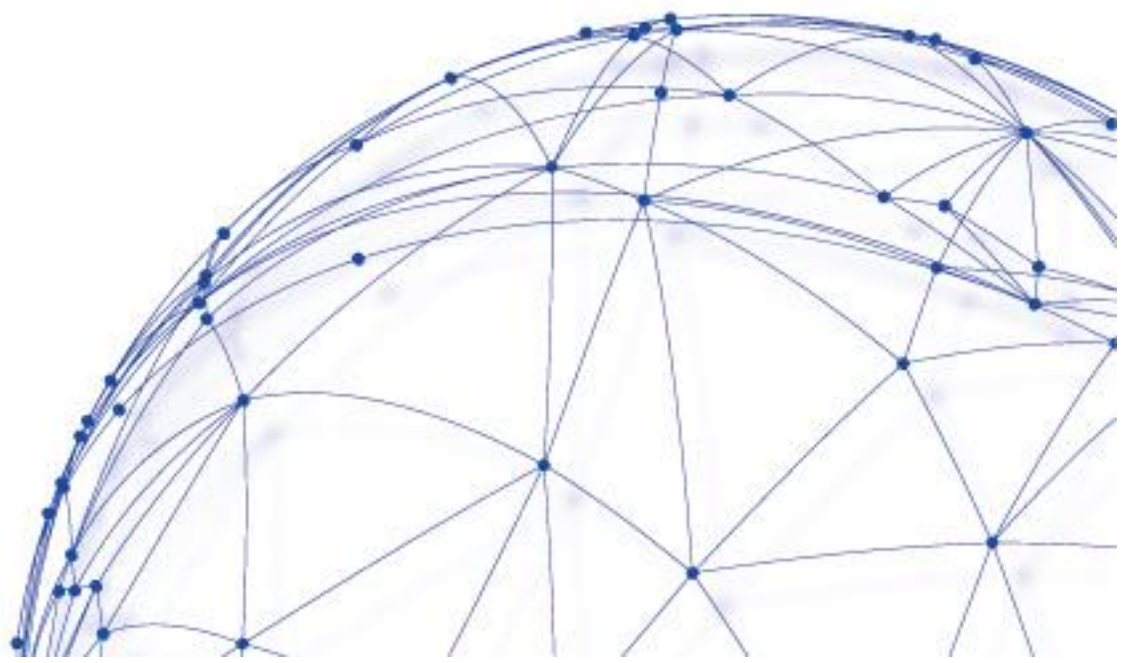


达梦技术手册

DM8_dmagent 使用手册

Service manual of DM8_dmagent



前言

概述

本文档主要介绍 DM Web 版管理工具 DEM 部署在远程主机上的代理软件 dmagent 的主要功能和使用方法。

读者对象




本文档主要适用于 DM 数据库的：

- 开发工程师
- 测试工程师
- 技术支持工程师
- 数据库管理员

通用约定

在本文档中可能出现下列标志，它们所代表的含义如下：

表 0.1 标志含义

标志	说明
 警告：	表示可能导致系统损坏、数据丢失或不可预知的结果。
 注意：	表示可能导致性能降低、服务不可用。
 小窍门：	可以帮助您解决某个问题或节省您的时间。



说明:

表示正文的附加信息，是对正文的强调和补充。

在本文档中可能出现下列格式，它们所代表的含义如下：

表 0.2 格式含义

格式	说明
宋体	表示正文。
Courier new	表示代码或者屏幕显示内容。
粗体	表示命令行中的关键字（命令中保持不变、必须照输的部分）或者正文中强调的内容。 标题、警告、注意、小窍门、说明等内容均采用粗体。
< >	语法符号中，表示一个语法对象。
::=	语法符号中，表示定义符，用来定义一个语法对象。定义符左边为语法对象，右边为相应的语法描述。
	语法符号中，表示或者符，限定的语法选项在实际语句中只能出现一个。
{ }	语法符号中，大括号内的语法选项在实际的语句中可以出现 0...N 次（N 为大于 0 的自然数），但是大括号本身不能出现在语句中。
[]	语法符号中，中括号内的语法选项在实际的语句中可以出现 0...1 次，但是中括号本身不能出现在语句中。
关键字	关键字在 DM_SQL 语言中具有特殊意义，在 SQL 语法描述中，关键字以大写形式出现。 但在实际书写 SQL 语句时，关键字既可以大写也可以小写。

访问相关文档

如果您安装了 DM 数据库，可在安装目录的“\doc”子目录中找到 DM 数据库的各种手册与技术丛书。

您也可以通过访问我们的网站 www.dameng.com 阅读或下载 DM 的各种相关文档。

联系我们

如果您有任何疑问或是想了解达梦数据库的最新动态消息，请联系我们：

网址: www.dameng.com

技术服务电话: 400-991-6599

技术服务邮箱: dmtech@dameng.com

目录

1 简介	7
2 主要功能	7
2.1 Java 外部函数的代理功能.....	7
2.2 SQL 命令行工具.....	7
2.3 build 工具	8
2.4 远程调用服务功能.....	8
2.5 收集 DEM 监控数据功能.....	8
2.6 dmagent 自动升级功能.....	8
2.7 smoke 工具.....	8
2.8 TPCC 测试工具	8
2.9 批量执行.....	9
2.10 升级替换.....	9
3 使用 dmagent	10
3.1 安装 dmagent.....	10
3.1.1 安装须知	10
3.1.2 获取 dmagent.....	10
3.1.3 安装示例	12
3.2 配置 dmagent.....	14

3.2.1 AGENT.INI 配置项	14
3.2.2 配置示例	38
3.3 启动与停止	39
3.3.1 start 启动脚本	39
3.3.2 服务	44

1 简介

dmagent 是 DM Web 版管理工具 DEM 部署在远程主机上的代理软件。DEM 通过 dmagent 可以监控远程主机的相关信息，也可以在远程主机部署 DM MPP、达梦数据守护和读写分离集群等集群系统。

