Oracle 采集代理说明书

DataPipeline

shao@datapipeline.com

DataPipeline

Table of contents

1. Oracle 实时数据采集代理安装与运维 3
2. 简介 4
2.1 支持 4
2.2 更新日志 6
3. 安装 13
3.1 系统要求 13
3.2 准备数据库 15
3.3 安装步骤 18
3.4 配置输出 20
3.5 配置参数 21
3.6 程序日志输出配置 24
3.7 文件下载 25
4. 使用 26
4.1 采集量及性能评估 26
4.2 资源限制 27
4.3 高可用 29
4.4 Agent 迁移程序 30
4.5 API 33
4.6 命令行工具 59
4.7 运维场景 63
5. 附录 69
5.1 LogMiner审计日志内容 69
5.2 指标监控 70
5.3 ASM磁盘信息配置 73

1. Oracle 实时数据采集代理安装与运维

该软件在本文中简称采集代理。PDF 版本下载

2. 简介

2.1 支持

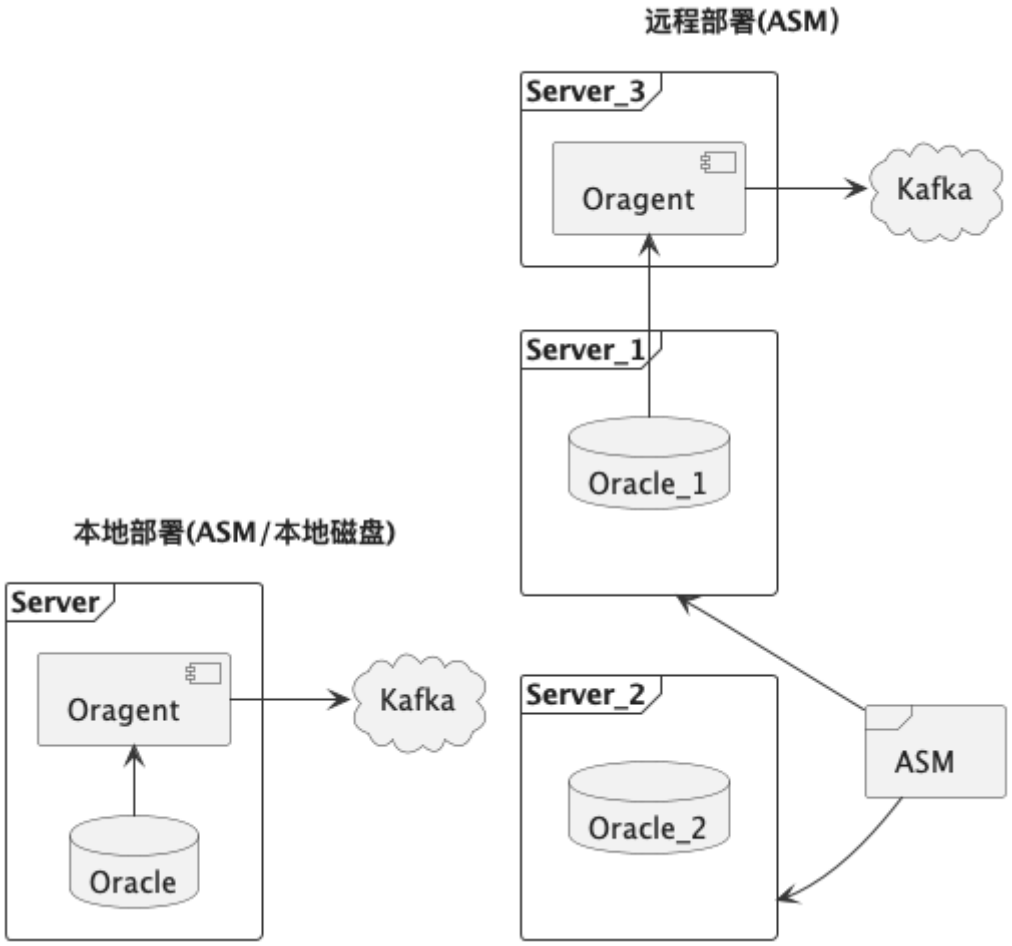
2.1.1 支持

描述	✓
数据库版本	9i, 10g, 11g, 12c, 19c
数据类型	CHAR, NCHAR, VARCHAR2, NVARCHAR2, DECIMAL, NUMBER, FLOAT, DATE, TIMESTAMP, TIMESTAMP_WITH_TIME_ZONE, TIMESTAMP_WITH_LOCAL_TIME_ZONE, CLOB, NCLOB, BLOB, LONG, RAW, BINARY_DOUBLE, BINARY_FLOAT, INTERVAL_YEAR_TO_MONTH, INTERVAL_DAY_TO_SECONDS
DDL	创建表, 删除表, 重命名表, 清空表, 添加字段, 删除字段, 重命名字段, 修改字段, 创建分区/子分区, 删除分区/子分区, 分裂分区/子分区, 合并分区/子分区, 交换分区, 修改分区 ,重命名分区
表类型	普通表，即堆组织表，含分区表，无压缩
数据库模式	多租户(PDB)，备库(ADG)
部署模式	远程，本地
操作系统	Windows, AIX 7, Linux

2.1.2 不支持

描述	✗
表类型	索引组织表，临时表，外部表，压缩表，嵌套表，对象表，簇表，列存储表，分布数表，内存表
数据类型	BFILE, UDT
部署模式	无 ASM 的远程
操作系统	AIX 5（不支持 java1.8）

2.1.3 部署架构


远程部署(ASM)
Server_3
Oragent
Kafka
Server_1
Oracle_1
本地部署(ASM/本地磁盘)
Server
Oragent
Kafka
Oracle
Server_2
Oracle_2
ASM

2.2 更新日志

2.2.1 [3.1.4] - 2025-08-31

- 修复安全漏洞扫描的相关问题。

2.2.2 [3.1.3] - 2025-07-31

- 日志中的敏感信息进行脱敏处理。
- 修复高可用切换后数据字典刷新的问题。

2.2.3 [3.1.2] - 2025-06-31

- 新增对日志中的敏感信息进行脱敏处理。
- 修复高可用切换后没有重新刷新数据字典的问题。
- 修复部分回滚数据没有正确处理的问题。

2.2.4 [3.1.1] - 2025-04-30

- 支持远程读取代理（RFA）模式

2.2.5 [3.1.0] - 2025-03-31

- 修复部分回滚操作没有正确处理的问题。
- 修复开启了并发解析后有概率报错的问题。
- 优化 ASM 连接因网络中断阻塞过长后自动重启重建连接池。

2.2.6 [3.0.17] - 2025-02-28

- 修复某种情况下删除数据所有列为空的问题。
- DataPipeline 受控停止功能适配。

2.2.7 [3.0.16] - 2025-01-30

- 修复某些数据整合问题。
- 修复某些情况下事务回滚导致处理过慢的问题。

2.2.8 [3.0.15] - 2024-12-31

- 修复某些数据整合问题。
- 修复解析 DDL 表刷新结构相关问题。

2.2.9 [3.0.14] - 2024-11-30

- 修复 Oracle 数据节点切换可能丢失数据的问题。
- 修复连续SQL变化导致漏掉头几条数的问题。

2.2.10 [3.0.13] - 2024-10-31

- 修复多 PDB 模式下获取 PDB 信息失败后的自动重启造成的连接泄露。
- 修复以错误的用户名密码重启后，再更新成正确的密码，没有重新获取表结构的问题。
- 修复 RAC 节点下，某些情况下事务提前输出的问题。
- 修复某些数据整合问题。

2.2.11 [3.0.12] - 2024-09-31

- 提升处理事务回滚的效率。
- RAC 节点宕机恢复后，自动恢复解析。
- 支持 VARCHAR (deprecated) 等同于 VARCHAR2 类型。
- 恢复 Kafka producer 的指标采集。
- 支持表名和用户名含有特殊符号。

2.2.12 [3.0.11] - 2024-08-31

- 修复密码含有特殊字符的问题。
- 支持指定 JDBC 驱动版本。

2.2.13 [3.0.10] - 2024-07-31

- 新增某些数据整合逻辑。
- 当网络异常时，重建连接池。
- 添加参数 txn_scn_sub 参数，控制缓存文件名是否包含 scn_sub 信息。默认为 false。

2.2.14 [3.0.9] - 2024-06-31

- Kafka Producer close 添加超时时间。

2.2.15 [3.0.8] - 2024-05-13

- 修复分区表在发生 DDL 变化后没有应用新表结构的问题。
- 增加参数 param.notification.send_ddl (默认为 false) 控制是否发送 DDL 事件。
- 增加参数 param.enable_partition_validation (默认为 false) 控制是否启用分区表验证。
- 将错误数据的发送间隔参数化：param.notification.

2.2.16 [3.0.7] - 2024-05-06

- 如果在处理带有 LOB 列的表时，获取列信息失败后将错误信息并发送通知。

2.2.17 [3.0.6] - 2024-04-12

- 修复高可用模式下，在激活和回复时有可能出现重复采集任务的问题。

2.2.18 [3.0.5] - 2024-04-02

- 修复遭遇错误重启时，有概率死锁的问题。影响范围: 3.0.1 - 3.0.4

2.2.19 [3.0.4] - 2024-03-08

- 新增 oracle.yml 中 src 下参数 connection_pool 用于自定义连接参数。

2.2.20 [3.0.3] - 2024-03-06

- 修改 Failed to merge 的日志级别为 WARN。

2.2.21 [3.0.2] - 2024-01-26

- 支持灾备相关功能，详见 灾备使用指南。

2.2.22 [3.0.1] - 2024-01-22

- 替换由于 JDK 版本而可能造成内存泄露的第三方 Semaphore 库。

- 修复远程读取 ASM 情况下，log 块大小 为 4096 时导致解析报错问题。

2.2.23 [1.1.37] - 2024-01-04

- 增加参数 mute_rac 支持禁用下线的 RAC 并继续解析上线的 RAC。
- 修复远程读取 ASM 情况下，log 块大小 为 4096 时导致解析报错问题。
- 修正采集 pdb 数据库某些情况下错误过滤问题。

2.2.24 [3.0.0] - 2023-12-28

- 支持多 PDB 模式采集。

2.2.25 [1.1.36] - 2023-12-28

- 修复 RAC 集群中 interval 分区表新增分区时，小概率丢失头几条数据的问题。

2.2.26 [2.1.5] - 2023-12-18

- 增加参数 mute_rac 支持禁用下线的 RAC 并继续解析上线的 RAC。

2.2.27 [2.1.4] - 2023-12-15

- 暂时禁用 Kafka producer_byte_rate metrics 指标，避免反复重启重建 Producer 后导致 micrometer metrics 线程无法释放最终导致 OOM 问题。
- 升级 kotlin SDK 至 1.8.20，避免在某些情况下 SemaphoreSegment 无法释放问题。

2.2.28 [2.1.3] - 2023-11-28

- 修复事务在提交后不会立马删除缓存的事务文件问题。影响版本 2.1.0 - 2.1.2。
- 增加 param 参数 tx_expire_hr （ 默认为 -1 即不回收 ），回收因上述缺陷引起的过大磁盘占用问题。请至少设置为 12 及以上。以防止误删未提交事务缓存。

2.2.29 [2.1.2] - 2023-11-22

- 修复在极端情况下，重置 IGNORE_KAFKA_EXCEPTION 设置可能造成的死锁问题。

2.2.30 [2.1.1] - 2023-11-14

- 新增配置 match_no_schema_sql 对于原始 SQL 语句中没有 schema 的情况，是否匹配到配置中的监听表对象。默认为 false，即不匹配。
- 增加对 record 的长度判断，防止解析到无效的 record。

2.2.31 [2.1.0] - 2023-11-01

- 支持读取与解析异步化及并发化。配置项名称与效果详见 配置详情。

2.2.32 [2.0.1] - 2023-10-27

- 支持 RAC 节点之一宕机后继续解析。

2.2.33 [2.0.0] - 2023-10-26

- 支持分布式高可用模式，详见 高可用使用指南。

2.2.34 [1.1.35] - 2023-10-24

- 增加子线程退出时的报错日志打印。
- 增加 Rowid 变化定期巡检功能。
- 修复 legacy_format = true 时，record version 冲突造成的兼容性问题。

2.2.35 [1.1.34] - 2023-10-17

- 修复 /tools/logs 接口执行 SQL 兼容 Oracle 10G 的问题。
- 修复错误处理遭遇异常时无法恢复的问题。

2.2.36 [1.1.33] - 2023-10-12

- 修复 RAC 集群中 interval 分区表新增分区时，小概率丢失头几条数据的问题。
- 修复 DROP, TRUNCATE 分区表事件没有捕获的问题。

2.2.37 [1.1.32] - 2023-09-28

- /config/validation 接口增加 verifyFilter 于 fixFilter 两个参数，用于检查表过滤功能是否正常及恢复。
- 屏蔽非法时间造成的异常延迟值。

2.2.38 [1.1.31] - 2023-08-22

- 区分手动启动(Agent start)与自动重启(Agent restart)的通知。
- 增加 local_src_login (默认为 false) 屏蔽 web 请求更新数据库连接信息的配置以应对某些特殊的网络情况。
- 修复 ASM 本地磁盘映射信息无法通过 web 请求保存的问题。

2.2.39 [1.1.30] - 2023-07-31

- 支持动态更新连接信息并重建连接。
- /export/health 接口若阻塞超过 3 秒则报错。
- 修复某些 ORP 被忽略的问题。

2.2.40 [1.1.29] - 2023-07-11

- 修复事务中违反约束的操作不进行输出。
- 解析失败的 Redo Record 产生最多 (err_files_max 默认为 0, 不记录）个文件置于 err 文件夹下。
- 新增 HTTP API /tools/download/err 下载 err 打包文件。

2.2.41 [1.1.28] - 2023-07-07

- 支持直接路径插入 (DIRECT-PATH INSERT)。
- 超过工作线程最大超时时间 (worker.max_blocking_ms ）后重建 Kafka Producer 实例。
- 增加配置项 missing_log_tolerance_times (默认为 1)。当发生主备较大延迟导致找不到日志时，在报错前可以容忍监听备库找不到日志的重启次数。
- 增加配置项 pdbs_schema 监听 PDB 数据库时，指定 DP_PDBS 表所在用户。
- 重构 CLI 部分代码。

2.2.42 [1.1.27] - 2023-06-15

- 修复默认值错误被整合的问题。

2.2.43 [1.1.26] - 2023-06-08

- ojdbc8 driver 降级为 12.2.0.1 规避兼容性问题.
- 修复某种数据整合问题。
- 增加选项 skip_validation: true 可使得 /config/validation 接口一直返回通过。

