

| 达梦技术手册

DM 企业管理器 DEM

Service manual of DM_DEM



前言

概述

本文档主要介绍 DM 企业管理器 DEM 的系统特性、基本概念、实现原理、主要功能，以及如何搭建并使用等。

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，右边为相对应的语法描述。 |
| | 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 引言 | 1 |
| 2 DEM 概述 | 2 |
| 2.1 系统架构 | 2 |
| 2.2 系统特性 | 4 |
| 2.3 DEM 主界面 | 5 |
| 3 DEM 部署 | 7 |
| 3.1 DEM 部署 | 7 |
| 3.1.1 环境准备 | 7 |
| 3.1.2 DEM 部署步骤 | 7 |
| 3.1.3 db.xml 配置参数说明 | 9 |
| 3.1.4 高可用集群搭建 | 10 |
| 3.2 DMAGENT 部署 | 11 |
| 3.2.1 dmagent 部署步骤 | 11 |
| 4 配置参数说明 | 13 |
| 5 用户管理 | 18 |
| 5.1 角色与权限 | 18 |
| 5.2 创建用户 | 18 |
| 5.3 用户启用与禁用 | 19 |
| 6 客户端工具 | 21 |
| 6.1 对象管理 | 21 |
| 6.1.1 系统结构 | 21 |
| 6.1.2 实现原理 | 21 |
| 6.1.3 功能介绍 | 23 |
| 6.2 SQL 编辑器 | 23 |

| | | |
|--------|----------------|----|
| 6.2.1 | 系统结构..... | 23 |
| 6.2.2 | 实现原理..... | 24 |
| 6.2.3 | 功能介绍..... | 24 |
| 6.3 | SQL 调试器 | 24 |
| 6.3.1 | 系统结构..... | 24 |
| 6.3.2 | 实现原理..... | 24 |
| 6.3.3 | 功能介绍..... | 25 |
| 6.4 | 集群管理 | 25 |
| 6.4.1 | 系统结构..... | 25 |
| 6.4.2 | 实现原理..... | 26 |
| 6.4.3 | 功能介绍..... | 26 |
| 6.4 | 数据迁移 | 47 |
| 6.5.1 | 系统结构..... | 48 |
| 6.5.2 | 实现原理..... | 48 |
| 6.5.3 | 功能介绍..... | 48 |
| 6.6 | PROXY 管理 | 50 |
| 6.7 | 批量执行 | 50 |
| 6.7.1 | 系统结构..... | 50 |
| 6.7.2 | 实现原理..... | 51 |
| 6.7.3 | 功能介绍..... | 51 |
| 6.8 | 升级管理 | 51 |
| 6.8.1 | 系统结构..... | 52 |
| 6.8.2 | 实现原理..... | 52 |
| 6.8.3 | 功能介绍..... | 53 |
| 6.9 | 迁移评估 | 57 |
| 6.9.1 | 系统结构..... | 57 |
| 6.9.2 | 实现原理..... | 58 |
| 6.9.3 | 功能介绍..... | 58 |
| 6.10 | 联机巡检 | 72 |
| 6.10.1 | 系统结构..... | 73 |

| | | |
|--------|-------------------|-----|
| 6.10.2 | 实现原理..... | 73 |
| 6.10.3 | 功能介绍..... | 74 |
| 6.11 | 联机大盘 | 81 |
| 6.11.1 | 系统结构 | 82 |
| 6.11.2 | 实现原理 | 82 |
| 6.11.3 | 功能介绍 | 83 |
| 7 | 智能运维 | 95 |
| 7.1 | DEM 监控 | 95 |
| 7.1.1 | 系统结构..... | 95 |
| 7.1.2 | 实现原理..... | 95 |
| 7.1.3 | 主机监控..... | 96 |
| 7.1.4 | 数据库监控..... | 113 |
| 7.2 | DEM 告警 | 142 |
| 7.2.1 | 系统结构..... | 143 |
| 7.2.2 | 实现原理..... | 143 |
| 7.2.3 | 创建告警配置..... | 144 |
| 7.2.4 | 查看告警状态和历史 | 147 |
| 7.2.5 | 告警通知..... | 148 |
| 7.2.6 | 停止告警..... | 150 |
| 7.3 | DEM 巡检 | 150 |
| 7.3.1 | 系统结构..... | 151 |
| 7.3.2 | 实现原理..... | 151 |
| 7.3.3 | 创建巡检配置..... | 152 |
| 7.3.4 | 查看巡检历史和巡检报告 | 154 |
| 7.3.5 | 停止巡检..... | 155 |
| 7.4 | DEM 备份 | 156 |
| 7.4.1 | 系统结构..... | 156 |
| 7.4.2 | 实现原理..... | 156 |
| 7.4.3 | 创建备份调度..... | 157 |

| | | |
|----------|-----------------------------|------------|
| 7.4.4 | 查看备份状态和历史 | 159 |
| 7.4.5 | 停止备份 | 160 |
| 8 | 代理 DMAGENT | 161 |
| 8.1 | 安装部署 | 161 |
| 8.1.1 | 安装说明 | 161 |
| 8.1.2 | 安装步骤 | 161 |
| 8.2 | 目录结构 | 161 |
| 8.3 | 配置文件说明 | 162 |
| 8.4 | 功能介绍 | 166 |
| 8.4.1 | Java 外部函数的代理功能 | 167 |
| 8.4.2 | 收集 DEM 监控数据功能 | 167 |
| 8.4.3 | 远程调用服务功能 | 170 |
| 8.4.4 | SQL 命令行工具 | 170 |
| 8.4.5 | TPCC 测试工具 | 170 |
| 8.4.6 | build 工具 | 171 |
| 8.4.7 | smoke 工具 | 171 |
| 8.5 | DMAGENT 升级 | 171 |
| 8.5.1 | 需要升级的情景 | 171 |
| 8.5.2 | 升级准备 | 172 |
| 8.5.3 | 升级时机 | 172 |
| 8.5.4 | 升级步骤 | 172 |
| 9 | 接口 | 174 |
| 9.1 | 基于 HTTP 协议开发的接口 | 174 |
| 9.1.1 | 管理 DEM 用户的接口 | 175 |
| 9.1.2 | 管理数据库连接的接口 | 183 |
| 9.1.3 | 管理 DEM 告警信息的接口 | 190 |
| 9.1.4 | 管理 prometheus 监控数据的接口 | 196 |
| 9.2 | 基于 UDP 协议开发的接口 | 199 |
| 9.2.1 | SNMP AGENT | 199 |

| | |
|------------------------------------|------------|
| 10 自动登录..... | 209 |
| 10.1 参数说明 | 209 |
| 10.2 参数配置 | 210 |
| 10.2.1 dem_params | 210 |
| 10.2.2 node_params | 210 |
| 10.2.3 panel_params..... | 210 |
| 10.3 参数传递方式..... | 211 |
| 10.4 自动登录参数加密..... | 212 |
| 10.4.1 示例 | 212 |
| 11 DEM 审计..... | 214 |
| 12 版本信息..... | 214 |
| 13 DEMSDK 文件 | 216 |
| 13.1 接口/类文件 | 216 |
| 13.1.1 INotify | 217 |
| 13.1.2 IPhoneNotify | 217 |
| 13.1.3 Notify..... | 219 |
| 13.1.4 NotifyUser..... | 220 |
| 13.1.5 IEncryptEngine | 222 |
| 13.1.6 DefaultEncryptEngine | 222 |
| 13.2 应用举例..... | 223 |
| 13.2.1 db.xml 相关密码字段加密..... | 223 |
| 13.2.2 自定义推送 | 225 |

1 引言

DM 企业管理器的英文全称 DM Enterprise Manager，简称 DEM。

DEM 提供一个通过 Web 界面来监控、管理并维护 DM 数据库的集中式管理平台，可以从任何可以访问 Web 应用的位置通过 DEM 来对 DM 数据库进行各种管理和监控。

DEM 不仅包含了传统桌面工具、管理工具 Manager 和数据迁移工具 DTS 等的功能，还提供了集群部署、监控和告警的功能。

本文主要介绍 DEM 的功能、概念、实现原理以及使用方法等。

2 DEM 概述

DEM 以 Web 应用的方式提供数据库管理、监控和维护的功能，实现远程管理和监控数据库实例，并且不局限于单个数据库实例的管理和监控，也能够对数据库集群（DM MPP、DMDSC 以及数据守护）进行管理和监控。

2.1 系统架构

DEM 由以下几个部分组成：

- ✚ DEM 服务器
- ✚ DEM 存储数据库
- ✚ 要管理和监控的数据库实例
- ✚ 数据库代理服务（dmagent）

其中 DEM 服务器为 DEM 应用服务器，负责处理客户端工具功能逻辑并存储 dmagent 收集到的数据到 DEM 存储数据库，同时展示数据给客户端。

DEM 存储数据库存储 DEM 的元数据和 dmagent 收集到的监控数据。

数据库代理服务 dmagent 为部署在远程机器上的代理，DEM 通过 dmagent 访问远程主机，同时 dmagent 收集监控信息发送给 DEM。

系统架构图如下：

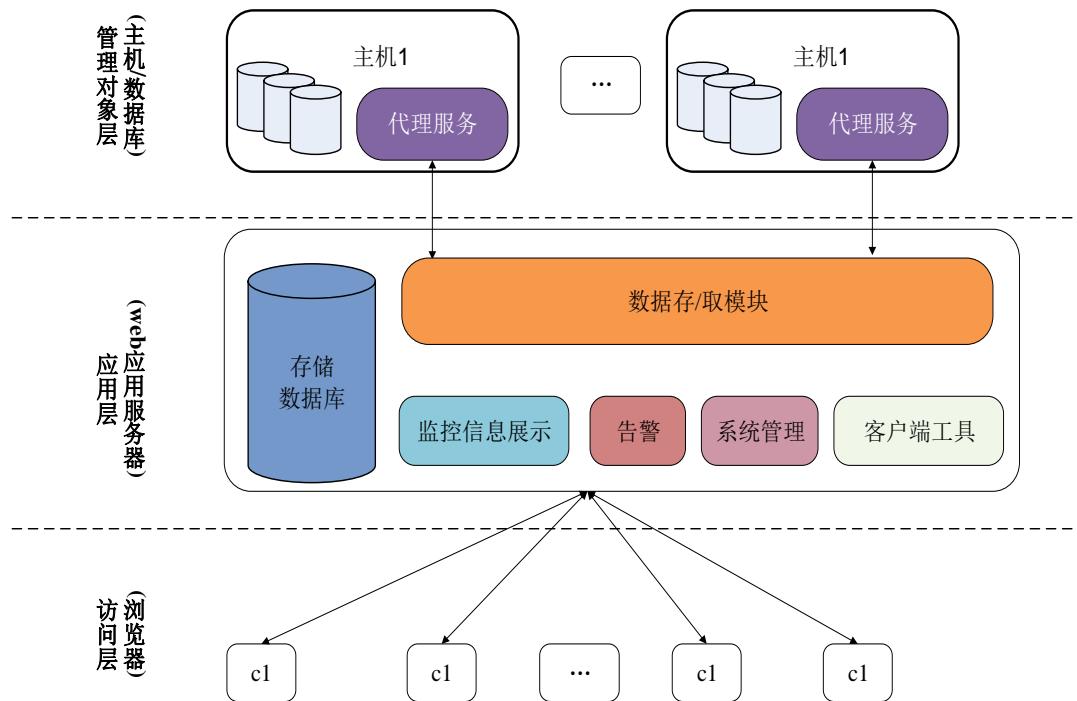


图 2.1 DEM 系统架构图

1. 管理对象层

管理对象层由所需监控的对象组成，包括需要监控的主机、主机上的数据库、以及运行在主机上的一个 dmagent。每个运行在主机上的 dmagent 负责收集自己的主机以及主机上运行的所有数据库的运行数据信息。DEM 通过 dmagent 访问和操作主机以及主机上的数据库实例。

2. 应用层

应用处理模块包括存储数据库、数据存/取模块、监控信息展示模块、告警模块、系统管理模块以及客户端工具六大模块。

- 存储数据库

存储数据库主要完成监控数据的存储。

- 数据存/取模块

数据存/取模块负责将各个主机上的代理服务发送过来的监控数据存入数据库，以及从数据库中检索数据给其他模块提供支持。

- 监控信息展示模块

监控信息展示模块负责组织整理监控信息，以表格或图形等多种便于用户查看的样式提供信息的展示，此外还对同属于一个集群系统的数据库进行分组管理和数据统计。

- 告警模块

告警模块负责提供告警策略的配置、对异常状态的检测以及告警信息的发布。

- 系统管理模块

系统管理模块负责对系统的统一管理，包括各种监控频率的控制、用户管理、权限管理、安全审计以及日志记录等。

- 客户端工具

客户端工具包括管理工具 Manager、数据迁移工具以及集群的部署工具等工具。

3. 访问层

访问层为终端用户的远程系统访问，系统采用了 B/S 架构，用户的访问主要通过基于 http 协议的浏览器请求来完成。

2.2 系统特性

DEM 的系统特性表现为五个方面：集中式管理、功能全面、高度可扩展、主动监控以及安全可靠。

1. 集中式管理

只需部署一套 DEM 服务器，用户就可以在任何地点通过网络访问 DEM 服务器，实现相应功能。

2. 功能全面

DEM 不仅提供了基本的数据库对象管理功能、数据迁移功能以及数据库集群部署功能，同时还提供了对主机和数据库的监控和告警功能。

3. 高度可扩展

DEM 按模块划分为数据库工具、监控和告警、系统管理 3 个模块，从模块上极易扩展。另外，数据库工具和监控告警模块可以管理和监控任意类型的数据库实例，无论是单实例还是集群实例，都可以按统一方式管理，同时在管理和监控的数据库实例数量上也极易扩展，可以轻松地从一个实例扩展到一千个实例。

