

YTS 拆分数据扫描定时任务到独立微服务设计文档

文档编号: yts-task-schedule

创建日期: 2023-10-11

最后修改日期: 2023-10-12

版本号: V1.0

电子版文件名: YTS 拆分数据扫描定时任务到独立微服务设计文档

文档修订摘要

[illegible]

目录

目录

YTS 拆分数据扫描定时任务到独立微服务设计文档 1

电子版文件名：YTS 拆分数据扫描定时任务到独立微服务设计文档 1

1. 概述 4

 现有方案的问题 4

 1.1 目的 4

 1.2 阅读对象 4

 1.3 相关资料 4

2. 整体设计 5

 2.1 整体设计 5

 2.2 优化 YTS-SDK 中 SPRING 定时任务线程池数 6

3. 集中扫描服务详细设计 7

 3.1 微服务 YTS 定时任务是否已集中统一服务中接口 7

 3.2 上报微服务数据保留天数 8

 3.3 服务端记录 YTS 已统一定时扫描的微服务列表记录 8

 3.4 YTS-SCHEDULE 定时扫描的逻辑 9

 3.5 YTS-SDK 增加异常事务重试的通用 RPC 服务 10

 2.3.6 YTS-SDK 增加服务启动后查询是否已经统一定时任务判断 10

 2.3.7 YTS-scheducle 服务查询未进行逻辑数据源共享的服务提醒共享 10

 2.4 方案 4 保留现有定时任务方案优化合包部署的定时任务 错误！未定义书签。

1.概述

现有方案的问题

- 1 解决 YTS-SDK 启动 11 个定时任务
- 2 定时任务绝大多数是数据库扫描和更新操作以及上报到服务端以及从服务端拉取数据
- 3 按服务器实例启动了 spring-schedule 的线程池(5 个线程)简强架构时需要合并线程池

1.1 目的

本文将阐述 YTS 的定时任务由原来仅支持 Spring 的定时任务框架触发拆分为数据库扫描上报部分迁移到 EOS 定时扫描的服务中，减少 YTS 的定时任务分布在每个微服务中占用线程池以及不能集中管理的问题，以及方便以后统一优化扫描逻辑。

