- ZooKeeper 启停及状态检查：得到 zk 机器地址登录 AG，切换到 admin 账号下，执行 search ads_zk，即可得到所有 zk 地址，并登录到相应机器。具体命令如下所示。
  - 启动：

```
/home/admin/zookeeper-3.4.6/bin/zkServer.sh start
```

  - 停止：

```
/home/admin/zookeeper-3.4.6/bin/zkServer.sh stop
```

  - 查看状态：

```
/home/admin/zookeeper-3.4.6/bin/zkServer.sh status
```

    系统显示如下：

```
JMX enabled by default
Using config: /home/admin/zookeeper-3.4.6/bin/../conf/zoo.cfg
Mode: follower
```

3.2.2 系统升级

3.2.2.1 升级GallardoServer

1. 登录 GallardoServer 所在的机器，切换到 admin 账号。
2. 将新的部署包放入*/home/admin/fuxi-service/20151117/gallardo-server.tar*目录下。
3. 比较部署包的 MD5：*md5sum /home/admin/fuxi-service/20151117/gallardo-server.tar* 。
4. 将原有两台 gallardo-server 所在机器上的*/home/admin/gallardo-server/lib* 文件夹用 */home/admin/fuxi-service/20151117/gallardo-server.tar* 中解压开的 lib 替换。

5. 停止 GallardoService 进程。

```
jps | grep GallardoServer | cut -f1 -d" " | xargs kill -9
```

6. 启动 Gallardo Server 进程。

```
/home/admin/install/gallardo-server/bin/startGallardoServer.sh
```

7. 查看 GallardoServer 状态，确认 server 已经重启。

```
jps | grep GallardoServer
```

3.2.2.2 升级GallardoUI

1. 登录 GallardoUI 所在的机器，切换到 admin 账号。
2. 将新的部署包存放在*/home/admin/fuxi-service/20151117/gallardo-ui.tar*目录下。
3. 比较部署包的MD5：*md5sum /home/admin/fuxi-service/20151117/gallardo-ui.tar* 。
4. 将原有两台 gallardo-ui 所在机器上的 */home/admin/gallardo-ui/lib* 文件夹用包*/home/admin/fuxi-service/20151117/gallardo-ui.tar* 中解压开的 lib 替换。
5. 停止 GallardoService 进程。

   `jps | grep GallardoUI | cut -f1 -d" " | xargs kill -9`
6. 启动 Gallardo UI 进程。

   `/home/admin/install/gallardo-ui/bin/startGallardoUI.sh`
7. 查看 GallardoUI 状态，确认 ui 已经重启。

   `jps | grep GallardoUI`

3.2.2.3 升级AM&Container

1. 登录AG 所在的机器，切换到 admin 账号。
2. 将新的部署包放在：*/home/admin/fuxi-service/20151117/gallardo-am.tar*。
3. 比较部署包的 MD5：md5sum /home/admin/fuxi-service/20151117/gallardo-am.tar 。
4. 上传盘古。

```
pu put /home/admin/fuxi-service/20151117/gallardo-am.tar pangu://localcluster/fuxi-service/am
/package/gallardo-am.tar
pu setreplica pangu://localcluster/fuxi-service/am/package/gallardo-am.tar n n
```

说明：

n 为 pangu 副本数，n=tubo 个数/3 取整。

5. 重启 AppMaster。

   a. 在 http://agip:8315/${clustername}/scheduler 中选择一个 AppMaster 所在的机器，地址一列为 Address，到对应机器上 kill -9 结束杀掉一个 AppMaster 进程，等待 5min。

   b. 进入页面 http://agip:8315/${clustername}/amapp，单击对应被kill 掉的 App，进入 detail 页面后，将地址栏中的 task 修改为 container。

   c. 查看该 AppMaster 以及所有Container版本是否更新正确，（AppMaster版本在页面最顶部，Container 版本在每一行最后一列）。

   d. 版本验证正确后，升级所有 App，步骤为在 AG 上执行如下命令 kill 掉所有 Fuxi Service AppMaster 进程。

   ```
   for ip in `sh search tubo `; do echo $ip ; ssh $ip 'source ~/.bash_profile; jps | grep ContainerAppMaster | cut -f1 -d" " | xargs kill -9'; done;
   ```

   进一步确认所有 AppMaster 进程被成功 kill。

   ```
   for ip in `sh search tubo `; do echo $ip ; ssh $ip 'source ~/.bash_profile; jps | grep ContainerAppMaster | cut -f1 -d" " '; done;
   ```

3.2.2.4 升级Fuxi-Service Rm & Nm

1. 登录AG 所在的机器，切换到 admin 账号。
2. 将新的部署包放在：*/home/admin/fuxi-service/20151117/app_lib_wrapper_test_release_64.tar.gz*。
3. 比较部署包的 MD5：*md5sum /home/admin/fuxi-service/20151117/app_lib_wrapper_test_release_64.tar.gz*。
4. 上传盘古。

   ```
   r pl
   r rp package://garuda_am
   r ap package://garuda_am /home/admin/fuxi-service/20151117/app_lib_wrapper_test_release_64.tar.gz
   ```

5. 重启 FuxiService Rm。

查看 rm 所在机器。

```
r wheream garuda/garudaAppMaster
```

kill Rm 进程。

```
ssh ${rmhost} "jps | grep ResourceManagerServer | xargs kill -9"
```

3.2.2.5 ResourceManager/Builder升级步骤

ResourceManager 和 Builder 是作为 Analytic DB 组件单独启动的，以 RM 为例升级步骤如下。

1. 执行/home/admin/garuda/bin/garuda.sh stop 停止进程。
2. 执行ps -ef|grep virgo命令，查看进程是否完全停止。
3. 将*/home/admin/garuda/virgo/pickup/*目录下的 jar 包替换成最新的 jar 包。
4. 执行/home/admin/garuda/bin/garuda.sh start命令，启动进程。
5. 执行ps -ef|grep virgo命令，查看进程是否启动。

3.2.2.6 ComputeNode/FrontNode/BufferNode升级步骤

ComputeNode/FrontNode/BufferNode 都是由飞天拉起的模块，三个模块的升级方式大致相同。以 FrontNode 为例说明升级流程。

1. 生成 pangu 上garuda.tar 包路径。

   pu cpdir pangu://localcluster/garuda/repository-{clustername}/frontnode/{version_old} pangu://localcluster/garuda/repository-{clustername}/frontnode/{version_new}

   其中 repository-{clustername} 为 pangu 上存放每个版本代码包的路径，可以从 zk 的 */global/config/repository* 得到。

   说明：

   version_old 为最近的版本。

   version_new 为将要发布的 jar 包中的版本号。

2. 更新 pangu 上 version_new 路径下的 garuda.tar 包。

   a. 从盘古上下载 garuda.tar： *pu get pangu://localcluster/garuda/repository-dtdream/frontnode/0.8.4/package/garuda.tar /home/admin/adsdeploy/fn/garuda.tar*

b. 更新 garuda.tar 中 lib 目录下的 jar 包。

```
tar -xf /home/admin/adsdeploy/fn/garuda.tar && rm -rf /home/admin/adsdeploy/fn/lib/*
mv /home/admin/com.taobao.garuda.mergenode.xxx.jar /home/admin/adsdeploy/fn/lib/
cd /home/admin/adsdeploy/fn && rm -rf garuda.tar && tar cf garuda.tar ./*
```

3. 将 garuda.tar 目录更新到第一步产生的新的盘古路径上。

   a. 删除新路径下的老包： *pu rm pangu://localcluster/garuda/repository-{clustername}/frontnode/{version_old}package/garuda.tar*。

   b. 将新 garuda.tar 包放到最新路径下： *pu cp /home/admin/adsdeploy/fn/garuda.tar pangu://localcluster/garuda/repository-{clustername}/frontnode/{version_new}/package/*。

4. 检查 pangu 路径上的 dbmanager 目录版本。pangu 上 dbmanager 的版本目前是和 ResourceManager 的版本绑定的，如果 ResourceManager 的版本更新了，那么就待更新 pangu 上 dbmanager 路径的版本号：*pu cpdir pangu://localcluster/garuda/repository-{clustername}/dbmanager/{version_old} pangu://localcluster/garuda/repository-{clustername}/dbmanager/{version_new}*

   说明：
   version_new 为 ResourceManager 的最新版本号。

5. 发送升级命令。

   ```
   curl -d "sql=ALTER SYSTEM UPGRADE SERVICE {module} PACKAGE {$package_version } CONFIG {config_version} IN DATABASE all" rm_master_ip:9999/api/command
   ```

   命令中有以下参数。

   - module：所要发布的模块 -- FRONTNODE/COMPUTENODE/BUFFERNODE。
   - package_version：为本次发布的 FrontNode 的版本号。
   - config_version：为本次发布所使用的配置文件的版本号，是 resource_group 这张表中 admin DB 的 conf_version。
   - rm_master_ip：主 master 的 IP 地址，从 console 的 /global/dbmanager 下即可获取。

6. 检查build 上ComputeNode包是否需要更新。发布ComputeNode时，需同时更新builder上的localnode jar 包。

   a. 将 localnode jar 包放到每台 builder 机器的/home/admin/garuda/localnoderepo 下。

   b. 更新 zk 上 ComputeNode 的版本配置`/global/config/localnode version`为最新的 localnode jar 包的配置。