4. 主动监控

DEM 通过 dmagent 定时收集数据库实例的信息，存储数据库实例的历史数据，提供分析依据，并能及时发现问题且发送告警通知。

5. 安全可靠

DEM 本身提供审计功能，可以查看详细的审计信息。

2.3 DEM 主界面

DEM 主界面如下图所示：

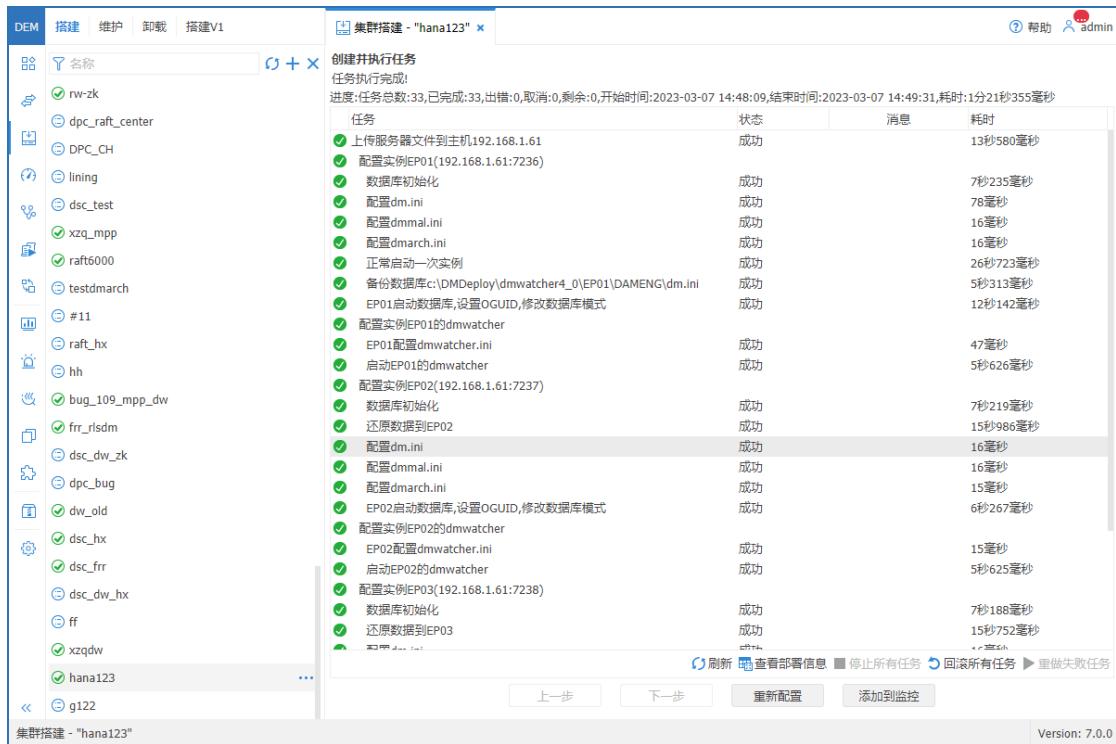


图 2.2 DEM 主界面

DEM 主界面可以分为三部分，具体划分方式见下图 2.3。左侧导航栏展示了 DEM 提供的全部功能；中间部分显示功能模块中存在的对象列表，名称前的图标展示了其当前的状态；右侧显示 DEM 的具体功能模块页面，进行相关功能的信息展示。用户可以通过点击左侧导航栏进行功能切换。

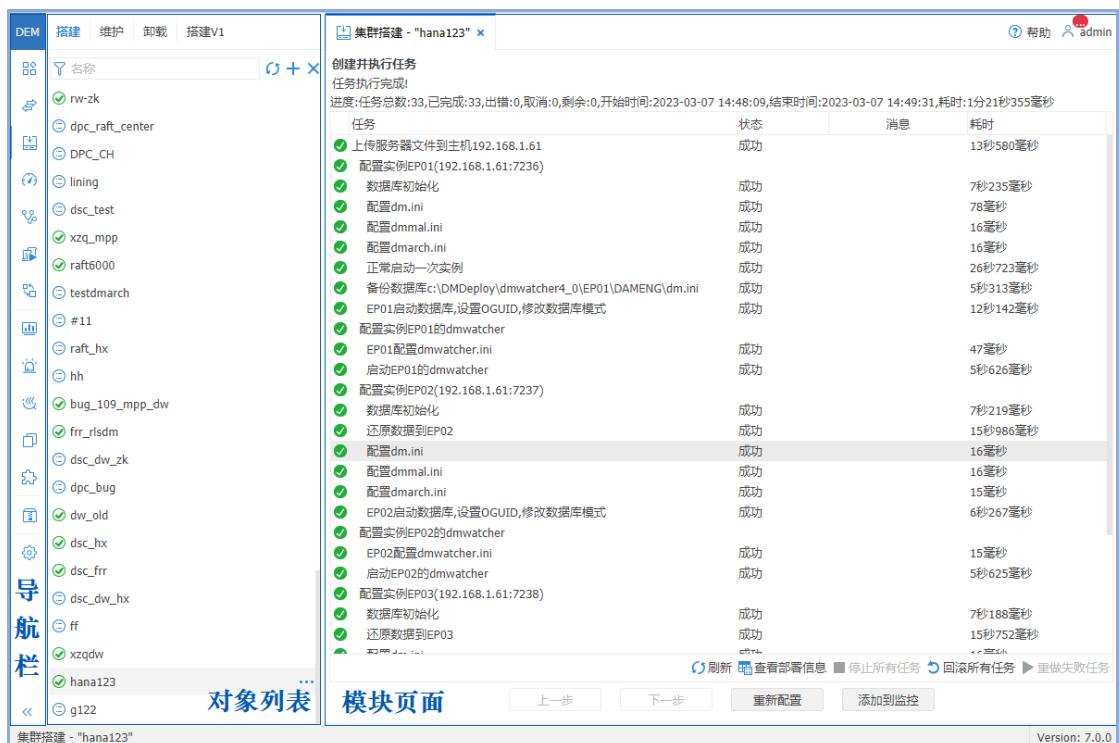


图 2.3 DEM 界面划分

DEM 所提供的功能如下：

- 对象管理
- 数据迁移
- 集群管理
- 联机大盘
- 联机巡检
- 批量执行
- 升级管理
- 资源监控
- 告警配置
- 自动巡检
- 备份调度
- 自定义监控
- 资源包
- 系统管理

DEM 提供的具体功能将在下面章节进行详细介绍。

3 DEM 部署

DEM 与达梦数据库代理（简称 dmagent）需同时部署。dmagent 是 DEM 部署在远程主机上的代理，只有在主机上启动了 dmagent，达梦企业管理平台软件才可以访问远程主机，并监控远程主机的相关信息。

下面将分别介绍 DEM 和 dmagent 的部署步骤。

3.1 DEM 部署

3.1.1 环境准备

DEM 部署需要准备：

- ✚ DEM WAR 包
- ✚ Tomcat
- ✚ JAVA 1.8
- ✚ 达梦数据库

其中 DEM WAR 包在达梦安装目录下 /web/dem.war。

Tomcat 需要自己准备，与要部署的机器版本匹配即可。

需要配置 JAVA 环境，JAVA 版本必须为 JAVA 1.8。

达梦数据库为 DEM 的后台库，故需保证 DEM 所在机器能访问达梦数据库。达梦数据库安装及实例创建详见《DM8 安装手册》。

3.1.2 DEM 部署步骤

1. 修改 DEM 后台数据库 DM.INI 参数配置，推荐配置：

```
MEMORY_POOL      = 200
BUFFER          = 1000
KEEP            = 64
SORT_BUF_SIZE   = 50
```

2. 在 DEM 后台库中执行 SQL 脚本 dem_init.sql。

dem_init.sql 路径为达梦数据库安装目录下的/web/dem_init.sql, 此 SQL 脚本编码为 UTF-8。



如果使用 **disql** 执行 SQL 脚本, 请设置 **set CHAR_CODE UTF8**。

注意:

```
服务器[localhost:5237]:处于普通打开状态
登录使用时间: 256.165<毫秒>
disql v8
SQL> set CHAR_CODE UTF8
SQL> start e:/dmdb/web/dem_init.sql
```

图 3.1 执行 SQL 脚本 dem_init.sql

3. 配置 tomcat

- 1) 在 conf/server.xml 中<Connector port="8080" protocol="HTTP/1.1" ... 追加属性字段 maxPostSize="-1";
- 2) 修改 jvm 启动参数;

Linux:

```
bin/catalina.sh -> JAVA_OPTS="-server -Djava.library.path=/opt/dmdbms/bin"
```

Windows:

```
bin/catalina.bat-> set java_opts=-server -Djava.library.path=c:\dmdbms\bin
```

- 3) 将 dem.war 放到 tomcat 的 webapps 目录下;
- 4) 启动 tomcat (必须先启动 tomcat, 才会解压缩 war 包, 当然也可以直接在 dem.war 中配置, 该方式则无需启动 tomcat)。

4. 修改 DEM 连接后台数据库配置文件 db.xml

配置 DEM 后台数据库的连接信息: ip、port、用户名、密码、连接池大小以及 SSL 登录信息等, 在文件 WEB-INF/db.xml (在 dem.war 中, 解压后在 tomcat 的路径 webapps/dem/WEB-INF/db.xml 下) 中配置。

若要以 SSL 安全方式连接后台数据库, 则需配置 SSLLDir 和 SSLPassword:

默认在 WEB-INF/sslDir 目录中存有密钥对, WEB-INF/db.xml 配置客户端连接使用的密钥文件路径 (SSLLDir) 为 WEB-INF/sslDir/client_ssl/SYSDBA, 密码 (SSLPassword) 为空, 对应 WEB-INF/db.xml 中配置的登录用户 SYSDBA。

```
<?xml version="1.0" encoding="UTF-8"?>

<ConnectPool>
```

```

<Server>localhost</Server>

<Port>5236</Port>

<User>SYSDBA</User>

<Password>SYSDBA</Password>

<InitPoolSize>5</InitPoolSize>

<CorePoolSize>10</CorePoolSize>

<MaxPoolSize>50</MaxPoolSize>

<KeepAliveTime>60</KeepAliveTime>

<DbDriver></DbDriver>

<DbTestStatement>select 1</DbTestStatement>

<SSLDir>.../sslDir/client_ssl/SYSDBA</SSLDir>

<SSLPassword></SSLPassword>

</ConnectPool>

```

db.xml 详细的配置参数参见 [3.1.3 db.xml 配置参数说明](#)。

5. 启动/重启 tomcat

如果步骤 3 中启动了 tomcat，则需重启 tomcat，否则启动 tomcat。

6. 登录验证

假设部署 DEM 的机器 IP 为 192.168.0.104，则 DEM 访问地址:
址: <http://192.168.0.104:8080/dem/>。

默认用户名和密码: admin/888888。

3.1.3 db.xml 配置参数说明

db.xml 配置参数包括：

表 3.1 db.xml 配置参数说明

| 参数名 | 说明 |
|----------|--|
| PoolType | 连接池类型, 1 :druid 2:达梦连接池, 默认不配置使用 druid |
| Dbtype | 数据库类型。dm8, dm, 已移除, 不需要配置 |
| Server | 数据库服务器地址 |
| Port | 数据库端口 |

| | |
|--------------------|--|
| DbName | 数据库名, 已移除 |
| User | 数据库登录用户名 |
| Password | 数据库登录密码。通过属性 engine 设置自定义加解密类, 可配置加密的密码。自定义加解密类参见 demsdk。使用 demsdk 自带的加解密类, 配置如下: <Password engine="com.dameng.dem.server.util.DefaultEncryptEngine">A72125717E5EF759 </Password> |
| InitPoolSize | 连接池初始连接数。缺省值 5 |
| CorePoolSize | 允许的最小空闲连接数。缺省值 10 |
| MaxPoolSize | 允许的最大连接数。缺省值 50 |
| KeepAliveTime | 空闲连接存活时间, PoolType=2 时有用, 缺省值 60 |
| WaitConnectionTime | 最大建立连接等待时间。 |
| DbDriver | Dm jdbc 驱动路径, 缺省为 WEB-INF/lib/ |
| DbTestStatement | 测试 SQL。缺省为 select 1 |
| SSLDir | 配置 ssl 的目录, 可设置相对路径 (相对于 db.xml) 或绝对路径 |
| SSLPASSWORD | 配置 ssl 的密码, 可设置加解密引擎, 同 Password 参数 |