同时，dmagent 也提供了 SQL 命令行查询工具和 TPCC 测试工具。

2 主要功能

本节介绍 dmagent 提供的主要功能。

2.1 Java 外部函数的代理功能

Java 外部函数的执行均通过 dmagent 进行，dmagent 转载指定的 jar 包，执行函数，并返回结果给达梦数据库服务器。

可通过 ap_enable 参数启用或关闭该功能。

外部函数使用请参见《DM8_SQL 语言使用手册》。

2.2 SQL 命令行工具

Isql 是用 Java 编写的客户端工具，可以连接和操作数据库。Isql 不仅能连 DM，还可以连其他类型的数据库（Mysql、Oracle 等）。

2.3 build 工具

编译打包工具。专门提供给达梦公司内部人员使用。

2.4 远程调用服务功能

远程调用服务功能是指 dmagent 辅助其他工具完成远程调用功能，例如，辅助 DEM 实现集群部署功能。

可通过 AGENT.INI 参数 service_enable 启用或关闭该功能。

2.5 收集 DEM 监控数据功能

dmagent 对每个监控项都会启动一个 gather 线程, gather 线程会定时的收集每个监控项数据，发送给 DEM。

可通过 AGENT.INI 参数 gather_enable 启用或关闭该功能。

2.6 dmagent 自动升级功能

提供 dmagent 自动升级功能。

可通过 AGENT.INI 参数 upgrade_enable 启用或关闭该功能。

2.7 smoke 工具

冒烟测试代理工具。专门提供给达梦公司内部人员使用。

2.8 TPCC 测试工具

提供 TPCC 数据库性能测试工具。

2.9 批量执行

批量执行功能包括批量执行脚本（Linux 的 SHELL 脚本、Windows 的 bat 脚本）、批量执行 SQL 语句，批量执行 SSH 客户端命令。

2.10 升级替换

提供了 repalce 插件来升级替换相关文件。

3 使用 dmagent

3.1 安装 dmagent

3.1.1 安装须知

- 一台机器可以启动一个或多个 dmagent。但如果启动多个，则 center_url 建议配置成不同的，否则发送数据时会相互影响。或者为了避免这种影响，可以只开启一个 dmagent 的数据收集功能。
- 在远程主机使用 dmagent，需首先手动将 dmagent 拷贝到远程主机；然后根据不同的平台，选用 service(服务方式)或 start(命令行方式)运行 dmagent。
- 运行 dmagent 需要用户设置 JRE_HOME 或 JAVA_HOME 环境变量，dmagent 未自带 JAVA 环境，因此需要用户自行配置 JAVA 环境，JAVA 版本必须为 JAVA 1.8。
- 建议以非 root 用户运行 dmagent，但如果部署集群需要创建系统服务，则只能以 root 用户运行。
- 运行 dmagent 的部分功能需要用到 service.key 文件，且需要安装配置了 service.key 才能启动 dmagent。可以通过 install_service_key 功能和 uninstall_service_key 功能来生成和删除 service.key。

3.1.2 获取 dmagent

可以通过 2 种方式获取到 dmagent 程序。下面分别介绍：

方式一 达梦数据库安装目录的 tool 下存有 dmagent。例如 D:\dmdbms\tool\dmagent。

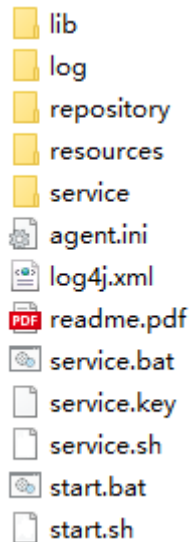


图 3.1 dmagent 目录

目录说明：

lib 目录：存放 dmagent 运行所需要的 jar 包。

log 目录：存放 dmagent 生成的日志文件。

repository 目录：仓库目录，存放各插件生成的临时文件，包含下载文件、上传文件等等。

resources 目录：资源目录，存放各插件运行时需要的相关资源文件。此目录在 dmagent 运行时为只读，即运行时不会在该目录下生成文件或修改目录下的文件。

service 目录：存放服务相关的执行文件。

agent.ini：dmagent 配置文件。

log4j.xml：日志配置文件。

readme.pdf：dmagent 使用说明文档。

service.key：远程 RPC 服务调用时所使用的校验 key，dmagent 作为服务提供方。

DEM 系统配置中所用的 service_key 使用的也是该文件。注意：service.key 需要用户在安装 dmagent 之后手动生成，生成方式参考 3.3.1 小节中的 install_service_key 功能说

明；若需要删除 service.key，参考 3.3.1 小节中的 uninstall_service_key 功能说明。

service.bat/service.sh: dmagent 服务脚本。service.bat 为 Windows 环境下使用；service.sh 为 Linux(Unix)环境下使用。脚本的具体使用请查看 3.3 小节。

start.bat/start.sh: dmagent 启动脚本。start.bat 为 Windows 环境下使用；start.sh 为 Linux(Unix)环境下使用。脚本的具体使用请查看 3.3 小节。

方式二 登录部署好的 DEM，在“DEM-资源包”中下载 dmagent。拷贝 dmagent 到要部署的机器上。

3.1.3 安装示例

本节将 dmagent 安装在 IP 为 192.168.0.105 的机器上。

安装 dmagent 的具体步骤为：

一. 为 IP: 192.168.0.105 机器配置 JAVA 1.8 环境。

dmagent 的运行需要依赖 JAVA 环境，因此部署 dmagent 前需要配置 JAVA 环境，JAVA 版本必须为 JAVA 1.8。

二. dmagent 部署步骤。

dmagent 部署步骤如下：

1. 获取 dmagent

将获取到的 dmagent 拷贝到要部署的机器上。

2. 修改 dmagent 配置

配置 dmagent 目录下的 AGENT.INI。更多的配置请参考 [3.2 配置 dmagent](#)。

下面以收集 DEM 监控数据功能参数为例。

```
center_url=http://192.168.0.104:8080/dem      #DEM 地址
```

gather_enable=true	#开启收集监控数据功能，默认打开
service_enable=true	#开启远程调用功能，部署时需要打开，

3. 核对 dmagent 和 DEM 所运行机器时间

此步骤为 DEM 专用。其他环境可跳过此核对时间步骤。

一般要求 dmagent 和 DEM 所运行机器时间一致，最多不要相差 M 分钟，M 取 5 分钟和主机失效时间两者中的最小值。其中，5 分钟为 dmagent 消息的有效时间，即 DEM 收到 dmagent 的消息时，如果收到的 dmagent 发送的消息的时间与 DEM 运行机器当前时间相差 5 分钟，则认为该消息无效，不处理该消息。主机失效时间 mf_invalid_time（单位：秒）即在主机失效时间内没有收到 dmagent 发送的信息，则认为主机失效。

若要 dmagent 可用，须保证 DEM 可以收到 dmagent 的消息，并且 dmagent 本身不是失效状态（失效状态即为 DEM 主机管理页面显示连接失败状态），因此 dmagent 与 DEM 所运行机器时间的差值取 5 分钟和主机失效时间的最小值，但建议 dmagent 与 DEM 所运行机器时间同步。

4. 生成并安装 service.key

dmagent 目录下：

Windows：

```
start.bat INSTALL_SERVICE_KEY 999999 #999999 为指定的 key
```

linux：

```
./start.sh INSTALL_SERVICE_KEY 999999 #999999 为指定的 key
```

5. 启动 dmagent

dmagent 目录下：

Windows：

```
start.bat <DMAGENT_HOME>/agent.ini    #进程方式运行
```

```
service.bat start/restart              #服务方式运行
```

linux:

```
./start.sh <DMAGENT_HOME>/agent.ini    #进程方式运行
```

```
./service.sh start/restart              #服务方式运行
```

若要部署集群并创建系统服务, 在 Linux (Unix) 环境下应以 root 用户运行 dmagent, 在 Windows 环境下应以 Administrator 用户运行 dmagent。

若要以服务方式运行 dmagent, 必须先注册服务(install), 再执行启动(start)、停止(stop)、删除(delete)、查看服务状态(status)以及打印 service 脚本帮助信息(help)等操作。

了解更多的启动和停止内容请参考 [3.3 启动与停止](#)。

3.2 配置 dmagent

用户需要通过 AGENT.INI 配置文件来配置 dmagent, AGENT.INI 配置文件存放在 AGENT_HOME 目录下(注: AGENT_HOME 目录表示 dmagent 程序的所在目录)。