其他模块发布时，待将上述步骤中的 frontnode 改为其他模块的名称，如 computenode/buffernode。

3.2.3 掉电启动

专有云或一体机环境中，可能会遇到整机群或整机掉电的情况，此时需按如下步骤恢复整个集群。

1. 确认所有机器已经启动。
2. 检查飞天状态。

   登录AG，执行/home/admin/dayu/bin/allapsara status命令。

   如果发现集群飞天有部分角色未启动，则执行如下命令 ，重启整个集群。

   /home/admin/dayu/bin/allapsara stop

   /home/admin/dayu/bin/allapsara start

   /home/admin/dayu/bin/allapsara status
3. 启动 zk。

   登录AG，执行 search ads_zk 得到所有 zk 部署的机器 ip，登录机器执行如下命令。

   1. 执行/home/admin/zookeeper-3.4.6/bin/zkServer.sh start命令，启动zk。
   2. 执行/home/admin/zookeeper-3.4.6/bin/zkServer.sh status命令，查看状态。

      系统显示结果如下。

      ```
      JMX enabled by default
       Using config: /home/admin/zookeeper-3.4.6/bin/../conf/zoo.cfg
       Mode: follower
      ```

   三台机器中正常状态应该是2台机器是 follower，一台是leader。
4. 确认元数据库可以正常连接。

   一体机中，执行mysql -hlocalhost -P3306 -uapsara -pxxx命令，登录AG。
5. 启动 console。

   登录AG，执行如下命令。

   /home/admin/garuda-console/bin/startup.sh

   /home/admin/redis/src/redis-server /home/admin/redis/work/redis.conf
6. 启动 gallardo。

登录gallardo server/ui/rmui 所在机器，一般情况下 server/ui 都和 fuximaster 混布，rmui 在AG上。

1. 执行/home/admin/install/gallardo-server/bin/startGallardoServer.sh命令，启动 GallardoServer。
2. 执行/home/admin/install/gallardo-ui/bin/startGallardoUI.sh命令，启动 GallardoUI。
3. 执行/home/admin/install/rmui/bin/startRmUI.sh命令，启动 RMUI。

7. 启动 ADS ResourceManager/Builder 节点。

   登录AG，切换到 admin 账号下，执行 search ads_rm/search ads_bu，即可得到控制节点的部署地址。

   - 启动：/home/admin/garuda/bin/garuda.sh start
   - 查看状态：ps -ef|grep virgo

8. 确定以上步骤都执行成功后，查看已拉起的 DB 状态，DB 包括 agentdb，sysdb 和所有用户 DB，检查以下内容。

   - 各 DB 节点个数是否完整，如 agentdb ComputeNode 的个数为计算节点的个数。
   - Console 上，配置管理 -- 高级 -- /lnmg/db/, /mnmg/db, /unmg/db 下，每个 db 的 zk 节点下，注册的节点个数是否完整。

   如果有问题，则按以下步骤进行恢复。

   1. Remove 有问题 DB 的资源。

      ```
      curl -d "sql=ALTER DATABASE ${dbName} PROPERTIES(databaseId=${dbid}) REMOVE RESOURCE OPTIONS(resource_type='ecu',ecu_type='${ecuType}',ecu_count=${ecuCnt})" ${RM_Master_IP}:9999/api/command
      ```

   2. ADD 有问题 DB 的资源。

      ```
      curl -d "sql=ALTER DATABASE ${dbName} PROPERTIES(databaseId=${ecuID}) ADD RESOURCE OPTIONS(resource_type='ecu',ecu_type='${ecuType}',ecu_count=${ecuCnt})" ${RM_Master_IP}:9999/api/command
      ```

9. 掉电问题诊断问题：DB 进程数（instance）在Gallardo上不对，一般都是进程数少或进程完全无法拉起。

   可能原因： DNS 问题，导致在某台机器上无法通过 hostname（机器名）链接到其他机器，飞天强依赖机器名登录缺少进程的机器，ping 其他机器的 hostname，检查是否通过机器名能够连通所有机器。
   解决方法： 解决 DNS 问题，或直接绑定机器名。

3.2.4 生成并部署配置文件

Analytic DB 各个模块之间的通信都遵守一定的安全协议，为了保证系统通信的安全性，会将所需访问系统的用户名和密码进行加密生成秘钥。Analytic DB 的安全认证体系由 Analytic DB 独立的认证签名算法实现，采用公私钥对的方式对所待访问系统的用户名和密码进行加密。

config.key.pub：公钥，用于加密，在生成配置文件或其他秘钥时使用。

config.key：私钥，用于解密，程序内部用来反解配置文件得到真实签名。

config.ini:：加密之后的文件，存储所有 Analytic DB 内部和 Analytic DB 外部系统鉴权所用的用户名和密码。

config.ini.plain ：config.ini 的明文。

3.2.4.1 config.ini内容介绍

config.ini 文件是 ADS 内部鉴权及外部系统用户名和密码的集合，其明文 config.ini.plain 内容如下。

```
[zookeeper]
 connecturl=xxx
sessionTimeout=30000
connectTimeout=5000
retryPolicyName=org.apache.curator.retry.ExponentialBackoffRetry
retryPolicyArgs=int,500;int,10
[keyList]
zookeeper=zookeeperKey
meta=mysqlKey
havana=havanaKey
pangu=panguKey
odps=odpsKey
tfs=tfsKey
umm=ummKey
h2=h2Key
dump=odpsKey
server=serverKey
mergenode-server=mergenodeServerKey
update-server=updatenodekey
taskmanager-server=taskmanagerKey
sysdb=sysdbKey
message=messageKey
yaochi=yaochiKey
aliyun=aliyunKey
oms=sysdbKey
[zookeeperKey]
accessId=
accessKey=
[mysqlKey]
accessId=
accessKey=
[h2Key]
accessId=
accessKey=
```

```
[havanaKey]
accessId=
accessKey=
[panguKey]
accessId=garuda
accessKey=
[odpsKey]
accessId=
accessKey=
[tfsKey]
accessId=
accessKey=
[ummKey]
accessId=
accessKey=
[mergenodeServerKey]
accessId=mergenode-server
accessKey=
[updatenodekey]
accessId=updatenode-server
accessKey=
[taskmanagerKey]
accessId=taskmanager-server
accessKey=
[serverKey]
accessId=mergenode-server
accessKey=
[sysdbKey]
accessId=
accessKey=
[messageKey]
accessId=
accessKey=
[aliyunKey]
accessId=
accessKey=
[yaochiKey]
accessId=
accessKey=
```

config.ini 整体分为三部分。

- ZK 配置。

  zk 链接地址：connecturl=zkip1:zkport,zkip2:zkport,zkip3:zkport/app/garuda/${clustername}。

  zk 默认参数。

  sessionTimeout=30000

  connectTimeout=5000

  retryPolicyName=org.apache.curator.retry.ExponentialBackoffRetry

  retryPolicyArgs=int,500;int,10

- 所需的 key 内容。

[zookeeperKey] -- 连接 ADS zk 的用户名和密码。

accessId=

accessKey=

[mysqlKey] -- ADS 元数据库的用户名和密码。

accessId=

accessKey=

[h2Key] -- frontnode 通过 JDBC（h2）访问 computenode 认证时使用。

accessId=

accessKey=

[havanaKey] -- havana 的用户名和密码。

accessId=

accessKey=

[panguKey] -- ADS 读写 pangu 所用的用户名和密码。

accessId=garuda

accessKey=为/apsara/security/internal_capability/InternalCapabilityForYu.txt文件的内容。

[odpsKey] -- ADS 使用ODPS时的用户名和密码

accessId=

accessKey=

[tfsKey] -- ADS 使用tfs（taobao file system）的用户名和密码

accessId=

accessKey=

[ummKey] -- ADS访问umm使用的accessid和accesskey

accessId=

accessKey=

[mergenodeServerKey] -- ADS mergenode 连接认证

accessId=mergenode-server

accessKey=公钥 config.key.pub 中除去第一行和最后一行剩下的内容

[updatenodekey] -- ADS buffernode 连接认证

accessId=updatenode-server

accessKey=公钥 config.key.pub 中除去第一行和最后一行剩下的内容

[taskmanagerKey] -- ADS ResourceManager 连接认证

accessId=taskmanager-server

accessKey=公钥 config.key.pub 中除去第一行和最后一行剩下的内容

[serverKey] -- 服务器秘钥

accessId=mergenode-server

accessKey=

[sysdbKey] -- meta 访问 JDBC（h2），访问 SYSDB 的 frontnode 时使用

accessId=

accessKey=

[messageKey] -- ADS message 库的用户名和密码

accessId=

accessKey=

[aliyunKey] -- ADS 访问 aliyun aas 使用的用户名和密码

accessId=

accessKey=

[yaochiKey] -- ADS 访问瑶池所使用的用户名和密码

accessId=

accessKey=

- key 名字与 key 内容的映射关系。

  zookeeper=zookeeperKey

  meta=mysqlKey

  havana=havanaKey

pangu=panguKey

odps=odpsKey

tfs=tfsKey

umm=ummKey

h2=h2Key

dump=odpsKey

server=serverKey

mergenode-server=mergenodeServerKey

update-server=updatenodekey

taskmanager-server=taskmanagerKey

sysdb=sysdbKey

message=messageKey

yaochi=yaochiKey

aliyun=aliyunKey

oms=sysdbKey

3.2.4.2 生成config.ini

ADS 有一套生成加密文件的工具：config-crypter-tool.zip，内容如下。

config-crypter.sh lib/

此工具基本操作。

1. 生成公私钥对。

   ```
   sh config-crypter.sh g -o config.key
   ```

   会在当前目录下生成 config.key 和 config.key.pub，其中 config.key.pub 是公钥，用户加密，config.key 为私钥，用于解密。
2. 准备 config.ini.plain 明文根据上述章节中提到的明文内容，准备 config.ini.plain 文件。

3. 加密。

```
sh config-crypter.sh e -k config.key.pub -i config.ini.plain -o config.ini
```

4. 解密。

```
sh config-crypter.sh d -k config.key -i config.ini -o config.ini.plain
```