3.1.4 高可用集群搭建

DEM 作为一种监控运维工具, 需要长时间且不间断地收集数据和对外提供服务, 有高可用的需求。但 DEM 作为一个 Web 应用程序, 自身无法提供高可用的功能, 故需通过搭建高可用的 Web 应用服务器集群来配合实现 DEM 的高可用。

以下为常见的几种高可用 Web 应用服务器集群搭建方式:

1. Tomcat + Nginx
2. Tomcat + Keepalived
3. Tomcat + Nginx + Keepalived

具体用于实现 DEM 高可用的高可用 Web 应用服务器集群搭建方式需要根据真实的系统环境而定, 详情请咨询系统管理员。

3.2 dmagent 部署

在要进行监控和管理的机器或数据库所在机器上部署 dmagent。

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。

3.2.1 dmagent 部署步骤

dmagent 部署步骤如下:

1. 获取 dmagent

可以通过 2 种方式获取 dmagent:

- 1) 达梦数据库安装目录的 tool 下存有 dmagent;
- 2) 登录部署好的 DEM, 在监控和告警模块下单击主机, 打开主机面板, 在工具栏选择下载代理, 下载 dmagent。

拷贝 dmagent 到要部署的机器上。

2. 修改 dmagent 配置

配置 dmagent 目录下的 agent.ini:

```
center.url=http://192.168.0.104:8080/dem      #DEM 访问地址
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

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. 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。具体内容可以参考《DM8_dmagent 使用手册》。

5. 验证 DEM 与 dmagent 是否连通

登录 DEM 选择监控和告警模块的“主机”，打开主机面板，检查是否可以看到自己部
署的 dmagent。

如果 dmagent 正常启动，但在 DEM 中未看到，则检查以下事项：

- 1) dmagent 和 DEM 所运行的机器时间是否一致，最多不超过 min (5 分钟，主机失
效时间)（该值由参数 “mf_invalid_time” 配置，详见[第 4 章 配置参数说明](#)）。
- 2) dmagent 所运行机器是否可以访问 DEM 所运行机器（检查是否可以 ping 通）。

6. 配置 dmagent 内外网信息

登录 DEM 选择监控和告警模块的“主机”，打开主机面板，找到自己部署的 dmagent，
在最右侧“操作”栏，打开下拉列表，选择“配置网络”，打开配置网络对话框，配置内外
网信息。

其中，一般 DEM 访问 dmagent 用外网，dmagent 之间或 dmagent 所在机器的数据
库之间通讯用内网。

4 配置参数说明

DEM 可以通过一些配置参数来设置功能和性能选项，包括客户端工具配置、监控告警配置、监控频率配置、系统邮箱配置、自定义推送配置以及安全配置。

具体如下（管理员或超级管理员用户可在 DEM 页面上查看“系统配置”）：

表 4.1 DEM 配置参数说明

| 模块类别 | 参数 | 默认值 | 描述 |
|---------|---------------------------|------|--------------------------------------|
| 客户端工具配置 | db_session_check_freq | 60 | 客户端工具中的数据库连接有效性检查频率(秒)，最小值1 |
| | db_session_invalid_time | 1800 | 客户端工具中的数据库连接的失效时间(秒)，最小值60 |
| | db_session_pool_count | 11 | 用于管理所有用户的数据库会话的池大小(重启生效)，最小值2 |
| | dem_page_size | 40 | 前端页面中的表格分页时的页大小，最小值1 |
| | dem_chart_max_points | 1440 | 前端页面中的折线图最大点数，0不限制 |
| | dashboard_refresh_freq | 5 | 联机大盘监测状态刷新频率(秒)，最小值1 |
| | dashboard_points | 30 | 联机大盘页面中保留的监测次数，最小值1 |
| 监控告警配置 | dem_page_refresh_freq | 60 | 主机和数据库页面状态刷新频率(秒)，最小值60 |
| | alert_his_re_notify_time | 600 | 同一主机或数据库的同类警告在该时间间隔内不再重复发送(秒)，最小值600 |
| | alert_check_thread_count | 2 | 监控告警监测任务线程数(重启生效)，最小值1 |
| | alert_notify_thread_count | 2 | 监控告警通知任务线程数(重启生效)，最小值1 |
| | alert_rule_reload_freq | 60 | 内存中告警规则缓存的刷新频率(秒)，最小值60 |
| | db_invalid_time | 300 | 指定时间内没有收到主机信息，则认为数据库失效(秒)，最小值120 |
| | mf_invalid_time | 180 | 指定时间内没有收到主机信息，则认为主机失效(秒)，最小值20 |
| | mf_process_invalid_time | 300 | 指定时间内没有收到主机信息，则认为进程失效(秒)，最小值120 |
| | db_license_threshold | 100 | license 信息到期告警提示阈值 |

| | | | |
|-----------------------|-----------------------------|--------|--|
| | db_sqllog_datasource | 0 | 获取数据库 SQL 日志信息的后台表，0 存放动态视图数据的后台表，1 存放日志文件数据的后台表 |