2.2.44 [1.1.25] - 2023-05-26

- 添加校验查询权限及连接校验接口 /config/validation。
- 修复部分 URP default 值为空的问题。
- 重构参数配置相关代码。

2.2.45 [1.1.24] - 2023-05-19

- 升级 ojdbc8 driver 至 21.3.0.0。

2.2.46 [1.1.23] - 2023-05-14

- 修复部分数据错乱问题。
- 优化大事务内部排序。
- 修复监听备库且备库只有单个 REDO 起效时的定期滞后解析问题。

2.2.47 [1.1.22] - 2023-03-31

- 新增工作线程阻塞超时参数 worker.max_blocking_sec，默认值 1800。超时后报错重启，设置方式参见配置文档。

2.2.48 [1.1.21] - 2023-03-23

- 修复 DEFAULT 列值在 UPDATE 的 before 中某些时候为空的问题。

2.2.49 [1.1.20] - 2023-03-16

- 修复 SRC_ID 被错误的初始化导致发送 topic 无效的问题。
- 修复 QmiSlot 列表为空导致的报错问题。

2.2.50 [1.1.19] - 2023-03-07

- 添加监控默认配置文件。
- 修复隐藏索引列导致寻找列 ID 失败。
- 取消不必要的事务提交排序步骤。

2.2.51 [1.1.18] - 2023-02-27

- 修复某些情况下文件句柄泄露问题。
- 适配 /metrics 接口与节点页面监控。

2.2.52 [1.1.17] - 2023-02-13

- 添加 /metrics 接口监控各项关键指标。
- 添加 recovery_mode = reset (默认为 none) 时，VirtualMachineError 导致的程序报错自动重置至最新进度。
- 添加 skip_error = true (默认为 true) 时，自动忽略损坏事务造成的错误。
- 修复非当前在线日志没有跳过无效日志段的问题。
- 修复提交事件乱序造成的问题。
- 遭遇 Failed to merge 时打印前后五条相关 record。
- HikariCP 连接池可编辑 param.yml 中的 connection_pool 项覆盖默认配置。

2.2.53 [1.1.16] - 2023-02-06

- 修复 legacy_format = true 时，即使用 string 序列化方式时的 json 格式问题。
- 修复获取 CLibLocalFile.size() 时文件描述符泄露问题。
- 修正需要整合的 record 跨事务文件导致整合不全的问题。
- 修复某些监控指标没有重置为 0 的问题。
- 添加 sonarqube 插件

2.2.54 [1.1.15] - 2023-01-18

- 若 Kafka Producer 发送错误 BufferExhaustedException 进入报错重启。
- 若抛出 VirtualMachineError，（例：Java heap space），程序退出。

2.2.55 [1.1.14] - 2023-01-04

- 调整日志文件大小判断逻辑：当日志文件为本地非 ASM 归档文件时，优先使用本地文件的方式读取其文件大小。
- 新增参数 `log_size_from_db` 默认值为 false，代表强制使用数据库计算得到的日志文件大小。(修改为 true 为了复现偶发日志解析不推进问题，非测试环境不要修改)

2.2.56 [1.1.13] - 2022-12-30

- 修复部分整合失败（Failed to merge）导致的数据错乱问题（含 255 列以上表）。
- 修复 AIX 配置下偶发读取无法推进问题。
- 修复 latency 指标无效问题。
- 修复触发新增子分区的第一条数据丢失问题。
- 调整 windows 环境脚本。
- 修复 KafkaProducer 占用发送完毕前未释放内存问题。
- 在 Kafka 消息中加入 xid (LONG) 字段代表事务 ID。
- 修复某些情况下更新 License 无效问题。

2.2.57 [1.1.12] - 2022-12-12

- 新增 web 页面查看状态及日志字典打包下载。
- 当进入偶发进度陷入死循环超过 `max_read_retries`（默认：5）后，内部重启，内部重启 `max_restart_before_reset`（默认：3）次后仍无法推进进度时，重置进度为当前最新。
- 新增 `validate_number_type`（默认：false）参数在明文化时检查 number 类型的 scale 部分是否合法。不合法打印错误日志及对应数据内容。

2.2.58 [1.1.11] - 2022-11-29

- 新增数据内容明文化的并发转换
- 调整连接池最大连接数为 10
- 优化事务内操作排序逻辑

2.2.59 [1.1.10] - 2022-11-20

- 调整 `conv_parallelism` 并发转换的线程数，当为 0 时，不开启并发转换。当小于 0 时，使用当前 CPU 核数。
- 删除 `cache_mb` 配置项，最大缓存文件大小保持与 `out_mb`（最大一次性输出大小）一致。
- 新增 Kafka Producer `delivery.timeout.ms` 默认值至 3600000 及 `request.timeout.ms` 默认值至 300000。防止发送超时导致丢数。
- Reset (API 调用或触发自动重置) 时，清空 dict 目录并重新获取表对应的 object id 列表。
- 新增 cli 工具包中 Kafka 相关 session 及文档。
- 新增监控指标 cmt-to-kafka-sent-lag。
- 调整 `print_cost` 的输出精度纳秒。
- Windows 运行环境下不再依赖本地 PID 文件做单例验证。
- scnTime 缺省值取上一次最大提交时间。

2.2.60 [1.1.9] - 2022-11-06

- 新增配置 `delete_trans_if_low_diskspace`。默认为 `false`。
- 新增 /export/health 接口中磁盘内存使用情况及许可证过期时间, 临近过期 7 天内会发送过期报警日志。
- 新增大事务打印阈值 `print_large_tx_threshold`，默认为 `10mb`
- `auto_reset_policy` 为 `earliest` 时 SQL 格式不正确导致的重置为当前最新进度的问题。
- 修复某种整合校验的误报问题 (Failed to merge)。

2.2.61 [1.1.8] - 2022-11-04

- 修复报错重启后不释放数据库连接问题。

2.2.62 [1.1.7] - 2022-11-02

- 修复部分数据 scn_time 为空的问题。
- 对于 Failed to merged 的数据，发送通知。
- 增加 AIX 系统裸设备读取支持。并添加文档说明。

2.2.63 [1.1.5] - 2022-10-27

- 删除不必要的日志打印
- 将 log4j.properties 与 JVM 参数配置移至 conf 文件夹，以便升级。
- 调整 JVM 调整相关文档说明
- 新增升级步骤文档

2.2.64 [1.1.4] - 2022-10-26

- 修复 ADG 模式需要某些表无法查询的问题。

2.2.65 [1.1.3] - 2022-10-26

- 修复 lob 类型乱码问题。

2.2.66 [1.1.2] - 2022-10-25

- 修复增删子分区 DDL 类型的解析逻辑。
- 新增节点 ASM 连接信息的文档说明。

2.2.67 [1.1.1] - 2022-10-23

- 修复从未有数据监听过的表 DROP 后 DDL 没有发送的错误。
- 修复创建表 DDL 没有正确发送的错误。
- 修复某些小写表没有正确监听 DDL 的错误。
- 调整 `retry_backoff_ms` 默认值从 10 变为 30。
- 调整 `start.sh` 补充 -h 选项说明，删除不必要的代码。
- 新增 ASM 远程读取相关错误处理的文档说明。
- 新增动态调整程序打印级别接口。

2.2.68 [1.1.0] - 2022-10-20

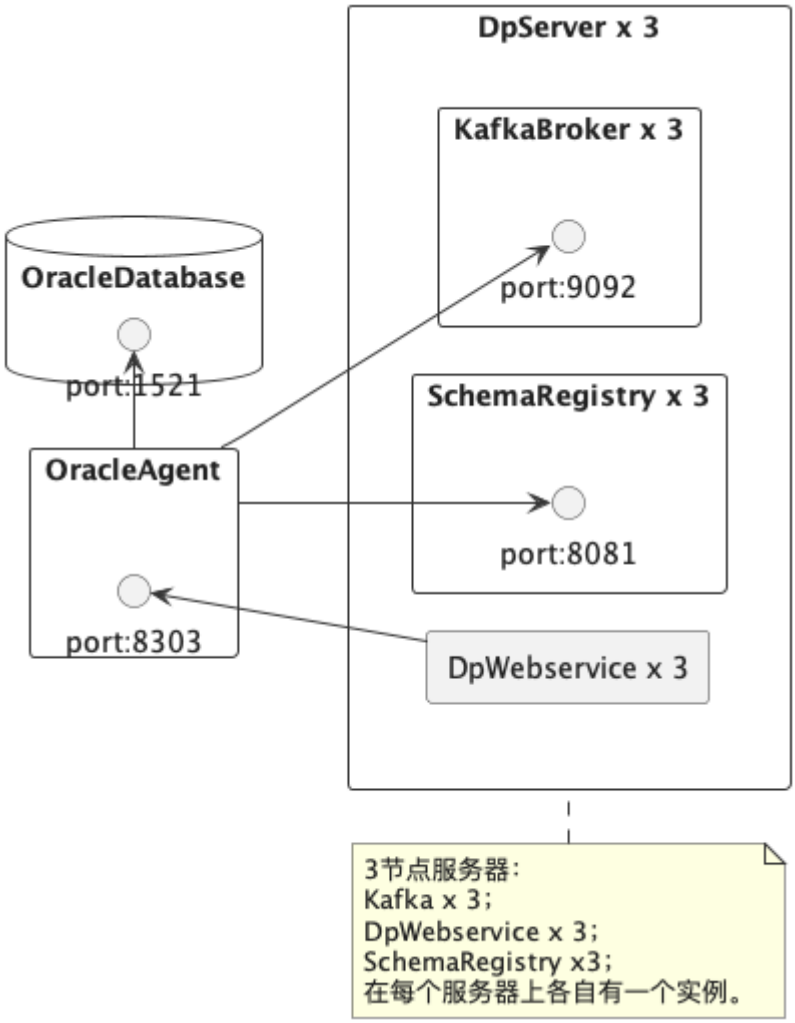
- 部分整合逻辑导致的数据为空或数据解析失败。
- 调整 API 文档，关闭某些高危 API。
- 调整本地文件登陆信息密文化。
- 新增资源控制参数。
- 新增 `jmx_prometheus_javaagent-0.17.0.jar` 于安装包中，用于开放 HTTP 接口用来暴露本地 JVM 指标。默认关闭。
- 新增事件类消息发送模块。

3. 安装

3.1 系统要求

网络要求

当使用 Kafka 作为采集代理的目的地时，需要为 Kafka producer 配置相应的连接信息。



- 须编辑部署所在服务器 /etc/hosts 做 kafka1，kafka2，kafka3 的对应域名-IP 映射。具体参照 Kafka输出配置
- 须访问 DataPipeline Kafka、Schema Registry 所在服务器的 9092（Kafka Borker 端口）和 8081 端口（Schema Registry 端口）。
- 须被 DataPipeline DpThrallWebService 所在服务器访问 8303 端口。（端口号在 param.yml 中可配置，默认为 8303）。

内存要求

采集代理程序运行在 JVM 中，系统剩余内存至少2G 以上。程序内存使用限制参考 内存限制。

磁盘要求

磁盘使用量主要组成为 1. 超过程序限定输出内存大小（param/out_mb）的事务解析后存储在本地的缓存文件，和数据库大事务行为有关。 2. 程序运行日志（固定大小，固定回收，通过配置 log4j.properties 文件指定单个大小和回收策略，默认单个文件 10M，30个文件） 与 DML 数据相关的文件在输出到 Kafka 后会自动删除回收。

程序默认当前磁盘量少于（ min_diskspace 默认 10G ）后会报错磁盘空间不足的错误，并停止解析。间隔（ retry_backoff_ms 默认 10000 ）毫秒后自动重启，重新检查是否有足够磁盘空间。

设置磁盘限制方法参考 磁盘限制。

系统设置

最大文件打开数（ ulimit -n ）建议设置为 65535。

系统用户

如果是本地部署，且安装启动采集代理的系统用户非 root 或 oracle 用户，由于程序需要对数据库本地的日志文件有读权限，该用户需加入 oracle 用户所在的用户组中。 所有采集代理 oragent 文件夹下的变更执行操作（启停，编辑配置文件）等都必须在改用户下。 注意不得以其他用户执行或修改运行目录下的文件。

```
$ useradd [用户名] -d [home目录，一般会在 /data 这种磁盘上] # 创建一个用户
$ groups oracle # 得到 oracle 用户所在用户组。可有多个。
oracle : oinstall dba pip install mkdocs-material
# 上面命令的返回结果
$ usermod -a -G oinstall,dba [用户名] # 将 【用户名】加入 -G 后面用逗号分隔的用户组中。
```

3.2 准备数据库

3.2.1 准备连接

对于有 ASM 的 Oracle 采集代理需要连接 ASM 实例。需开启 ASM 实例的远程监听。若是配置静态监听则需在 listener.ora 中有类似如下配置。

listener.ora

```
SID_LIST_LISTENER_ASM =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = ASM)
      (ORACLE_HOME = /u01/app/grid)
      (SID_NAME = +ASM1)
    )
)
```

Oracle 节点 ASM 连接信息配置

ASM 用户名

此处用户为 agentasm 且须持有 SYSDBA 角色。用户名格式为:

agentasm as sysdba

ASM 连接地址

节点 ASM 连接地址格式有两种：官方 Oracle JDBC 连接格式说明

- 使用 SERVICE: //HOST:PORT/SERVICE 例子：//10.0.10.169:1521/+ASM
- 使用 SID: HOST:PORT:SID 例子： 10.0.10.169:1521:+ASM1

如果是单机 ASM，SID 一般为 +ASM，如果是 RAC，则一般为 +ASM(x)，x 为节点数，比如 +ASM1。具体连接地址和 SID 请咨询对应 DBA。

ENDIAN 类型

优先选择大端。如果发现日志不推进，选择小端。POST /export/stop 后，在节点选择小端，并重启任务。

创建 Oracle 实例连接用户

为采集代理创建 Oracle 连接用户。

```
--******************用户相关******************
-- 创建用户名/密码
create user agent1 identified by agent1;
-- 创建角色
create role agent1_role identified by agent1_role;
-- 赋登陆、查表、查字典权限给该角色
grant connect,select any table,select any dictionary,create session to agent1_role,agent1;
-- 赋予该用户该角色
grant agent1_role to agent1;
-- 选择该用户表空间
alter user agent1 quota 2m on users;
```

创建 ASM 实例连接用户

如果 Oracle 使用了 ASM 存储。则使用 SYSASM 角色的用户登陆 ASM 数据库后创建采集代理连接 ASM 的用户。

```
-- 创建用户名/密码
create user agentasm identified by agentasm;
-- ASM 用户须有 SYSDBA 角色
grant sysdba to agentasm;
```

准备系统视图

```
create or replace view XKCCLE as select * from sys.x$KCCLE;
create or replace view XKCCCP as select * from sys.x$KCCCP;
create or replace view DP_XKSPPI as select * from X$KSPPI;
```

```
create or replace view DP_XKSPPSV as select * from X$KSPPSV;
create or replace view DP_KSPPCV as select * from X$KSPPCV;
```

如果是 PDB 环境，则须登陆或切换至该 PDB 后执行

```
create table sys.dp_pdbs as select * from sys.v$pdbs;
```