5. 帮助文档。

```
sh config-crypter.sh
Usage: config-crypter command [options] Command:
g generate config key pair e encrypt config
d decrypt config Options:
-k keyfile config key file to encrypt/decrypt, optional
-i input intput file, optional
-o output output file, optional Sample:
config-crypter g -o rsa.key
config-crypter e -k config.key.pub -i test.cfg.plain -o test.cfg config-crypter d -k config.key -i
test.cfg -o test.cfg.plain
```

config-crypter d -k config.key -i test.cfg -o test.cfg.plain

ADS 所有模块都需部署 config.ini 文件，按照部署方法可以分为两种类型，如后续两章所述。

3.2.4.3 ResourceManager&Builder

ResourceManager 和 builder 都是手动拉起，因此配置存放目录非常固定： */home/admin/garuda/virgo/etc/*。发布 config.ini 时，直接替换此目录下的 config.ini，然后重启进程即可。

3.2.4.4 ComputeNode&FrontNode&BufferNode

这三个模块都是飞天拉起，更新 config.ini 需升级配置，步骤如下。

1. 生成 config.tar 文件。

```
mkdir etc
 mv config.ini etc/tar cf config.tar etc/
```

2. 在 pangu 上创建新版本号的配置路径。

```
pu mkdir pangu://localcluster/garuda/repository-${clustername}/config/${version}/
```

3. 将 config.tar 上传 pangu。

```
pu put config.tar pangu://localcluster/garuda/repository-${clustername}/config/${version}/
```

4. 发布配置。

```
curl -d "sql=ALTER SYSTEM UPGRADE SERVICE COMPUTENODE PACKAGE ${package
_version} CONFIG ${config_version} IN DATABASE all" rm_master_ip:9999/api/command

curl -d "sql=ALTER SYSTEM UPGRADE SERVICE FRONTNODE PACKAGE ${package
```

```
_version} CONFIG ${config_version} IN DATABASE all" rm_master_ip:9999/api/command

curl -d "sql=ALTER SYSTEM UPGRADE SERVICE BUFFERNODE PACKAGE ${package
_version} CONFIG ${config_version} IN DATABASE all" rm_master_ip:9999/api/command
```

其中 ComputeNode/BufferNode/FrontNode 的${packageversion} 为当前线上的各模块正在运行的版本，可从 resource_group 这张元数据表中查出。

3.2.5 常见问题诊断

3.2.5.1 常见问题诊断

- ResourceManager/Builder 无法启动。诊断。

  1. 更改启动文件。

     a. cd /home/admin/garuda/virgo/bin/

     b. vim startup.sh

     c. 将 exec "$SCRIPT_DIR"/"$EXECUTABLE" start "$@" >& /dev/null 改为 exec "$SCRIPT_DIR"/"$EXECUTABLE" start "$@" >>/home/admin/garuda/logs/garuda.log。

  2. 启动ResourceManager。

     /home/admin/garuda/bin/garuda.sh start

  3. 查看/home/admin/garuda/logs/garuda.log 日志文件的末尾，报错内容的具体信息。

  4. 将 /home/admin/garuda/virgo/bin/startup.sh 里的 exec "$SCRIPT_DIR"/"$EXECUTABLE" start "$@" >>/home/admin/garuda/logs/garuda.log 重新改为exec "$SCRIPT_DIR"/"$EXECUTABLE" start "$@" >& /dev/null。

- **pu 命令**

  pu 命令完整路径为：/apsara/deploy/pu。pu 命令中，通过 subcommand option 来指定选项。

  pu <command options> subcmd <subcmd options>`

  注意：command options 和 subcmd options 是顺序相关的。command options 必须写在 subcmd 之前

  pu 命令支持默认的盘古路径名，在大部分命令中，如果不指定盘古路径，会假设是 pangu://localcluster/，也可以只指定一个相对路径，默认会假设根是 pangu://localcluster/。

  并且 pu 支持交互式命令模式，通过 pu -c 进入交互模式，操作盘古文件系统更便利，还支持更多的子命令（sub commands）和更丰富的 option 如 find 命令和 du 命令，put/get 等。

更多信息请参考 pu --help。

- **ls 命令**

  功能：列出指定目录下的文件及目录。

  命令：pu ls [Option]

  参数说明： *可选参数Option可以为：-r 或者-l -r 表示递归的列出子目录下的文件。 -l表示列出该目录下文件或者目录的meta信息（不递归）。

  > **说明**：
  > 文件的 meta 信息包括：MinCopy-MaxCopy，FileLength，AppName，PartName，FileType，CompressType，ReferenceType, CreateTime，LastModifyTime。

  > **说明**：
  > 目录的 meta 信息包括：该目录下所有文件的 FileLength，FileNumber，DirNumber, Pinned。

  > **说明**：
  > - 同时使用 -l 和 -r 只有 -l 起作用。
  > - 必要参数 PanguDir 和 PanguFile 的格式为：pangu://localcluster/dir1/dir2/.../dirN/。
  > - 当未指定 PanguDir 时，默认其值为 pangu://localcluster/。
  >
  > 例子
  >
  > pu ls pangu://localcluster/pengjianhong/
  >
  > pu ls -l pangu://localcluster/pengjianhong/
  >
  > pu ls -r pangu://localcluster/pengjianhong/

- **lsfilegroup 命令**

  命令 : pu lsfilegroup [fileGroupName]

  参数说明

  - *可选参数 fileGroupName 是用户自己指定的 FileGroupName，FileGroupName 是用户指定的盘古用来存放 TempFile 的目录名称。用户应该认为，Filegroup 是一种隐藏文件，使用 pu ls 是看不到的。

- 如果不写 fileGroupName，该命令列出所有的 fileGroupName。

  例子：

  pu lsfilegroup

  pu lsfilegroup PanguQuotaSmokeTest

- **rm 命令**

  功能：删除盘古文件。

  命令：pu rm [-p] PanguFile

  参数说明：* 可选参数只能是-p，表示强制删除，即不先保存在回收站中。 * PanguFile 的形式为：pangu://clusterName/dir1/dir2/.../dirN/FileName。

  例子

  pu rm pangu://localcluster/pengjianhong/pjh.log2

  > **说明**：
  > **删除的文件会放到盘古回收站中（deleted目录），例如：pangu://localcluster/deleted/pengjianhong/pjh.log2-1338442922_0**

- **mv 命令**

  功能：移动文件命令：pu mv PanguFile1 PanguFile2/PanguDir 参数说明：两个必选参数都是盘古地址，第一个参数必须是盘古文件，第二个参数可以为盘古文件或者为盘古目录。如果第二个参数是盘古目录，该目录不存在，则该命令会自动建立该目标目录。 例子

  pu mv pangu://localcluster/pengjianhong/pjh.log2 pangu://localcluster/pengjianhong/Test/

- **mvdir 命令**

  功能：移动目录命令：pu mvdir panguDir1 panguDir2 参数说明：panguDir1 必须存在，panguDir2 必须不存在，由该工具自动新建 panguDir2。例子

  pu mvdir pangu://localcluster/pengjianhong/PEManual/ pangu://localcluster/pengjianhong/PEManual2/

- **cp 命令**

  功能：文件拷贝。

  命令

pu cp File1 File2 [--pangu_tool_defaultMinCopy=MINCOPY] [--pangu_tool_defaultMaxCopy=MAXCOPY][--pangu_tool_defaultFileType=RandomAccessFile?] [--pangu_tool_isPiops=PiopsFlag?] [--pangu_tool_piops_space=PiopsSpace?] [--pangu_tool_piops_iops=PiopsIops?] [-m CPMODE] [-t FILETYPE]

参数说明

本命令支持以下三种模式。

模式1：本地文件拷贝到盘古文件(upload)，此时 File1 为本地文件地址，File2 为盘古文件地址。

模式2: 盘古文件拷贝到本地(download)，此时 File1 为盘古文件地址，File2 为本地文件地址，此时 File2 可以为.表示下载盘古文件到本地当前目录。

模式3：盘古文件内部拷贝，此时 File1 和 File2 都为盘古文件地址 (2) 本命令可以使用-m选项来设置拷贝模式（CPMODE），对于模式1和模式3有效，CPMODE 可以有以下选择。

- append：当拷贝的源和目的的文件名相同时，拷贝时采用断电续传技术，之前传过的部分（以长度进行判断）不再继续传递。这也是默认的选项。
- overwrite：当拷贝的源和目的的文件名相同时，拷贝时会覆盖目的地址的旧文件。
- exclude：当拷贝的源和目的的文件名相同时，提示错误退出。不进行拷贝。

说明：

使用默认的append模式可能出现以下问题。本地有一个文件 a.txt，内容是：Hello,Apsara.Hello,Pangu，将 a.txt 上传到盘古，名字也为 a.txt。此时盘古中的 a.txt的内容和本地一样。之后我在本地将 a.txt 的内容更改为：Hello,Apsara,Pangu,Hello,Pangu.Hello,Fuxi，将更改后的 a.txt 上传到盘古，名字还是 a.txt。这样，盘古上的 a.txt 可能会变为：Hello,Apsara.Hello,Pangu,angu.Hello,Fuxi。

补充说明

- 对于模式 1 和模式 3，目的地址都是盘古地址，如果 File2（即dstFile）不存在，该命令会自动创建文件。minCopy 和 maxCopy，默认值都是3。默认的 app-part 属性都是 BIGFILE。
- 若设置 pangu_tool_defaultFileType 这个 flag 的值为 RandomAccessFile，则拷贝到盘古后文件类型为 RandomAccessFile。

- 若设定了pangu_tool_isPiops 为 true，则盘古将保证此文件的 IO 性能达到 pangu_tool_piops_space 和 pangu_tool_piops_iops 这两个所设定的指标。其中 PiopsFlag 可设为 true 或 false，PiopsSpace 为存储空间大小值，如 20G，PiopsIops 为 IO 性能值，如 1000 此 flag 只对目的地址是盘古地址的拷贝有效。
- -t 选项可以设定文件类型，FILETYPE 可以为'normal'或 'raid'

  例子

  pu cp ./a.txt pangu://localcluster/pengjianhong/Test/a.txt pu cp pangu://localcluster/pengjianhong/Test/a.txt pangu://localcluster/pengjianhong/Test/b.txt pu cp pangu://localcluster/pengjianhong/Test/a.txt .