1.2 阅读对象

此文档将适合如下人员阅读：
领域设计工程师。
产品开发工程师。
产品测试工程师。
其他相关人员。

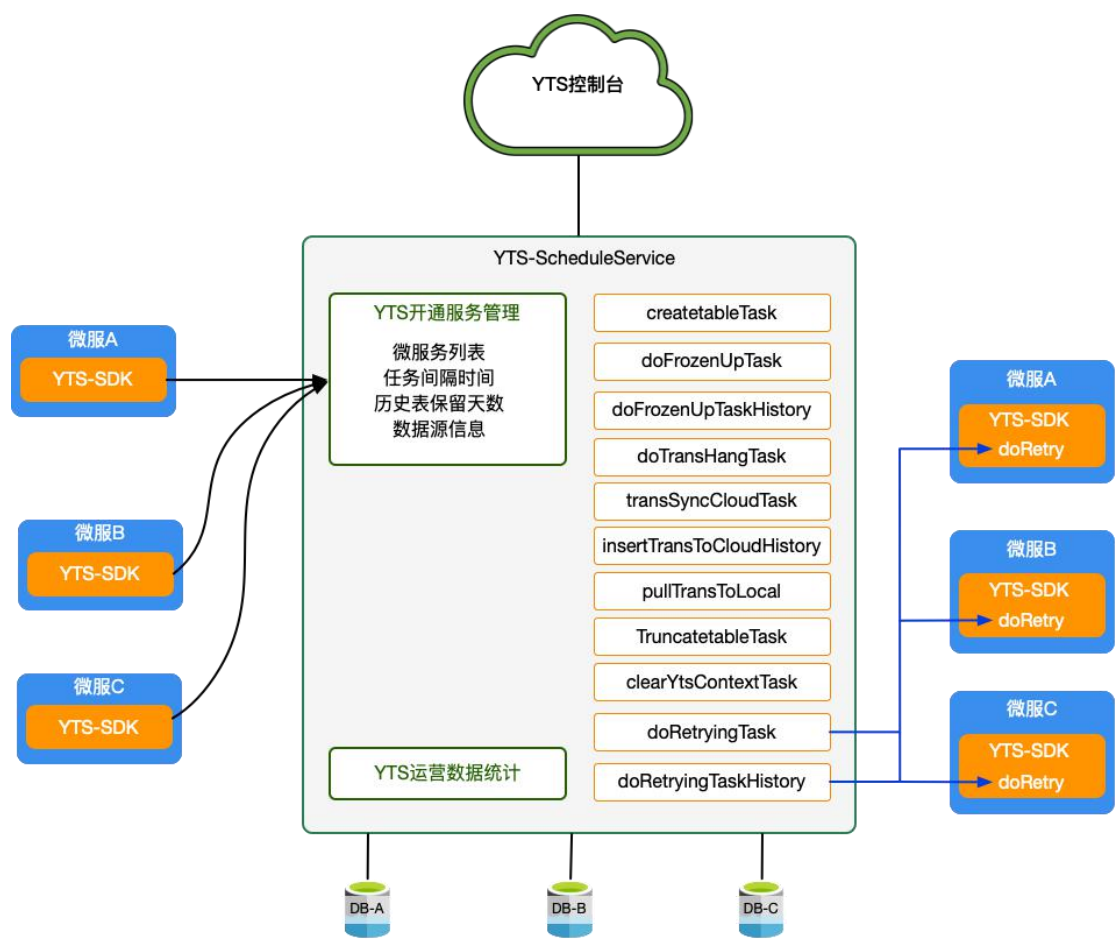
1.3 相关资料

《一致性框架(YTS)设计文档》

2.整体设计

为了版本兼容以及防止集中扫描服务异常导致 YTS 无法正常工作，保留原来的 YTS 中集成的 spring 定时任务，在服务启动时请求集中扫描服务的接口，查询该服务是否进行集中扫描，如果没有启用则仍旧启动原有的 spring 的定时任务使 YTS 功能正常。