3.2.1 AGENT.INI 配置项

本节详细介绍 AGENT.INI 的各配置项。

3.2.1.1 general (通用参数)

3.2.1.1.1 general 参数

参数名	默认值	描述
-----	-----	----

DM8_dmagent 使用手册

id_policy	0	dmagent 的 ID 生成策略。0:使用 MAC 和 IP 生成; 1:使用 IP 生成
center_url	http://192.168.2.10:8080/dem	dem 的 url 地址
ip_list	[]	指定 dmagent 的 ip 地址列表, 主要应用于存在网络地址映射的环境
ip_list_effect_id_generate	false	是否使用 ip_list 生成 dmagent ID
inner_ip		内网 IP
outer_ip		外网 IP
dm_bin_path		JNI 调用的 BIN 路径
repository_path		仓库路径, 指定各工具生成文件的路径
collect_data_by_command	true	是否启用操作系统命令收集服务器资源数据 (仅支持 Linux 系统)。 true:启用; false:关闭
deploy_umask	0077	用于部署完成后修改执行码文件权限, 仅在 Linux 环境下支持

3.2.1.1.2 collect_data_by_command 详解

在 AGENT.INI 中 collect_data_by_command 参数设置为 true 时, Linux 环境下数据监控功能通过操作系统命令实现, 启用数据监控功能前请通过 com.dameng.common.util.system.SystemInfoTest 类功能进行检查。下面详细介绍:

1. SystemInfoTest 类功能介绍:

检测监控收集功能涉及的命令和文件是否可用。

打印命令输出和文件内容。

打印操作系统信息和硬件信息。

2. SystemInfoTest 类使用方法

方法如下：

SystemInfoTest [options]

Options:

-c, --check	检查命令和文件是否可用
-d, --data	打印命令输出和文件内容
-s, --systeminfo	打印操作系统信息和硬件信息
-h, --help	打印帮助信息

3. 使用 check 功能进行检查：

通过 java 命令调用 agent.jar 执行 SystemInfoTest 类的 check 功能，检查监控相关命令及文件是否可以正常运行或读取。若输出结果全为 success 则表示可正常执行监控功能，若出现 fail 则表示检查失败会影响监控功能的正常使用，请咨询系统管理员。

例 假设 agent 部署目录为 /opt/dmagent

```
java -cp /opt/dmagent/lib/*:/opt/dmagent/lib/agent-6.0.1.jar  
com.dameng.common.util.system.SystemInfoTest -c
```

输出结果：

```
checking command df.....success!  
checking command getconf.....success!  
checking command getent.....success!
```


3.2.1.2 ap（JAVA 外部函数代理功能）

3.2.1.2.1 ap 参数

参数名	默认值	描述
ap_enable	true	是否启用外部函数功能。true:启用；false:关闭
ap_port	6363	外部函数端口。使用外部函数时要和 dm.ini 中的 EXTERNAL_JFUN_PORT 保持一致
ap_class_loader_life_time	120	类加载生命周期，单位 s

3.2.1.2.2 配置示例

启用该功能需配置：

```
ap_enable=true
```

```
ap_port 配置与 DM.INI 中的 EXTERNAL_JFUN_PORT 一致
```

3.2.1.3 isql（SQL 命令行工具）

3.2.1.3.1 isql 参数

参数名	默认值	描述
isql_auto_commit	true	是否自动提交
isql_rs_batch_show	true	结果集是否分批显示
isql_rs_batch_show_size	50	结果集分批显示时，每批显示的行数
isql_col_max_width	50	每列最大显示字符数，超过用部分...显示
isql_col_auto_wrap	true	当列字符数超过 isql_col_max_width 时，是否自动换行以显示全部内容

isql_col_escape	true	是否将查询结果中的\t \r \n 特殊字符转为空格
isql_highlight_sql	true	sql 语句是否对关键字进行高亮标注
isql_file_encoding		文件编码

3.2.1.3.2 isql 工具使用示例

isql 工具启用方式:

```
start.bat/start.sh isql agent.ini
```

具体使用命令，在启动 isql 后，输入 help，查看帮助信息，具体如下：

```
C:\dmdbms1\tool\dmagent>start.bat isql agent.ini
Agent-V7.4.1.27499-2024.08.01-DEV
ISQL> help
"help"          - show this usage message
"login"         - login database, alias: "conn"
"logout"        - logout database
"edit"          - edit sql, alias: "ed"
"exit"          - exit this program
"last"          - last execute sql execute again
"..."         - execute sql file
"desc ..."    - get table or view columns description
"explain ..." - get sql execute plan
```

图 3.2 dmagent isql 帮助信息

3.2.1.4 build（编译工具）

build 参数专门提供给达梦公司内部人员使用。

参数名	默认值	描述
build_config_path	build.xml	编译的配置文件路径

3.2.1.5 service（远程调用服务）

3.2.1.5.1 service 参数

参数名	默认值	描述
-----	-----	----

service_enable	true	是否启用 dmagent 远程调用服务功能。true:启用；false:关闭
service_port	6364	服务端口号
service_key	999999	填充当前调用服务默认使用的认证 key（dmagent 作为客户端访问远端）

3.2.1.5.2 配置示例

启用该功能需配置：

```
service_enable =true

service_port=任一没被占用的端口
```

3.2.1.6 gather（搜集 DEM 监控数据）

3.2.1.6.1 gather 参数

每个 dmagent 的 gather 线程收集信息频率由下表中的配置参数控制。

参数名	默认值	描述
gather_enable	true	是否启用 dmagent 信息收集服务功能。true:启用；false:关闭
gather_center_servlet	/dem/dma_agent	dem 信息收集的 servlet 地址
gather_offline_save_local	false	是否启用信息收集通信失败本地保存功能。true:启用；false:关闭
gather_offline_data_directory	gather	本地保存目录
gather_offline_flush_freq	60	通信失败时数据刷新频率

DM8_dmagent 使用手册

gather_offline_send_freq	300	通信失败时数据再次发送频率
gather_connection_pool_size	50	数据库连接池大小
gather_connection_life_time	10	数据库连接生命周期（秒）
gather_mf_info_freq	[60]	主机基本信息收集频率（秒）
gather_mf_stat_freq	[60]	收集主机状态信息的频率（秒）
gather_mf_core_freq	[60]	收集主机 CORE 信息的频率（秒）
gather_mf_group_info_freq	[60]	收集主机组基本信息的频率（秒）
gather_mf_group_stat_freq	[60]	收集主机组状态信息的频率（秒）
gather_process_stat_freq	[60]	收集进程状态信息的频率（秒）
gather_process_group_stat_freq	[60]	收集进程组状态信息的频率（秒）
gather_db_info_freq	[60]	数据库基本信息收集频率（秒）
gather_db_stat_freq	[60]	收集数据库状态信息的频率（秒）
gather_db_deadlock_freq	[60]	收集数据库死锁信息的频率（秒）
gather_db_trxwait_freq	[60]	收集数据库事务等待信息的频率（秒）
gather_db_event_freq	[60]	收集数据库事件信息的频率（秒）
gather_db_danger_event_freq	[60]	收集数据库重要事件或行