赋权用户访问系统视图

```
grant select on sys.ts$ to agent1;
grant select on sys.tabsubpart$ to agent1;
grant select on sys.obj$ to agent1;
grant select on sys.seg$ to agent1;
grant select on sys.coltype$ to agent1;
grant select on sys.xkccle to agent1;
grant select on sys.xkcccp to agent1;
grant select on sys.DP_XKSPPI to agent1;
grant select on sys.DP_XKSPPSV to agent1;
grant select on sys.DP_KSPPCV to agent1;
grant select on sys.seq$ to agent1;
grant select on sys.user$ to agent1;
grant select on sys.col$ to agent1;
grant select on sys.ind$  to agent1;
grant select on sys.tab$ to agent1;
grant select on sys.lob$ to agent1;
grant select on sys.indpart$ to agent1;
grant select on sys.lobfrag$ to agent1;
grant select on sys.user$ to agent1;
grant select on sys.source$ to agent1;
grant select on sys.view$  to agent1;
grant select on sys.ts$  to agent1;
grant select on sys.profname$ to agent1;
grant select on sys.profile$  to agent1;
grant select on sys.sysauth$ to agent1;
grant select on sys.props$ to agent1;
grant select on sys.idl_ub1$ to agent1;
grant select on sys.user$ to agent1;
grant select on sys.ntab$  to agent1;
grant select on sys.sum$ to agent1;
grant select on sys.com$ to agent1;
grant select on sys.partlob$ to agent1;
grant select on sys.attrcol$ to agent1;
grant select on sys.ccol$ to agent1;
grant select on sys.subpartcol$ to agent1;
grant select on sys.tabpart$ to agent1;
grant select on sys.indpart$ to agent1;
grant select on sys.indcompart$ to agent1;
grant select on sys.defrole$ to agent1;
grant select on sys.v_$instance to agent1;
grant select on SYS.v_$database to agent1;
grant select on sys.v_$datafile to agent1;
grant select on sys.V_$SESSION  to agent1;
grant select on sys.v_$log  to agent1;
grant select on sys.v_$logfile  to agent1;
grant select on sys.gv_$logfile  to agent1;
```

3.2.2 准备数据库日志

实时数据捕获原理是解析重做日志。 当有激增数据量时，重做日志可能还来不及解析就被切换， 您需要为 Oracle 配置开启归档模式，这样被切换的重做日志会被归档。采集代理仍可以继续解析归档日志而不丢失数据。

确认归档日志模式

1. 查询是否开启归档日志

```
SQL> select log_mode from v$database;
LOG_MODE
------------
ARCHIVELOG
```

```
SQL> archive log list;
```

2. 如果上述查询结果为否，强烈建议开启归档日志。启用归档模式具体步骤请由 DBA 操作。

确认日志补全

采集代理正常运行需要 Oracle 数据库需要开启最小补全日志。 对于 DataPipeline 数据应用来讲，需要对监听的表开启日志补全，日志补全的方式可以是数据库级别，也可以是表级别。 如果是监听 ADG 备库，则主库需开启补全日志。且主备重做日志大小应当一致。

最小级别补全日志和强制日志记录

1. 以 SQL*Plus 登陆拥有 `ALTER SYSTEM` 权限的用户。
2. 以下命令查询数据库是否开启最小补全及是否处于记录所有日志模式。

```
SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;
```

3. 如果查询结果是 `NO` ，使用以下命令开启数据库日志最小补全、开启强制日志记录。

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> ALTER DATABASE FORCE LOGGING;
```

4. 再次执行第一步查询确认结果。

```
SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;
```

5. 切换日志生效。

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

为获取 row update 事件所有列的 before 的值， 采集代理时需要为数据库开启 (ALL) 补全日志，或开启表级别 (ALL) 补全日志。

开启数据库级别日志 **ALL** 补全日志

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

开启表级别日志 **ALL** 补全日志

```
SQL> ALTER TABLE <schema name>.<table name> ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

如果只开启表级别日志补全，需要在节点中加入 JDBC 连接参数 `SUPPLEMENTAL_ALL_CHECK=false` ，跳过库 ALL 级别日志补全检查。并请为想要监听的表开启 （ALL) 表级别日志补全。否则校验会不通过。

3.3 安装步骤

3.3.1 首次安装

1. 系统准备阶段: 确保采集代理运行所在服务器符合 系统要求。
2. 数据库准备阶段: 根据 准备数据库 指导，确认数据库模式符合要求，创建数据库用户及赋权等等。
3. 配置 **JDK** 环境: 安装 JDK二进制包。
4. 安装采集代理: 解压 采集代理安装包.
5. 配置 **kafka** 输出: 配置 kafka
6. 启动: `./start.sh -w`

3.3.2 指定 ojdbc8 驱动版本

从 3.0.11 版本开始，可以指定 ojdbc8 的版本，默认使用 12.2.0.1 版本。下载页面提供了其他版本的驱动下载。将驱动解压后放置于 libs 文件夹。并使用 driver.sh 脚本进行设置。

```
## 解压后置于 libs 文件夹 下，例如：libs/21.13.0.0
./driver.sh <驱动目录路径>
```

其他版本官网下载 ojdbc8 地址 https://repo1.maven.org/maven2/com/oracle/database/jdbc/ojdbc8/

从 Oracle 12C 后，高级国际化需求需要同时配置 orai18n 路径 https://repo1.maven.org/maven2/com/oracle/database/nls/orai18n/

3.3.3 替换软件版本（升级/降级）

```
# 进入需要升级/降级的采集代理运行文件夹下
cd oragent
# 解压安装至本文件夹，忽略第一层文件夹，忽略 conf 文件夹。
tar -xf <新安装包路径> --strip-components=1 --exclude conf
```

3.3.4 安装配置远程文件读取

RFA（Remote-File-Accessor) 服务器用于实现对远程文件的访问操作，配合Oracle采集代理可实现跨网段文件访问。

1. 下载与安装

请从内部仓库下载最新版RFA安装包：

```
# 从内部仓库下载最新版RFA服务器
wget <RFA安装包下载地址>
```

2. 启动和管理

```
# 启动RFA服务器(默认端口为9081)
./rfa-server.sh start

# 指定端口启动
./rfa-server.sh 9082 start

# 查看运行状态
./rfa-server.sh status

# 重启服务
./rfa-server.sh restart

# 停止服务
./rfa-server.sh stop
```

3. 配置**Oracle**采集代理使用**RFA**

在Oracle采集代理的配置文件中，需要添加RFA服务器的地址和端口信息：

```
# RFA服务器地址和端口
param:
  rfa:
    host: 172.0.1.125
    port: 9081
```

3.3.5 本地无密登录

agent 用户须在 grid 用户所在用户组

```
# 切换至安装采集代理的用户
su - agent
# 下载 oragent 安装包 及对应 instant_client 压缩包
# 解压 oragent 安装包 后得到 oragent 目录
# 进入 oragent 文件夹
cd oragent
# 把 instant_client 的两个包放至 oragent 目录下并解压得到 instant_client_21_3 文件夹
unzip instantclient-basic-linux.x64-21.3.0.0.0.zip
unzip instantclient-sqlplus-linux.x64-21.3.0.0.0.zip
```

编辑 conf/jvm.option 在 JVM 参数中添加

```
-Djava.library.path=<instant_client_21_3完整路径>
```

编辑 ~/.bash_profile 添加一行

```
export LD_LIBRARY_PATH=<instant_client_21_3完整路径>:$LD_LIBRARY_PATH
# 使其在当前 session 生效
source ~/.bash_profile
```

在节点 ASM 连接参数里填写

```
用户名 sys as sysdba
密码 oracle(随便填)
地址 ASM_BEQ
ORACLE_HOME 填写 grid 用户下 echo $ORACLE_HOME 的值
ORACLE_SID 填写 grid 用户下 echo $ORACLE_SID 的值
```

若手工编辑 conf/oracle.yml，需要把登录信息改为 ``` asm_login: sys as sysdba/oracle@ASM_BEQ

3.4 配置输出

3.4.1 Kafka Producer 配置

采集代理解析后输出目的地为 Kafka 时，需要配置 Kafka 的连接信息。

发送的消息格式使用 Avro 格式，需要配置 Schema Registry 的地址。

支持标准 Kafka Producer 参数，详细信息请参照 Kafka 官方文档。

依赖版本为:

io.confluent:kafka-avro-serializer:5.2.0

org.apache.kafka:kafka-clients:3.1.0

param.yml 中的 kafka 配置：

```
kafka:
  # 须配置对应 /etc/hosts 域名映射
  bootstrap.servers: kafka1:9092,kafka2:9092,kafka3:9092
  # 如果 Schema Registry 和 Kafka 在同一服务器上，可以共用域名，否则直接用其所在服务器的 IP 即可
  schema.registry.url: http://kafka1:8081,http://kafka2:8081,http://kafka3:8081
  acks: 1
```

bootstrap.servers（必填）

形式为 kafka1:9092,kafka2:9092,kafka3:9092, 以逗号为分隔。

Kafka 地址必须与 DP Kafka broker 配置的 KAFKA_ADVERTISED_LISTENERS 中监听域名匹配。

举例 Kafka1 在 docker-compose 文件中的配置：KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://kafka1:9092。代表他可被 client 指向 kafka1 所访问。 如果是3节点，则必须3个节点的域名映射都配置，否则由于 Kafka 是根据 advertised.listener 来寻找 leader partition 所在的 broker，连接 broker 发送请求第一次会返回 topic leader partition 所在 broker 对应的 advertised.listeners 中的地址。如果访问不了就无法发送。

域名映射

需要在 /etc/hosts 中配置 kafka1, kafka2, kafka3 域名映射。

schema.registry.url (必填)

填写 schema registry 的地址，可填多个，逗号分隔: 假设 Schema Registry 和 Kafka1 部署在同一台机子上，那可以直接用对应的域名来替代 IP

```
schema.registry.url: http://host1:8081,http://host2:8081,http://host3:8081
```

acks (选填)

默认为 1。参照 Kafka Producer `acks` 的设置说明。

3.5 配置参数

conf/param.yml

```yaml
# -------- output ---------
# 输出通道
# channel 模式:
#   - kafka     输出到 Kafka
#   - file(.v)  输出到本地文件(是否打印列值)
#   - console   输出到屏幕（标准输出）
#   - null      无输出
channel: file

# zookeeper 地址，当需要高可用功能时必须配置
zookeeper: ""

# -------- env config ---------
# charset_form 为 1 的编码，对应数据库 NLS_CHARACTERSET
charset1: gbk

# charset_form 为 2 的编码，对应数据库 NLS_NCHAR_CHARACTERSET
charset2: utf-8

# clob 的编码
charset_clob: utf-8

# web HTTP 端口
port: 8303

# 本机 host 地址，默认取本地主机 ip 地址
host: 127.0.0.1 # 替换为实际的默认本地主机IP

offmore: false

# -------- interval ---------
# 当读到头的时候，多久 query 一次 le
change_interval_sec: 1.0

# 等到更新监听规则至生效的超时时间
map_update_timeout_sec: 300

# 发生错误后自动重启间隔
retry_backoff_ms: 30000

# -------- print ---------
# 对某个 obj no 做打印
print_objno: []

# 对某个 obj no 做打印级别
print_level: 1

# 详细打印限制
print_detail_cooldown_ms: 3000

# 打印 [exec] 的耗时
print_cost: false

# 打印 merge 的结果
print_merge: false

# 打印大 trans 的结果
print_large_tx_threshold: 10m

# -------- resource ---------
# 最小磁盘空间
min_diskspace: 10g

# 当异常退出时，是否删除缓存的事务文件
delete_trans_if_low_diskspace: false

# 一次性读取大小
read_mb: 10.0

# 一次性输出大小/最大缓存文件大小
out_mb: 10.0

# 是否开启并发转换
conv_parallelism: 0

# producer 数量
producer_num: 1

# cpu% 限制
cpu_limit: 0

# deal 限速
deal_byte_rate: 0
```

```yaml
# conv 限速
conv_byte_rate: 0

# -------- misc ----------
# 当日志被回收时自动策略:earliest/latest
auto_reset_policy: none

# 是否在 deal 阶段就过滤
filter_ahead: true

# 是否使用 DIRECT_MEMORY 当本地部署时
direct_local: false

# 使用 老 agent 创建 view
legacy_view: false

# 指定 dp_pdbs 表所在用户
pdbs_schema: sys

# converter 用 avro(false) 还是 string(true)
legacy_format: false

# 是否跳过序列化错误
skip_error: true

# 是否忽略 update 255列的列值
ignore_255_update: false

# 当异常退出时，是否保存当前 ORACLE REDO 文件
save_on_quit: false

# 跳过整合失败数据
skip_merge_fail: true

# 故意的 lag
lag_barrier: -1

# 读取本地日志文件时的读取方式
local_mode: none

# 对归档和在线日志使用同样的本地读取方式
local_mode_all: true

# 当使用 local_mode = clib 时，是否使用数据库查询得到的日志文件大小
log_size_from_db: false

aix_stat_size_off: -1

# 是否在转换时验证数字类型精度
validate_number_type: false

# 当读取进度不推进时，最多重试几次后重启
max_read_retries: 5

# 最多几次由于读取进度不推进重启时，重置进度
max_restart_before_reset: 3

log_p_max: -1

# 监视程序监视间隔
monitor_interval_sec: 30

# 是否忽略 Kafka Producer 报错
ignore_kafka_exception: false

# 恢复模式: reset | none
recovery_mode: none

# 指定获取数据库当前正在写的日志进度的 SQL
lesql: ""

# 是否匹配不带 schema 的原始 SQL 到具体监听表
match_no_schema_sql: false

# 缓存事务文件的过期时间（小时）
tx_expire_hr: 0

# 缓存的 lob 数据过期时间（分钟）
lob_expire_min: 2

# 当发现当前解析日志已被切换后，是否关闭当前文件并重新打开
close_on_switch: true

# 通知相关的配置
NotificationSpec:
  # 是否发送 DDL 消息
  send_ddl: false

  # 对不同错误类型的发送频率设置
  error_record_report_interval: {}

# 是否验证分区表的分区合法性
enable_partition_validation: true
```

```yaml
# 工作线程配置对象
WorkerSpec:
 # 最大阻塞时间
 maxBlockingSec: 1800

# 是否忽略单独的 ORP record
ignore_orp_only: false

# /config/validation 接口跳过实际校验连接信息
skip_validation: false

# 日志中是否对连接串做打码
mask_creds: true

# 找不到日志时，多久报一次错
missing_log_tolerance_times: 1

# 最大的错误文件缓存数量
err_files_max: 0

# 允许本地编辑的 login 信息不被节点信息覆盖
local_src_login: false

# 高可用容灾延迟
ha_failover_delay: 5000

# 是否开启并发解析
deal_parallelism: 0

# 是否开启并发读取
read_parallelism: 0

# read 限速
read_byte_rate: 0

# 对象池大小
max_record_pool: 10000

# 当多久进度没变化时，尝试查询数据库 open 状态
no_change_count: 10

# 开放修改在线日志的查询过滤条件
online_conditions: ""

# 开放修改归档日志的查询过滤条件
archive_conditions: ""

# 屏蔽的 rac 节点 id 列表
mute_rac: []

# 是否发送安全 scn
safe_scn: false

# 发送安全 scn 的间隔（毫秒）
safe_scn_interval_ms: 10000

# 是否优先使用本地归档文件
local_arch: false

# 是否使用 scn sub 作为缓存文件名的组成部分
tx_scn_sub: false

# 远程读取 asm 时最大的读取长度
asm_length: 30720

# 是否使用 OCI 连接 asm
asm_oci: false

# Kafka Producer 标准参数在这里进行设置。参照 Kafka 官方文档.
kafka:
 # 须配置对应 /etc/hosts 域名映射
 bootstrap.servers: kafka1:9092,kafka2:9092,kafka3:9092
 # 如果 Schema Registry 和 Kafka 在同一服务器上，可以共用域名，否则直接用其所在服务器的 IP 即可
 schema.registry.url: http://kafka1:8081,http://kafka2:8081,http://kafka3:8081
 acks: 1
```