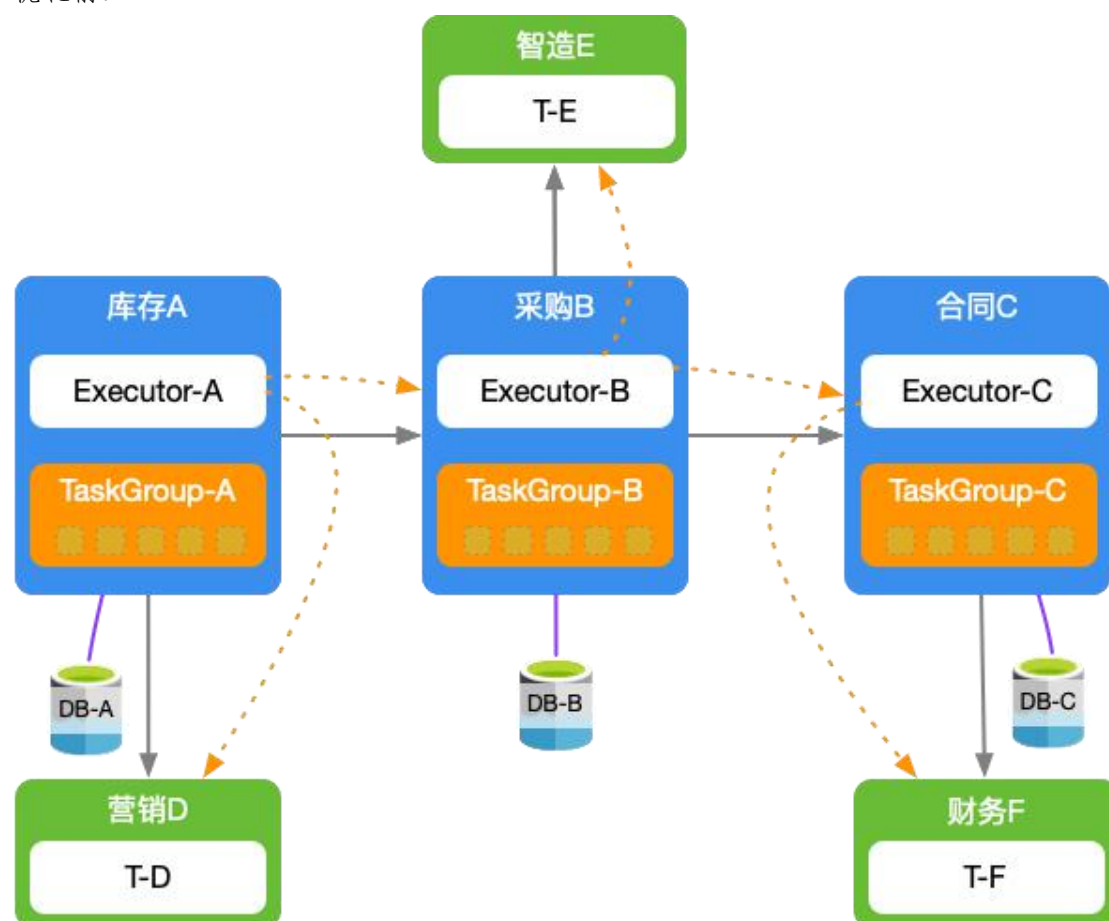
2.1 整体设计



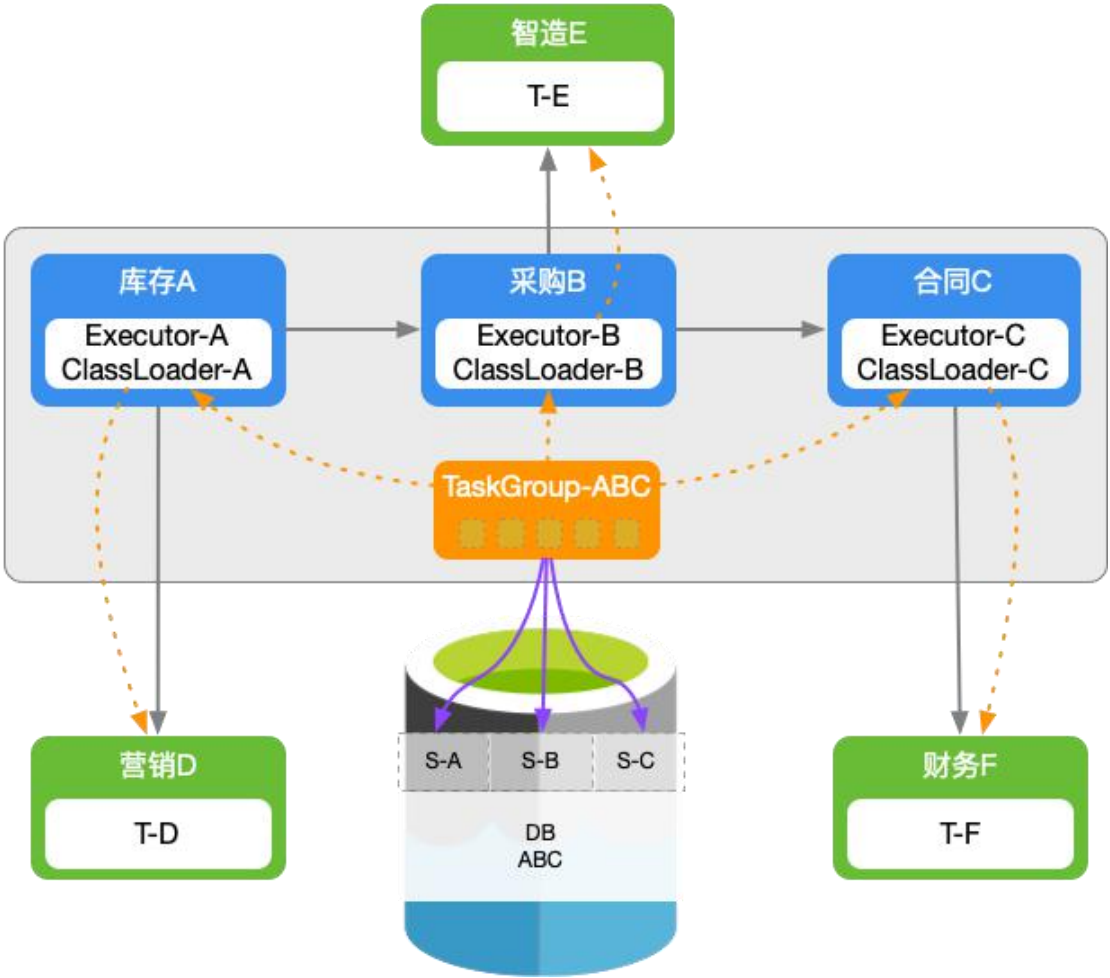
2.2 优化 YTS-SDK 中 spring 定时任务线程池数

但在合包部署时需要优化 YTS-SDK 中 spring 定时任务线程数，使在合包部署时一个进程只开启一个 spring 的定时任务线程池。

优化前：



优化后：



3.集中扫描服务详细设计

3.1 微服务 YTS 定时任务是否已集中统一服务中接口

参数列表

参数编码	参数名说明	参数数据类型	是否必填
serviceName	微服务编码	string	是
env	部署的环境信息	string	是
logicDsCode	微服务逻辑数据源编码	srring	是
ytsVersion	yts 的版本信息	string	是

返回结果

参数编码	参数名说明	参数数据类型	是否必填
isYtsSchedule	是否已被统一调度	boolean	是

如果根据查询的微服务编码已经在统一的服务中进行统一定时服务中，并且已把服务的逻辑数据源共享则返回 **true** 否则返回 **false**

3.2 上报微服务数据保留天数

参数列表

参数编码	参数名说明	参数数据类型	是否必填
serviceName	微服务编码	string	是
env	部署的环境信息	string	是
ytsVersion	yts 的版本信息	string	是
truncateBeforeDays	数据保留天数	int	是

返回结果

参数编码	参数名说明	参数数据类型	是否必填
reportSuccess	是否上报成功	boolean	是

服务端查询该服务的记录，确认如果数据的保留天数和上报的不同则修改服务的保留天数的记录，修改成功或者记录的保留天数和上报的一致则返回成功，否则返回 **false**。

3.3 服务端记录 YTS 已统一定时扫描的微服务列表记录

记录 YTS 已统一定时扫描的服务列表

yts_schedule_service_info

属性	类型	说明
id	bigint	oid 主键
service_name	varchar(255)	微服务编码
logic_ds_code	varchar(128)	微服务逻辑数据源编码
ds_share_status	int	是否已共享数据源 如果没有共享数据源不开启集中扫描，启动原有的

		定时任务
truncate_before_days	int	YTS 记录保留的天数
sdk_version	varchar(50)	YTS 的版本
env	varchar(50)	部署的环境信息
report_time	datetime	第一次上报时间
update_time	datetime	最后一次上报时间