| | message_token_validity | 300000 | 远端 agent 发送消息的有效期(毫秒)，0 表示不限制 |
| dmagent 监控频率 配置 | gather_mf_info_freq | [60] | 主机基本信息收集频率(秒)，最小值 10 |
| | gather_mf_stat_freq | [60] | 收集主机状态信息的频率(秒)，最小值 10 |
| | gather_mf_disk_freq | [600] | 收集主机磁盘信息的频率(秒)，最小值 10 |
| | gather_mf_process_freq | [60] | 收集主机进程信息的频率(秒)，最小值 10 |
| | gather_db_info_freq | [60] | 数据库基本信息收集频率(秒)，最小值 60 |
| | gather_db_stat_freq | [60] | 收集数据库状态信息的频率(秒)，最小值 60 |
| | gather_db_deadlock_freq | [60] | 收集数据库死锁信息的频率(秒)，最小值 60 |
| | gather_db_event_freq | [60] | 收集数据库事件信息的频率(秒)，最小值 60 |
| | gather_db_session_freq | [60] | 收集数据库会话信息的频率(秒)，最小值 60 |
| | gather_db_tablespace_freq | [600] | 收集数据库表空间信息的频率(秒)，最小值 60 |
| | gather_db_user_freq | [600] | 收集数据库用户锁定信息的频率(秒)，最小值 60 |
| | gather_db_table_freq | [60] | 收集数据库表数据信息的频率(秒)，最小值 60 |
| | gather_db_audit_record_freq | [0] | 收集数据库审计记录的频率(秒)，最小值 60, 0 表示不收集 |
| | gather_db_log_history_freq | [60] | 收集数据库运行日志记录的频率(秒)，最小值 60 |
| | gather_db_rlog_history_freq | [60] | 收集数据库 Redo 日志信息的频率(秒)，最小值 60 |
| | gather_db_backup_freq | [60] | 执行数据库备份的检查频率(秒)，最小值 60 |
| | gather_db_arch_freq | [60] | 收集数据库归档信息的频率(秒)，最小值 60 |
| | gather_db_backupset_freq | [60] | 收集数据库备份集信息的频率(秒)，最小值 60 |
| | gather_db_ini_file_freq | [180] | 收集数据库INI 配置文件(如 DM.INI, dmarch.ini)的频率(秒), |

| | | | |
|---------|---------------------------------|---------------------------------------|---|
| | | | 最小值 60 |
| | gather_db_sqllog_file_freq | [60] | 收集数据库 SQL 日志（日志文件）的频率（秒），最小值 60 |
| | gather_db_sqllog_file_exec_time | 1000 | 收集数据库 SQL 日志（日志文件）的最小执行耗时（毫秒），最小值 0 |
| | gather_db_sqllog_file_buffer | 100 | 收集数据库 SQL 日志（日志文件）的缓存大小，最小值 100 |
| | gather_db_sqllog_view_freq | [60] | 收集数据库 SQL 日志（动态视图）的频率(秒)，最小值 60 |
| | gather_db_sqllog_view_exec_time | 1000 | 收集数据库 SQL 日志（动态视图）的最小执行耗时（毫秒），最小值 0 |
| | gather_db_group_dpc_stat_freq | [60] | 收集 DPC 集群状态信息的频率(秒)，最小值 60 |
| | gather_db_group_dsc_stat_freq | [60] | 收集 DSC 集群状态信息的频率(秒)，最小值 60 |
| | gather_db_group_mpp_stat_freq | [60] | 收集 MPP 集群状态信息的频率(秒)，最小值 60 |
| | gather_db_group_rw_stat_freq | [60] | 收集 RW 集群状态信息的频率(秒)，最小值 60 |
| | gather_db_group_dw_stat_freq | [60] | 收集 DW 集群状态信息的频率(秒)，最小值 60 |
| 系统邮箱配置 | mail_enable | true | (true/false)，开启时需完成邮箱配置 |
| | mail_sender | Dameng Database Develop Team | 系统邮件发送人 |
| | mail_address | dmoa@dame ng.shang ai | 系统邮箱地址 |
| | mail_password | 888888 | 系统邮箱密码 |
| | mail_smtp_host | 192.168.0 .212 | SMTP 服务器地址或者 SMTP 服务器地址:端口 |
| | mail_ssl | false | (true/false)，开启 SSL 加密，仅当 mail_smtp_host 中指定端口时有效 |
| 自定义推送配置 | mail_config | | 邮箱服务器连接配置属性，格式为 key=value，多个以;间隔 |
| | custom_notify_enable | false | (true/false)，开启时必须实现自定义推送功能 |
| | custom_notify_class | | 自定义推送类，需实现接口 com.dam |

| | | | |
|------|-----------------------------------|---|--|