3.6 程序日志输出配置

采集代理使用 log4j1.x 作为日志输出框架。 依赖版本为:

log4j:log4j:1.2.17

日志输出配置文件位于运行目录 conf 文件夹下:

conf/log4j.properties

```
log4j.rootLogger=info,file
# 控制台输出
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.target=System.out
log4j.appender.console.immediateFlush=true
log4j.appender.console.encoding=UTF-8
#log4j.appender.console.threshold=info
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.conversionPattern=%d{ISO8601} [%-5p] %m   [%t][%F:%L]%n
# 文件输出
log4j.appender.file=org.apache.log4j.appender.TimeAndSizeRollingAppender
# 每个文件10MB
log4j.appender.file.MaxFileSize=10MB
#log4j.appender.file.PreserveLogFileNameExtension=true
log4j.appender.file.DatePattern='.'yyyy-MM-dd
# 文件名
log4j.appender.file.file=log/export.log
# 保留 30个
log4j.appender.file.MaxRollFileCount=30
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ISO8601} [%-5p] %m   [%t][%F:%L]%n
log4j.appender.file.Encoding=utf-8
```

3.7 文件下载

3.7.1 Oracle 采集代理软件下载

安装包请咨询 **DataPipeline.Inc**

3.7.2 ojdbc8 下载

ojdbc8-21.13.0.0.tar

3.7.3 OpenJDK 下载安装

Eclipse OpenJDK 官网

Licenses:

Apache License, Version 2.0

Eclipse Distribution License 1.0 (BSD)

Eclipse Public License 2.0

— (Secondary) GNU General Public License, version 2 with OpenJDK Assembly Exception

— (Secondary) GNU General Public License, version 2 with the GNU Classpath Exception

Downloads

Linux x64 JDK 1.8

AIX PPC64 JDK 1.8

Installation

确保下载上述安装包至一个不会被移动或删除的文件夹，使用命令行 cd 进改文件夹。

LINUX

```
#解压
tar xzf <openjdk_binary>.tar.gz
#加入PATH变量
export PATH=$PWD/<解压后的文件夹>/bin:$PATH
#检查是否生效
java -version
```

持久化PATH变量，可编辑 ~/.bash_profile 添加 `export PATH=<解压文件夹绝对路径>/bin:$PATH`

AIX

```
#解压
gunzip -c <openjdk_binary>.tar.gz | tar xf -
#加入PATH变量
export PATH=$PWD/<extracted_directory>/bin:$PATH
#检查是否生效
java -version
```

4. 使用

4.1 采集量及性能评估

4.1.1 日志量与延迟关系

有多种因素影响处理速率，CPU 性能、内存大小，磁盘 IO、网络 IO、日志中数据特征、Kafka 吞吐性能等等。 一般来讲，源库日志产生速率在 20MB/s 以下，能保持在 1s - 2s 左右延迟。同时数据库流量不能太低。太低的库延迟也会稍高。有可能会达到 4s 左右。 同样 1 秒内的 20MB 原始日志文件，在 CPU、内存、IO 都相同的情况下，所包含的事务数量，DML数量，列数量，列类型等的区别都会影响处理速率。所以以实际运行评估并预测为准。

- 查询某时间段内归档日志每小时增长量

```
select trunc(COMPLETION_TIME,'HH') Hour,thread#, round(sum(BLOCKS*BLOCK_SIZE)/1024/1024) MB, count(*) Archives from (select * from v$archived_log where completion_time between
to_date('2022-10-24', 'yyyy-MM-dd') and  to_date('2022-10-26', 'yyyy-MM-dd'))  group by trunc(COMPLETION_TIME,'HH'),thread#  order by 1
```

查询每日归档日志大小

```
select * from (select trunc(completion_time) as ARC_DATE, count(*) as COUNT, round((sum(blocks * block_size) / 1024 / 1024), 2) as ARC_MB from v$archived_log group by trunc(completion_time) order by
trunc(completion_time) desc ) where rownum<=10
```

- 采集速率评估 ASM 远程读取受限较大。一般读取速率在 5 - 20 MB/s。本地读取可达 60 - 100 MB/s。 如果开启读取并发，一般来说 10 并发一下读取速率线性增长。

4.1.2 生成延迟预估图

输入每小时日志增长量(GB，逗号分隔):
4,10,20,30,100,200,10,5,1,1,1,1

平均推进速率(MB/s):

生成平滑延迟图

4.2 资源限制

4.2.1 CPU 使用率说明

采集代理在默认配置下（不开启并发）的情况下对 CPU 的使用大约为 10% - 200% 左右， 主要 CPU 时间消耗在于解析日志，转换 record 格式上。并会伴有一定间隔的 JVM 垃圾回收造成的 CPU 高峰，一般持续不到 1秒，该高峰可由 JVM 的 GC 线程数控制，见采集代理线程数控制。 在解析量很小时，即与数据库时间延迟不大时，CPU 使用较为空闲。 当落后数据库时间时，采集代理将满负荷运行以求尽快追上进度。 当开启并发处理（deal_parallelism）后，满负荷运行的 agent 可达到 1000% 或以上的 CPU 使用率，其具体值取决于运行所在服务器的总 CPU 核数。

4.2.2 内存使用说明

内存使用由 JVM 参数控制，可修改 `conf/jvm.option` 脚本中对应的 `HEAP_SIZE` 来限定内存使用量。当采集代理为本地部署时，程序会使用 2G 堆外内存来加速本地文件读取，该值也可以修改 `conf/jvm.option` 文件中参数进行调整。

conf/jvm.option

```
# JVM 堆内存大小
HEAP_SIZE="1024M"
# JVM 堆外内存大小
-XX:MaxDirectMemorySize=2G
```

4.2.3 磁盘使用量限制

采集代理会定期检查当前所在文件夹所在的磁盘剩余空间，当小于 `min_diskspace` 设定的值时，程序会不断报错并自动重启。等待磁盘空间够了，继续运行。

conf/param.yml

```
# 值为数字 + 单位(g/m/k)，默认为10g
min_diskspace: 10g
```

```
# 更新最小磁盘限制为 50g
curl -XPUT localhost:8303/param/min_diskspace/ -d '50g'
```

4.2.4 采集代理线程数控制

采集代理使用的线程数主要由 5 个主体线程（负责异步处理流程），及并发处理线程池（负责并发读取，解析，转换线程组成。并发数由 read/deal/conv parallelism 控制.

conf/param.yml

```
# 默认为值为 0，即不开启并发，-1 为 cpu 核数的并发，其他值为具体并发数。
read_parallelism: 0
deal_parallelism: 0
conv_parallelism: 0
```

同时由于采集代理运行于采用默认 ParallelGC 的 JVM 之上，有一部分线程会被 JVM 用于垃圾回收。该线程数默认与 cpu 核数相关，可通过 JVM 参数调整。

conf/jvm.option

```
# 设置并发垃圾回收线程数 = 4
-XX:ParallelGCThreads=4
```

```
# 更新 conv_parallelism 为 1
curl -XPUT localhost:8303/param/conv_parallelism/ -d '1'
```

4.2.5 采集代理日志读取速率限制

采集代理可以控制每秒读取日志的速率

conf/param.yml

```
# 默认为值为 0，即不限制
deal_byte_rate: 0
```

```
# 更新 deal_byte_rate 为 1mb/s
curl -XPUT localhost:8303/param/deal_byte_rate -d '1000000'
```

4.2.6 采集代理处理速率限制

采集代理可以控制每秒处理的数据速率（过滤非监听表事件后）。该控制也会影响Kafka Producer 实际发送的数据量。

conf/param.yml

```
# 默认为值为 0，即不限制
conv_byte_rate: 0
```

```
# 更新 conv_byte_rate 为 1mb/s
curl -XPUT localhost:8303/param/conv_byte_rate -d '1000000'
```

4.2.7 Kafka Producer 数量控制

程序会使用固定数量的 Kafka Producer 并发发送数据。数量控制参数为 `producer_num` 。提高 Producer 数量可提高总发送速率。

conf/param.yml

```
# 默认为值为 1
producer_num: 1
```

```
# 更新 producer 数量为 5
curl -XPUT localhost:8303/param/producer_num -d '5'
```

4.2.8 Kafka Producer 配额限制

通过在 DataPipeline zookeeper 容器中的 `kafka-configs.sh` 脚本，可在采集代理配置的 Kafka 集群中配置 Kafka Producer 发送速率配额来限制网络带宽。

```
# 在部署有 zk1 容器的服务器上 进入 Zookeeper 容器执行命令
docker exec -it zk1 bash
# 将节点 id (dp web url: /data-nodes/{id}) 对应的采集代理的 Kafka producer 总速率限制在 1024 bytes/s
bin/kafka-configs.sh  --zookeeper localhost:2181 --alter --add-config 'producer_byte_rate=1024' --entity-type clients --entity-name agent_5
```

4.2.9 CPU 配额限制

采集主体为 java 程序。其 CPU 使用受操作系统 cpu 分配控制。但仍可在程序内部监控 cpu 使用量 (cpu time)，测算出当前使用百分比，从而设置一定的 sleep 时间，降低 cpu 占用。

控制参数为 `cpu_limit`，值为 0 或 100 时代表不限制。

conf/param.yml

```
# 值为数字，代表百分数%
cpu_limit: 50
```

```
# 更新 cpu_limit 限制为 80%
curl -XPUT localhost:8303/param/cpu_limit -d '80'
```

生产不推荐设置的低于30%，因为系统会自动为其他高优先级的进程分配更多 cpu 时间。只有在 cpu 资源特别紧张时，发生采集代理和其他进程明显资源争抢时，再考虑配置该参数。实际推荐值需要结合系统配置及限制后性能效果做判断。

4.3 高可用

4.3.1 基于 Zookeeper 的分布式高可用模式

支持基于 Zookeeper 的分布式高可用。保证多个采集代理运行时中只有一个采集代理运行实例在工作。

要求

采集代理版本 2.0.0 及以上。对应 DataPipeline 产品版本 4.2。同时 DataPipeline 的 Zookeeper 集群开放 2181 端口供采集代理访问。

- 须指定 conf/oracle.yml 中的 src_id 来确定集群关系。
- 须指定 conf/param.yml 中的 host 来确定采集代理访问地址。
- 须指定 conf/param.yml 中的 zookeeper 来确定 Zookeeper 访问地址。

配置步骤

- 编辑 `conf/oracle.yml` 中，编辑 src_id 使其与数据节点 ID 一致。

```
src:
  id: <数据节点ID>
```

- 编辑 `conf/param.yml` 中，添加如下配置项：

```
param:
  zookeeper: <zookeeper 地址，须带有端口号，只有需要多个采集代理时需要>
  host: <webservice 能访问的 agent 地址，不要加端口号，DP 大于 4.2 必须配置>
```

4.3.2 单机 Keep-alive

通过配置 crontab 调用采集代理下的 `keep-alived.sh` 对目录下 PID 文件对应的进程进行存在判断。如果进程消失，则重启。 检查结果和重启历史记录在目录下 `restart_history` 文件中。

配置步骤

- 切换到采集代理使用的系统用户
- 编辑 crontab -e

```
# 通过 cron 定义检查周期
* * * * * <agent目录>/keep-alived.sh <agent目录>
```

4.3.3 单机 Keep-running

通过配置 crontab 调用采集代理下的 `keep-running.sh` 检查 `save/<src_id>/cfg.loginfo` 文件的修改时间。如果在规定时间内未更新，则重启采集代理。该脚本用于监控采集代理是否正常运行，因为 `cfg.loginfo` 是重启恢复的进度点，会定期更新。 检查结果和重启历史记录在目录下 `restart_history` 文件中。
注意：keep-running.sh 使用 Linux GNU `stat` 与 `date` 命令。在 BSD/macOS 上需替换为等价命令。

配置步骤

- 切换到采集代理使用的系统用户
- 编辑 crontab -e

```
# 使用默认10分钟超时检查
* * * * * <agent目录>/keep-running.sh <agent目录>

# 第二个脚本参数设定超时时间（分钟） cron 定义检查周期，建议大于等于超时时间。至少大于 1 分钟（由于重启初始化阶段可能由于大量监听表的初始化导致一开始不会推进进度）
*/15 * * * * <agent目录>/keep-running.sh <agent目录> 15
```

4.4 Agent 迁移程序

4.4.1 一、迁移程序简介

1、目录结构

文件（夹）名称	作用	备注
agent-migrate-1.0-SNAPSHOT.jar	程序 jar 包	代码修改可以直接更新此 jar 包
conf（文件夹）	存放配置文件 config.yml	
result（文件夹）	存放迁移过程中持久化的一些信息	如 config.json 、 data.json
start.sh	启动脚本	

4.4.2 二、迁移步骤

1、确认新 agent 实例状态

以新 agent 的主机 ip : 192.168.0.14 ，端口 : 8888 为例

1、执行命令查看新 agent 的健康状态

```
$ curl 192.168.0.14:8888/export/health
```

2、若显示 Connection Refused，代表新 agent 的 web 服务未开启，需在新 agent 部署路径执行

```
$ ./start.sh -w
```

3、若查看健康状态的接口返回类似这样的结果，则代表 web 已启动

```
{"status":"DOWN","checks":[...]}
```

4、若 status 的值为 "UP" ，代表新 agent 已经在解析日志了。
迁移所做的操作可能会影响其运行状态，须执行下面的命令停止

```
$ curl -XPOST 192.168.0.14:8888/export/stop
```

2、确认旧 Agent 实例状态

在 DP 节点中可查看旧 Agent 源端的状态，确认其是「运行中」状态

3、备份

备份 DP 中各个 dp-thrall-mysql 容器的数据库信息

4、解压迁移程序安装包

执行解压命令：

```
tar -xvf agent-migrate-linux-1.0-SNAPSHOT.tar
```

新建出 agent-migrate 文件夹。

5、修改配置文件

进入 agent-migrate 文件夹中的 conf 文件夹，按照示例修改配置文件 config.yml

```
# 旧 agent 的主机 ip、web 端口、部署根路径、token 信息
old:
  host: 82.157.57.48
  web_port: 8303
```

```
  path: /home/zc-agent/
  token: dVUL0Yinqodc4FjYC7Lw1LUANgN7XHtb
# 新 agent 的 srcId（填写 DP 的节点 ID ）
# 新 agent 的 主机 ip、web 端口、是否使用旧 agent 的序列化方式
new:
  src_id: 6
  host: 82.157.57.48
  web_port: 8888
  # string 代表使用旧 agent 的序列化方式，默认值为 avro
  migrate_topic_format: string
# DP的主机 ip、web 端口、kafka 地址、token 信息
dp:
  host: 81.70.62.93
  web_port: 5000
  kafka_bootstrap_servers: 192.168.0.31:9092
  token: ECswrmlZn05LQcFa/2ww8bnnICYKMZtK6MYzrqON6NmuGpyHs0vQNAkrVYH1QX+1
```