表结构设计要遵循数据库设计规范

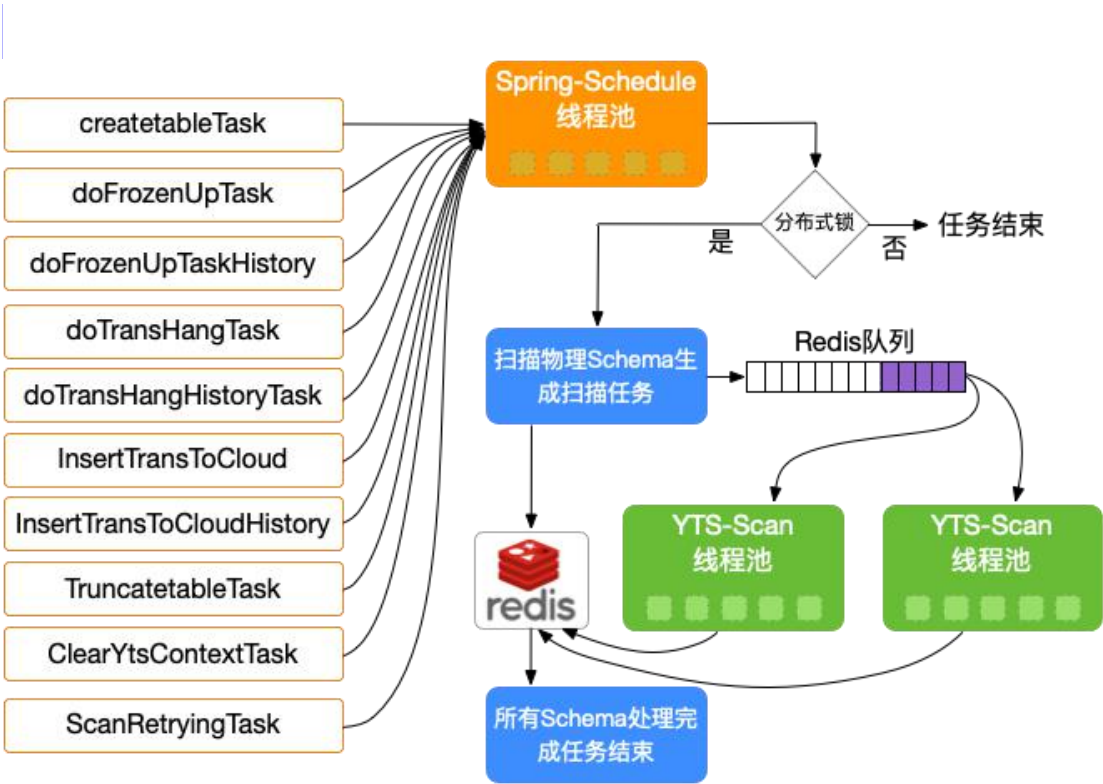
服务合并的场景（对比领域服务合并文档和 yts 接入文档需要处理 YTS 相关事项）

合并场景包括 rpc 短路和数据库事务合并场景的区分。

3.4 YTS-Schedule 定时扫描的逻辑

根据 YTS 定时任务设置的间隔，定时运行 YTS 的定时任务逻辑。

定时任务的逻辑根据微服务名遍历逻辑数据源中的所有 schema 扫描数据，并完与 YMS-console 云服务的交互，如果有需要重试的定时任务则调用该微服务重试 RPC 接口完成该异常事务的重试操作。每个定时任务的主要功能参考（3.8 章节）



* 建表的定时任务如果 YTS-Scan 线程池的任务队列满则等待，其他任务直接忽略，等待下一轮扫描。

3.5 YTS-SDK 增加异常事务重试的通用 RPC 服务

```
/**  
 * 重试异常事务  
 * @param gtxId  
 * @return  
 */
```

```
ProcessResult retry(String gtxId);
```

该服务有 YTS-schedule 的统一定时任务根据微服务名进行调用

3.6 YTS-SDK 增加服务启动后查询是否已经统一定时任务判断

服务启动后 YTS-SDK 增加从 YTS-schedule 服务中查询是否已经统一定时任务如果已经统一定时任务则无需启动内部的定时任务，交由统一的定时任务服务扫描数据库并 YMS-console 进行交互，本服务有需要重试的异常事务时有统一定时任务调用该服务的重试接口进行重试。

启动后上报本服务的相关信息到 YTS-schedule 服务中。

3.7 YTS-schedule 服务查询未进行逻辑数据源共享的服务提醒

YTS-schedule 微服务查询已经上报信息而且没有进行逻辑数据源共享的服务，提醒该微服务进行数据源共享。

3.8 YTS 定时任务

定时任务扫描所有开启集中扫描的微服务，为每个微服务额所有物理数据源进行定时任务的操作。

3.8.1 定时建表的定时任务(CreateTableTask)

默认执行时间为凌晨 1:05/2:05/3:05/4:05/5:05/6:05 执行 6 次避免由于某个时间段数据库被锁等原因无法建表成功的问题。运行建表成功后，将标记进行缓存，下次任务

执行时直接跳过，无需进行多次数据库建表操作，建表类型有 2 个一个为 yts_trans_yyyyMMdd 一个为 yts_rel_yyyyMMdd，每个类型表均创建未来 10 天的表，避免服务某些原因异常导致定时任务没有正常执行而无法建表成功的问题。

3.8.2 删除过期历史表的定时任务(TruncateTableTask)

根据微服务设置的保留历史天数的设置，删除保留天数以前的 yts_trans_yyyyMMdd,yts_rel_yyyyMMdd 两种类型的表。

3.8.3 删除过期 YTS 上下文数据的定时任务(CleanYtsContextTask)