DM8_dmagent 使用手册

		为信息的频率
gather_db_session_freq	[60]	收集数据库会话信息的频率（秒）
gather_db_tablespace_freq	[600]	收集数据库表空间信息的频率（秒）
gather_db_user_freq	[600]	收集数据库用户锁定信息的频率（秒）
gather_db_audit_record_freq	[0]	收集数据库审计记录的频率（秒）
gather_db_serverlog_freq	[60]	收集数据库运行日志信息的频率（秒）
gather_db_serverlog_distinct_enable	true	是否对收集的数据库运行日志去重
gather_db_serverlog_infolevel_enable	false	是否收集数据库 INFO 级别的运行日志信息
gather_db_rlog_freq	[60]	收集数据库重做日志信息的频率（秒）
gather_db_backup_freq	[60]	收集数据库备份信息的频率（秒）
gather_db_arch_freq	[60]	收集数据库归档信息的频率（秒）
gather_db_backupset_freq	[60]	收集数据库备份集信息的频率（秒）
gather_db_backupset_check_days	3	收集数据库指定天数内的备份集信息
gather_db_ini_file_freq	[180]	收集数据库配置文件信息的频率（秒）
gather_db_sqllog_file_freq	[60]	收集数据库 SQL 日志文件

DM8_dmagent 使用手册

		信息的频率（秒）
gather_db_sqllog_file_exectime	1000	收集数据库 SQL 日志文件中 SQL 的最小执行时间(毫秒)
gather_db_sqllog_file_buffer	100	收集数据库 SQL 日志文件中 SQL 的缓存大小
gather_db_sqllog_file_param_record	false	是否收集数据库 SQL 日志文件中 SQL 的参数信息
gather_db_sqllog_file_param_stmt_count	10000	收集数据库 SQL 日志文件中 SQL 的参数信息的缓存大小
gather_db_sqllog_file_param_batch_count	1	收集数据库 SQL 日志文件中 SQL 批量绑定参数大小
gather_db_sqlverify_enable	false	是否开启 SQL 审核
gather_db_sqlverify_rule_context		SQL 审核的规则上下文信息
gather_db_sqllog_view_freq	[60]	收集数据库 SQL 动态视图信息的频率（秒）
gather_db_sqllog_view_exectime	1000	收集数据库动态视图中 SQL 的最小执行时间（毫秒）
gather_db_job_history_freq	[60]	收集数据库作业历史信息的频率（秒）
gather_db_index_freq	[60]	收集数据库作业历史信息

		的频率（秒）
gather_db_group_info_freq	[60]	收集数据库集群基本信息的频率（秒）
gather_db_group_dpc_sql_freq	[60]	收集 DPC 集群 SQL 信息的频率（秒）
gather_db_group_dpc_sql_plan_freq	[60]	收集 DPC 集群 SQL 执行计划的频率（秒）
gather_db_group_stat_freq	[60]	收集数据库集群状态信息的频率（秒）
gather_db_group_session_freq	[60]	收集数据库集群会话信息的频率（秒）
gather_db_lock_freq	[60]	收集数据库事务锁信息的频率（秒）

3.2.1.6.2 配置示例

启用该功能需配置：

```
gather_enable=true  
center_url=http://<dem_ip>:<dem_port>/dem  
gather_center_servlet=dem/dma_agent
```

Gather 线程收集信息频率配置格式为[freq(s), year, month, day, hour, minute, second]。freq 表示收集频率，单位为秒； year, month, day, hour, minute, second 表示开始收集时间，该参数可以选择配置，但必须保持顺序。即可以配置[freq(s)], [freq(s), year], [freq(s), year, month], [freq(s), year, month, day, hour], [freq(s), year, month, day, hour, minute], [freq(s), year, month, day, hour, minute, second]这几种情况，month 默认值为 1，day 默认值为 1，hour 默认值为 0，minute 默认值为 0，second 默认值为 0。



注意：

使用前需检查各监控收集功能是否可用。WINDOWS 环境下没有平台环境的差异，都是使用统一的接口，因此不用检查。Linux 环境下不同 CPU 芯片对操作系统的影响不同，因此使用之前需要进行监控功能检查。下面详细介绍 Linux 环境下监控收集检查方法：

在 Linux 环境下 DEM 数据监控功能通过 Linux 系统命令实现，启用 DEM 数据监控功能前请通过 com.dameng.common.util.system.SystemInfoTest 类功能进行检查。

1. SystemInfoTest 类功能介绍：

检测监控收集功能涉及的命令和文件是否可用。

打印命令输出和文件内容。

打印操作系统信息和硬件信息。

2. SystemInfoTest 类使用方法：

```
SystemInfoTest [options]
```

Options:

```
-c, --check      检查命令和文件是否可用
-d, --data       打印命令输出和文件内容
-s, --systeminfo 打印操作系统信息和硬件信息
```


-h, --help 打印帮助信息

3. 使用 check 功能进行检查:

通过 java 命令调用 agent.jar 执行 SystemInfoTest 类的 check 功能, 检查监控相关命令及文件是否可以正常运行或读取。若输出结果全为 success 则表示可正常执行监控功能, 若出现 fail 则表示检查失败会影响监控功能的正常使用, 请咨询系统管理员。

例如, 假设 agent 部署目录为 /opt/dmagent, 输入检查命令。

```
java -cp /opt/dmagent/lib/*:/opt/dmagent/lib/agent-6.0.1.jar  
com.dameng.common.util.system.SystemInfoTest -c
```

输出结果:

```
checking command df.....success!  
checking command getconf.....success!  
checking command getent.....success!  
checking command iostat.....success!  
checking command ls.....success!  
checking command lsblk.....success!  
checking command lscpu.....success!  
checking command taskset.....success!  
checking command uname.....success!  
checking command vmstat.....success!  
checking file /proc .....success!  
checking file /proc/cpuinfo .....success!  
checking file /proc/meminfo .....success!  
checking file /proc/mounts .....success!  
checking file /proc/stat .....success!  
checking file /proc/vmstat .....success!  
checking file /proc/sys/fs/file-nr .....success!  
checking file /proc/53510/cmdline .....success!
```

```

checking file /proc/53510/cwd .....success!
checking file /proc/53510/exe .....success!
checking file /proc/53510/fd .....success!
checking file /proc/53510/io .....success!
checking file /proc/53510/stat .....success!
checking file /proc/53510/status .....success!
checking file /proc/53510/environ .....success!
checking file /proc/53510/statm .....success!
checking file /proc/self/stat .....success!
checking file /proc/uptime .....success!