- **cpDir 命令**

  功能：文件夹拷贝。

  命令

  pu cpdir srcDir dstDir [--pangu_tool_defaultMaxCopy=MAXCOPY][--pangu_tool_defaultFileType=RandomAccessFile?] [--pangu_tool_isPiops=PiopsFlag?] [--pangu_tool_piops_space=PiopsSpace?] [--pangu_tool_piops_iops=PiopsIops?] [-m CPMODE] [-t FILETYPE]

  参数说明

  - 本命令支持以下三种模式。

    模式1：本地目录上传到盘古目录（upload）：此时要求 disDir 为盘古目录形式。

    模式2：盘古目录下载到本地目录（download）：此时要求srcDir为盘古目录形式。

    模式3：盘古目录拷贝到另一个盘古目录：此时 srcDir 和 dstDir 都要求为盘古目录形式。
  - CPMODE 和 cp 命令中的 CPMODE，具体说明如下。
    - 在模式 1 或者模式 3 下，如果目标文件夹已经存在并且是 exclude 模式，则提示出错退出。
    - 如果是 append 模式，则对于该目录及其子目录下下的所有文件的拷贝都是用 append 模式。
    - 如果是 overwrite 模式，则对于该目录及其子目录下下所有文件的拷贝都使用 overwrite 模式。
    - 默认值为 exclude 模式（和 cp 不同），即在不使用 global flag 改变默认的 CPMODE 时，dstDir 待是一个未建立的目录。

说明：

cpDir 会递归的拷贝子目录下的所有文件。

- 其他参数适用于该目录及其子目录下的所有文件，参数用法及含义与 cp 命令中的参数相同。

  例子

  假设当前位于 /apsarapangu/disk2/pengjianhong/Test2/ 目录下。

  pu cpdir . pangu://localcluster/pengjianhong/NewTest/ pu cpdir pangu://localcluster/pengjianhong/NewTest/ pangu://localcluster/pengjianhong/NewTest2/ pu cpdir pangu://localcluster/pengjianhong/NewTest2/ ../Test3/。

- **mkdir 命令**

  功能：创建目录。

  命令

  pu mkdir panguDir

  参数说明：panguDir 是盘古格式的目录。

  例子

  pu mkdir pangu://localcluster/pengjianhong/Test2/

- **cat 命令**

  功能：查看盘古文件。

  命令

  pu cat panguFile

  参数说明：panguFile 是一个已经存在的盘古格式的文件路径。

  例子

  pu cat pangu://localcluster/pengjianhong/Test2/HelloWorld

- **meta 命令**

  功能：查看盘古文件的 meta 信息。

  命令

  pu meta panguFile

参数说明：panguFile 是一个已经存在的盘古格式的文件路径。文件的 meta 信息包括：MinCopy-MaxCopy，FileLength，AppName，PartName，FileType，CompressType，ReferenceCount, CreateTime，LastModifyTime。

例子

pu meta pangu://localcluster/pengjianhong/Test2/HelloWorld

注意：该命令只能查看文件的 meta，不能查看目录的 meta 信息。如要查看某目录的 meta 信息，请参见 **dirmeta 命令**

- **touch 命令**

  功能：创建盘古文件。

  命令

  pu touch panguFile [minCopy maxCopy appName partName]

  参数说明：必选参数 panguFile 是一个待创建的盘古格式的文件路径。如果该文件已经存在，则命令会出错。可选参数 minCopy 默认值是1，maxCopy 默认值是1，app 和 part 的默认值都是 BIG_FILE。

  例子

  pu touch pangu://localcluster/pengjianhong/Test2/HelloWorld3

  pu touch pangu://localcluster/pengjianhong/Test2/HelloWorld3 3 5 PJH_APP PJH_PART

- **setReplica 命令**

  说明：设置文件的 minCopy 和 maxCopy。

  命令：pu setreplica panguFile newMinCopy newMaxCopy

  例子

  pu setreplica pangu://localcluster/pengjianhong/Test2/HelloWorld3 1 2

- **setHint 命令**

  说明：设置文件的 appName 和 partName。

  命令

  pu sethint panguFile newAppName newPartName

例子

pu sethint pangu://localcluster/pengjianhong/Test2/HelloWorld3 JHP_APP JHP_PART

- **Quota 命令**

  说明：列出某个目录的 Quota 信息，即该目录及其子目录的 FileNumber Limit（文件数最大限额）和 Used Value（已经存在的文件数），FileLength（文件长度最大限额）和 Used Value（已经存在的文件长度）。如果没有设置 Quota，则会显示 unlimited。

  命令

  pu quota panguDir

  例子

  pu quota pangu://localcluster/pengjianhong/Test2/

- **restore 命令**

  说明：rm 删除的文件，如果没有采用 -p 参数，会暂时放在盘古的回收站中，一段时间之后（默认是1天）才会彻底清除。期间，可以采用 restore 命令进行恢复。

  命令

  pu restore deletedPanguFile

  例子

  pu rm pangu://localcluster/pengjianhong/Test2/HelloWorld

  说明：
  此时文件被放到回收站中。

  pu restore

  pangu://localcluster/deleted/pengjianhong/Test2/HelloWorld_1338442922_0

  此时去执行pu ls pangu://localcluster/pengjianhong/Test2/又可以看到 HelloWorld 文件了。

- **dirMeta 命令**

  说明：得到指定目录的 meta 信息，包括：该目录下（包括其子目录下）文件的总长度，文件数，以及目录数。

  命令

pu dirmeta panguDir

例子

pu dirmeta pangu://localcluster/pengjianhong/

说明：

dirmeta 和 quota 命令都可以得到某目录（递归其子目录）下所有的文件数和文件长度，但两者的结果往往是不同的。因为 Quota 的文件长度反应了文件的拷贝数。而 Quota 的文件数包含了子目录数。dirMeta 操作对盘古的负担要远远大于 Quota 的负担，建议尽量使用 Quota 命令。

- **_touchfilegroup命令**

  说明：创建一个 Filegroup。

  命令

  pu _touchfilegroup panguFileGroupName

  命令说明：必选参数 panguFileGroupName 是盘古某个 FileGroup 的名字，格式为 pangu://localcluster/filegroup/YourFileGroupName，注意没有最后的/，盘古将 FileGroup 作为一个文件看待。

- **rmfilegroup命令**

  说明：删除一个 FileGroup。

  命令

  pu rmfilegroup panguFileGroupName

  例子

  pu lsfilegroup

  得到所有的filegroup

  pu rmfilegroup pangu://localcluster/filegroup/abc

  注意：通过pu lsfilegroup列出来可能有是FileGroup文件，例如abc，有的是FileGroup目录，例如Fuxi/，此时只能删除例如abc形式的FileGroup文件。

- **find命令**

  功能：根据命名或修改时间查找文件或文件夹。

命令

pu find panguDir [-f keyChar] [-d keyChar] [-m maxEditTime,minEditTime]

参数说明

- panguDir：在目录 panguDir 及其子目录中进行查找。
- -f ：只查找文件。
- -d：只查找目录。
- keyChar：查找的文件名或目录名待包含的字符串。
- -m maxEditTime,minEditTime：查找在最近 minEditTime 分钟到 maxEditTime 分钟之间被修改过的文件。

  例子：

  pu find pangu://localcluster/zhuhongyu -f test

  pu find pangu://localcluster/zhuhongyu -d test

  pu find pangu://localcluster/zhuhongyu -m10,1

- **ln 命令**

  功能：为盘古文件建立硬链接。

  命令

  pu ln panguSrcFile panguDstFile

  例子

  pu ln pangu://localcluster/zhuhongyu/a.txt pangu://localcluster/zhuhongyu/la

- **put 命令**

  功能：从本地拷贝文件到盘古系统，是 pu cp 命令的一种拷贝类型。

  命令

  pu put localFile panguFile

  参数： 1. localFile：本地文件路径，文件必须存在。 2. panguFile：盘古文件路径，文件可存在可不存在。 3. 其他可选参数同pu cp命令的参数。

  例子

  pu put b.txt pangu://localcluster/zhuhongyu/b.txt

- **get 命令**

  功能：拷贝盘古文件到本地，是 pu cp 命令的一种拷贝类型。

  命令

  pu get panguFile localFile

  参数说明： 1. panguFile：盘古文件路径，文件必须存在。 2. localFile：本地文件路径，文件必须不存在。 3. 其他可选参数同pu cp命令的参数。

  例子

  pu get pangu://localcluster/zhuhongyu/b.txt ./b.txt

- **cptf 命令**

  功能：将本地文件作为 Tempfile 拷贝到盘古，或将盘古 Tempfile 拷贝到本地。

  命令

  pu cptf <localfile> <cs addr> <filegroup> <index> 或 cptf <cs addr> <filegroup> <index> < localfile>

  参数说明： 1. localfile：本地文件地址。 2. cs addr：chunk server的ip地址及端口号。 3. filegroup：Tempfile 所属的file group name，格式为'/fgName'或'volName@serverName/fgName'。 4. index：Tempfile在所属file group中的index。

  例子

  pu cptf ./a.txt tcp://10.101.xxx.xxx:10260 /fg1 2

  pu cptf tcp://10.101.xxx.xxx:10260 /fg1 2 ./a.txt

- **lstf 命令**

  功能：列出指定chunk server上的Temp file。

  命令

  pu lstf cs_address [-g] [-v VolName?]