根据微服务设置的保留历史数据天数清理 yts_context 表中过期的 YTS 上下文的数据，清理数据时根据 gtxId 中的日期清理过期数据，清理需要分批并且在执行过程可以根据设置间隔休眠一定时间，避免对数据库形成过大的压力。

3.8.4 防挂起(当前日期)处理的定时任务 (DoTransHangTask)

默认间隔为 10 分钟一次，查询当天 yts_trans_yyyyMMdd 表中状态为 confirming 的 Root 节点的数据，根据调用下游的 rel 记录完成该事务的后续逻辑，使该事务处理能完成处理。

3.8.5 防挂起(历史)处理的定时任务(DoTransHangHistoryTask)

默认间隔时间为 1 小时一次，查询该服务保留天数并遍历查询 yts_trans_yyyyMMdd 表中状态为 confirming 的 Root 节点的数据，根据调用下游的 rel 记录完成该事务的后续逻辑，使该事务处理能完成处理。

3.8.6 重试的定时任务(DoRetryTransTask)

默认间隔时间为 10 分钟，服务从 YTS 云端控制台查询该服务需要重试的异常事务额列表，将异常事务更新状态为待重试，查询数据库中状态为 retrying 的事务然后遍历调用该服务重试接口（3.5 章节）进行重试。

3.8.7 解锁业务数据(当天)的定时任务(DoFrozenUpTask)

默认执行时间间隔为 10 分钟，查询当天有待解锁的业务数据记录，调用解锁接口进行解锁操作。

3.8.7 解锁业务数据(历史)的定时任务(DoFrozenUpHistoryTask)

默认执行时间间隔为 10 分钟，根据微服务设置的保留历史天数，遍历查询所有历史表中待解锁的历史记录调用解锁接口进行解锁操作。

3.8.8 上报当天异常事务的定时任务(ReportToCloud)

默认执行时间为 10 分钟，上报当天异常事务数据或者当天重试后状态有变更的数据，以及该服务业务成功但上游业务失败的事务数据到云端控制台，方便云端进行管理，以及重试等操作。

3.8.9 上报历史异常事务的定时任务(ReportToCloudHistory)

默认间隔时间为 1 小时，根据微服务设置的历史数据保留天数查询所有历史表中需要上报的异常事务数据上报到云端。

4. 服务部署

由于调用 YTS 的重试任务需要使用 RPC 的服务调用，所以部署未和微服务对应的环境每个环境部署一份，先期该集中扫描的服务功能集成到 EOS 的统一服务中。

5.附录

5.1 已接入 YTS 的微服务列表：

微服务：86

Shema：76

领域	微服务名	schema	专属化 schema
平台	iuap-apdoc-customer	ugoods	iuap_apdoc_coredoc
	iuap-apdoc-material		iuap_apdoc_coredoc
	ubpm-web-rest	ubpm	
基础数据	yonbip-am-ambd	amc_ambd	amc_ambd
	yonbip-am-aum	amc_aum	amc_aum
	yonbip-am-rmm	amc_mro	amc_mro
采购云	yonbip-cpu-bcprice	cpu_baseservice	cpu_baseservice
	yonbip-cpu-bcres		cpu_baseservice
	yonbip-cpu-bcsourcing	cpu_pubbiz_sourcing	cpu_pubbiz_sourcing
	yonbip-cpu-bid	cpu_bid	cpu_bid
	yonbip-cpu-cooperation	cpu_cooperation	cpu_cooperation
	yonbip-cpu-lawbid	cpu_lawbid	cpu_lawbid
	yonbip-cpu-mall	cpu_mall	cpu_mall
	yonbip-cpu-sourcing	cpu_sourcing	cpu_sourcing
	yonbip-cpu-suppliermgr	cpu_suppliermgr	cpu_suppliermgr
	yonbip-cpu-synergy	cpu_synergy	cpu_synergy
	yonbip-cpu-yop	yuncai_yop	yuncai_yop
财务	yonbip-fi-arap	fiproduct	fiproduct
	yonbip-fi-ctmbam	ctmbam	ctmbam
	yonbip-fi-ctmcam	ctmcam	ctmcam
	yonbip-fi-ctmcmp	ctmcmp	ctmcmp
	yonbip-fi-ctmcmp-import	ctmcmp	
	yonbip-fi-ctmcspl	ctmcspl	ctmcspl
	yonbip-fi-ctmdrft	ctmdrft	ctmdrft
	yonbip-fi-ctmfdtr	ctmfdtr	ctmfdtr