```

3.2.1.7 upgrade (dmagent 自动升级)

3.2.1.7.1 upgrade 参数

参数名	默认值	描述
upgrade_enable	true	是否启用自动升级 dmagent 功能。true:启用；false:关闭
upgrade_center_servlet	/dem/dma_agent	升级 Servlet 路径
upgrade_version_check_freq	600	dmagent 检查升级频率，单位：秒。设置为 0 表示不检查

3.2.1.7.2 配置示例

若要启动 dmagent 自动升级，须提前配置如下 AGENT.INI 参数。

例 为 dmagent 配置自动升级。其中，center_url 为 dem 的 url 地址。

```

center_url=http://<dem_ip>:<dem_port>/dem  --必配项，下载新版 dmagent 软件的地址
upgrade_enable =true

```

```
upgrade_center_servlet=dem/dma_agent
upgrade_version_check_freq=600
```

3.2.1.8 smoke（冒烟工具）

smoke 参数专门提供给达梦公司内部人员使用。

参数名	默认值	描述
smoke_test_case_path		冒烟测例文件目录
smoke_config_path	smoke.xml	冒烟测试配置文件路径
smoke_terminals	1	冒烟测试并行线程数
smoke_terminate_on_error	true	冒烟测试遇到错误是否终止程序
smoke_online_log	false	冒烟测试日志是否发送至远程数据库

3.2.1.9 tpcc（性能测试）

3.2.1.9.1 tpcc 参数

参数名	默认值	描述
tpcc_step	[create_table,load_data,create_index,run_tpcc]	TPCC 执行步骤，分别为创建表、加载数据、创建索引、执行测试。步骤可选，但必须保证顺序
tpcc_db	dameng	数据库类型
tpcc_driver	dm.jdbc.driver.DmDriver	数据库驱动
tpcc_url	jdbc:dm://localhost:5236	数据库 URL
tpcc_user	SYSDBA	数据库用户名
tpcc_password	sysDBA*00	数据库密码
tpcc_warehouses	100	TPCC 仓库数量

DM8_dmagent 使用手册

tpcc_load_workers	10	TPCC 加载数据工作线程数
tpcc_terminals	5	TPCC 终端数
tpcc_terminal_warehouse _fixed	false	TPCC 终端仓库是否固定。 true:固定; false:不固定
tpcc_run_mins	5	TPCC 运行时间, 单位: 分钟
tpcc_run_trxs_per_termin al	0	TPCC 每个终端执行的事务数
tpcc_limit_trxs_per_min	0	TPCC 每分钟事务数限制
tpcc_new_order_weight	45	TPCC 新订单事务数权重
tpcc_payment_weight	43	TPCC 支付事务数权重
tpcc_order_status_weight	4	TPCC 订单状态事务数权重
tpcc_delivery_weight	4	TPCC 交付事务数权重
tpcc_stock_level_weight	4	TPCC 库存状态事务数权重
tpcc_file_location		TPCC 加载数据文件的路径
tpcc_result_directory_ena ble	false	TPCC 执行信息和执行结果是 否保存。true:保存; false:不 保存
tpcc_os_collector_script		系统信息收集脚本
tpcc_os_collector_interval	0	系统信息收集间隔
tpcc_os_collector_ssshadr		系统信息收集 SSH 地址
tpcc_os_collector_devices		系统信息收集的设备
tpcc_csv_null_value		TPCC 信息 CSV 文件默认空值

tpcc_gui_enable	false	是否启用 TPCC GUI。true:启用; false:关闭
tpcc_gui_interval	1	TPCC GUI 数据点采集时间间隔, 单位: 秒
tpcc_gui_max_data_number	720	TPCC GUI 面板最大容纳数据点个数, 达到最大个数之后保存面板之后重绘

3.2.1.9.2 tpcc 测试工具使用示例

tpcc 工具启用方式:

```
start.bat/start.sh tpcc agent.ini #按 agent.ini 中配置的 tpcc_step 步骤执行

start.bat/start.sh tpcc_table agent.ini #tpcc 创建表

start.bat/start.sh tpcc_load agent.ini #tpcc 装载数据

start.bat/start.sh tpcc_index agent.ini #tpcc 创建索引

start.bat/start.sh tpcc_run agent.ini #tpcc 测试
```

3.2.1.10 批量执行

3.2.1.10.1 command(批量执行脚本)

此处的脚本是指 Linux 的 SHELL 脚本和 Windows 的 bat 脚本。

参数名	默认值	描述
command_item_configs: 用于配置批量执行脚本的信息, 主要包括下述参数:		
id	0	批量执行 id,不可重复

DM8_dmagent 使用手册

dependency_id		此批量执行依赖的其他批量执行 id 集合,格式:id1,id2,...
user	test	Linux 环境下,执行 SHELL 脚本的用户名
password	test	Linux 环境下,执行 SHELL 脚本用户的密码
agent_service_url	rpc://localhost:6364	目标主机的 dmagent 服务地址
working_directory		SHELL 脚本进程的工作目录
timeout	0	执行 command 命令超时时间
commands_file		待执行的命令文件。文件包含一条或多条使用 shell语法或bat语法编写的命令。Windows环境使用bat语法, Linux环境使用SHELL语法
commands		待执行的一条或多条 command 命令。须使用 shell 语法或 bat 语法编写的命令。Windows 环境使用 bat 语法, Linux 环境使用 SHELL 语法

3.2.1.10.2 sql(批量执行 sql)

参数名	默认值	描述
sql_max_fetch_rows	100	查询语句的最大提取行数
sql_col_max_width	50	列的最大显示宽度
sql_col_auto_wrap	true	超过最大显示宽度时是否自动换行
sql_col_escape	true	是否转义特殊列值
sql_file_encoding		SQL 文件编码

sql_item_configs: 用于配置批量执行 SQL 的信息, 主要包括下述参数:

id	0	批量执行 id,不可重复
----	---	--------------

dependency_id		此批量执行依赖的其他批量执行 id 集合，格式:id1,id2,...
user	SYSDBA	数据库登录用户名
password	sysDBA*00	数据库登录密码
url	jdbc:dm://localhost:5236	Jdbc 连接串，格式:jdbc:dm://host:port[?propertyName=propertyValue1][&propertyName2=propertyValue2]...
continue_when_error	false	执行 SQL 语句出错是否继续
timeout	0	执行 SQL 语句的超时时间
sqls_file		待执行的 SQL 脚本。SQL 脚本中包括一条或多条 SQL 语句
sqls		待执行的一条或多条 SQL 语句

3.2.1.10.3 ssh(批量执行 ssh 客户端命令)

专门用于 Linux 环境中。

参数名	默认值	描述
ssh_item_configs: 用于配置批量执行 SSH 客户端命令的信息，主要包括以下参数:		
id	0	批量执行 id，不可重复

dependency_id		此批量执行依赖的其他批量执行 id 集合，格式： id1,id2,...
user	test	Linux 环境下，SSH 连接的用户名
password	test	Linux 环境下，SSH 连接用户的密码
ip	localhost	SSH 连接机器的 IP
port	22	SSH 连接机器的端口号
timeout	0	SSH 连接超时时间
commands_file		待执行的 SSH 脚本文件。SSH 脚本文件包括一条或多条 SSH 命令
commands		待执行的一条或多条 SSH 命令。比如 ll、pwd 等命令

3.2.1.11 sqlverify (SQL 审核)