  参数说明： 1. cs_address：chunk server 的 ip 地址及端口号。 2. -g：只列出 file group 3. -v：指定volume name。

  例子

  pu lstf tcp://10.101.xxx.xxx:10260

- **lsfilegroup**

  功能：列出指定 file group 目录下的 file group。

  命令

  pu lsfilegroup FgDir?

  参数说明

  FgDir：指定的 file group 目录名，FgDir可以指定Volume Name和Cluster Name，如'VolA@localcluster/myfg'。

  > **说明**：
  > FgDir 不能以pangu:// 开头，且FgDir 不能为空。

  例子

  pu lsfilegroup VolA@localcluster/myfg

  pu lsfilegroup fg

  pu lsfilegroup /（列出master上所有的file group）

- **rmfilegroup**

  功能：删除指定的 file group。

  命令

  pu rmfilegroup fgName

  参数说明： fgName：要删除的file group名字，要以'/'开头。

  例子

  pu rmfilegroup fg1

- **rmfgdir**

  功能：删除指定的 file group 目录。

  命令

  pu rmfgdir /fgDirName/

  参数说明

fgDirName;要删除的file group目录的名字（注：当前版本无法构造出能使用这条命令的场景，因为fgDirName必须为空目录，而当一个file group目录下不再有file group存在时mater会自动删除这个空目录）。

例子

pu rmfgdir /myfgDirName/

- **du**

  和quota命令相同

- **dev**

  功能：列出 pu 支持的开发命令

  命令：pu dev

- **_open4append**

  功能：打开一个文件，打印出打开这个文件的耗时。

  命令

  pu _open4append FileName?

  参数说明

  PanguFileName：要打开文件的地址，文件可以为本地文件或盘古文件。

  例子

  pu _open4append a.txt

- **_checkReplicaNumberRecursivly**

  功能：递归的检查指定目录下文件的 minCopy 和 maxCopy 是否是指定的值，输出满足条件的文件信息。

  命令

  pu _checkReplicaNumberRecursivly PanguDirPath? c_minCopy c_maxCopy

  参数说明： 1. PanguDirOrFilePath?：待检查的目录地址。 2. c_minCopy c_maxCopy：所指定的 minCopy 和 maxCopy 值。

  例子

pu _checkReplicaNumberRecursivly pangu://localcluster/test/ 2 2

- **_modifyReplicaNumberRecursivly**

  功能：递归的将指定目录下文件的 minCopy 和 maxCopy 是所指定值的文件的 minCopy 和 maxCopy 设置为新的值。

  命令

  pu _modifyReplicaNumberRecursivly PanguDirPath? o_minCopy o_maxCopy n_minCopy n_maxCopy

  参数说明： 1. PanguDirOrFilePath?：要设置文件新的 minCopy 和 maxCopy 值的目录地址 2. o_minCopy o_maxCopy：所指定的原有的 minCopy 和 maxCopy 值 3. n_minCopy n_maxCopy ：要设置成的新 minCopy 和 maxCopy 值。

  例子

  pu _modifyReplicaNumberRecursivly pangu://localcluster/test/ 2 2 3 3

- **_readChunkFile**

  功能：读取所指定的 chunk file 的内容。

  命令

  pu _readChunkFile csAddr fileID chunkIndex chunkVersion readFileLength

  参数说明： 1. csAddr：chunk serve r的 ip 地址+端口。 2. fileID：要读取的文件的 File ID 。 3. chunkIndex：要读取的文件所在的 chunkIndex 4. chunkVersion：要读取的文件所属的 chunkVersion。 5. readFileLength：要读取的字节数，可以少于文件的总字节数。

  > **说明**：
  > 参数（2）到参数（5）可以通过puadmin gfi 命令获得。

  例子

  pu _readChunkFile tcp://xxx.xxx.xxx.xxx:10260 109955457744897 0 1 50

4 大数据应用加速器

4.1 运维工具系统

DTBoost的运维主要是DTBoost自带的管理运维后台。

DTBoost的运维后台，可以方便地查看当前应用的运行状态。同时也可以提供模块的更新发布、版本切换、系统日志查询等服务。

4.2 例行维护

日常的例行维护通过DTBoost运维后台完成。

4.2.1 系统用户管理

DTBoost 系统装机之后，账号体系集成了Base的账号体系。

第一个进入DTBoost的base用户，可以将自己设置为DTBoost的root账号（一般在安装完之后就会执行这步骤，请记录这个root账号），这个账号日后会用来管理DTBoost上其他账号的权限设置和账号开关。

账号管理入口：在页面上，选择**dtboost服务** > **设置** > **用户管理**。

4.2.2 运维后台管理

进入运维管理后台的方法如下所示：

1. 在浏览器地址栏中，输入系统管理后台的地址：*dtboost* 服务域名:*9999* 端口。

   说明：
   您可登录天基，导航至DTBoost服务，即可查询DTBoost服务域名。

2. 输入管理员的初始账号，包含用户名及密码。
3. 单击**登录**，选择运维集群（默认 **专有云默认集群**）进入运维后台，如图 *4-1:* 运维后台所示。

图 4-1: 运维后台

运维后台主要提供应用列表、应用发布、系统监控、集群管理、配置管理和日志查询六个功能。

应用列表

应用列表主要提供以下功能：

- 查看目前安装的模块，以及运行状况。
- 管理运行的模块，包括启动、关停、删除和重启，如图 *4-2:* 应用列表所示。

图 4-2: 应用列表

name	version	proc num	publish at	status	action
__ADMIN__	0.0.0_0	1/1		online	
__PROXY__	0.0.0_0	2/2		online	
admin	4.0.3_7	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
analyse-fe	2.2.0_21	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
analysis	2.1.16_17	0/0	6/29/2017, 2:54:41 PM	offline	download delete start
	2.1.16_20	4/4	7/4/2017, 7:15:17 PM	online	download stop restart
console	2.1.0_92	0/1	6/29/2017, 2:54:41 PM	offline	download delete start
factory	0.0.1_12	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
ruleengine	2.0.0_1	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
service	1.0.2_1	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
smartmove_v2	2.1.2_13	1/1	6/29/2017, 2:54:41 PM	online	download stop restart
system	3.2.0_7	1/1	6/29/2017, 2:54:41 PM	online	download stop restart