提示信息（重要！！！）

```
配置文件中的 migrate_topic_format: string 属性，在 DP 以下版本中生效。
如果不满足版本要求还将属性值设置为 string 的话，可能会有 DP 任务按照错误的反序列化方式消费的风险。
3.5版本：0.23.48及以上
3.6版本：0.24.10及以上
```

如何填写配置中的 **TOKEN** 信息

```
旧 agent：
1. 登录旧 agent 的源端
2. 打开开发者模式
3. 点击同步对象
4. 点击 Network（网络）中的 getusermap 请求
5. 查看 Request Headers（请求标头） 中 Cookie 的 token 值

DP：
1. 登录 DP
2. 打开开发者模式
3. 点击 Application（应用） 中的 Local Storage（本地存储空间）
4. 查看 DP 所在服务器的 token 值
```

6、执行迁移

在 agent-migrate 文件夹下执行 ./start.sh

默认按照 配置迁移 **->** 进度迁移 **->** 数据迁移 顺序依次执行

以下命令可指定从某个特定迁移步骤开始执行：

```
# 从 进度迁移 开始执行
./start.sh -p
# 从 数据迁移 开始执行
./start.sh -l
```

4.4.3 三、迁移步骤及回退方案

配置迁移

作用： ####

迁移同步对象信息和数据库信息

异常回滚步骤： ####

1. 迁移程序自动将新 agent 实例的 map.yml 和 oracle.yml 恢复至默认值

回退方案： ####

1. 因为不会影响到旧 agent 和 DP 的运行，无需特别处理。

备注： ####

持久化信息存储在 result 文件夹中的 config.json 文件中

old_conf ：旧 agent 实例处获取到的数据库配置信息

node_config ：新 agent 实例存储的数据库配置信息

进度迁移

作用： ####

结合旧 agent 的日志解析进度（ cfg.loginfo ）及未提交的事务列表（ translist ）设置解析进度

异常回滚步骤：

1. 迁移程序自动将新 agent 实例的解析进度恢复至默认值
2. 迁移程序重启旧 **agent** 实例

回退方案： ####

1. 迁移程序会尝试重启旧 agent 实例，但需要人工确认旧 agent 实例是否已正确重启。如果未正确重启，则需要在旧 agent 的 web 界面手动重启。

数据迁移

作用： ####

完成节点配置信息的更新及链路读取模式的修改

异常回滚步骤：

1. 迁移程序会查看是否有链路已经被修改为新 Agent 的读取模式。如果有的话，则将其读取模式恢复为旧 Agent 的读取模式
2. 迁移程序重启旧 **agent** 实例

回退方案：

1. 停止新 agent 实例
2. 迁移程序会尝试重启旧 agent 实例，但需要人工确认旧 agent 实例是否已正确重启。如果未正确重启，则需要在旧 agent 的 web 界面手动重启。
3. 如果节点关联的链路、任务较少，可人工确认并将关联链路的读取模式调整为旧 Agent 读取模式，并手动重启任务。关联的链路、任务较多的话，可以直接使用备份文件恢复数据库。

备注： ####

持久化信息存储在 result 文件夹中的 data.json 文件中

topicNames ：运行中任务所对应的 agent topic 的名称列表

nineOffsets ：旧 agent 实例向 agent topic 写入数据的 offset 列表

endOffsets ：topic 对应 endOffset 的 map

topicConfigs ：topic 对应其配置信息的 map

topicOffsets ：topic 对应 offset 的列表

4.5 API

4.5.1 Oracle Agent Web API v3.0.12

Scroll down for code samples, example requests and responses. Select a language for code samples from the tabs above or the mobile navigation menu.

管理参数配置，管理运行实例启停状态，迁移接口以及一些辅助工具

Base URLs:

- https://localhost:8303

Email: Support

4.5.2 export

控制运行实例

启动实例（可初始化**Oracle**节点信息）

`POST /export/start`

Body parameter


```
{
  "host": "10.0.10.169",
  "port": 1521,
  "database": "orcl",
  "username": "agent1",
  "password": "agent1",
  "dpToken": "9lK2YSFlp2XU/pfrdvWd4ublEKFTIM3qdAY/l47x69GawPZjH343e7mNU+86XclkE0X4uoajqOidgEEjrUIN3g==",
  "oragentConfig": {
    "uniqueId": 10,
    "mode": "RAW",
    "host": "10.0.10.169",
    "port": 1521,
    "connectionString": "10.0.10.169:1521:+ASM",
    "asmUser": "agentasm",
    "asmPassword": 123456,
    "bigEndian": false
  }
}
```


PARAMETERS

Name	In	Type	Required	Description
reset	query	string	false	启动前重置, 读取起始点设置成 latest/earliest/ timestamp 归档日志位置
time	query	string	false	起始点时间戳
body	body	OracleNodeConfig	false	Oracle节点信息（包含许可证）
host	body	string(ip)	false	地址
port	body	integer(int32)	false	端口
database	body	string	false	数据库
username	body	string	false	用户名
password	body	string	false	密码
dpToken	body	string	false	许可证
oragentConfig	body	OragentConfig	false	agent配置
» uniqueId	body	integer(int32)	false	ID

Name	In	Type	Required	Description
» mode	body	string	false	读取模式
» host	body	string(ip)	false	地址
» port	body	integer(int32)	false	端口
» connectionString	body	string	false	JDBC连接串
» asmUser	body	string	false	ASM用户
» asmPassword	body	string	false	ASM密码
» bigEndian	body	boolean	false	大小端

Enumerated Values

Parameter	Value
reset	earliest
reset	latest
reset	time
» mode	RAW
» mode	DB
» mode	ASM

RESPONSES

Status	Meaning	Description	Schema
200	OK	启动该实例	None
400	Bad Request	许可证过期	None
409	Conflict	实例已在运行中/发生错误	None
500	Internal Server Error	启动失败	None

This operation does not require authentication

停止实例

`POST /export/stop`

RESPONSES

Status	Meaning	Description	Schema
200	OK	停止成功	None
404	Not Found	无实例存在	None

This operation does not require authentication

清除运行数据

`POST /export/clean`

删除了中断点信息及残留本地缓存及程勋运行日志。必须要在实例停止的情况下运行。相当于运行 ./clean.sh.

RESPONSES

Status	Meaning	Description	Schema
200	OK	清除成功	None
409	Conflict	实例还在运行	None

This operation does not require authentication

获取Oracle节点信息

`GET /export/config`

RESPONSES

Status	Meaning	Description	Schema
200	OK	节点信息（包含许可证）	None
404	Not Found	找不到对应配置	None

This operation does not require authentication

校验配置信息

`GET /export/config/validation`

Body parameter


```
{
  "host": "10.0.10.169",
  "port": 1521,
  "database": "orcl",
  "username": "agent1",
  "password": "agent1",
  "dpToken": "9lK2YSFlp2XU/pfrdvWd4ublEKFTIM3qdAY/l47x69GawPZjH343e7mNU+86XclkE0X4uoajqOidgEEjrUIN3g==",
  "oragentConfig": {
    "uniqueId": 10,
    "mode": "RAW",
    "host": "10.0.10.169",
    "port": 1521,
    "connectionString": "10.0.10.169:1521:+ASM",
    "asmUser": "agentasm",
    "asmPassword": 123456,
    "bigEndian": false
  }
}
```


PARAMETERS

Name	In	Type	Required	Description
privsOnly	query	boolean	false	仅校验权限，不校验 Kafka 及 ASM 配置
body	body	OracleNodeConfig	true	Oracle节点信息
host	body	string(ip)	false	地址
port	body	integer(int32)	false	端口
database	body	string	false	数据库
username	body	string	false	用户名
password	body	string	false	密码
dpToken	body	string	false	许可证
oragentConfig	body	OragentConfig	false	agent配置
» uniqueId	body	integer(int32)	false	ID
» mode	body	string	false	读取模式

Name	In	Type	Required	Description
» host	body	string(ip)	false	地址
» port	body	integer(int32)	false	端口
» connectionString	body	string	false	JDBC连接串
» asmUser	body	string	false	ASM用户
» asmPassword	body	string	false	ASM密码
» bigEndian	body	boolean	false	大小端

Enumerated Values

Parameter	Value
» mode	RAW
» mode	DB
» mode	ASM

RESPONSES

Status	Meaning	Description	Schema
200	OK	校验通过	None
500	Internal Server Error	校验未通过	None

This operation does not require authentication

设置许可证

`POST /export/license`

Body parameter

```
9lK2YSFlp2XU/pfrdvWd4ublEKFTIM3qdAY/l47x69GawPZjH343e7mNU+86XcIkE0X4uoajqOidgEEjrUIN3g==
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	更新成功	None
400	Bad Request	非法许可证	None

This operation does not require authentication

实例健康状态

`GET /export/health`

Example responses

200 Response


```
{
  "status": "UP",
  "checks": [
    {
      "id": 1,
      "status": "string",
      "data": {
        "id": 1,
        "deployment-id": "58c67b73-ba1a-4a92-b229-9c42086a472a",
        "delay-sec": 2,
        "system": "string",
        "err": "string"
```

```
    }
   }
  ]
}
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	返回健康状态	Health

This operation does not require authentication

设置起始点

`POST /export/savepoint`

Body parameter

```
[
  {
    "thr": 0,
    "seq": 0,
    "cbk": 0,
    "cbo": 0,
    "lwn_block_num": 0,
    "name": "string"
  }
]
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	成功更新起点	None
400	Bad Request	更新起点失败	None

This operation does not require authentication

当前起点

`GET /export/savepoint`

Example responses

200 Response

```
"RAC:1 SEQ:10709 BLK:8616 BKO:0 LBK:8616 LSC:1183982414"
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	当前起点	string
404	Not Found	起点不存在	None

This operation does not require authentication

设置监听表

`POST /export/tables`

Body parameter

```
[
  "OWNER.TABLE"
]
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	更新成功（无变化）	None
201	Created	更新成功（有变化）	None

This operation does not require authentication

获取监听表

GET /export/tables

Example responses

200 Response

```
"map:\n  ts: 1657356113010\n  map:\n  - table: OWNER.TABLE"
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	监听表	string

This operation does not require authentication

依照给定的配置和进度信息恢复 **Agent**

POST /export/config/recover

Body parameter

```
{
  "mapConfig": {
    "map.ts": 1667269111284,
    "map.map": [
      {
        "table": "OWNER.TABLE"
      }
    ]
  },
  "srcConfig": {
    "src": {
      "id": 1,
      "login": "DP_TEST/123456@192.168.0.14:1521/orcl"
    },
    "dp_token": "8qQvIp26WZh05LcX+H8uDj5JLPCtkCAAa4apNb0dnXPqjikCA9NXwZmBF+pCAPvSyF+U6PQ+fUO1TQnGVzSKFS76ldmwPjOtOCvpXSQERb4=",
    "asm.login": "/as sysdba",
    "asm.oracle_home": "",
    "asm.oracle_sid": "",
    "asm.mode": "db",
    "asm.exadata": false,
    "asm.subAsm": []
  },
  "paramConfig": {
    "param.channel": "kafka",
    "param.kafka": {
      "bootstrap.servers": "kafkazc:9092",
      "schema.registry.url": "http://192.168.0.31:8081",
      "acks": "1"
    },
    "param.charset1": "gbk",
    "param.charset2": "utf-16",
    "param.charset_clob": "gbk",
    "param.port": 8998,
    "param.offmore": false,
    "param.change_interval_sec": 1,
    "param.map_update_timeout_sec": 300,
    "param.retry_backoff_ms": 10000,
    "param.print_objno": [],
    "param.print_level": 1,
    "param.print_detail_cooldown_ms": 3000,
    "param.print_cost": false,
    "param.print_merge": false,
    "param.print_large_tx_threshold": "10m",
    "param.min_diskspace": "10g",
    "param.delete_trans_if_low_diskspace": false,
    "param.out_mb": 20,
```

```
    "param.conv_parallelism": 0,
    "param.producer_num": 1,
    "param.cpu_limit": 0,
    "param.deal_byte_rate": 0,
    "param.conv_byte_rate": 0,
    "param.supplemental_all_only": false,
    "param.auto_reset_policy": "none",
    "param.filter_ahead": true,
    "param.direct_local": true,
    "param.legacy_view": false,
    "param.legacy_format": false,
    "param.skip_error": true,
    "param.ignore_255_update": false,
    "param.save_on_quit": false,
    "param.skip_merge_fail": true,
    "param.lag_barrier": -1,
    "param.local_mode": "none",
    "param.local_mode_all": true,
    "param.os": "linux",
    "param.lx": false,
    "param.aix_stat_size_off": -1,
    "param.record_head_len": -1
  },
  "safeScn": 1265702183
}
```


PARAMETERS

Name	In	Type	Required	Description
body	body	RecoverConfig	true	none
mapConfig	body	object	false	none
srcConfig	body	object	false	none
paramConfig	body	object	false	none
safeScn	body	integer(int64)	false	none

RESPONSES

Status	Meaning	Description	Schema
200	OK	成功地恢复配置及进度，并重启 Agent 实例	None
500	Internal Server Error	恢复时遇到异常或未成功重启 Agent 实例	None

This operation does not require authentication

4.5.3 param

控制运行参数

获取单个参数值

`GET /param/{key}`

PARAMETERS

Name	In	Type	Required	Description
key	path	string	true	参数名

Example responses

200 Response


```
50
```


RESPONSES

Status	Meaning	Description	Schema
200	OK	值	string

Status	Meaning	Description	Schema
404	Not Found	不存在	None

This operation does not require authentication

更新单个参数值

`PUT /param/{key}`

Body parameter

```
50
```

PARAMETERS

Name	In	Type	Required	Description
key	path	string	true	参数名

Example responses

200 Response

```
"string"
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	更新成功	string
400	Bad Request	非法参数或值	None

This operation does not require authentication

获取(某类)参数

`GET /params/{type}`

PARAMETERS

Name	In	Type	Required	Description
type	path	string	true	参数类型

Enumerated Values

Parameter	Value
type	data-output
type	logging
type	environment
type	resource-limit
type	error-handling
type	performance
type	others

Example responses

200 Response

```
"{\n  \"cpu_limit\": 50,\n  \"memory_limit\": 50\n}"
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	所有参数 properties 格式	string

This operation does not require authentication

更新(某类)参数

`PUT /params/{type}`

Body parameter

```
|-
{
  "cpu_limit": 50,
  "memory_limit": 50
}
```

PARAMETERS

Name	In	Type	Required	Description
type	path	string	true	参数类型

Enumerated Values

Parameter	Value
type	data-output
type	logging
type	environment
type	resource-limit
type	error-handling
type	performance
type	others

Example responses

200 Response

```
"string"
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	更新成功	string
400	Bad Request	非法参数或值	None

This operation does not require authentication

4.5.4 tools

工具 API

重启运行时进程(JVM)

`POST /tools/restart`

RESPONSES

Status	Meaning	Description	Schema
200	OK	执行重启	None

This operation does not require authentication

查看 log/export.log 日志

`GET /tools/view/log`

PARAMETERS

Name	In	Type	Required	Description
head	query	integer	false	头几行
tail	query	integer	false	尾几行
keyword	query	string	false	关键字，支持 regex

Example responses

200 Response

"[{\"thr\":1,\"seq\":10728,\"name\":\"/u01/app/oracle/oradata/orcl/redo02.log\",\"resetlog_no\":1408558}]"

RESPONSES

Status	Meaning	Description	Schema
200	OK	startScn 对应所在 Redo logs.	string

This operation does not require authentication

升级 agent

`POST /tools/upgrade/{secret}`

Body parameter