| | | | eng.dem.server.util.[INotify 或 IPhoneNotify] |
| 安全配置 | user_login_limit_count | 5 | 限制用户登录连续错误次数，若超过将锁定账号，最小值 3 |
| | user_login_lock_time | 60 | 账号锁定的时间（分钟），最小值 10 |
| | user_keep_alive_time | 0 | 账户存活时间（秒），最小值 60, 0 表示不限制 |
| | audit_enable | false | 是否开启系统审计 (true/false) |
| | audit_flush_freq | 30 | 审计信息刷盘频率（秒），最小值 1 |
| 其他功能 | prometheus_metric_nodes | | 响应 prometheus 信息拉取请求的主机列表，以;间隔，信息拉取频率由 prometheus 自身配置 |
| | rometheus_metric_db_no des | | 响应 prometheus 信息拉取请求的数据库列表，以;间隔，信息拉取频率由 prometheus 自身配置 |
| | subscription_push_freq | 3600 | 订阅推送频率(秒)，最小值 600 |
| | subscription_sql_top | 10 | 订阅信息包含高频 SQL, 慢 SQL, 错误 SQL 和死锁 SQL 数目, 最大值 20 |
| | auditrecord_receiver_address | | 审计日志接收服务器地址，格式：IP:PORT |
| | auditrecord_target_db | | 发送目标数据库的审计日志 |
| | auditrecord_mark_config_path | | 连接审计目标数据库所使用的密标配置文件路径 |
| | auditrecord_send_freq | 120 | 审计日志发送频率(秒)，最小值为 60 |
| | database_baseinfo_udp_send_freq | 60 | 数据库基础信息的 UDP 协议发送频率(秒)，最小值为 60 |
| | database_configinfo_udp_send_freq | 60 | 数据库配置信息的 UDP 协议发送频率(秒)，最小值为 60 |
| | database_statusinfo_udp_send_freq | 60 | 数据库状态信息的 UDP 协议发送频率(秒)，最小值为 60 |
| | database_info_udp_reciver_address | | 数据库信息的 UDP 协议接收网络地址，格式：IP: PORT |
| | database_info_udp_send_db_list | | 数据库信息的 UDP 协议发送限定的数据库列表，列表项以;分隔 |
| | api_crypt_class | com.damen.g.dem.ser.ver.util.DefaultEncryptEngine | 实现接口 com.dameng.dem.server.util.IEncryptEngine, 用于用户密码的加解密 |

DM 企业管理器 DEM

| | | | |
|--|---------------|-------|----------------------------------|
| | snmp_enable | false | (true/false), 开启时必须实现 SNMP 协议 |
| | snmp_udp_port | 161 | SNMP 协议监听网络端口号 |

5 用户管理

用户管理包括以下几个方面，创建用户需要指定用户权限：

- ◆ 角色与权限
- ◆ 创建用户
- ◆ 用户启用与禁用

5.1 角色与权限

当创建一个 DEM 用户时，必须给每个用户指定一个角色，用户角色决定了用户权限。

DEM 中设置了 4 种角色：普通用户、高级用户、管理员以及超级管理员。

部署 DEM 时预设的 admin 为超级管理员，超级管理员只有 admin，之后创建的用户只可为普通用户、高级用户或管理员。

每个角色的权限如下：

表 5.1 DEM 角色权限说明

| 角色 | 客户端工具权限 | 监控及告警权限 | 系统管理权限 |
|-------|---------|---------|--------|
| 普通用户 | √ | | |
| 高级用户 | √ | √ | |
| 管理员 | √ | √ | √ |
| 超级管理员 | √ | √ | √ |

以管理员或超级管理员用户登录 DEM，在系统管理->角色管理中也可查看角色权限信息。

5.2 创建用户

管理员或超级管理员必须为每个需要登录 DEM 的人创建一个用户账户，并为每个用户指定一个角色，角色不同，权限不同。

管理员或超级管理员在 DEM 页面可以创建用户账户，具体步骤如下：

1. 在系统管理下，点击用户管理，打开用户管理页面；

2. 在用户管理页面工具栏，点击“添加”按钮；

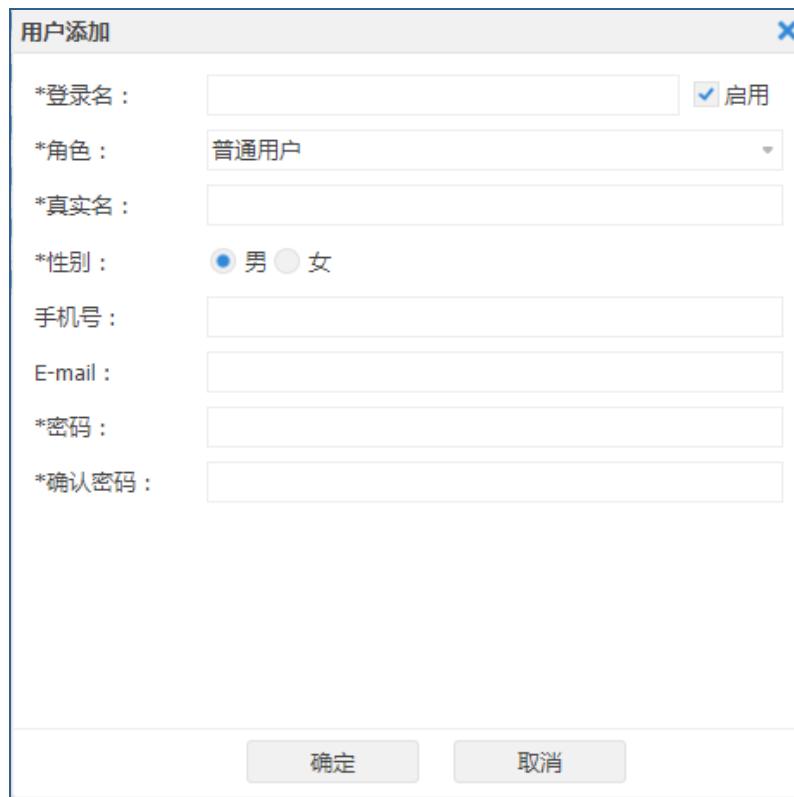


图 5.1 用户添加界面

3. 输入唯一用户登录名和密码；

4. 指定用户角色；

5. 将带星号项填写完毕后，点击确定，保存用户信息。

创建用户账户后也可对用户信息进行重新编辑。

用户登录后可在 DEM 工具栏点击最右侧的用户名，修改自己的真实名、性别、邮箱、手机号以及密码等信息。

5.3 用户启用与禁用

管理员和超级管理员可以在用户管理页面启用或禁用用户，用户被禁用后无法登录 DEM。若用户登录连续错误 5 次（默认值 5，管理员和超级管理员可自行配置，详见参数 user_login_limit_count），系统将自动禁用该用户，用户被锁住则无法登录，默认锁住时间为 60 分钟（默认值 60 分钟，管理员和超级管理员可自行配置，详见参数 user_login_lock_time）。



注意：

如果登录用户超过账号存活时间（默认值为 0，表示不做限制，管理员和超级管理员可自行配置，详见参数 `user_keep_alive_time`）且未做任何请求操作，则会被强制退出。

6 客户端工具