说明：

- 支持多版本无缝切换，每个模块至少要保持一个版本在线。
- 在线的版本，需要先停止（stop），才能删除（delete）。

应用发布

如图 *4-3: 发布页面*所示，应用发布主要用于升级更新模块，将官方提供的应用设计包，通过这个页面发布上去即可完成快速升级。

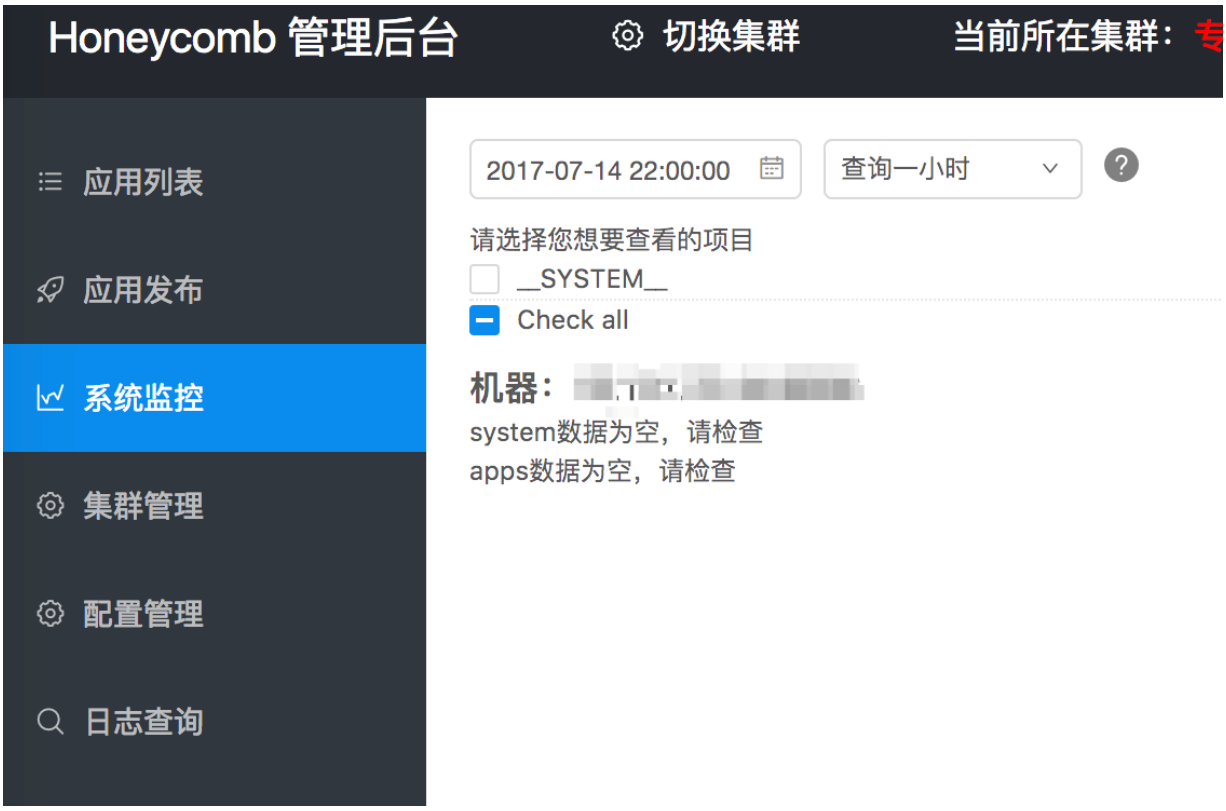
图 4-3: 发布页面

系统监控

选择对应时间范围内机器上各系统应用的状态。

如图 *4-4: 监控系统的状态*所示，可以查看系统监控数据。

图 4-4: 监控系统的状态

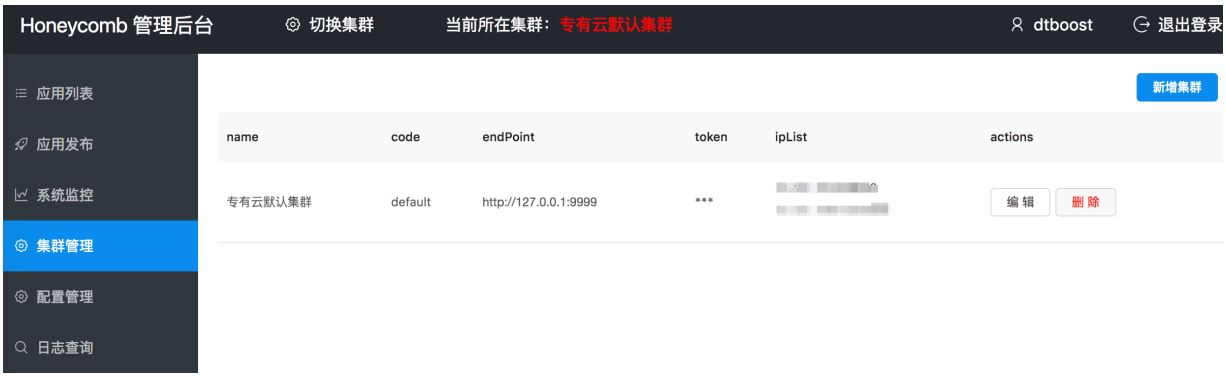


集群管理

管理专有云集群里机器配置信息，新加或更改机器。

如图 *4-5: 集群管理*所示，可以查看和编辑机器信息。

图 4-5: 集群管理



配置管理

管理应用内各 APP 的配置，可以新加配置。

如图 *4-6:* 配置管理所示，可以查看和编辑机器信息。

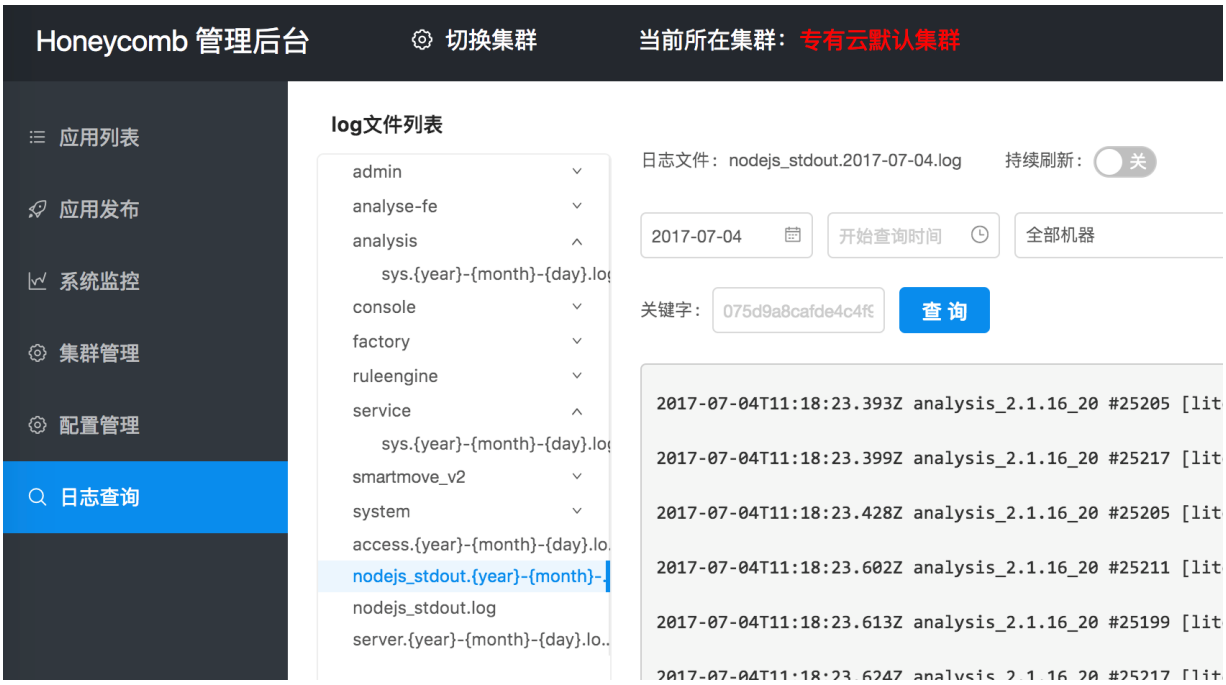
图 4-6: 配置管理

日志查询

根据日期、关键词、request_id 等来查询各系统的日志。

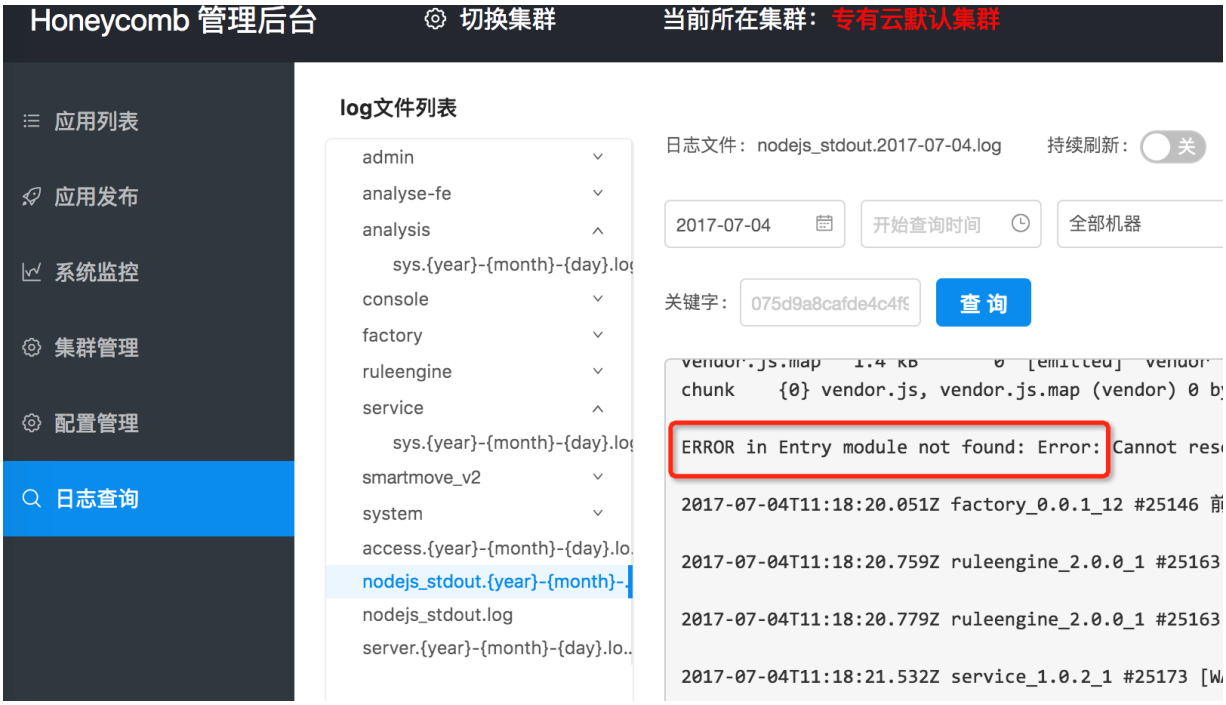
如图 *4-7: 查看服务的log*所示，可以查看服务的log。

图 4-7: 查看服务的log



如图 *4-8: 查看各模块的log*所示，日志中有 ERROR输出，可以反馈到工单中。

图 4-8: 查看各模块的log



4.3 备份与恢复

4.3.1 备份数据

dtboost的meta库自带高可用，无需备份数据。

如需要完整的备份数据，请联系阿里云的技术支持寻求帮助。

4.3.2 恢复数据

暂无。

4.4 故障处理

4.4.1 常见故障处理

4.4.1.1 断电恢复

请提交工单向技术支持寻求帮助。

4.4.1.2 物理设备损坏

请提交工单向技术支持寻求帮助。

4.4.1.3 应用故障

4.4.1.3.1 访问故障

如果出现无法访问的情况，可以先检查DTBoost运维后台的模块是否都运行正常。

如果不正常，可以选择重启服务。

如果重启仍然无法解决，请提交工单向技术支持寻求帮助。

4.4.1.3.2 登录故障

操作步骤

1. 如果出现无法登录的情况，请先清理浏览器的缓存、cookie，重新尝试登录。
2. 如果登录框提示登录异常，请根据提示异常检查：
   - 是否密码不对
   - 是否账号锁定
   - 是否账号被关停
3. 如果还是无法定位，请提交工单寻找技术支持。