3.2.1.11.1 sqlverify 参数

参数名	默认值	描述
sql_verify_sql_file_path	无	待审核 SQL 文件路径；文件内容为一个或多个待审核的 SQL 语句
sql_verify_jdbc_url	无	提供 JDBC 连接串，以支持和加强部分规则的审核；连接串包含用户名和密码，故不再设置其参数
sql_verify_config_path	无	用户自定义规则项的文件路径；文件内容是固定格式的 SQL 审核规则，规则以 JSON 串的方式写入文本并读取反序列化
sql_verify_url	无	SQL 审核日志回显路径；命令行端不可见

sql_verify_id	无	SQL 审核执行的标识信息；命令行端不可见
sql_verify_execute_id		

3.2.1.11.2 配置示例

该功能在 DEM 使用过程中对用户是透明的，不需要用户自己进行配置，以下配置只是一个示例。

使用该功能时需配置：

sql_verify_sql_file_path=待审核 SQL 文件路径，如 C:\Users\dameng\Desktop\Test.sql

sql_verify_jdbc_url=JDBC 连接串，如 jdbc:dm://localhost:5236

sql_verify_config_path=用户自定义规则项的文件路径

3.2.1.12 replace（替换）

替换操作的相关参数详情如下：

参数名	默认值	描述
replace_show_detail	false	显示替换的详细信息
replace_remote	true	<p>是否在远程服务上执行替换。</p> <p>replace_remote=false，升级本地单机的目录或文件；</p> <p>replace_remote=true，根据 xxx 升级远端服务器的目录或文件</p>
replace_simple_output	false	<p>本地替换时是否输出简单信息。</p> <p>replace_simple_output=true，替换过程输出简易信息，如进度条</p>

DM8_dmagent 使用手册

		replace_simple_output=false, 替换过程输出详细信息
--	--	---

replace_item_configs: 用于配置替换的信息, 主要包括以下参数:

id	0	批量执行 id, 不可重复
dependency_id		此批量执行依赖的其他批量执行 id 集合, 格式:id1,id2,..
agent_service_url	rpc://localhost:6364	目标主机的 dmagent 服务地址
source_file	空	来源文件 (类型为 zip)
dest_directory	空	目标文件夹路径 (如果 replace_remote=true, 指定的是远端文件夹路径)
backup_enable	true	是否备份被替换的文件
owner		文件所有者 (仅支持 Linux 系统)
group		文件所属组 (仅支持 Linux 系统)
permission		文件权限 (仅支持 Linux 系统)
retain_exist_file_props	true	是否保留已存在文件的属性 (所有者、组、 权限, 仅支持 Linux 系统)
replace_readonly_file_enable	false	是否替换只读文件
user		执行脚本的用户名
password		执行脚本的用户密码
pre_script_path		在替换动作执行之前, 执行此脚本内容
post_script_path		在替换动作执行之后, 执行此脚本内容

下面给出了 replace(替换)的一个配置示例:

```
set service_key=1628114E75763801 //设置环境变量, 新版 agent 强制校验
```

```
start.bat/start.sh replace agent.ini #按 agent.ini 中配置的 replace 执行
```

3.2.1.13 inspect

相关参数详情如下：

参数名	默认值	描述
inspect_output_directory	无	巡检报告保存路径

inspect_databases：用于配置待巡检数据库的信息，主要包括下述参数：

url	jdbc:dm://localhost:5236	JDBC 连接串
user	SYSDBA	数据库登录用户名
password	sysDBA*00	数据库登录密码
backup_directory		备份集搜索路径，多个路径间用分号隔开
agent_service_url	rpc://localhost:6364	目标主机的 dmagent 服务地址
archive_days	30	查询指定日期内的数据库归档文件
backup_days	30	巡查询指定日期内的数据库备份集
backup_check_days	3	校验指定日期内的数据库备份集
dead_lock_days	30	查询指定日期内的数据库死锁信息
instance_log_days	30	查询指定日期内的数据库运行日志信息

3.2.1.14 stack

相关参数详情如下：

参数名	默认值	描述
-----	-----	----

stack_item_configs：用于配置导出堆栈的信息，主要包括下述参数：

id	0	批量执行 id，不可重复
dependency_id		此批量执行依赖的其他批量执行 id 集合,格

DM8_dmagent 使用手册

		式:id1,id2,..
agent_service_url	rpc://localhost:6364	目标主机的 dmagent 服务地址
user	test	执行脚本的用户名
password	test	执行脚本的用户密码
resource		资源，用于标识收集的堆栈信息，格式：实例名(ip:port)
pid		进程号
command		待执行的脚本，支持以'\$pid\$'格式自定义进程号变量

3.2.1.15 log

相关参数详情如下：

参数名	默认值	描述
log_item_configs：用于配置导出运行日志的信息，主要包括下述参数：		
id	0	批量执行 id，不可重复
dependency_id		此批量执行依赖的其他批量执行 id 集合,格式:id1,id2,..
agent_service_url	rpc://localhost:6364	目标主机的 dmagent 服务地址
instance_name		实例名，用于标识收集的运行日志信息
server_log_path	../log	运行日志路径
begin_time		运行日志的开始时间
end_time		运行日志的结束时间

3.2.1.16 index_optimize (索引优化)

相关参数详情如下：

参数名	默认值	描述
index_optimize_sql_file_path		待优化的 SQL 文件 路径
index_optimize_jdbc_url	jdbc:dm://localhost:5236?username =SYSDBA&password=sysDBA*00	JDBC 连接串, 包含 用户名和密码
index_optimize_update_database_statistic	false	是否更新库的统计 信息
index_optimize_update_table_statistic	false	是否更新表的统计 信息

3.2.1.17 dem_client (DEM 客户端工具)

dem-client 工具需要配合 dem-server(以命令行方式部署的 DEM 服务器)使用, 主要功能包含显示系统配置信息、主机信息、数据库信息, 以及导入系统配置信息、数据库信息等功能。

3.2.1.17.1 dem_client 参数

相关参数详情如下：

参数名	默认值	描述
dem_client_config_file_path		配置文件路径

3.2.1.17.2 配置示例

DEMCLIENT 插件支持的具体功能与命令如下：

1) 显示 DEMSERVER 的系统配置信息。

```
start demclient show sysconfig agent.ini
```

2) 导入 DEMSERVER 的系统配置信息。

```
start demclient import sysconfig agent.ini
```

3) 显示 DEMSERVER 监控的主机信息。

```
start demclient show mainframe agent.ini
```

4) 显示 DEMSERVER 监控的数据库信息。

```
start demclient show database agent.ini
```

5) 导入需要监控的数据库信息。

```
start demclient import database agent.ini
```

3.2.2 配置示例

本章节主要介绍以下两种功能的配置方法：

- 部署与监控功能
- 外部函数功能

部署与监控功能

配置部署与监控功能需要 dmagent 与 DEM 进行连接，连接 DEM 需要修改 AGENT.INI

配置文件的 center_url 参数），将连接地址修改为 DEM 的 URL 地址。

示例如下：

```
#general  
  
id_policy = 0    #agent id general policy, 0: mac & ip, 1: ip  
  
center_url = http://192.168.1.60:8080/dem  
  
ip_list = []
```

AGENT.INI 配置文件中其他关于部署与监控功能的参数根据功能需求进行调整，修改完成后启动 dmagent，登录 DEM 后即可进行部署与监控。

外部函数功能

配置外部函数功能需要打开外部函数功能开关，即修改 AGENT.INI 配置文件中的参数 ap_enable 和 ap_port。其中，参数 ap_enable 需设置为 true，参数 ap_port 的值需与 dm.ini 文件中的参数 EXTERNAL_JFUN_PORT 的值保持一致。