客户端工具包括管理维护数据库和数据库相关组件的工具。包括对象管理、SQL 编辑器、SQL 调试、部署集群、数据迁移以及 Proxy 管理。

6.1 对象管理

DEM 的对象管理功能与桌面版 manager 的对象管理功能一致。可通过 DEM 直接访问要管理的数据库实例，通过导航树方式展示数据库对象，并通过右键操作方式操作数据库对象。

6.1.1 系统结构

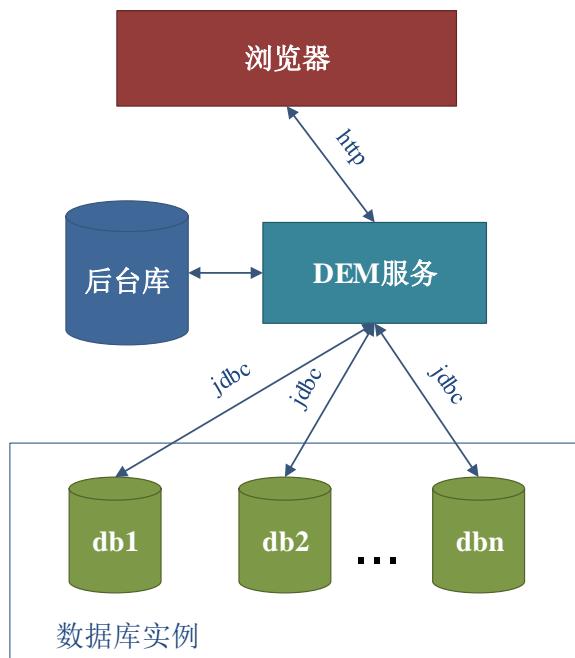


图 6.1 对象管理系统结构图

对象管理工具可通过 DEM 直接利用 jdbc 连接所要管理的数据库进行访问。

6.1.2 实现原理

浏览器中的每次数据库操作都需要有对应的数据库连接（DBSession），若在每次操作时创建 DBSession，性能较差，故通常将建立的 DBSession 缓存起来，供下次操作重复使用。DEM 中也是采用这种思想，创建的 DBSession 会分配一个全局唯一 ID，并被缓

存到连接池（SessionPool）中，浏览器中的操作请求会附带该 DBSession 的 ID，发至 DEM，DEM 通过该 ID 在 SessionPool 中找到对应的 DBSession，执行相应的数据库操作。为了提升并发性能，DEM 内置多个 SessionPool 来均衡负载。如下图：

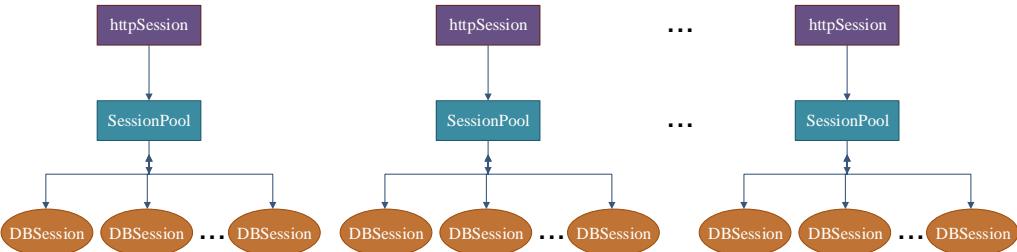


图 6.2 对象管理实现原理图

每个请求包含一个对应的 httpSession 信息，根据 httpSession 映射到 SessionPool，从对应 SessionPool 中取出对应的 DBSession。

其中 httpSession 为前端存储 session 相关信息，DBSession 为实际的对应的数据库连接，SessionPool 为数据库会话管理池。

数据库会话管理池大小即为 SessionPool 的数量，管理员和超级管理员可以通过 DEM 界面的系统配置页面进行配置，路径：客户端工具配置-> db_session_pool_count，默认 11。详细请参考[第 4 章 配置参数说明](#)。

DBSession 的增删查：

1. 增加 DBSession

在对象管理中新建连接，都会创建一个 DBSession，然后根据 DBSession ID，存储在对应 SessionPool 中。

2. 删除 DBSession

删除 DBSession 有 2 种方式：

1) 在 DEM 页面的对象管理中断开连接，会根据 DBSession ID 将 DBSession 从 SessionPool 中移除，并断开连接；

2) 定时检查有效性，检查频率由参数“db_session_check_freq”控制，有效时间由“db_session_invalid_time”控制，默认 1800s，即如果 1800s 没有被使用过，则认为失效。这 2 个参数可由管理员或超级管理员配置。

3. 查询 DBSession

在对象管理中，新建连接，创建 DBSession 后，会在相对应对象保留 DBSession，并传入对应前端存储的 HttpSession 中，后续每次操作，都会传入 HttpSession，然后通

过 HttpSession 中的 DBSession ID，查找到对应的 SessionPool，然后再根据 DBSession ID 从 SessionPool 中取出 DBSession。

6.1.3 功能介绍

对象管理功能与桌面版管理工具 manager 的对象管理功能一致，使用方式详细请参见管理工具 manager 的联机帮助手册。

下图为使用 DEM 对象管理功能对本地库进行操作的示例。

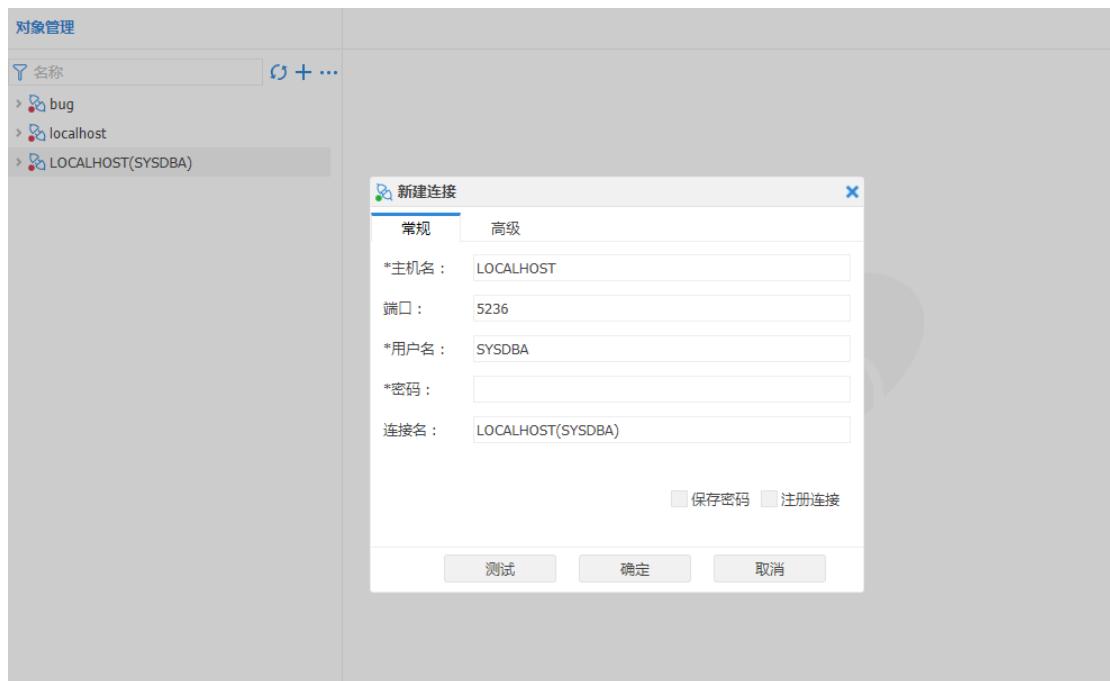


图 6.3 DEM 对本地库进行操作

6.2 SQL 编辑器

SQL 编辑器功能与桌面版 manager 的 SQL 编辑器功能一致。SQL 编辑器通过 DEM 直接访问要管理的数据库实例，在页面编辑 SQL，通过 jdbc 连接数据库实例执行 SQL，且同样提供了全新的 SQL 助手 2.0，能够对用户输入的 SQL 语句进行实时的语法检查，定位错误的 SQL 语句，同时能够对用户输入 SQL 进行实时的智能提示。

6.2.1 系统结构

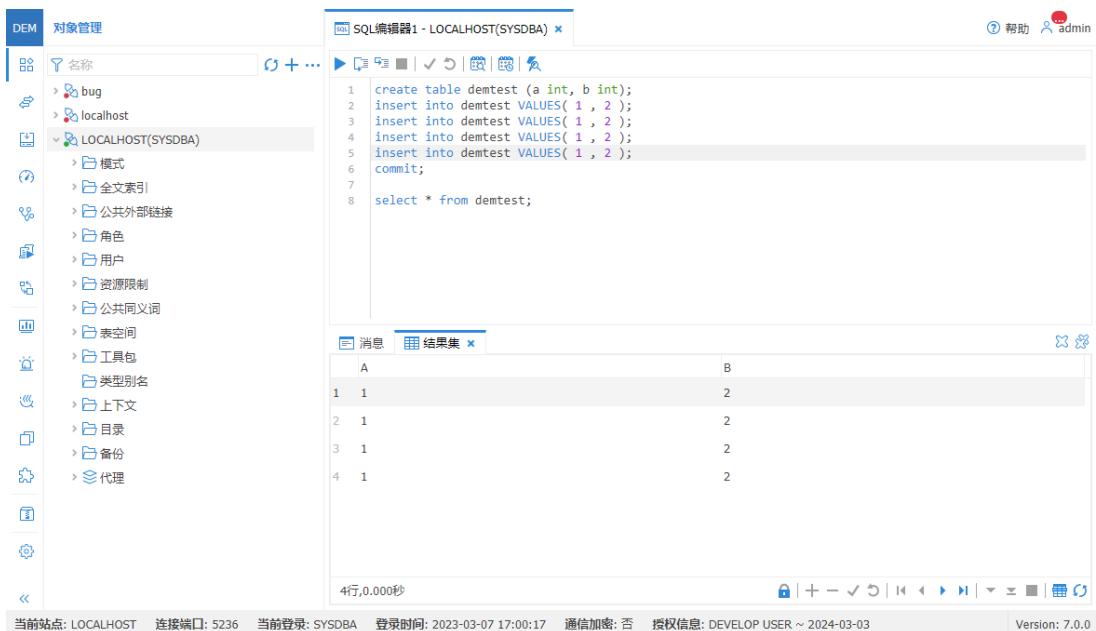
参见 [6.1 对象管理](#)。

6.2.2 实现原理

参见 [6.1 对象管理](#)。

6.2.3 功能介绍

与桌面版管理工具 manager 的 SQL 编辑器功能一致，详细请参见管理工具 manager 的联机帮助手册。



6.3 SQL 调试器

SQL 调试功能与桌面版 manager 的 SQL 调试功能一致。通过 DEM 直接访问要管理的数据库实例。

6.3.1 系统结构

参见 [6.1 对象管理](#)。

6.3.2 实现原理

参见 [6.1 对象管理](#)。

6.3.3 功能介绍

与桌面版管理工具 manager 的 SQL 调试功能一致，详细请参见管理工具 manager 的联机帮助手册。

6.4 集群管理

达梦集群往往环境比较复杂、组件多、配置文件和配置项多，部署集群工具采用了友好向导式部署方式，能实现远程部署，用户只需访问 DEM 即可完成在多台机器上部署集群组件。DEM 目前支持部署单实例、分布式、大规模并行处理、共享存储、多副本、数据守护（大规模并行处理主备、实时主备、读写分离）。



说明：数据守护需要与服务器版本相匹配，即最新的支持数据守护 v4.0，则需要上传的服务器必须支持数据守护 v4.0，服务器版本信息详细参见《DM8 数据守护和读写分离集群 v4.0》。

6.4.1 系统结构

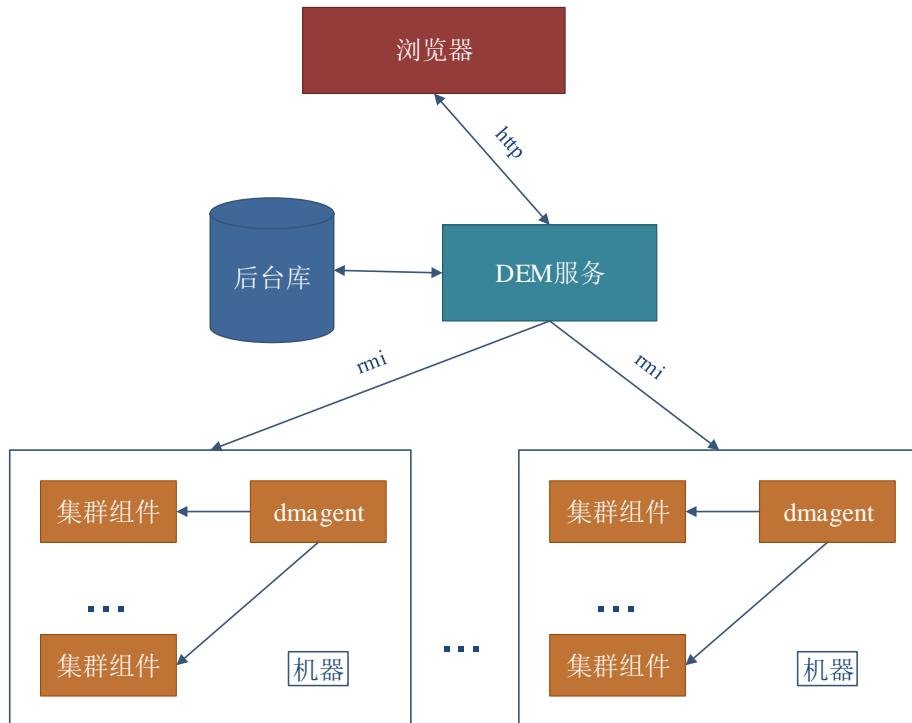


图 6.4 部署集群系统结构图

6.4.2 实现原理

集群部署分 3 层：前端浏览器、DEM 服务以及要部署集群组件的机器。

用户通过前端浏览器配置集群，配置好后，DEM 按配置信息，创建部署任务，将部署任务分发各要部署的机器上的 dmagent 执行。