```
file: string
```

PARAMETERS

Name	In	Type	Required	Description
secret	path	string	true	升级密钥
file	body	string(binary)	true	none

RESPONSES

Status	Meaning	Description	Schema
200	OK	升级成功	None
403	Forbidden	非法密钥	None
500	Internal Server Error	升级失败	None

This operation does not require authentication

回滚 agent

`POST /tools/rollback/{secret}`

PARAMETERS

Name	In	Type	Required	Description
secret	path	string	true	回滚密钥

RESPONSES

Status	Meaning	Description	Schema
200	OK	回滚成功	None
403	Forbidden	非法密钥	None
500	Internal Server Error	回滚失败	None

This operation does not require authentication

调整 root logger 打印级别

`PUT /tools/loglevel/{level}`

PARAMETERS

Name	In	Type	Required	Description
level	path	string	true	打印级别

Enumerated Values

Parameter	Value
level	INFO
level	ERROR
level	TRACE
level	DEBUG
level	WARN

RESPONSES

Status	Meaning	Description	Schema
200	OK	设置成功	None

This operation does not require authentication

获取 scn

`GET /tools/scn`

如果提供 time 参数，则返回该事件对应的 scn，否则返回当前 scn

PARAMETERS

Name	In	Type	Required	Description
time	query	string	false	该时间对应的scn

Example responses

200 Response

```
1186387875
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	scn	integer

This operation does not require authentication

获取该 **scn** 对应所在 **Redo logs.**

`GET /tools/logs`

PARAMETERS

Name	In	Type	Required	Description
startScn	query	integer	true	scn

Example responses

200 Response

"[{\"thr\":1,\"seq\":10728,\"name\":\"/u01/app/oracle/oradata/orcl/redo02.log\",\"resetlog_no\":1408558}]"

RESPONSES

Status	Meaning	Description	Schema
200	OK	startScn 对应所在 Redo logs.	string

This operation does not require authentication

打包下载 **log**（日志）或者 **dict**（数据字典）文件

`POST /tools/download/{type}`

Body parameter


```
{
  "wild": "*export.log.2022-10*",
  "start": "2022-10-10T10:10:10.000Z",
  "end": "2022-10-10T10:10:10.000Z",
  "tar": true
}
```


PARAMETERS

Name	In	Type	Required	Description
type	path	string	true	文件类型，只能是 log 或者 dict，代表日志和数据字典文件
body	body	FindFileRequest	false	文件查找条件
wild	body	string	false	文件名通配符，必须包含 .log 或者 .json
start	body	string(date-time)	false	最小文件修改时间(ISO8601)，默认为最小值
end	body	string(date-time)	false	最大文件修改时间(ISO8601), 默认为最大值
tar	body	boolean	false	是否打包，否的话只传第一个文件

Enumerated Values

Parameter	Value
type	log
type	dict

Example responses

200 Response

RESPONSES

Status	Meaning	Description	Schema
200	OK	文件	string

This operation does not require authentication

4.5.5 migrate

迁移时获取旧 agent 的配置和进度等信息

根据给定的旧 **agent** 部署路径获取配置信息

`GET /migrate/legacy/config`

PARAMETERS

Name	In	Type	Required	Description
path	query	string	true	旧 agent 部署路径

Example responses

200 Response


```
{
  "src_id": "1",
  "src_login": "DP_TEST/123456@dn-oracle-source:1521/ORCL",
  "param_exadata": "0",
  "asm_login": "agentasm as sysdba/agentasm@//10.10.10.20:1521/+ASM",
  "asm_oracle_sid": "+ASM1",
  "asm_oracle_home": "/u01/app/19.0/grid",
  "asm_mode": "RAW",
  "asm_device_id": "1 , 2 , 3",
  "asm_disk": "/dev/asm-diskc , /dev/asm-diskd , /dev/asm-diske",
  "asm_dev": "/dev/sdb1 , /dev/sdb2 , /dev/sdb3"
}
```


RESPONSES

Status	Meaning	Description	Schema
200	OK	旧 agent 配置信息	LegacyConfig

This operation does not require authentication

根据给定的旧 **agent** 部署路径获取 **Savepoint**

`GET /migrate/legacy/{srcId}/savepoint`

PARAMETERS

Name	In	Type	Required	Description
srcId	path	integer	true	none
path	query	string	true	旧 agent 部署路径

Example responses

200 Response


```
[
  {
    "thr": 0,
    "seq": 0,
```

```
    "cbk": 0,
    "cbo": 0,
    "lwn_block_num": 0
  }
]
```


RESPONSES

Status	Meaning	Description	Schema
200	OK	旧 agent 的 Savepoint	Inline

RESPONSE SCHEMA

Status Code **200**

各个 *Thread* 对应的 *Savepoint*

Name	Type	Required	Restrictions	Description
anonymous	[Savepoint]	false	none	各个 Thread 对应的 Savepoint
» thr	integer(int32)	false	none	线程 ID
» seq	integer(int64)	false	none	日志文件号
» cbk	integer(int64)	false	none	当前块号
» cbo	integer	false	none	当前块偏移量
» lwn_block_num	integer(int64)	false	none	下一个 lwn 的开始位置

This operation does not require authentication

根据给定的旧 **agent** 部署路径获取事务列表

`GET /migrate/legacy/{srcId}/translist`

PARAMETERS

Name	In	Type	Required	Description
srcId	path	integer	true	none
path	query	string	true	旧 agent 部署路径

Example responses

200 Response


```
[
  {
    "trans_id": 0,
    "scn_bgn0": 0,
    "scn_bgn": {
      "scn": 0,
      "scn_sub": 0,
      "len": 10
    },
    "scn": {
      "scn": 0,
      "scn_sub": 0,
      "len": 10
    },
    "scn_max": {
      "scn": 0,
      "scn_sub": 0,
      "len": 10
    },
    "scn_time_bgn": 0,
    "scn_time": {
      "value": 0
    },
    "scn_time_max": 0,
    "thread_id": 0,
    "uba": 0,
    "dba": 0,
    "slt": 0,
```

```
    "rba": 0,
    "is_ddl": 0,
    "is_cmted": true,
    "changed": 0,
    "cached_len": 0,
    "total_len": 0,
    "flag": 0,
    "cur_fno": 0,
    "max_fno": 0
  }
]
```


RESPONSES

Status	Meaning	Description	Schema
200	OK	旧 agent 的事务列表	Inline

RESPONSE SCHEMA

Status Code **200**

事务信息

Name	Type	Required	Restrictions	Description
anonymous	[Trans]	false	none	事务信息
» trans_id	integer(int64)	false	none	事务 ID
» scn_bgn0	integer(int64)	false	none	起始 SCN
» scn_bgn	Scn	false	none	none
»» scn	integer(int64)	false	none	none
»» scn_sub	integer(short)	false	none	none
»» len	integer(int32)	false	none	none
» scn	Scn	false	none	none
» scn_max	Scn	false	none	none
» scn_time_bgn	integer(int64)	false	none	起始 SCN 时间
» scn_time	object	false	none	none
»» value	integer(int64)	false	none	SCN 时间值
» scn_time_max	integer(int64)	false	none	最大 SCN 时间
» thread_id	integer(int32)	false	none	线程 ID
» uba	integer(int64)	false	none	Undo 块地址
» dba	integer(int64)	false	none	数据块地址
» slt	integer(int32)	false	none	none
» rba	integer(int64)	false	none	Redo 块地址
» is_ddl	integer(int32)	false	none	是否是 DDL
» is_cmted	boolean	false	none	是否 commit
» changed	integer(int64)	false	none	none
» cached_len	integer(int64)	false	none	缓存长度
» total_len	integer(int64)	false	none	总长度
» flag	integer(int32)	false	none	标志位
» cur_fno	integer(int32)	false	none	当前缓存文件序号
» max_fno	integer(int32)	false	none	最大缓存文件序号

This operation does not require authentication

根据给定的旧 **agent** 部署路径获取事务的最早 **SCN** 值

`GET /migrate/legacy/{srcId}/translist/scn`

PARAMETERS

Name	In	Type	Required	Description
srcId	path	integer	true	none
path	query	string	true	旧 agent 部署路径

Example responses

200 Response

```
[
  0
]
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	旧 agent 最早 SCN 值的列表	Inline

RESPONSE SCHEMA

Status Code **200**

最早 *SCN* 值

Name	Type	Required	Restrictions	Description

This operation does not require authentication

4.5.6 admin

管理员命令

获取 **log** 文件列表

`GET /admin/logs`

Example responses

200 Response

```
[
  "export.log.2022-10-10"
]
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	文件列表	Inline

RESPONSE SCHEMA

This operation does not require authentication

选择多个文件名并打包下载

`POST /admin/logs/download`

Body parameter

```
[
  "export.log.2022-10-10"
]
```

Example responses

200 Response

RESPONSES

Status	Meaning	Description	Schema
200	OK	日志打包文件	string

This operation does not require authentication

获取 log 文件内容

GET /admin/logs/{name}

PARAMETERS

Name	In	Type	Required	Description
name	path	string	true	文件名

Example responses

200 Response

RESPONSES

Status	Meaning	Description	Schema
200	OK	文件	string

This operation does not require authentication

4.5.7 metrics

获取agent的metric

GET /metrics

PARAMETERS

Name	In	Type	Required	Description
format	query	string	false	metrics的格式

Example responses

200 Response

```
[
  {
    "name": "string",
    "help": "string",
    "type": "string",
    "unit": "string",
    "samples": [
      {
        "lables": [
          {
            "label": "string",
            "value": "string"
          }
        ],
        "value": "string"
      }
```

```
    ]
  }
]
```

RESPONSES

Status	Meaning	Description	Schema
200	OK	agent metrics	Inline

RESPONSE SCHEMA

Status Code **200**

agent metric

Name	Type	Required	Restrictions	Description
anonymous	[Metric]	false	none	agent metric
» name	string	false	none	name
» help	string	false	none	none
» type	string	false	none	none
» unit	string	false	none	none
» samples	[any]	false	none	none
»» lables	[any]	false	none	none
»»» label	string	false	none	none
»»» value	string	false	none	none
»» value	string	false	none	none

This operation does not require authentication

4.5.8 Schemas

OragentConfig

```
{
  "uniqueId": 10,
  "mode": "RAW",
  "host": "10.0.10.169",
  "port": 1521,
  "connectionString": "10.0.10.169:1521:+ASM",
  "asmUser": "agentasm",
  "asmPassword": 123456,
  "bigEndian": false
}
```

agent配置

PROPERTIES

Name	Type	Required	Restrictions	Description
uniqueId	integer(int32)	false	none	ID
mode	string	false	none	读取模式
host	string(ip)	false	none	地址
port	integer(int32)	false	none	端口
connectionString	string	false	none	JDBC连接串
asmUser	string	false	none	ASM用户
asmPassword	string	false	none	ASM密码

Name	Type	Required	Restrictions	Description
bigEndian	boolean	false	none	大小端

Enumerated Values

Property	Value
mode	RAW
mode	DB
mode	ASM

OracleNodeConfig


```
{
 "host": "10.0.10.169",
 "port": 1521,
 "database": "orcl",
 "username": "agent1",
 "password": "agent1",
 "dpToken": "9lK2YSFlp2XU/pfrdvWd4ublEKFTIM3qdAY/l47x69GawPZjH343e7mNU+86XclkE0X4uoajqOidgEEjrUlN3g==",
 "oragentConfig": {
  "uniqueId": 10,
  "mode": "RAW",
  "host": "10.0.10.169",
  "port": 1521,
  "connectionString": "10.0.10.169:1521:+ASM",
  "asmUser": "agentasm",
  "asmPassword": 123456,
  "bigEndian": false
 }
}
```


Oracle节点配置

PROPERTIES

Name	Type	Required	Restrictions	Description
host	string(ip)	false	none	地址
port	integer(int32)	false	none	端口
database	string	false	none	数据库
username	string	false	none	用户名
password	string	false	none	密码
dpToken	string	false	none	许可证
oragentConfig	OragentConfig	false	none	agent配置

RedoLog


```
{
 "thr": 0,
 "seq": 0,
 "cbk": 0,
 "cbo": 0,
 "lwn_block_num": 0,
 "name": "string"
}
```


重做日志

PROPERTIES

Name	Type	Required	Restrictions	Description
thr	integer	false	none	线程号
seq	integer(int64)	false	none	序列号
cbk	integer(int64)	false	none	当前块号

Name	Type	Required	Restrictions	Description
cbo	integer	false	none	当前块偏移量
lwn_block_num	integer(int64)	false	none	下一个 lwn 的开始位置
name	string	false	none	日志名

Data


```
{
 "id": 1,
 "deployment-id": "58c67b73-ba1a-4a92-b229-9c42086a472a",
 "delay-sec": 2,
 "system": "string",
 "err": "string"
}
```


PROPERTIES

Name	Type	Required	Restrictions	Description
id	string	false	none	ID
deployment-id	string	false	none	部署ID
delay-sec	string	false	none	延迟（秒）
system	string	false	none	系统检查项
err	string	false	none	错误

Check


```
{
 "id": 1,
 "status": "string",
 "data": {
  "id": 1,
  "deployment-id": "58c67b73-ba1a-4a92-b229-9c42086a472a",
  "delay-sec": 2,
  "system": "string",
  "err": "string"
 }
}
```


PROPERTIES

Name	Type	Required	Restrictions	Description
id	string	false	none	检查项
status	string	false	none	状态
data	Data	false	none	none

Health


```
{
 "status": "UP",
 "checks": [
  {
   "id": 1,
   "status": "string",
   "data": {
    "id": 1,
    "deployment-id": "58c67b73-ba1a-4a92-b229-9c42086a472a",
    "delay-sec": 2,
    "system": "string",
    "err": "string"
   }
  }
 ]
}
```

PROPERTIES

Name	Type	Required	Restrictions	Description
status	string	false	none	状态
checks	[Check]	false	none	所有检查项

Enumerated Values

Property	Value
status	UP
status	DOWN

FindFileRequest


```
{
  "wild": "*export.log.2022-10*",
  "start": "2022-10-10T10:10:10.000Z",
  "end": "2022-10-10T10:10:10.000Z",
  "tar": true
}
```


PROPERTIES

Name	Type	Required	Restrictions	Description
wild	string	false	none	文件名通配符，必须包含 .log 或者 .json
start	string(date-time)	false	none	最小文件修改时间(ISO8601)，默认为最小值
end	string(date-time)	false	none	最大文件修改时间(ISO8601), 默认为最大值
tar	boolean	false	none	是否打包，否的话只传第一个文件

ApiResponse


```
{
  "code": 0,
  "type": "string",
  "message": "string"
}
```


PROPERTIES

Name	Type	Required	Restrictions	Description
code	integer(int32)	false	none	none
type	string	false	none	none
message	string	false	none	none