示例如下：

```
#ap
ap_enable = true      #whether enable ap plugin
ap_port = 6363        #ap listen tcp/ip port
```

AGENT.INI 配置文件修改完成后启动 dmagent，即可使用 dmagent 的外部函数功能。

3.3 启动与停止

dmagent 存在两种启动模式。

1. 运行 start.bat(sh) 脚本启动 dmagent。
2. 运行 service.bat(sh) 脚本安装 dmagent 服务后，通过系统服务命令或者 service.bat(sh) 脚本启动 dmagent 服务。

3.3.1 start 启动脚本

在 AGENT_HOME 目录下存在启动脚本 start.bat 和 start.sh，运行 start 脚本可以以命令行的方式启动 dmagent。

start.bat 专门在 Windows 平台下使用，start.sh 专门在 Linux (Unix) 平台下使用。

针对不同的平台，start 启动脚本的启动参数略有不同，下面分别介绍。

3.3.1.1 Windows 环境

下面介绍 Windows 环境下 dmagent 启动脚本 start.bat 的使用方法。

语法如下:

start.bat [**<命令参数>**] **<AGENT.INI 路径>**

<命令参数>::=

command|decrypt|deploy|encrypt|inspect|install_service_key|isql|replace|smoke|sql|sqlverify|
|tpcc|tpcc_index|tpcc_load|tpcc_run|tpcc_table|uninstall_service_key

参数介绍:

<agent_ini 路径>: 指定 AGENT.INI 配置文件的路径。

<命令参数>: 运行 dmagent 各种功能, 具体如下所示, 各参数对应的具体用法可以

查看打印的帮助信息:

- command: 批量执行工具, 连接主机执行脚本, 关于 command 工具的相关配

置信息请参考 [3.2.1 AGENT.INI 配置项](#)。

- decrypt: 进行解密操作。
- deploy: 进行集群搭建操作。
- encrypt: 进行加密操作。
- inspect: 生成巡检报告。
- install_service_key: 生成 service.key。
- isql: 连接数据库并打开命令行查询工具 ISQL, 关于 ISQL 工具的相关配置信息请

参考 [3.2.1 AGENT.INI 配置项](#)。

- replace: 替换文件, 关于 replace 工具的相关配置信息请参考 [3.2.1 AGENT.INI](#)

[配置项](#)。

- smoke: 运行冒烟测试, 关于冒烟测试的配置参数请参考 [3.2.1 AGENT.INI 配置项](#)。
- sql: 批量执行工具, 连接数据库执行 SQL, 关于 sql 工具的相关配置信息请参考

[3.2.1 AGENT.INI 配置项](#)。

- sqlverify: SQL 审核, 关于 sqlverify 工具的相关配置信息请参考 [3.2.1 AGENT.INI 配置项](#)。

- tpcc: 运行 TPCC 测试程序, 关于 TPCC 测试程序的相关配置信息请参考 [3.2.1 AGENT.INI 配置项](#), 按照其中的配置参数 tpcc_step 执行相应步骤。

- tpcc_index: 运行 TPCC 测试程序的创建索引步骤。
- tpcc_load: 运行 TPCC 测试程序的装载数据步骤。
- tpcc_run: 运行 TPCC 测试程序的测试步骤。
- tpcc_table: 运行 TPCC 测试程序的建表步骤。
- uninstall_service_key: 删除 service.key。

通过 start.bat 脚本启动 dmagent, 并运行 dmagent 各种功能, 示例如下:

1. 启动 dmagent。

```
D:\dmagent>start.bat agent.ini
Agent-V7.5.1.30535-2024.11.15-DEV
System is ready.
```

2. 启动 dmagent 命令行查询工具 isql。

```
D:\dmagent>start.bat isql agent.ini
Agent-V7.5.1.30535-2024.11.15-DEV
ISQL>
```

3. 运行 TPCC 功能。

```
D:\dmagent>start.bat tpcc agent.ini
Agent-V7.5.1.30535-2024.11.15-DEV

Worker 000: Loading ITEM
Worker 001: Loading Warehouse 1
Worker 002: Loading Warehouse 2
Worker 003: Loading Warehouse 3
Worker 004: Loading Warehouse 4
Worker 005: Loading Warehouse 5
Worker 006: Loading Warehouse 6
Worker 007: Loading Warehouse 7
Worker 008: Loading Warehouse 8
Worker 009: Loading Warehouse 9
Worker 000: Loading ITEM done
Worker 000: Loading Warehouse 10
```

4. 打印 dmagent 帮助信息。

```
D:\dmagent>start.bat
Agent-V7.5.1.30535-2024.11.15-DEV
Usage:
start plugin ...
start agent.ini

plugin
- "command" - execute commands
- "decrypt" - decrypt ciphertext
- "deploy" - build DM clusters
- "encrypt" - encrypt plaintext
- "inspect" - generate inspect report
- "install_service_key" - install service key file to agent
- "isql" - start sql command
- "replace" - replace files
- "smoke" - run smoke test
- "sql" - execute sqls
- "sqlverify" - execute sql verify
- "tpcc" - tpcc, run steps configed in agent.ini
- "tpcc_index" - tpcc, run create table index
- "tpcc_load" - tpcc, run load data
- "tpcc_run" - tpcc, run test
- "tpcc_table" - tpcc, run create table
- "uninstall_service_key" - uninstall service key file from agent
```

3.3.1.2 Linux(Unix)环境

下面介绍 Linux (Unix) 环境下 dmagent 启动脚本 start.sh 的使用方法。

语法如下：

```
./start.sh [-d][<命令参数>] <AGENT.INI 路径>
```

或

```
./start.sh [--list | -l]
```

<命令参数>::=

```
command|decrypt|deploy|encrypt|inspect|install_service_key|isql|replace|smoke|sqlsqlverify|
|tpcc|tpcc_index|tpcc_load|tpcc_run|tpcc_table|uninstall_service_key
```

参数介绍：

--list | -l: 打印正在运行的 dmagent 进程相关信息。

-d: 后台启动 dmagent 进程。

<命令参数>、<AGENT.INI 路径>: 请参考 [3.3.1.1 Windows 环境](#)。

通过 start.sh 脚本启动 dmagent，并运行 dmagent 各种功能的示例如下：

1. 启动 dmagent。

```
[root@localhost dmagent]# ./start.sh /opt/dmagent/agent.ini
Agent-V5.0.1.5667-2021.02.08-DEV starting...
```

2. 后台启动 dmagent。

```
[root@localhost dmagent]# ./start.sh -d /opt/dmagent/agent.ini
Starting dmagent...
.....
The dmagent(pid: 29307) started successfully.
[root@localhost dmagent]#
```

3. 打印正在运行 dmagent 进程。

```
[root@localhost dmagent]# ./start.sh --list
dmagent process list:
root      29307      1 99 08:55 pts/0    00:05:05 java -Xms64m -Xmx2048m -cp ./opt/dmagent/lib/*
```