集群部署采用先配置后执行的方式，以向导式方式引导用户配置集群相关信息，然后 DEM 通过 RMI 方式访问远程机器的 dmagent，将具体的部署任务例如部署配置文件、启停集群组件以及备份还原等分发给 dmagent 执行，这样既能实现集中式部署，实现只需在 DEM 上操作就可以完成所有集群组件的部署，又能实现多台机器并发的执行部署任务，使得在没有依赖关系前提下，多台机器可以并发执行部署任务。

6.4.3 功能介绍

下面将根据不同的集群种类分别介绍其部署过程，其中集群分为：DM MPP、数据守护和 DMDSC。

6.4.3.1 MPP 部署

DM MPP 分为普通 MPP 和 MPP 主备，下面将详细介绍此 2 种 MPP 集群的部署步骤。

6.4.3.1.1 普通 MPP 部署

在左侧导航栏选择集群管理，中间栏选择搭建后点击加号，打开新建集群搭建对话框：

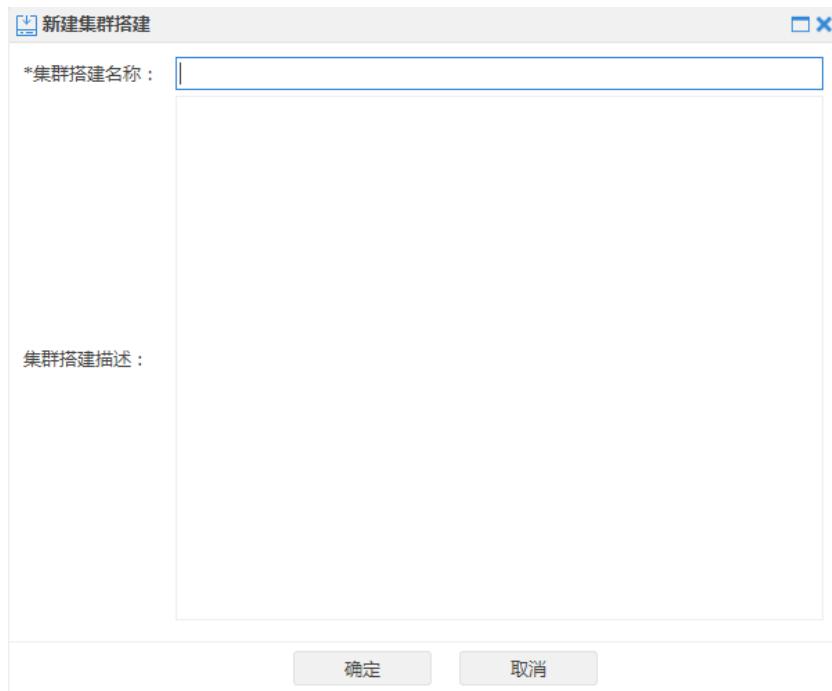


图 6.5 新建集群部署界面

给定集群搭建名称后，跳转到“选择集群类型”界面，选择“大规模并行处理”：



图 6.6 选择部署集群类型界面

部署步骤：

1. 选择主机

如下图所示：

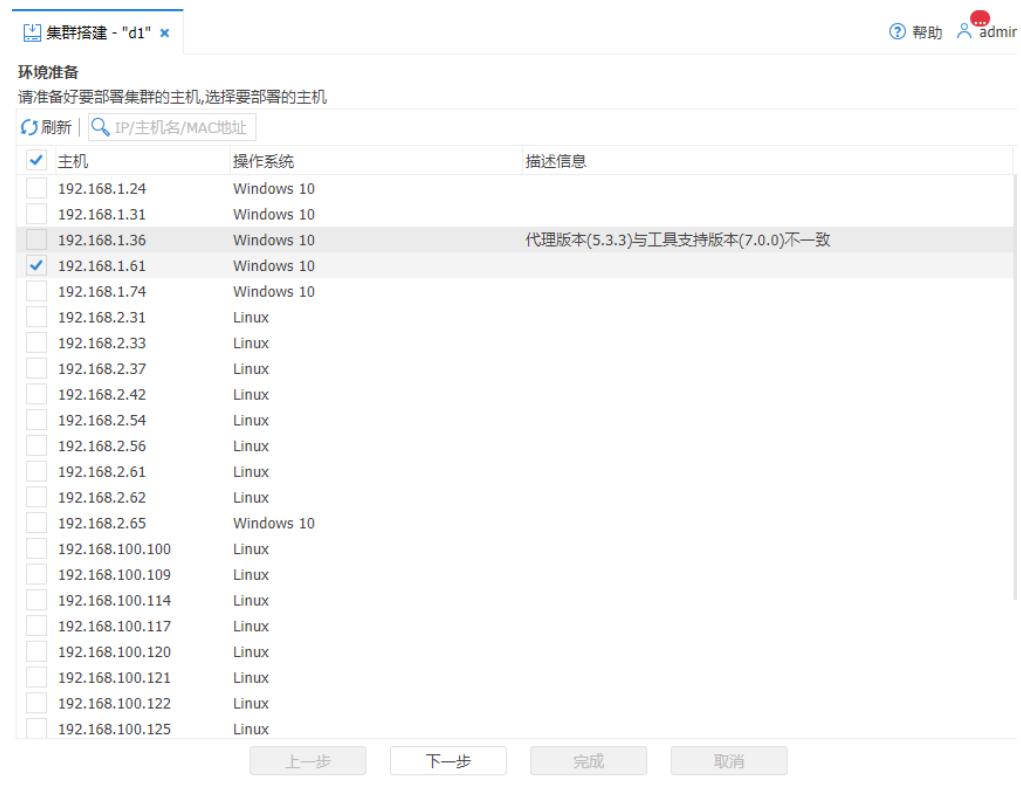


图 6.6 环境准备界面

先准备好要部署机器，并在机器上部署 dmagent，具体 dmagent 部署请参见 [3.2 dmagent 部署](#)。

选中要部署的主机，点击“下一步”。

2. 实例划分

如下图所示：



图 6.7 实例规划界面

配置实例的主机分布，以及实例相关端口划分。

默认的 MPP 在每台所选中的主机上部署一个实例。

通过“添加实例”按钮可以添加实例，通过“删除实例”按钮可以删除实例的配置。

可以在表格中对每个实例的相关参数进行编辑。

同时也可以通过“参数配置”，对所有实例进行统一编辑，如下图：



图 6.8 参数配置界面

如果要将实例及相关组件部署为服务，则选择“注册服务”，并可点击“配置服务”，对服务名相关信息进行配置。如下图所示：



图 6.9 服务名配置界面

服务名有固定前缀或后缀，用户可以自定义配置服务名的部分。

实例配置完成后，点击“下一步”。

3. 数据准备

如下图所示：

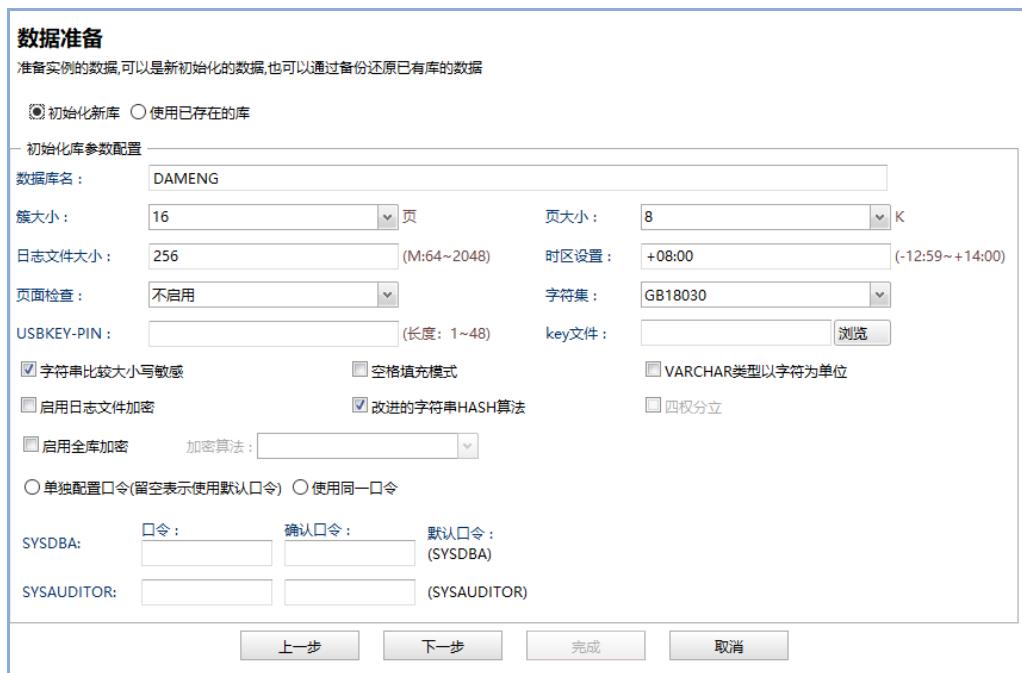


图 6.10 数据准备界面

准备实例的数据，有 2 种方式：初始化新库和使用已存在的库。

- 初始化新库

利用 dminit 工具配置初始化参数进而初始化新库（详细参见《DM8_dminit 使用手册》）。

- 使用已存在的库

如下图：

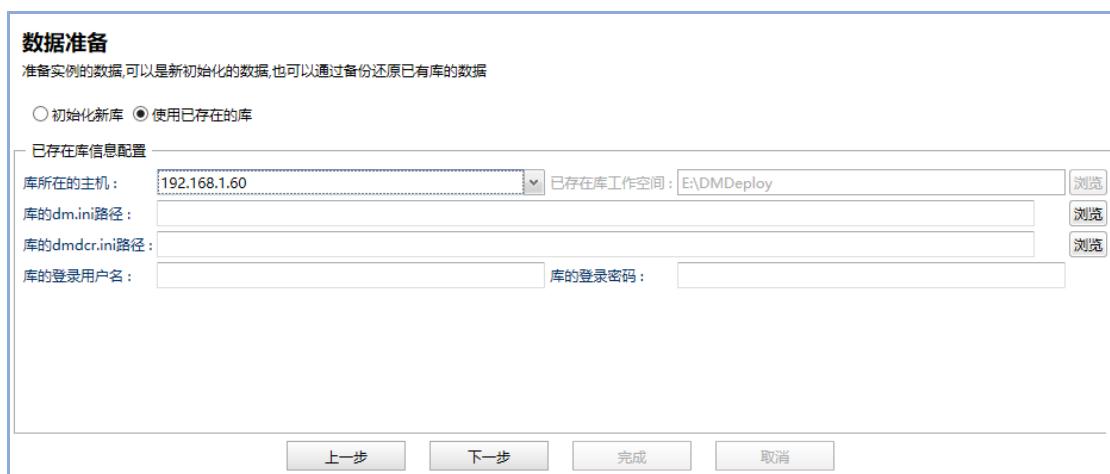


图 6.11 使用已存在的库进行数据准备

通过脱机备份还原，准备数据。

配置好相关参数后，点击“下一步”。

4. dmmail.ini 参数配置

如下图所示：

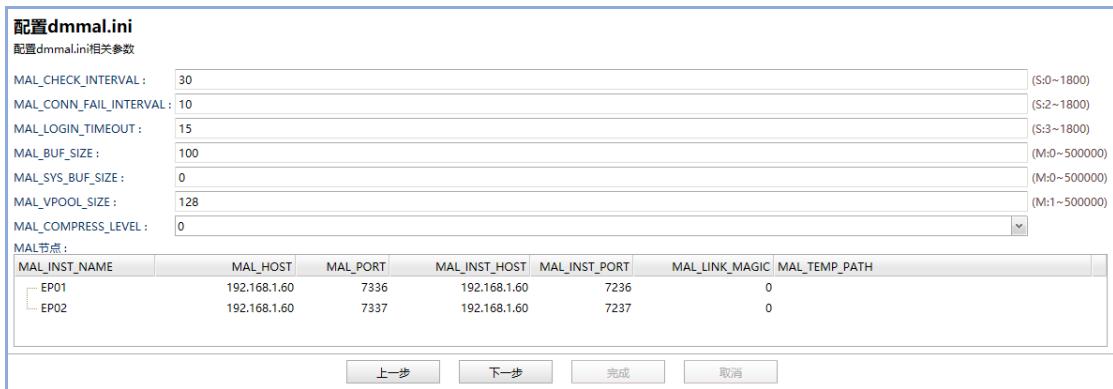


图 6.12 dmmail.ini 参数配置

dmmail.ini 详细参数参见《DM8 集群代理使用手册》。

5. dmarch.ini 参数配置

如下图所示：

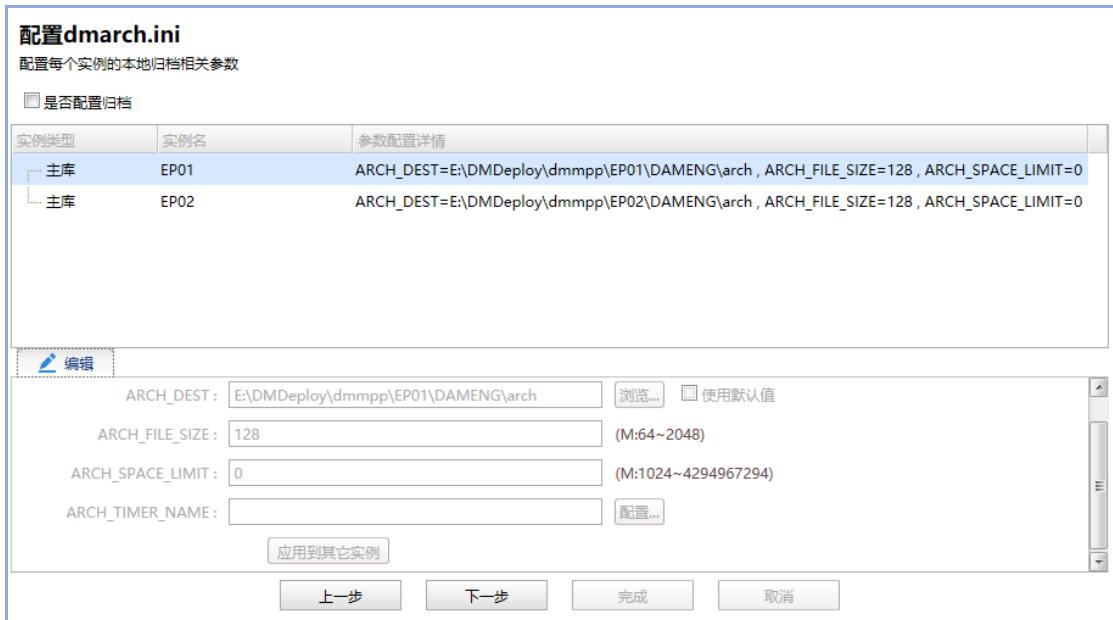


图 6.13 dmarch.ini 参数配置

MPP 环境下并不要求必须配置归档,但用户也可以根据应用实际需要进行配置,选中“是否配置归档”选择框,即可进行配置。

点击“下一步”。

6. 上传服务器文件

如下图所示：



图 6.14 上传服务器文件

上传服务器文件仅支持.zip, ISO, bin 文件。可以“各节点使用同一的达梦数据库服务器文件”，即所有 linux 的主机上传一份服务器文件，所有 windows 的主机上传一份服务器文件。也可以“各节点单独配置达梦数据库服务器文件”，即所有的节点分别上传服务器文件。

为版本统一，建议使用“各节点使用同一的达梦数据库服务器文件”，但如果使用了 linux 不同平台，则使用的服务器版本也会不同，无法使用同一份达梦数据库服务器文件，则需每个节点单独上传服务器文件。

上传完成后，点击“下一步”。

7. 部署信息预览

如下图所示：



图 6.15 部署信息预览

预览所有的配置信息。

确认无误后，点击“下一步”

8. 执行部署

如下图所示：

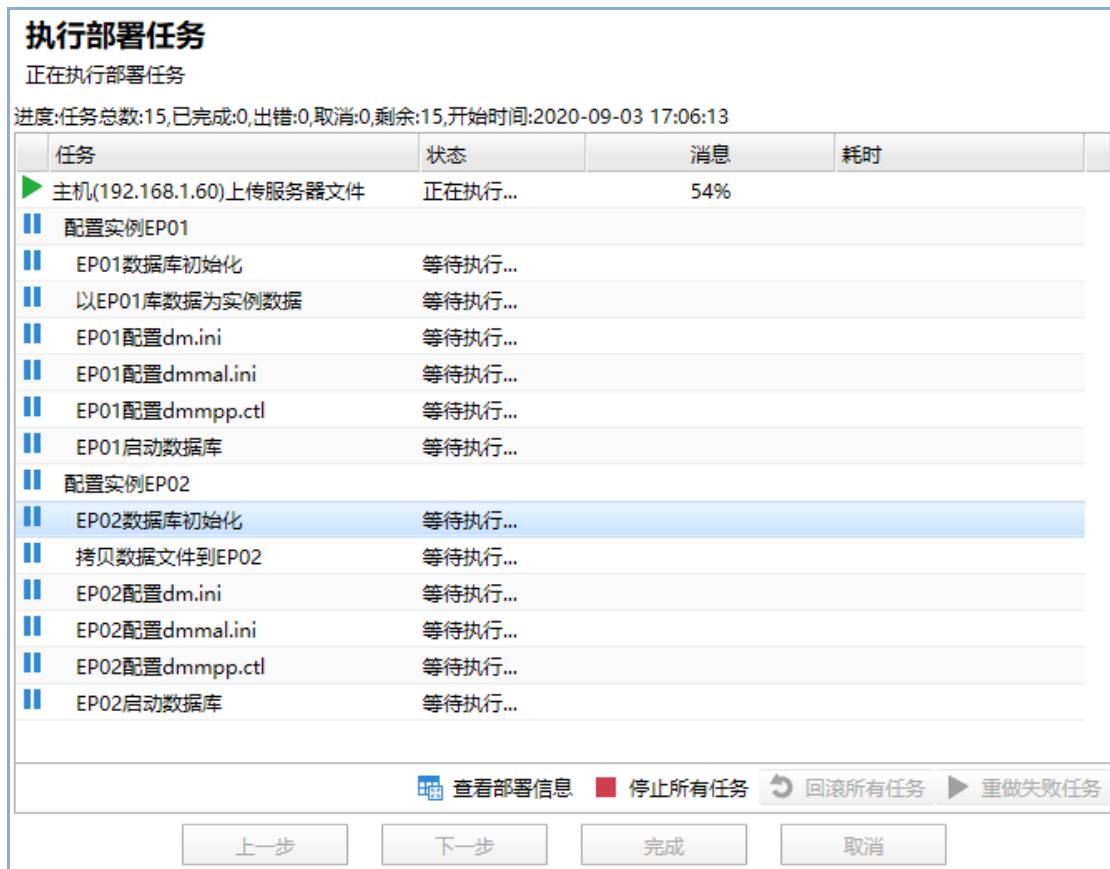


图 6.16 执行部署任务

开始执行部署任务。

点击“查看部署信息”，可以对部署配置信息进行详细浏览。

执行过程中，点击“停止所有任务”，将停止执行所有任务。

执行结束后，点击“回滚任务”，可以回滚所有执行的任务，清除环境。

执行结束后，如果有任务执行失败，点击“重做失败任务”，即可重新执行失败的任务。

若执行结束且部署成功，则如下图所示：



图 6.17 部署成功

点击“添加到监控”，将跳转到“数据库监控”页面，同时会将部署的集群添加到数据库监控列表中。

6.4.3.1.2 MPP 主备部署

参见 [6.4.3.2 数据守护部署](#)。

6.4.3.2 数据守护部署

数据守护分为：实时主备、MPP 主备、读写分离、DMDSC 数据守护以及异步备库。下面将详细展开介绍此 5 种集群的部署步骤。

6.4.3.2.1 实时主备部署

点击“集群部署”，打开新建集群部署对话框，给定名称后，打开“选择部署集群类型”界面，选择“数据守护 v4.0”下面“实时主备”：



图 6.18 选择部署集群类型界面

部署步骤：

1. 选择主机

参见章节 [6.4.3.1.1 1. 选择主机](#)。

2. 实例划分

参见章节 [6.4.3.1.12. 实例划分](#)，仅配置的参数有所不同，具体参数配置参见《DM8 数据守护与读写分离集群 v4.0》。

3. 主备库划分

如下图所示：

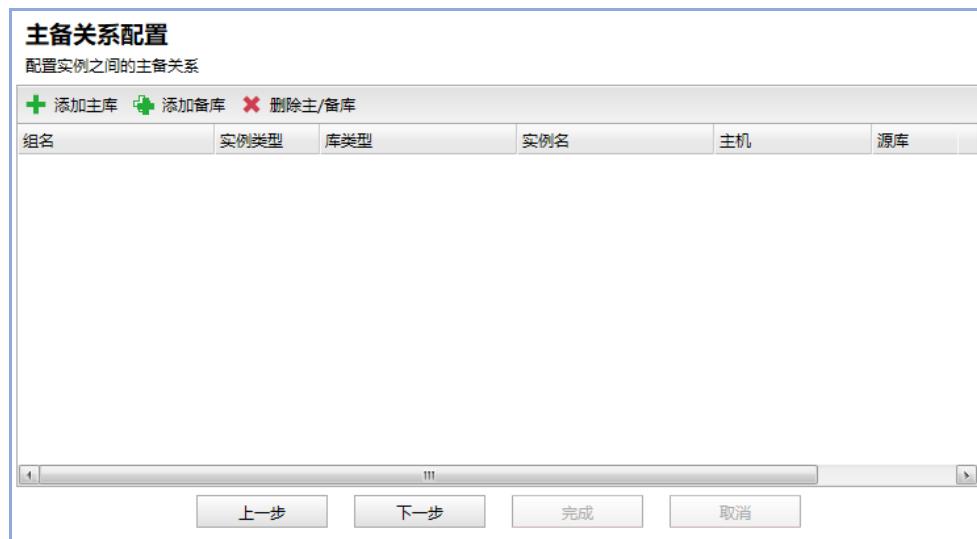


图 6.19 主备关系配置界面

首先通过点击“添加主库”配置哪个实例作为主库，如下图所示：

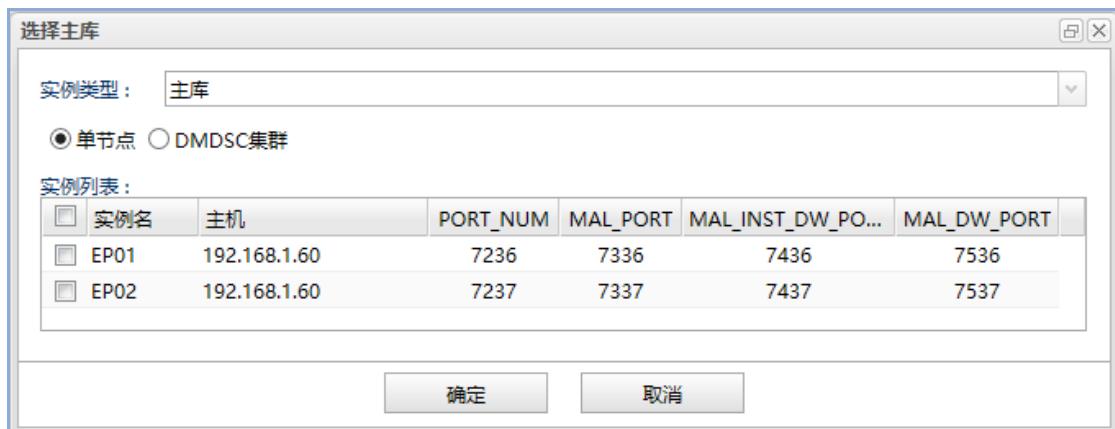


图 6.20 选择主库界面

实时主备，主库可以为单节点，也可以为 DMDSC 集群。