	yonbip-fi-ctmgrm	ctmgrm	ctmgrm
	yonbip-fi-ctmlcm	ctmlcm	ctmlcm
	yonbip-fi-ctmlgm	ctmlgm	ctmlgm
	yonbip-fi-ctmstct	ctmstct	ctmstct
	yonbip-fi-ctmstwb	ctmstwb	ctmstwb
	yonbip-fi-ctmtlm	ctmtlm	ctmtlm
	yonbip-fi-eafrecord	yms_pre	idoc
	yonbip-fi-eafsys	yms_pre	idoc
	yonbip-fi-etl	figl	figl
	yonbip-fi-exsapct	apct	apct
	yonbip-fi-expsrbsm	znbz	znbz
	yonbip-fi-fa	fiefa	fiproduct
	yonbip-fi-gl	figl	figl
	yonbip-fi-ia	fieia	fi_report
	yonbip-fi-it	fiproduct	fiproduct
	yonbip-fi-miderp	fiproduct	fiproduct
	yonbip-fi-otp	figl	figl
	yonbip-fi-sepmbcdc	epmbcdc	epmp_bcdc
	yonbip-fi-taxit	diwork_input_tax	diwork_input_tax
	yonbip-fi-taxot	diwork_piaoeda_client	diwork_piaoeda_client
人力	yonbip-hr-contract	corehr	corehr
	yonbip-hr-paybiz	diwork_wa_mdd	diwork_wa_mdd
	yonbip-hr-paybizo	diwork_wa_split	diwork_wa
	yonbip-hr-staff	iuap_cloud_basedoc	iuap_apdoc_basedoc
	yonbip-hr-staffmgo	iuap_cloud_basedoc	iuap_apdoc_basedoc
	yonbip-hr-staffmgr	corehr	corehr
	yonbip-hr-staffproc		corehr
共享服务	yonbip-mid-sscpf	shared_service	shared_service
营销	yonbip-mkt-crm	crm	crm
	yonbip-mkt-dst	omsdata	omsdata
	yonbip-mkt-dstbiz		omsdata
	yonbip-mkt-mkc2b	uorders	uorders
	yonbip-mkt-retailbiz	uretaildata	uretaildata
	yonbip-mkt-retailweb		uretaildata
	yonbip-mkt-usmp	usmp	usmp
制造	yonbip-mm-mfed	engineering_data	engineering_data
	yonbip-mm-mfmr	requirements_planning	requirements_planning
	yonbip-mm-mfpo	production_order	production_order

	yonbip-mm-qic	qms_dfm	qms_dfm
	yonbip-mm-qms	qms_qit	qms_qit
	yonbip-mm-sfc	imp_sfc	imp_sfc
项目云	yonbip-pm-budgetme	pmcloud	pmcloud
	yonbip-pm-commonbd		pmcloud
	yonbip-pm-planme		pmcloud
	yonbip-pm-project		pmcloud
	yonbip-pm-projectme		pmcloud
	yonbip-pm-revexp		pmcloud
	yonbip-pm-workhours		pmcloud
供应链	yonbip-scm-inttrade	uinttrade	uinttrade
	yonbip-scm-inttradebiz		uinttrade
	yonbip-scm-pu	upurchase	upurchase
	yonbip-scm-pubiz		upurchase
	yonbip-scm-sact	sact	sact
	yonbip-scm-scmbd	scmbd	scmbd
	yonbip-scm-scmmp	scmmp	scmmp
	yonbip-scm-scmpub	ustock	ustock
	yonbip-scm-scmpubbiz	uscmpub	ustock
	yonbip-scm-scmsa	uorders	uorders
	yonbip-scm-stock	ustock	ustock
	yonbip-scm-stockbiz		ustock
	yonbip-scm-stockimport		ustock