4. 启动 dmagent 命令行查询工具 isql。

```
[test@test208 dmagent]$ ./start.sh isql ./agent.ini
Tip: if using the deployment feature, it is recommended to start dmagent with the '-d' parameter.
Agent-V5.0.1.5892-2021.03.08-DEV starting...
ISQL>
```

5. 运行 TPCC 功能。

```
[test@test208 dmagent]$ ./start.sh tpcc agent.ini
Tip: if using the deployment feature, it is recommended to start dmagent with the '-d' parameter.
Agent-V5.0.1.5892-2021.03.08-DEV starting...

Worker 000: Loading ITEM
Worker 001: Loading Warehouse      1
Worker 002: Loading Warehouse      2
Worker 003: Loading Warehouse      3
Worker 004: Loading Warehouse      4
Worker 005: Loading Warehouse      5
Worker 006: Loading Warehouse      6
Worker 007: Loading Warehouse      7
Worker 008: Loading Warehouse      8
Worker 009: Loading Warehouse      9
```

6. 打印 dmagent 帮助。

```
[test@test114 dmagent-7.4.2]$ ./start.sh
tip: if using the deployment feature, it is recommended to start dmagent with the '-d' parameter.
Agent-V7.5.1.30535-2024.11.15-DEV
Usage:
start plugin ...
start [-d] agent.ini
start (--list | -l)

-d                - start agent in the background
--list | -l      - show agent process list
plugin
|- "command"      - execute commands
|- "decrypt"      - decrypt ciphertext
|- "deploy"       - build DM clusters
|- "encrypt"      - encrypt plaintext
|- "inspect"      - generate inspect report
|- "install_service_key" - install service key file to agent
|- "isql"         - start sql command
|- "replace"      - replace files
|- "smoke"        - run smoke test
|- "sql"          - execute sqls
|- "sqlverify"    - execute sql verify
|- "tpcc"         - tpcc, run steps configed in agent.ini
|- "tpcc_index"   - tpcc, run create table index
|- "tpcc_load"    - tpcc, run load data
|- "tpcc_run"     - tpcc, run test
|- "tpcc_table"   - tpcc, run create table
|- "uninstall_service_key" - uninstall service key file from agent
```

3.3.2 服务

在 AGENT_HOME 目录下存在 service 脚本 service.bat 和 service.sh。service.bat 专门在 Windows 平台下使用，service.sh 专门在 Linux（Unix）平台下使用。

运行 service 脚本可以注册、删除、启动、停止 dmagent 的服务，dmagent 的默认服务名为 DmAgentService。

service 脚本使用方法。

语法如下：

SERVICE.BAT {INSTALL|DELETE|START|STOP|STATUS|HELP}

或

SERVICE.SH {INSTALL|DELETE|START|STOP|STATUS|HELP}

参数介绍:

INSTALL: 注册。

DELETE: 删除。

START: 启动。

STOP: 停止。

STATUS: 查看状态。

HELP: 查看帮助信息。

3.3.2.1 Windows 环境

下面介绍 Windows 环境下 service 脚本 service.bat 的使用方法。

1. 注册服务。

```
d:\dmagent>service.bat install
input agent home [d:\dmagent] :
input agent.ini path [d:\dmagent\agent.ini] :
Installation the service DmAgentService completed.
```

2. 删除服务。

```
d:\dmagent>service.bat delete
Delete the service DmAgentService completed.
```

3. 启动服务。

```
d:\dmagent>service.bat start
DmAgentService 服务正在启动 .
DmAgentService 服务已经启动成功。
```

4. 停止服务。

```
d:\dmagent>service.bat stop
DmAgentService 服务正在停止。
DmAgentService 服务已成功停止。
```

5. 查看服务状态。

```
d:\dmagent>service.bat status
The service 'DmAgentService' is running.
```

6. 打印 service 脚本帮助信息。

```
d:\dmagent>service.bat help
Usage: service.bat { start | stop | delete | status | restart | install }
service.bat start          #start agent service
service.bat stop           #stop agent service
service.bat delete         #delete agent service
service.bat restart        #restart agent service
service.bat status         #show agent service's status
service.bat install        #install agent service
```



注意：操作 dmagent 服务(DmAgentService)需要管理员权限。

3.3.2.2 Linux(Unix)环境

下面介绍 Linux(Unix)环境下 service 脚本 service.sh 的使用方法。

1. 注册服务。

```
[root@localhost dmagent]# ./service.sh install
input agent home [/opt/dmagent] :
input agent.ini path [/opt/dmagent/agent.ini] :
Installation the service DmAgentService completed.
[root@localhost dmagent]# █
```

2. 删除服务。

```
[root@localhost dmagent]# ./service.sh delete
Redirecting to /bin/systemctl stop DmAgentService.service
Delete the service DmAgentService completed.
[root@localhost dmagent]# █
```

3. 启动服务。

```
[root@localhost dmagent]# ./service.sh start
Redirecting to /bin/systemctl start DmAgentService.service
[root@localhost dmagent]#
```

4. 停止服务。

```
[root@localhost dmagent]# ./service.sh stop
Redirecting to /bin/systemctl stop DmAgentService.service
[root@localhost dmagent]#
```

5. 查看服务状态。

```
[root@localhost dmagent]# ./service.sh status
Redirecting to /bin/systemctl status DmAgentService.service
● DmAgentService.service - Dm Agent Service
   Loaded: loaded (/usr/lib/systemd/system/DmAgentService.service; enabled; vendor preset
   Active: active (running) since 二 2021-02-09 09:18:16 CST; 33s ago
   Process: 31149 ExecStart=/opt/dmagent/service/DmAgentService start (code=exited, status
   Main PID: 31195 (java)
   Tasks: 44
   CGroup: /system.slice/DmAgentService.service
           └─31195 java -Xms64m -Xmx2048m -cp ./opt/dmagent/lib/* -Ddameng.log.file=/opt

2月 09 09:18:11 localhost.localdomain systemd[1]: Starting Dm Agent Service...
2月 09 09:18:16 localhost.localdomain DmAgentService[31149]: Starting dmagent[ 确定 ]
2月 09 09:18:16 localhost.localdomain systemd[1]: Started Dm Agent Service.
[root@localhost dmagent]#
```

6. 打印 service 脚本帮助信息。

```
[root@localhost dmagent]# ./service.sh help
Usage: service.sh { start | stop | delete | status | restart | install }
       service.sh start           #start agent service
       service.sh stop            #stop agent service
       service.sh delete          #delete agent service
       service.sh restart         #restart agent service
       service.sh status          #show agent service's status
       service.sh install         #install agent service

[root@localhost dmagent]#
```



注意:

操作 dmagent 服务(DmAgentService)需要管理员权限。

咨询热线：400-991-6599

技术支持：dmtech@dameng.com

官网网址：www.dameng.com



武汉达梦数据库股份有限公司
Wuhan Dameng Database Co.,Ltd.

地址：武汉市东湖新技术开发区高新大道999号未来科技大厦C3栋16—19层

16th-19th Floor, Future Tech Building C3, No.999 Gaoxin Road, Donghu New Tech Development Zone,Wuhan,Hubei Province,China

电话：(+86) 027-87588000 传真：(+86) 027-87588810