4.4.1.3.3 服务接口异常

如果是服务接口无法访问，参考访问故障。

如果提示签名错误，请检查签名算法 **secretToken**。

如果提示签名过期，请检查服务器的系统时间是异常，调整好系统时间，即可修复此问题。

5 大数据管家

5.1 例行维护

大数据管家（Big Data Manager）日常的例行维护主要通过天基完成。

- 天基Portal巡检

  打开天基 Portal，找到 BCC 这个 Project，查看所有容器是否是达到终态。

  在 BCC Project 的 DashBorad 里确认该 Project 没有其他报错信息。

- 监控项以及告警处理

  - 硬件监控

    磁盘报警：正常情况下，系统会自动回收 log（保留最近30天），但不排除异常情况下，日志暴增导致系统磁盘告急，目前这种情况下请直接寻找技术支持。

  - 系统异常

    根据巡检提示信息，在 I+运维平台处理异常。如提示不明确，请联系阿里云技术支持获得帮助。

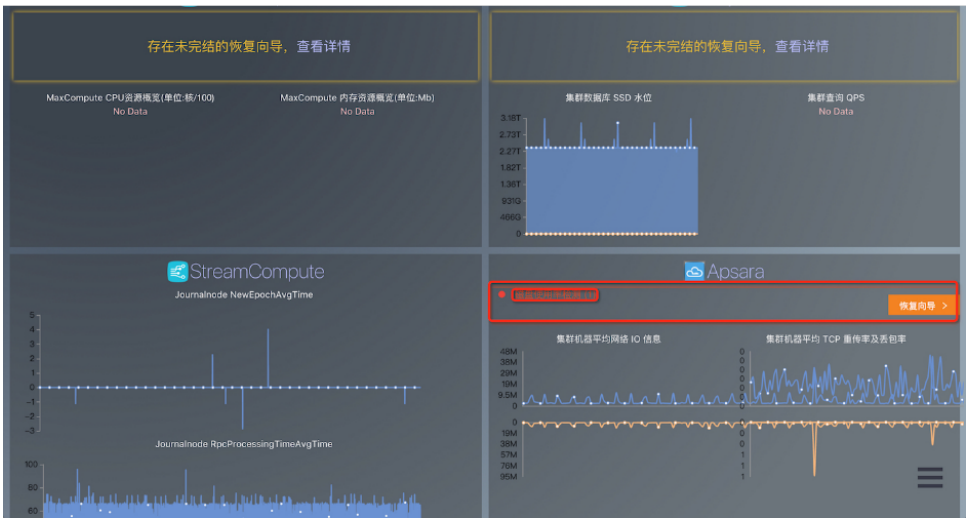
5.1.1 自检

自检是产品服务提供的服务自检查的功能，提供定时调度运行检查，将检查的结果通过页面进行展示，让您更好地知道服务当前状态的一种功能。自检包含状态及数量显示、服务下自检项信息、手动运行自检项、自检项运行的机器和状态、自检项运行详情、自检配置（在配置功能中体现）。

首页

在首页中可以看到产品列表，产品当前状态（绿色为自检均正常，红色为不正常），左边为状态，为告警和系统错误的数量（红色表示错误），右侧有**恢复向导**。

图 5-1: 产品列表



服务页面

服务页面上，产品或服务前面的颜色点表示当前服务下子自检状态，红色为有自检运行不正常，绿色为运行都正常。有的服务下面并没有挂自检行，即没有自检项在服务对应的机器运行，服务的健康历史显示为**当前产品（服务）暂时没有问题，很健康**，这种情况下产品或服务前面的颜色点均为绿色（如图服务正常）。而对于子服务状态，子服务前面的颜色点也具备子服务自检状态的体现，而验颜色点仅体现本服务状态，其子服务状态不体现；子服务右侧有显示颜色及数字同产品列表中意义，数量统计的是本服务和子服务数量总和（如图服务总量）。

图 5-2: 服务页面

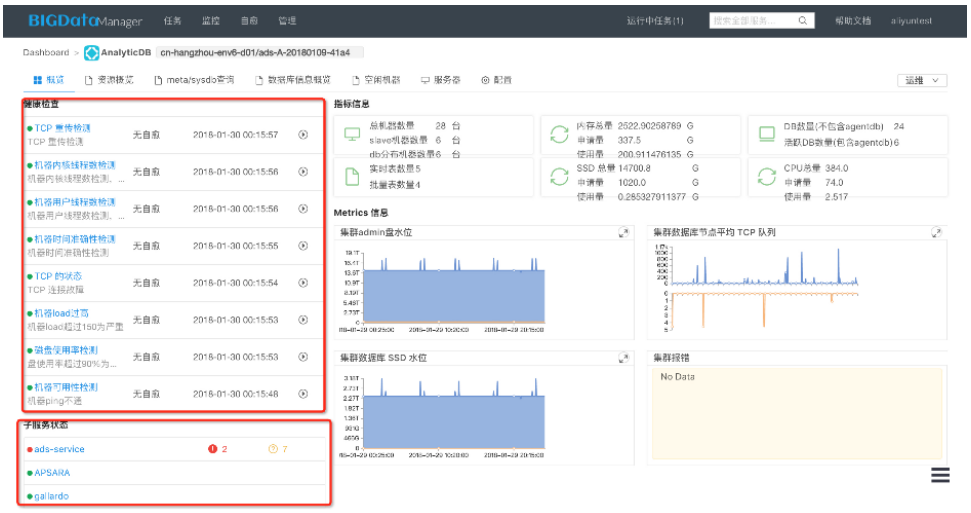


图 5-3: 服务正常

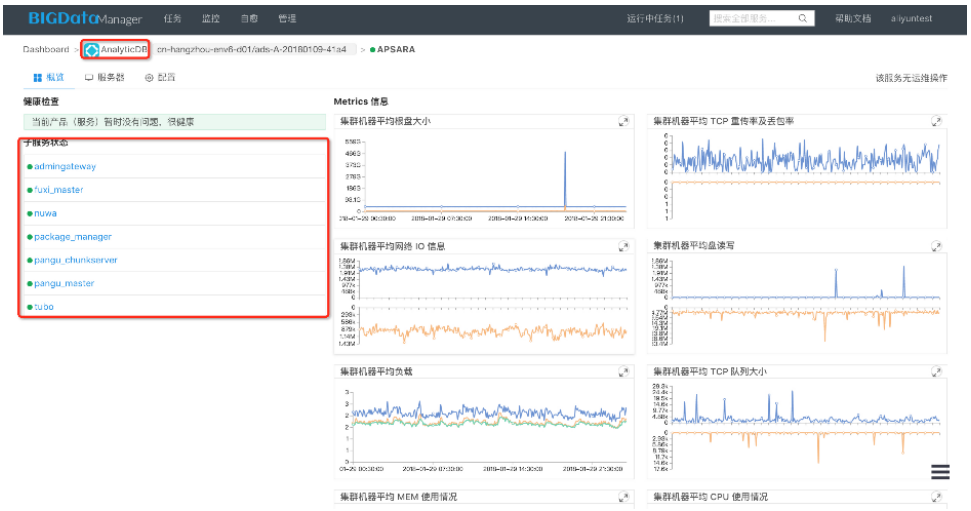
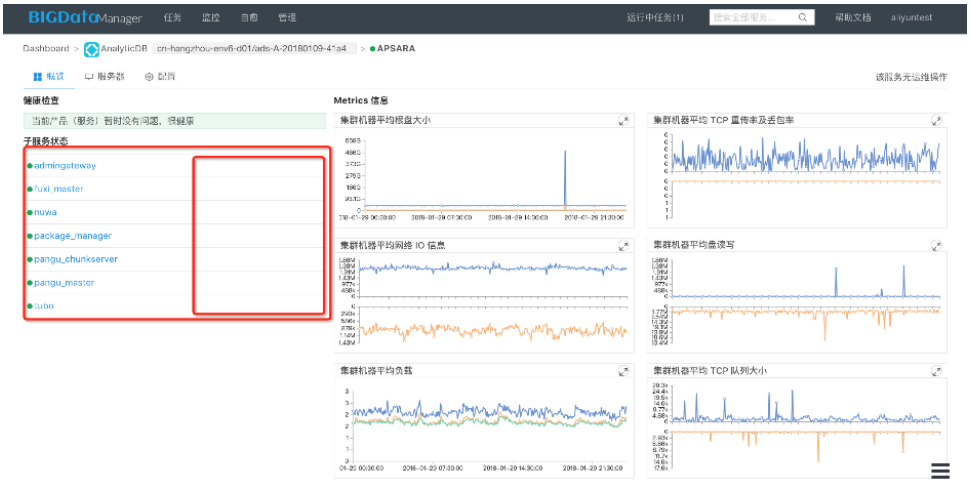


图 5-4: 服务总量



服务下自检项信息：如图服务正常所示，每个服务下健康检查列表中存在有列表数据的，都是为服务健康做的自检，每个自检项都包含状态（红色为自检项中有至少在一个机器上运行不正常，绿色为运行都正常）、自检名称和说明、最近运行完成时间、手动运行按钮。健康检查显示自检项策略：如果服务自检大于八项，自检不正常数量大于八的，不正常自检项均显示，不正常数量小于八个项时，只显示八项，包括不正常的和最近完成的正常自检项，如果都正常，那么就显示八项正常的；如果服务自检不足八项，则全部显示出来。状态异常的再上，状态一样的时间越新显示越靠上。