LegacyConfig


```
{
  "src_id": "1",
  "src_login": "DP_TEST/123456@dn-oracle-source:1521/ORCL",
  "param_exadata": "0",
  "asm_login": "agentasm as sysdba/agentasm@//10.10.10.20:1521/+ASM",
  "asm_oracle_sid": "+ASM1",
  "asm_oracle_home": "/u01/app/19.0/grid",
  "asm_mode": "RAW",
  "asm_device_id": "1 , 2 , 3",
  "asm_disk": "/dev/asm-diskc , /dev/asm-diskd , /dev/asm-diske",
  "asm_dev": "/dev/sdb1 , /dev/sdb2 , /dev/sdb3"
}
```

PROPERTIES

Name	Type	Required	Restrictions	Description
src_id	string	false	none	none
src_login	string	false	none	数据库登录信息
param_exadata	string	false	none	读取方式，小端为 0 ，大端为 1
asm_login	string	false	none	ASM 登录信息
asm_oracle_sid	string	false	none	ASM 对应 SID
asm_oracle_home	string	false	none	ASM 对应 ORACLE_HOME
asm_mode	string	false	none	读取模式
asm_device_id	string	false	none	ASM 设备号
asm_disk	string	false	none	ASM 磁盘路径
asm_dev	string	false	none	ASM 磁盘信息

Enumerated Values

Property	Value
param_exadata	0
param_exadata	1
asm_mode	RAW
asm_mode	DB
asm_mode	ASM

Savepoint

```
{
  "thr": 0,
  "seq": 0,
  "cbk": 0,
  "cbo": 0,
  "lwn_block_num": 0
}
```

PROPERTIES

Name	Type	Required	Restrictions	Description
thr	integer(int32)	false	none	线程 ID
seq	integer(int64)	false	none	日志文件号
cbk	integer(int64)	false	none	当前块号
cbo	integer	false	none	当前块偏移量
lwn_block_num	integer(int64)	false	none	下一个 lwn 的开始位置

Trans

```
{
  "trans_id": 0,
  "scn_bgn0": 0,
  "scn_bgn": {
    "scn": 0,
    "scn_sub": 0,
    "len": 10
  },
  "scn": {
    "scn": 0,
    "scn_sub": 0,
    "len": 10
  },
```

```
  "scn_max": {
    "scn": 0,
    "scn_sub": 0,
    "len": 10
  },
  "scn_time_bgn": 0,
  "scn_time": {
    "value": 0
  },
  "scn_time_max": 0,
  "thread_id": 0,
  "uba": 0,
  "dba": 0,
  "slt": 0,
  "rba": 0,
  "is_ddl": 0,
  "is_cmted": true,
  "changed": 0,
  "cached_len": 0,
  "total_len": 0,
  "flag": 0,
  "cur_fno": 0,
  "max_fno": 0
}
```


PROPERTIES

Name	Type	Required	Restrictions	Description
trans_id	integer(int64)	false	none	事务 ID
scn_bgn0	integer(int64)	false	none	起始 SCN
scn_bgn	Scn	false	none	none
scn	Scn	false	none	none
scn_max	Scn	false	none	none
scn_time_bgn	integer(int64)	false	none	起始 SCN 时间
scn_time	object	false	none	none
» value	integer(int64)	false	none	SCN 时间值
scn_time_max	integer(int64)	false	none	最大 SCN 时间
thread_id	integer(int32)	false	none	线程 ID
uba	integer(int64)	false	none	Undo 块地址
dba	integer(int64)	false	none	数据块地址
slt	integer(int32)	false	none	none
rba	integer(int64)	false	none	Redo 块地址
is_ddl	integer(int32)	false	none	是否是 DDL
is_cmted	boolean	false	none	是否 commit
changed	integer(int64)	false	none	none
cached_len	integer(int64)	false	none	缓存长度
total_len	integer(int64)	false	none	总长度
flag	integer(int32)	false	none	标志位
cur_fno	integer(int32)	false	none	当前缓存文件序号
max_fno	integer(int32)	false	none	最大缓存文件序号

Scn


```
{
  "scn": 0,
  "scn_sub": 0,
  "len": 10
}
```

PROPERTIES

Name	Type	Required	Restrictions	Description
scn	integer(int64)	false	none	none
scn_sub	integer(short)	false	none	none
len	integer(int32)	false	none	none

RecoverConfig


```
{
  "mapConfig": {
    "map.ts": 1667269111284,
    "map.map": [
      {
        "table": "OWNER.TABLE"
      }
    ]
  },
  "srcConfig": {
    "src": {
      "id": 1,
      "login": "DP_TEST/123456@192.168.0.14:1521/orcl"
    },
    "dp_token": "8qQvIp26WZh05LcX+H8uDj5JLPCtkCAAa4apNb0dnXPqjikCA9NXwZmBF+pCAPvSyF+U6PQ+fUO1TQnGVzSKFS76ldmwPjOtOCvpXSQERb4=",
    "asm.login": "/as sysdba",
    "asm.oracle_home": "",
    "asm.oracle_sid": "",
    "asm.mode": "db",
    "asm.exadata": false,
    "asm.subAsm": []
  },
  "paramConfig": {
    "param.channel": "kafka",
    "param.kafka": {
      "bootstrap.servers": "kafkazc:9092",
      "schema.registry.url": "http://192.168.0.31:8081",
      "acks": "1"
    },
    "param.charset1": "gbk",
    "param.charset2": "utf-16",
    "param.charset_clob": "gbk",
    "param.port": 8998,
    "param.offmore": false,
    "param.change_interval_sec": 1,
    "param.map_update_timeout_sec": 300,
    "param.retry_backoff_ms": 10000,
    "param.print_objno": [],
    "param.print_level": 1,
    "param.print_detail_cooldown_ms": 3000,
    "param.print_cost": false,
    "param.print_merge": false,
    "param.print_large_tx_threshold": "10m",
    "param.min_diskspace": "10g",
    "param.delete_trans_if_low_diskspace": false,
    "param.out_mb": 20,
    "param.conv_parallelism": 0,
    "param.producer_num": 1,
    "param.cpu_limit": 0,
    "param.deal_byte_rate": 0,
    "param.conv_byte_rate": 0,
    "param.supplemental_all_only": false,
    "param.auto_reset_policy": "none",
    "param.filter_ahead": true,
    "param.direct_local": true,
    "param.legacy_view": false,
    "param.legacy_format": false,
    "param.skip_error": true,
    "param.ignore_255_update": false,
    "param.save_on_quit": false,
    "param.skip_merge_fail": true,
    "param.lag_barrier": -1,
    "param.local_mode": "none",
    "param.local_mode_all": true,
    "param.os": "linux",
    "param.lx": false,
    "param.aix_stat_size_off": -1,
    "param.record_head_len": -1
  },
  "safeScn": 1265702183
}
```

PROPERTIES

Name	Type	Required	Restrictions	Description
mapConfig	object	false	none	none
srcConfig	object	false	none	none
paramConfig	object	false	none	none
safeScn	integer(int64)	false	none	none

Metric


```
{
 "name": "string",
 "help": "string",
 "type": "string",
 "unit": "string",
 "samples": [
  {
   "lables": [
    {
     "label": "string",
     "value": "string"
    }
   ],
   "value": "string"
  }
 ]
}
```


PROPERTIES

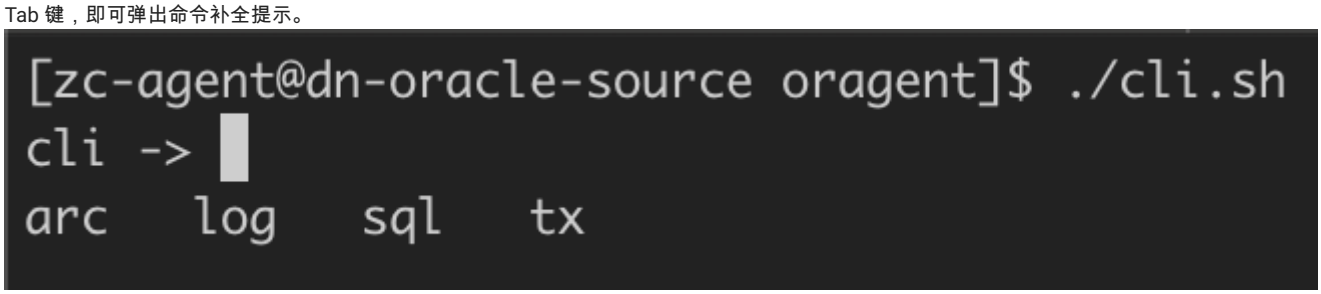
Name	Type	Required	Restrictions	Description
name	string	false	none	name
help	string	false	none	none
type	string	false	none	none
unit	string	false	none	none
samples	[any]	false	none	none
» lables	[any]	false	none	none
»» label	string	false	none	none
»» value	string	false	none	none
» value	string	false	none	none

4.5.9 swagger-ui

4.6 命令行工具

4.6.1 CLI 工具

1. 启动 `./cli.sh`
2. Tab 键，即可弹出命令补全提示。



3. 进入 session 输入 session 名称后，点击回车进入。
   选中命令，输入 - 或 -- 后点击 Tab 键，可显示参数补全提示。



4. 退出 session 输入 exit，点击回车，退出当前 session。

Session

ARC

归档文件相关。

列出归档文件

```
cli -> arc
arc -> ls -n 100 # 列出最新100个归档文件的 sequence 和 thread
```

下载归档文件

```
cli -> arc
arc -> cp --thr=1 --seq=680 # 下载thread = 1，sequence = 680 的归档文件至本地
```

目录打包

会生成【日期.tar】包含运行目录以及对应的数据库日志文件。

```
cli -> arc
arc -> tar
download current redo logs
```

本地离线/在解析日志片段

可以从指定位置开始解析离线文件或在线数据库日志。 命令相关选项参照 ps -?，会在本地路径下生成 cli.log 日志文件，从中查看 TRACE 级别日志。

```
cli -> arc
arc -> ps
Deal offline redo log segments and print to file.
```

KAFKA

Kafka 相关

列出 **topic**

```
cli -> kafka
kafka -> topic -t ABC # 列出所有包含 ABC 的 topic
```

查询 **topic** 内容

根据时间范围，过滤条件来查看 agent topic 中的数据内容。命令格式为: find -t `TOPIC` -f `FILTER` -r `START-TIME` / `END-TIME`

FILTER 格式

格式为 key=value 格式，可指定字段匹配。
如果没有 key= 前缀，则直接匹配所有内容。
支持多个 -f 选项，代表都须符合所有过滤条件。

目前支持字段为:
rid=<ROWID>
op=<OP_CODE>
offset=<KAFKA_OFFSET>
<字段名>=<字段值包含>

样例:
-f rid=AAAXXBBBDCCED
-f ID===123456
-f 2022-10-10

时间格式为 `YYYY-MM-DDTHH:mm:ss`

Example

kafka -> find -t dp_agent_1_AGENT1_SHAO -f value=haha -r 2022-02-02T02:02:02/2022-02-02T03:02:02

SQL

查询数据库

执行**SQL**查询

```
cli -> sql
sql -> query select * from shao
                    ID
                 19990
                 10000
                     1
                 20000
                 21002
                     1
                     1
                     1
                     1
                     1
                     1
                     1
Executed SQL: select * from shao
sql ->
```

输入 query <SQL>

TRANS

本地缓存事务文件相关。

打印 **trans**

```
cli -> tx
tx -> print -f <trans 文件名称>
```

LOG

日志相关

下载日志

```
cli -> log
# 获取 9月1日 至 9月9日 的 export 日志，存储至本地的 sep.tar 文件中
log -> get -s 2022-09-01 -e 2022-09-09 -n sep -t export
```

```
cli -> log
log -> get -s 2022-09-01 -e 2022-09-09 -n sep -t export
2022-09-09 08:04:01,786 [INFO ]

        *
        *  License expire at 2029-07-14T17:08:33.610.
        *

       [main][Export.kt:112]
2022-09-09 08:04:01,826 [INFO ] find [export] log from 2022-09-01 to 2022-09-09   [main][ControlVerticle.kt:337]
2022-09-09 08:04:01,827 [INFO ] tar command : tar -cf sep.tar ./export.log   [main][ControlVerticle.kt:365]
tar export log files between 2022-09-01 and 2022-09-09 to log/sep.tar
log ->
```

4.6.2 常用命令

通过 `le.sh` 查询并查看 Redo 文件头。

```
hexdump -e '4/4 "%04d" ","' -s $((`./le.sh | grep 4( 294967295 | awk '{print $3}'` * 512 + 0x30)) /)u01/app/oracle/oradata/DP/redo03.log -n 4
```

4.7 运维场景

4.7.1 通用故障处理流程

当我们在讨论采集代理无法工作的场景时，一般是指采集进度无法推进，或者推进得非常慢。这种情况下，我们需要对采集代理进行诊断，找出问题的原因，然后进行修复。 诊断需要一些关键信息，从运维的角度可以在恢复前通过一定的步骤将关键信息保留，再恢复后进行事后分析。

判断故障类型

- 如果是已知错误，可以参照常见错误的进行处理。
- 如果是未知错误，比如一直报类似 IndexOutOfBound，NullPointerException 之类的致命错误并不断自动重启无法恢复的情况，此种一般由于解析问题，或者事务文件损坏，导致无法恢复，需要保留现场后重置进度并恢复运行。

保留现场

当我们想在事后还原现场时，需要保留现场的一些要素，大致分类为三种：

1. 外部环境：包括系统负载，网络情况，数据库状态，当前正在采集的数据库日志对象等。
2. 内部环境：包括运行目录下配置文件，产生的临时文件，
3. 运行历史：包括了程序运行的历史日志，历史指标等。

基本要素包括：

- log 目录下的日志文件
- conf 目录的配置文件
- dict 目录下的数据字典文件
- trans 目录下的缓存事务
- arch 目录下的归档文件
- 当前 JVM 堆栈信息
- Java heapspace 等 OOM 报错后产生的 hprof 文件
- JVM 无故崩溃后的 hs_err.log 文件

收集程序堆栈快照：

使用 jstack 命令输出到当前目录，最后带上日期时间戳，以便后续分析。

```
# 将 agent pid 对应的进程的堆栈信息输出到当前目录，以当前时间戳命名
jstack $(cat pid) > thread-$(date +%Y%m%d_%H%M%S).txt
```

下载当前采集的归档文件：

如果需要保留当前正在采集的归档文件，首先需要查看当前的进度文件，得到当前采集的 RAC 和 THREAD 信息。然后使用 ./cli.sh 中的 arc session，使用 CP 命令将对应的归档文件下载到 arch 目录。

打包所有其他文件：

简单来说，只需要将当前运行目录复制即可。但是需要注意的是，如果 trans 目录很大，可能会导致复制时间过长。可以选择只复制 conf, dict, arch, log 目录。

如何恢复

在保留完现场后，如果要恢复运行，选择重置 agent。并根据需要指定起点，参照指定起点的方法。

外部环境

1. 系统负载

- top命令
- iostat命令

• vmstat命令

4.7.2 根据 scn 重置起点

场景介绍

名词解释：- `scn` System change number 数据库严格单调递增的变化序列号，类似于 timestamp。- `savepoint` 起点信息，格式为 `RAC:<thread#> SEQ:<sequence#> BLK:0 BKO:0 LBK:0 LSC:0`

sequence# 和 thread# 从数据库查询得到 `scn` 所属日志的 thread# 和 sequence#

在监听程序停止时更新 `savepoint` 。下次启动时，则以该 `savepoint` 为起点启动。

常用命令

```
# 获取当前 scn
$ curl localhost:8303/tools/scn
1156657092
# 根据时间得到 scn
$ curl localhost:8303/tools/scn?time=2022-10-21T16:18:20
1191761688
```