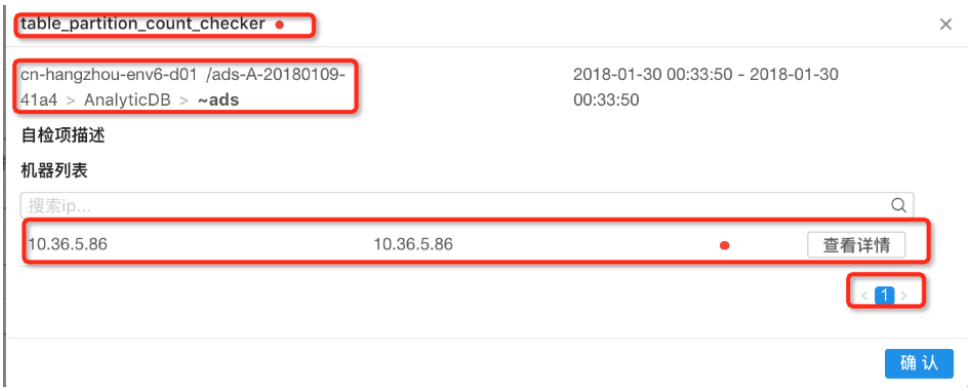
手动运行自检项：如图手动自检所示，每个自检项后面都有一个手动执行按钮，您可以通过单击按钮触发此自检项马上运行健康检查。此功能主要是为了让客户在某些场景下更快的发现服务问题，如有一些自检项运行十分耗时间和耗资源，您在配置自检的时候希望此类自检不经常运行，这

样一来客户查看的自检结果就丢失了一定的实时性，不能体现当前服务的自检状态。手动执行正是为了让客户能够在等待一定时间（自检运行时间）下，能够马上查看到自检状态。您只要单击提交手动执行按钮，过一段时间刷新即可。提交手动执行后，该自检项提示正在**手动执行中**字样，执行完毕后字样会消失。

图 5-5: 手动自检

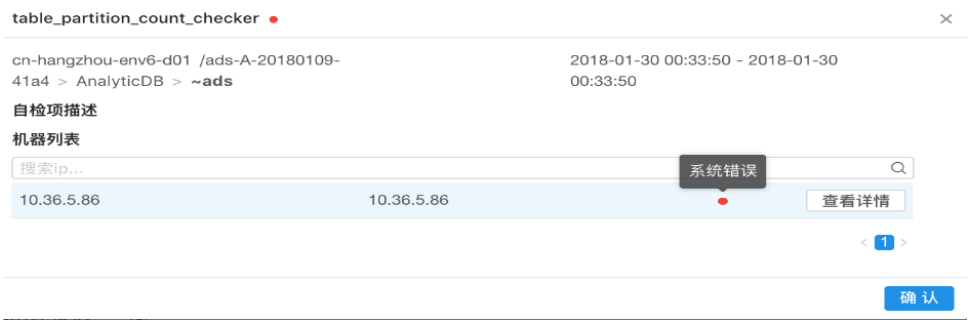
自检项运行的机器和状态：单击自检项名称后，会打开一个自检项运行的机器和状态列表的对话框（如图自检对话框），对话框中包含自检项名称、自检项所在的服务、自检项运行的起止时间（所有机器运行的起止时间）、运行类别（自助发起和手动执行）、自检项描述信息、分页的机器运行状态列表、IP搜索框。

图 5-6: 自检对话框



单击表示状态的图标，提示状态信息（如图显示信息）。

图 5-7: 显示信息



自检项运行详情：如图自检对话框，单击每个自检机器的**查看详情**，即可打开该自检项在此机器的运行详情（如图运行详情），分别有该自检项在机器上运行的起止时间，运行的脚本、脚本退出码、脚本运行结果和详细输入信息。

图 5-8: 运行详情

5.1.2 故障处理

常见故障处理

- **登录故障**

  如出现无法登录的情况，请先清理浏览器的缓存、cookie，重新尝试登录。

  如果登录框提示登录异常，请根据提示异常检查以下内容。

  - 是否密码不对
  - 是否账号锁定
  - 是否账号被关停

- **其他异常**

  请寻找技术支持。

5.2 备份与恢复

备份数据

大数据管家使用的数据库自带高可用，无需备份数据。如需要完整的备份数据，请联系技术支持寻求帮助。

恢复数据

暂无。

6 关系网络分析

6.1 运维

6.1.1 查看实例

通过对实例的查看和分析，了解实例的运行情况，对有问题的实例进行主备切换，清理日志等运维操作。

查看CE运行实例

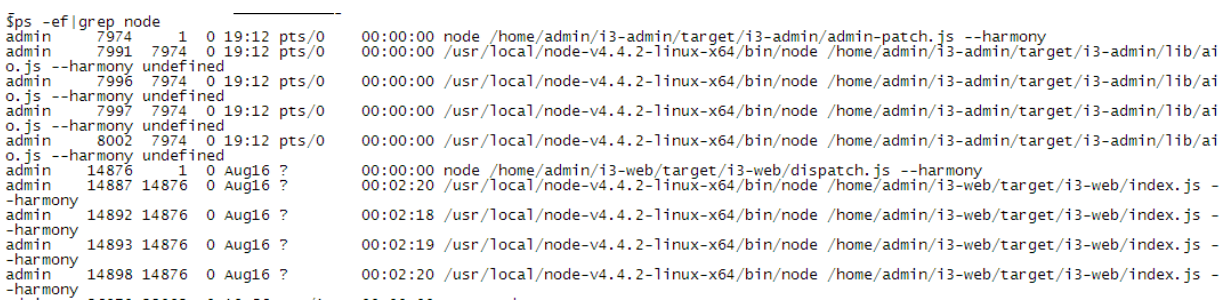
登录I+应用服务器，执行ps –ef|grep java命令，如果存在如图 *6-1: 查看CE运行实例*所示的进程表示I+后端服务正常。

图 6-1: 查看CE运行实例

查看node实例

登录I+应用服务器，执行ps –ef|grep node命令，如果存在如图 *6-2: 查看node实例*所示的进程表示I+ node服务正常。

图 6-2: 查看node实例



其中，i3-web表示I+前端正常，i3-admin表示I+管理前端服务正常，如果没有发布I+管理，可以没有i3-admin的进程。

6.1.2 文件日志

I3Eye应用及CE日志：

I3eye/CE的日志目录为：/home/admin/logs目录。

/home/admin/logs目录挂载了一块100G的数据盘，随着运行时间增长，日志会越来越多，需要做自动清理。清理分两种策略：

- 第一种策略：根据时间清理，也就是自动清理2周前的日志（文件的创建日期）。
- 第二种策略：根据整目录的日志大小，如果占用整块数据盘空间80%以上时，将自动清理最老的日志文件。

6.1.3 数据库日志

数据库日志记录i3相关程序执行时间轨迹，其中主要的为ADS执行的sql，包括成功与否、异常、执行时间。

登录到mysql的i3eye数据库。

查看最近一次操作i3eye程序执行的时间轨迹的sql语句如下：

```
SELECT * from i3eye_time_trace WHERE main_time_trace_id in (
SELECT max(main_time_trace_id) from i3eye_time_trace);
```

查询ads最近1小时执行的sql语句如下：

```
select * from i3eye_time_trace where name like 'com.alibaba.iplus.common.dal.manual%' and (
gmt_create < now() and gmt_create > date_sub(now(), interval 1 hour) );
```

查询ads最近一小时出错的sql语句如下：

```
select * from i3eye_time_trace where complete = 0 and name like 'com.alibaba.iplus.common.
dal.manual%' and (gmt_create < now() and gmt_create > date_sub(now(), interval 1 hour) );
```

6.1.4 停止服务

用admin用户登录I+服务器，执行启动脚本，用ps命令查看进程：

- 查看CloudEngine进程：ps –ef|grep java
- 查看node进程：ps –ef|grep node

将对应线程kill掉，就可以停止服务。

6.1.5 重启服务

用admin用户登录I+服务器，执行启动脚本：

- 直接部署启动i3eye，i3-web，i3-admin：i3eye-deploy.sh start
- 仅部署启动i3eye：i3eye-deploy.sh start_i3eye
- 仅部署启动i3web：i3eye-deploy.sh start_i3web
- 仅部署启动i3admin：i3eye-deploy.sh start_i3admin

6.2 安全维护

6.2.1 网络安全维护

网络安全维护包括设备安全和网络安全。

设备安全

- 检查网络设备，启用设备的安全管理协议和配置。
- 检查网络设备软件版本，及时升级到更安全的版本。
- 具体安全维护方法请参见各设备的产品文档。

网络安全

根据系统网络的现状，可以适当选配入侵检测系统（IDS）或入侵防御系统（IPS）对来自外部或者内部网络的数据流量进行检测，实时防御网络内的异常行为和攻击行为。

6.2.2 账号密码维护

账号密码包括I+系统管理员密码和设备密码。

为了保证账号安全，请定期修改系统和设备密码，并使用高复杂度的密码。

6.3 故障处理

6.3.1 故障响应机制

维护人员应该建立故障应急响应机制，以保证出现故障或者安全事故后，可以尽快排除故障，恢复生产。

6.3.2 故障处理方法

维护人员在日常维护中发现系统故障后，可以参考本文档的运维部分解决问题。

如遇不能解决的故障，请收集故障信息（包括系统信息、故障现象等），联系阿里云技术支持工程师，并在技术支持工程师的指导下解决问题。

故障解决后，维护人员应及时对问题进行检查、总结和改进。

6.3.3 常见故障处理

磁盘空间不足

可能原因：I+系统日志过大。

解决方法：如果是监控日志，一般放在/home/admin/logs，可以将之前的日志删掉。

机器维修或者下线

可能原因：机器硬件损坏或者机器过保。

解决方法：重新安装I+。

进程异常

可能原因：未自启动或者异常退出，可以查看/home/admin/logs下的日志，检查原因。

解决方法：重启I+。

6.3.4 硬件故障处理

磁盘故障

解决方案：由于I+一般采用集群部署，可以直接将I+所有线程结束，然后更换硬盘后启动。

内存、主板、CPU、电源等需要停机更换的故障

解决方案：

当需要进行停机维修时：

- 如果能进入到系统，则按停止服务的步骤来停止该机器上的I+服务。
- 如果不能进入系统，只有强制关机。