执行步骤

1. 获取起点

```
# 获取 `scn` 对应日志文件，即 savepoint
$ curl localhost:8303/tools/logs?startScn=1153604547
[{"thr":1,"seq":9329,"name":"2_/u01/app/oracle/product/12.2.0/db_1/dbs/arch1_9329_1023213783.dbf"}]
```

2. 停止实例

```
# 执行停止实例命令
$ curl -XPOST localhost:8303/export/stop
Undeploy c4a973e1-8bdd-4da6-9a87-931b62fbc550
# 确认停止后状态为 DOWN
$ curl localhost:8303/export/health
{"status":"DOWN","checks":[{"id":null,"status":null,"data":null}]}
```

3. 更新起点

```
# 确认 Step.1 的日志文件无误后，使用 Step.1 的返回内容更新 savepoint
$ curl -XPOST localhost:8303/export/savepoint -d'[{"thr":1,"seq":9329,"name":"2_/u01/app/oracle/product/12.2.0/db_1/dbs/arch1_9329_1023213783.dbf"}]'
savepoint updated
```

4. 确认起点

```
$ curl localhost:8303/export/savepoint
RAC:1 SEQ:9329 BLK:0 BKO:0 LBK:0 LSC:0
```

5. 启动实例

```
# 启动实例
$ curl -XPOST localhost:8303/export/start
Deploy Oragent instance.
# 查看健康状态
$ curl localhost:8303/export/health
{"status":"UP","checks":[{"id":"1","status":null,"data":{"id":"1","deployment-id":"bf6fdf37-bbcb-421f-97b4-ef75fb1c5761","delay-sec":"2463","system":"cpu=25","err":null}}]}
```

4.7.3 处理日志被回收

当程序发生报错 Failed to get log name 时，通常原因是采集代理来不及解析被回收的日志。 当发生此类事件时，已被回收的重做日志中的数据就丢失了。或者由 DBA 恢复日志，或者选择跳过。

同时需要重新审核数据库归档策略与采集代理解析能力是否匹配。 查询归档日志是否开启，归档日志保留时间，采集代理解析速度，归档产生速度等等因素，来评估是否满足场景。

- 查询是否开启归档参考 确认归档日志模式
- 归档日志保留时间一般由 RMAN 控制，可以询问 DBA 相关设置。

手动处理跳过缺失日志

跳过有两种方式：

命令行方式

在采集代理运行目录下（oragent）

```
# 停止采集代理程序
./stop.sh
# 清除进度及残余数据
./clean.sh
# 启动采集代理程序
./start.sh -w
# 激活实例
curl -XPOST localhost:8303/export/start
```

API 方式

```
# 停止当前采集代理实例
curl -XPOST localhost:8303/export/stop
# 重启并重置进度到最新
curl -XPOST localhost:8303/export/start?reset=latest
```

设置自动重置策略

当频繁发生时，可以设定归档日志尚未被采集解析就被回收导致程序报错时的的处理策略：

- none 程序报错并不断重启
- earliest 重置为能找到的最早的归档日志位置
- latest 重置为当前最新的归档日志位置

POST /param/auto_reset_policy/[earliest/latest/none]

```
#更新策略配置
curl -XPUT localhost:8303/param/auto_reset_policy -d 'earliest'
#查看该配置是否被更新
curl localhost:8303/param/auto_reset_policy
```

4.7.4 常见报错处理

激活任务报错或超时

首先查看节点关于采集代理的日志判断原因。如果没有信息，查看采集代理日志判断原因。一般原因会由于采集程序报错，导致无法完成更新监听列表请求导致。 若是查看日志发现是由于一次性新增监听表过多导致。可增加 `map_update_timeout_sec` ，该参数默认为 300 秒。

任务激活时报已被 **xxx** 占用

采集代理只能对应一个 DP 节点。 当被别的节点对应的任务所激活后。不能再被其他节点对应的任务使用了。 如要强制使用该节点，则需 `./stop.sh` 停止实例后执行 `./clean.sh` 删除数据。 `./start.sh -w` 启动后，再次激活任务。

节点日志中发现**ASM**连接失败

一般由于节点信息中 ASM 连接方式填写有误导致。按照准备连接的格式正确填写节点 ASM 连接参数后。 发送 `/export/stop` 请求停止实例。再次激活任务时可更新连接信息。

ASM 远程读取大小端设置不正确导致进度不推进

该配置默认为小端，如果通过 `/export/health` 接口发现 delay 没有变化，或通过 `/export/savepoint` 接口发现进度没有推进。则有可能是由于大小端设置错误导致。 此时，需要 `/export/stop` 停止实例后，更新节点中大小端配置，而后再次激活任务。

4.7.5 许可证问题

许可证到期时间

采集代理的使用许可证与 DataPipeline 数据平台保持一致。当 DataPipeline 数据任务激活时，平台的许可证将自动更新至活动的采集代理。

在 DataPipeline 数据平台更新许可证后，平台的许可证也会同步更新至活动的采集代理。如果采集代理处于未启动状态，数据平台更新许可证后，需要手动重新激活相关任务以更新采集代理的许可证。

如果采集代理在启动后检测到许可证已过期，将自动进入暂停状态。待相关数据任务激活并更新许可证后，采集代理将自动恢复正常运行。

5. 附录

5.1 LogMiner审计日志内容

LogMiner 是 Oracle 内置的日志审计工具。可以用来分析日志中的 DML 及 DDL。 该功能需要 SYSDBA 角色的用户，或者创建拥有使用 LogMiner 功能的角色

5.1.1 权限

```
-- 创建 logminer 使用者角色
create role cdc_logminer_privs;
grant create session,execute_catalog_role,select any transaction,flashback any table,select any table,lock any table,select any dictionary to cdc_logminer_privs;
grant select on SYSTEM.LOGMNR_COL$ to cdc_logminer_privs;
grant select on SYSTEM.LOGMNR_OBJ$ to cdc_logminer_privs;
grant select on SYSTEM.LOGMNR_USER$ to cdc_logminer_privs;
grant select on SYSTEM.LOGMNR_UID$ to cdc_logminer_privs;
grant select on V_$DATABASE to cdc_logminer_privs;
grant select_catalog_role to cdc_logminer_privs;
grant LOGMINING to cdc_logminer_privs;
-- 将上述角色赋予某用户
grant cdc_logminer_privs to agent1;
alter user agent1 quota unlimited on users;
```

5.1.2 使用步骤

选择日志文件

首先先确认需要查询的日志文件的文件路径，可以通过查询 v$archived_log，通过 scn 或时间戳过滤出需要查询的归档文件或指定本地文件。

选择数据字典提供方式

- 若使用当前数据库（产生日志的数据库）则使用 若是使用 DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG 即可。
- 当使用其他 Oracle 数据库（非产生日志的数据库时）分析日志文件时，需要再原数据库通过 LogMiner 生成一个字典日志文件，该命令会生成一个额外的归档文件。在 logmnr.add_logfile 时指定。

```
EXECUTE DBMS_LOGMNR_D.BUILD(OPTIONS=> DBMS_LOGMNR_D.STORE_IN_REDO_LOGS);
```

执行 **LogMiner** 并查询结果

```
-- 加上日志文件
exec DBMS_LOGMNR.ADD_LOGFILE(LOGFILENAME => '+DATA/ideasdbdg1/onlinelog/group_21.396.1027396781', OPTIONS => DBMS_LOGMNR.NEW);
-- 开启 LogMiner
exec DBMS_LOGMNR.START_LOGMNR(OPTIONS => DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG+DBMS_LOGMNR.COMMITTED_DATA_ONLY);
-- 查询 LOGMNR 视图，该查询会正式解析日志文件。返回时间由加入的日志量决定。
SELECT SCN, OPERATION, SQL_REDO, XID, TIMESTAMP, ROW_ID FROM V$LOGMNR_CONTENTS WHERE SEG_NAME = 'DP_TEST';
```

5.1.3 参考

LogMiner 使用方法： https://docs.oracle.com/en/database/oracle/oracle-database/19/sutil/oracle-logminer-utility.html#GUID-A0D89906-C787-4EB4-BA47-171A457445EC V$LOGMNR_CONTENTS 视图详情： https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/V-LOGMNR_CONTENTS.html#GUID-B9196942-07BF-4935-B603-FA875064F5C3

5.2 指标监控

5.2.1 采集代理指标监控

采集代理使用 jmx_exporter 及 Micrometer 暴露关键指标。 可通过 Prometheus + Grafana 构建实时及历史指标监控系统。

有两种暴露方式：

- 标准模式：默认开启，不使用额外的端口，仍使用 param.yml 中定义的 port 作为端口。
- 插件模式：使用了 jmx_prometheus_javaagent-0.17.0.jar 并自定义额外端口。启动时 ./start.sh -w -j <指定端口>

两种方式都使用同一个 endpoint /metrics ， 区别是第一种方式可额外使用 /metrics?format=json 得到 JSON 返回。

查询指标接口

返回指标当前值:

```
# 标准模式
curl agent_ip:<agent_web_port>/metrics[?format=json]
# 插件模式
curl agent_ip:<jmx_exporter_port>/metrics
```

Prometheus 配置

以 Linux 单机部署的 Prometheus 为例配置 Prometheus.yml。

```
- job_name: "oracle_agent"
  static_configs:
    # 采集代理的部署 IP 及端口
    - targets: [ "192.168.0.14:9999" ]
```

Grafana 配置

1. Grafana 配置 Prometheus 数据源，详情参见 Grafana 官方使用文档。
2. Grafana 的 dashboards 中导入 agent_metrics_grafana 的 json 文件：agent性能监控配置文件

指标定义

采集代理相关指标

指标名称	指标含义	指标单位
agent_cmt_to_send_latency_95th_percentile	从一个事务被agent解析到该事务准备发送到kafka的处理时间的95%分位数	ms
agent_cmt_to_send_latency_max	从一个事务被agent解析到该事务准备发送到kafka的处理时间的最大值	ms
agent_cmt_to_send_latency_mean	从一个事务被agent解析到该事务准备发送到kafka的处理时间的平均值	ms
agent_deallag	agent解析的积压（agent解析进程落后redo日志时间）	ms
agent_start_total	描述自上次agent start.sh -w 后agent的解析进程启动的次数	次数
agent_redolog_deal_quantity_oneMinute_rate	agent日志率解析速率	KB/s
agent_redolog_deal_quantity_count	描述自agent启动以来的日志解析量	KB
agent_process_trans_total	描述自agent启动以来的事务处理量	条
agent_cmt_to_send_lag_max	描述oracle事务提交到写入kafka的所用的时间	ms
kafka_producer_byte_rate	描述每个kafka producer 的kafka写入速率	B/s

同时 Micrometer 提供系统相关指标：

指标组	描述
JvmGcMetrics	JVM GC 指标
JvmMemoryMetrics	JVM 内存指标
JvmThreadMetrics	JVM 线程指标
DiskSpaceMetrics	磁盘空间指标
FileDescriptorMetrics	文件句柄指标
ProcessorMetrics	处理器指标
UptimeMetrics	运行时间指标

关键性系统指标：

指标名	指标含义	单位
java_lang_Memory_HeapMemoryUsage_max	最大堆内存量	Byte
java_lang_Memory_HeapMemoryUsage_used	堆内存使用量	Byte
java_lang_Memory_NonHeapMemoryUsage_max	最大堆外内存	Byte
java_lang_Memory_NonHeapMemoryUsage_used	堆外内存使用量	Byte
java_lang_GarbageCollector_CollectionTime{name="G1 Young Generation"}	新生代回收耗时	Byte
java_lang_GarbageCollector_CollectionTime{name="G1 Old Generation"}	老生代回收耗时	Byte
java_lang_GarbageCollector_CollectionCount{name="G1 Young Generation"}	新生代回收次数	Byte
java_lang_GarbageCollector_CollectionCount{name="G1 Old Generation"}	老生代回收次数	Byte
java_lang_ClassLoading_TotalLoadedClassCount	总载入类	
java_lang_OperatingSystem_SystemCpuLoad	系统 cpu load	
java_lang_OperatingSystem_ProcessCpuLoad	进程 cpu load	
metrics_processCpuUsage_Value{type="gauges"}	进程 cpu 占用率	
metrics_systemCpuUsage_Value{type="gauges"}	系统 cpu 占用率	
java_lang_Threading_ThreadCount	线程数	
java_lang_Threading_DaemonThreadCount	守护线程数	
metrics_jvmThreadsLive_Value{type="gauges"}	活动线程数	
metrics_jvmThreadsDaemon_Value{type="gauges"}	守护线程数	

具体指标名可通过接口返回过滤关键字

```
curl -s /metrics | grep -v '#' | grep -i <关键字>
```

比如 file（文件句柄相关）thread（线程相关）对照其名称得到具体名称。 例如:

```
curl -s localhost:7303/metrics  | grep -v '#' | grep -i file
metrics_processFilesMax_Number{type="gauges"} 655350.0
metrics_processFilesMax_Value{type="gauges"} 655350.0
metrics_processFilesOpen_Value{type="gauges"} 49.0
java_lang_OperatingSystem_OpenFileDescriptorCount 49.0
metrics_processFilesOpen_Number{type="gauges"} 49.0
java_lang_OperatingSystem_MaxFileDescriptorCount 655350.0
```

配置输出规则

采集代理会保留所有上述指标及输出固定格式。其配置文件放置在 `export-<version>.jar` 中。 可通过配置文件 `conf/metrics_v1.yml` 覆盖默认配置文件。 详细解释及使用方法请参考 jmx_exporter 配置方法。

匹配输入

此处摘抄自 jmx_exporter 配置方法

Pattern 匹配格式为：

```
domain<beanpropertyName1=beanPropertyValue1, beanpropertyName2=beanPropertyValue2, ...><key1, key2, ...>attrName: value
```

组成部分	描述
domain	Bean 名，JMX Object 逗号前的部分。
beanPropertyName/Value	Bean 属性 (key/Value)。 JMX Object 名逗号后的部分。
keyN	如果是复合或表格式数据, 属性名会在此列表中，类似列名
attrName	属性名
value	属性值

默认输出格式

```
domain_beanPropertyValue1_key1_key2_...keyN_attrName{beanpropertyName2="beanPropertyValue2", ...}: value
```

注：micrometer 会将程序中定义的 label (例如 src_id) 拼接在 name (beanProperty) 上，可通过正则表达式匹配后还原。 例如：

```
- pattern: metrics<name=agentRedologDealQuantity\.src\_id\.(\d+), type=meters.*><>OneMinuteRate
  name: agent_redolog_deal_quantity_oneMinute_rate
  labels:
    src_id: "$1"
  type: GAUGE
# domain：metrics
# beanpropertyName1：name
# beanpropertyName2：type
# attrName：OneMinuteRate
```

5.3 ASM磁盘信息配置