主库添加完成后，点击“添加备库”，如下图所示：

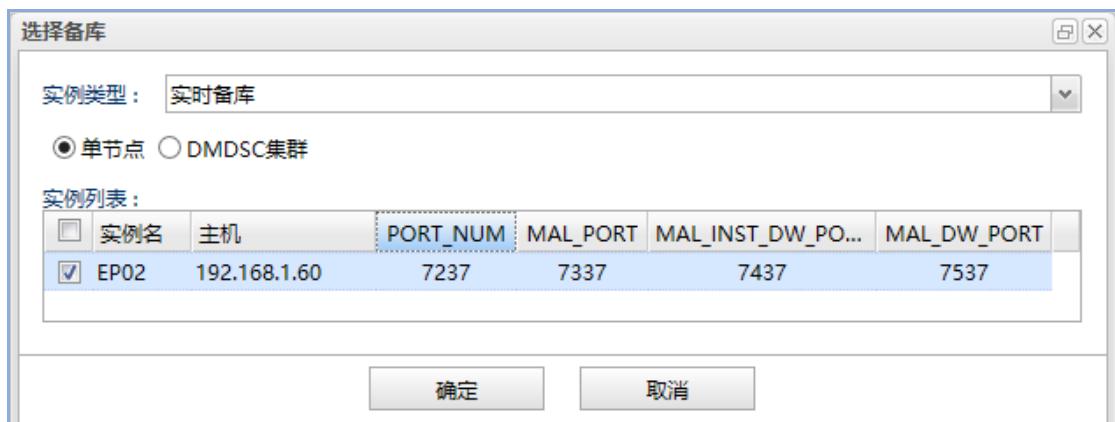


图 6.21 选择备库界面

可以选择备库类型，包括 MPP 备库、实时备库、即时备库以及异步备库。

这里是实时主备，故选择“实时备库”。

同样，备库可以是单节点也可以是 DMDSC 集群。

4. 数据准备

参见章节 [6.4.3.1.1 3. 数据准备](#)。

5. DM.INI 参数配置

如下图所示：

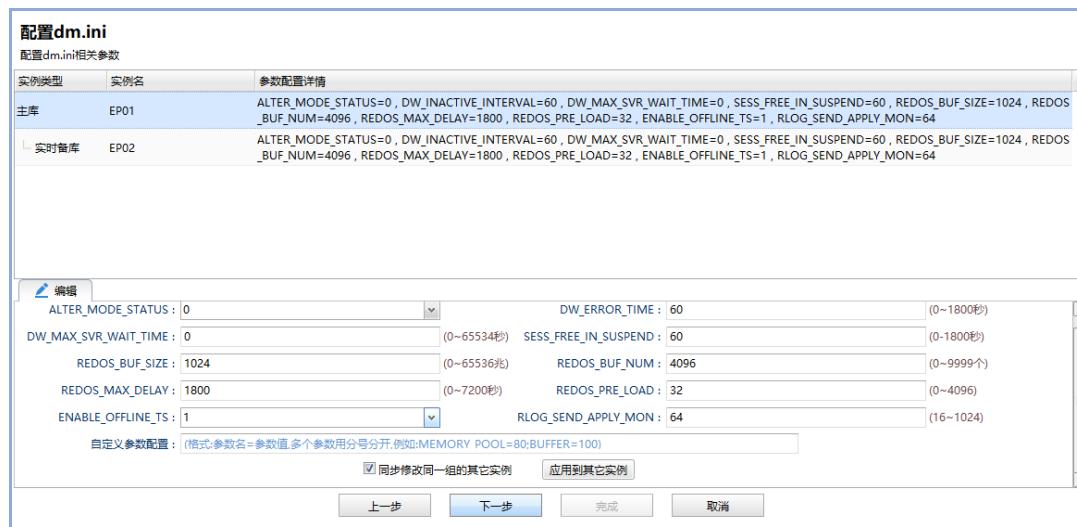


图 6.22 配置 DM.INI 界面

配置实时主备相关的参数信息，详细参见《DM8 数据守护与读写分离集群 v4.0》。

也可以通过自定义参数配置，配置其他 DM.INI 参数信息。

6. dmmal.ini 参数配置

参见章节 [6.4.3.1.1 4.dmmal.ini 参数配置](#)。

7. dmarch.ini 参数配置

参见章节 [6.4.3.1.1 5.dmarch.ini 参数配置](#)。

实时主备必须配置归档信息，因此没有“是否配置归档”选择框。

8. dmwatcher.ini 参数配置

如下图所示：

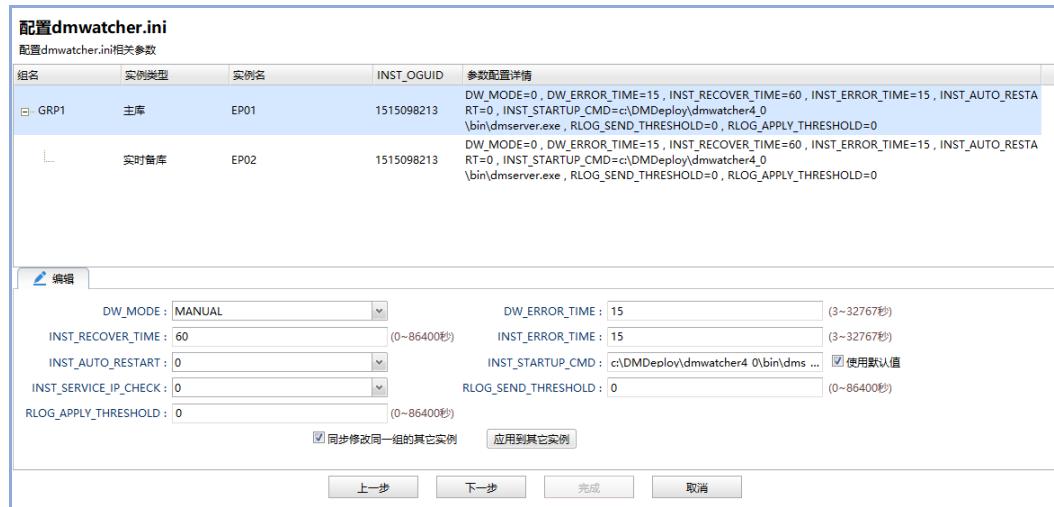


图 6.23 dmwatcher.ini 参数配置界面

配置 dmwatcher.ini 相关参数。

点击“下一步”。

9. 配置监视器 dmmonitor

如下图所示：

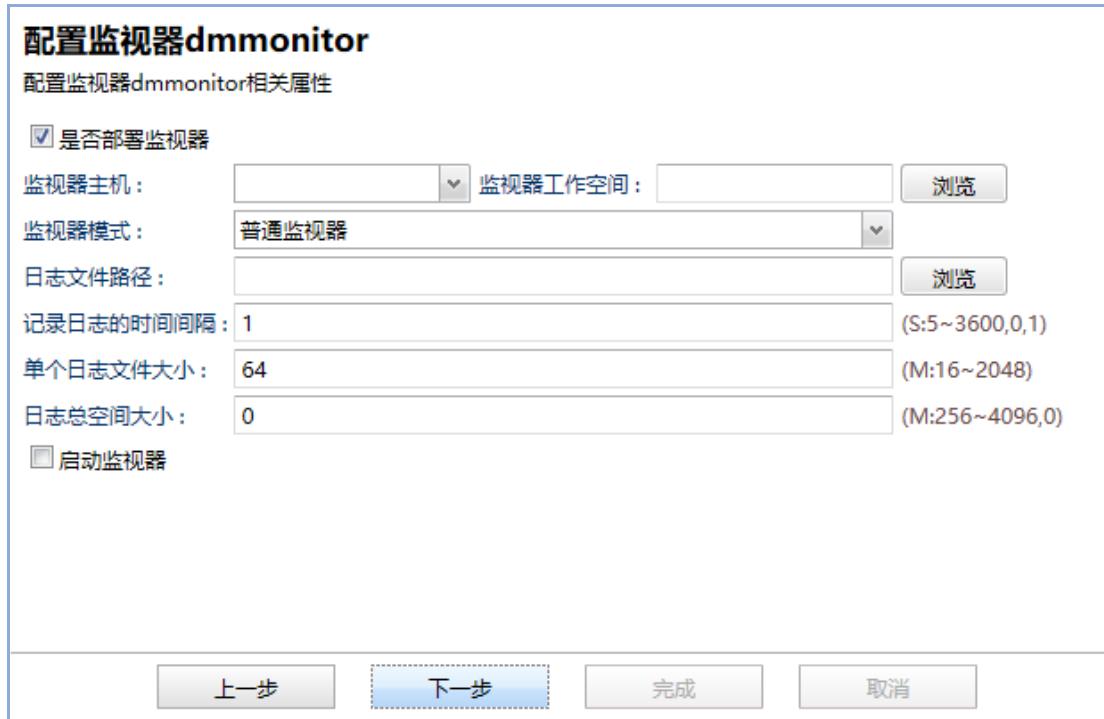


图 6.24 配置监视器 dmmonitor 界面

默认选择部署监视器，但需要用户指定监视器部署在哪台机器上，如果配置的是确认监视器，默认部署完后会启动监视器，用户可以通过选择勾选“启动监视器”决定是否在部署完成后启动监视器。

10. 上传服务器文件

参见章节 [6.4.3.1.1 6.上传服务器文件](#)。

11. 部署信息预览

参见章节 [6.4.3.1.1 7.部署信息预览](#)。

12. 执行部署

参见章节 [6.4.3.1.1 8.执行部署](#)。

6.4.3.2.2 MPP 主备部署

点击“集群部署”，打开新建集群部署对话框，给定名称后，打开“选择部署集群类型”界面，选择“数据守护 v4.0”下面“大规模并行处理主备”：

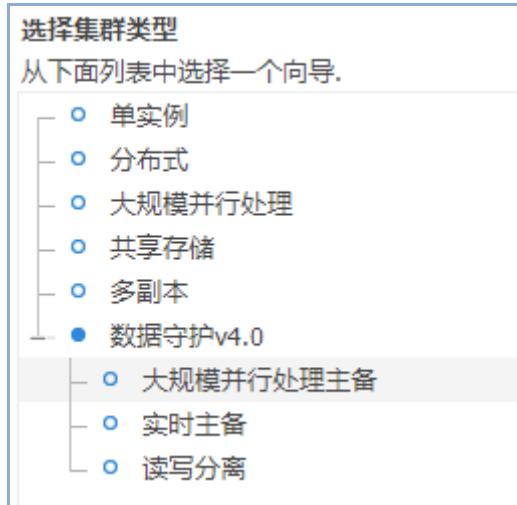


图 6.25 选择部署集群类型界面

部署步骤同 [6.4.3.2.1 实时主备部署](#)的步骤，在“主备库划分”时选择“MPP 备库”即可。

6.4.3.2.3 读写分离部署

点击“集群部署”，打开新建集群部署对话框，给定名称后，打开“选择部署集群类型”界面，选择“数据守护 v4.0”下面“读写分离”：

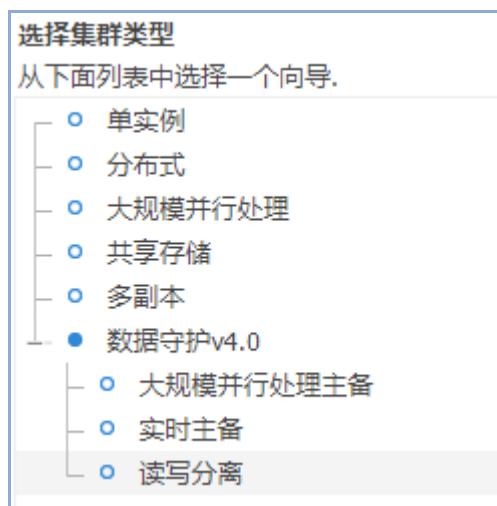


图 6.26 选择部署集群类型界面

部署步骤同 [6.4.3.2.1 实时主备部署](#)的步骤，在“主备库划分”时选择“即时备库”即可。

6.4.3.2.4 DMDSC 数据守护部署

图 6.27 中“数据守护 v4.0”下的“实时主备”和“读写分离”节点都可以配置为 DMDSC。

6.4.3.2.5 异步备库部署

选择图 6.27 中“数据守护 v4.0”下的“实时主备”、“读写分离”或者“大规模并行处理主备”，在“主备库划分”时，添加异步备库，如下图所示：

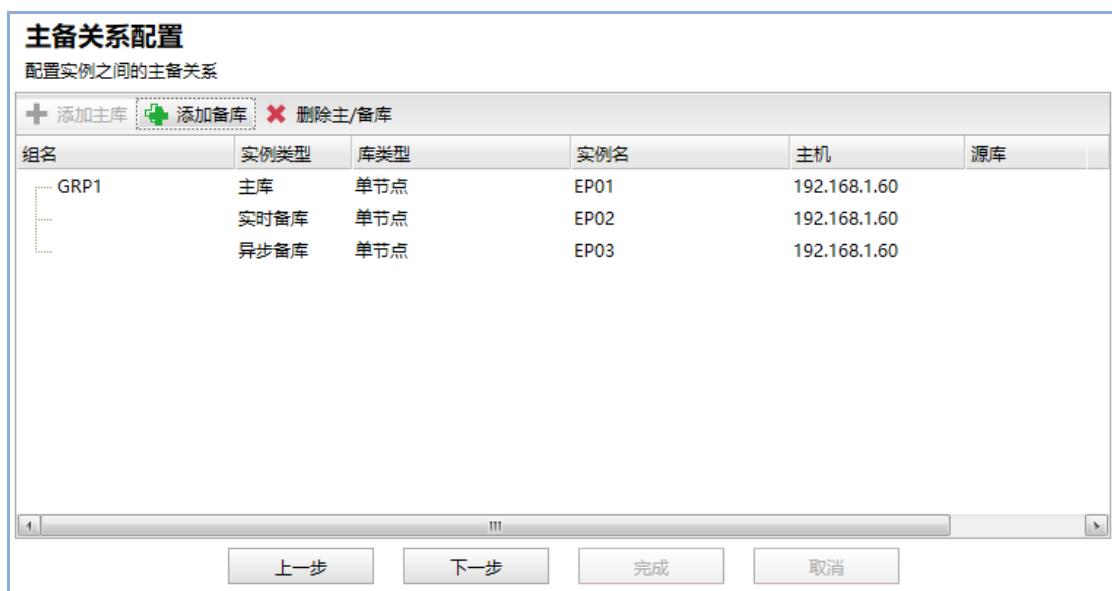


图 6.27 主备关系配置界面

异步备库需要配置对应“源库”。

现在异步备库配置源库策略为：除了异步备库外，其他主备库都需要作为源库使用，同时，异步备库可以级联配置，即异步备库的源库可以为另一个异步备库，因此，上述配置中，默认不配置，为空，表示其他主备库都作为源库使用，但也可指定源库为其他的异步备库。

6.4.3.3 DMDSC 部署

下面将详细展开介绍 DMDSC 集群的部署步骤。

点击“集群部署”，打开新建集群部署对话框，给定名称后，打开“选择部署集群类型”界面，选择“共享存储”：

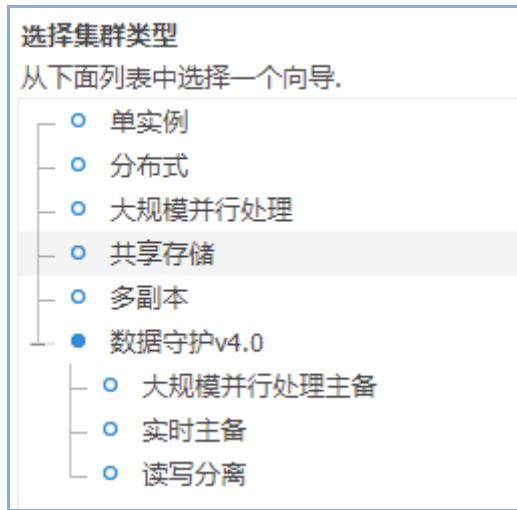


图 6.28 选择部署集群类型界面

部署步骤：

1. 选择主机

参见章节 [6.4.3.1.1 1. 选择主机。](#)

2. 实例划分

如下图所示：

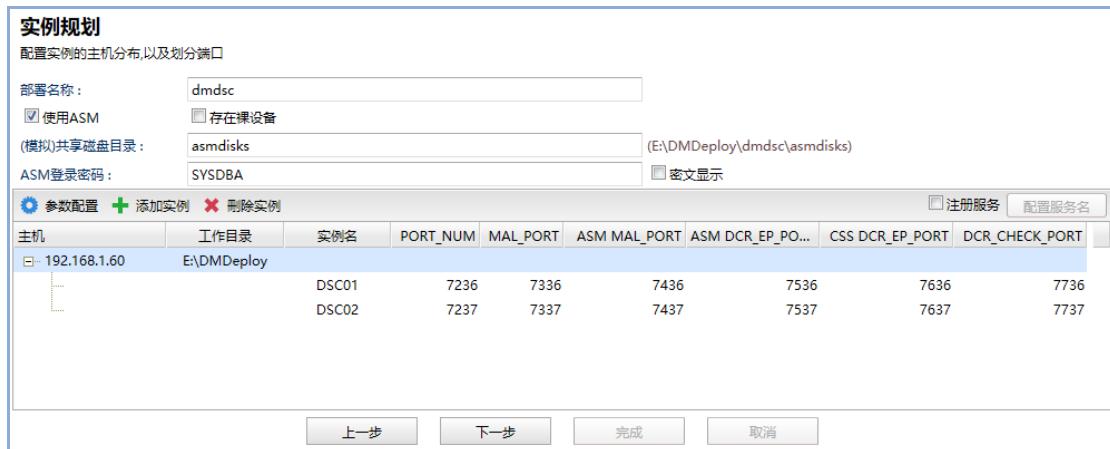


图 6.29 实例规划界面

默认的选择多个主机，每个主机上一个实例，但如果只选择一个主机，则默认会在这个主机上配置 2 个实例，用户根据实际情况添加和删除实例。

DMSC 可以选择是否使用 ASM，如果使用，则需要配置共享磁盘目录和 ASM 登录密码，而且 ASM 登录密码没有默认值，必须手动配置，“密文显示”按钮控制密码以明文显示还是密文显示。

如果存在裸设备，共享磁盘目录为裸设备目录，需要指定“裸设备目录”，否则需要通过模拟共享目录，模拟共享目录需要指定“（模拟）共享磁盘目录”，这里只需指定目录名

称，因为为统一管理，都会在部署目录下，故完整目录为：部署目录+部署名称+（模拟）共享磁盘目录。

点击“下一步”，配置 dmdcr_cfg.ini 参数。

3. dmdcr_cfg.ini 参数配置

如下图所示：

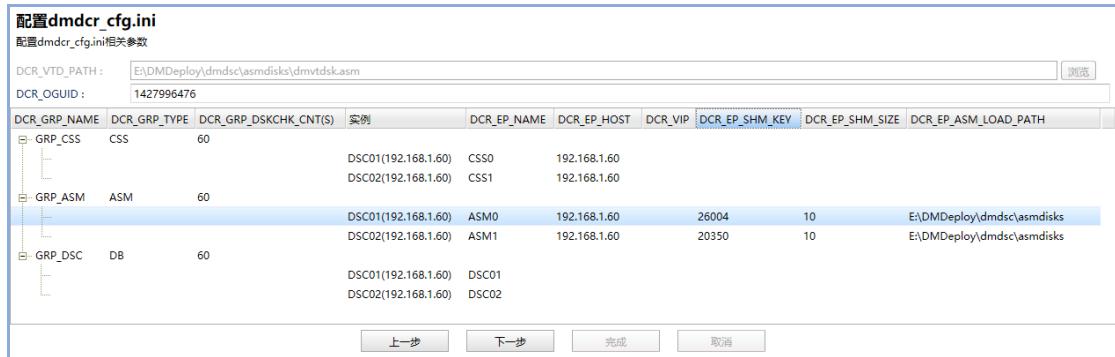


图 6.30 配置 dmdcr_cfg.ini

dmdcr_cfg.ini 详细参数配置说明参见《DM8 共享存储集群》。

点击“下一步”。

4. dmavrmal.ini 参数配置

如下图所示：

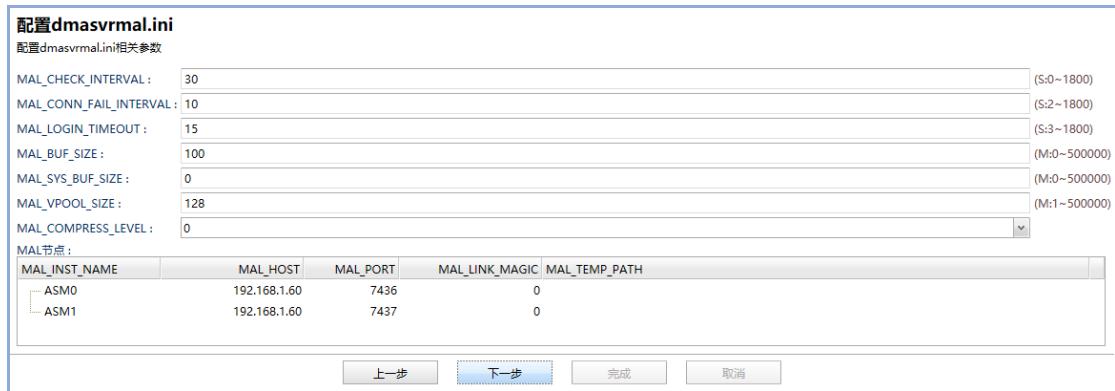


图 6.31 配置 dmavrmal.ini

dmavrmal.ini 详细参数配置说明参见《DM8 共享存储集群》。

点击“下一步”。

5. dmdcr.ini 参数配置

如下图所示：

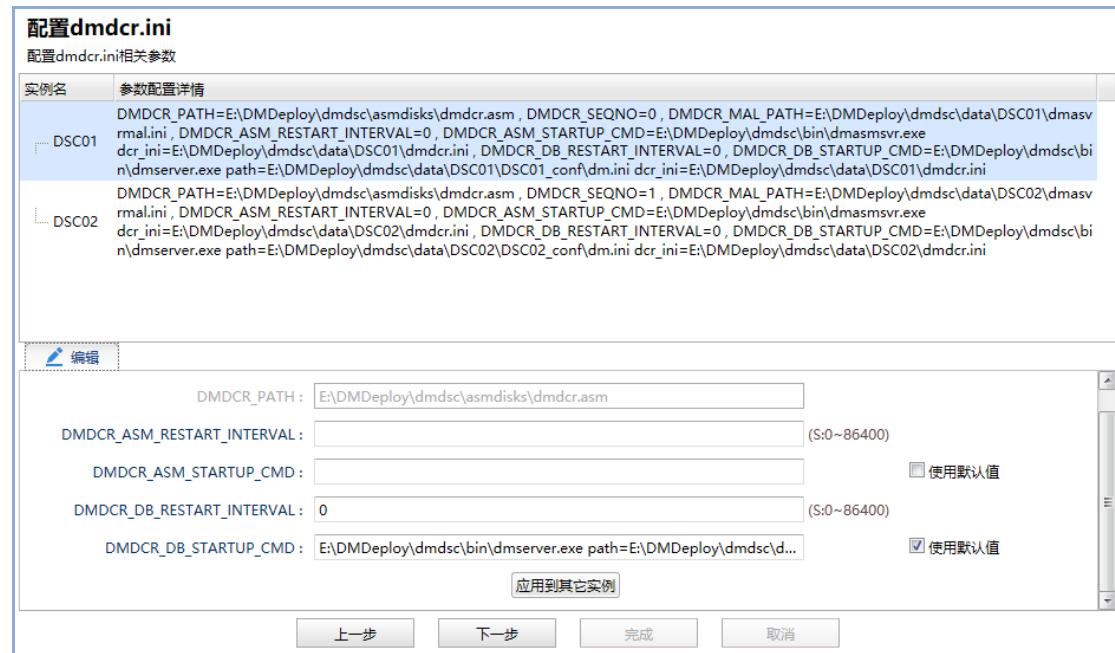


图 6.32 配置 dmdcr.ini

dmdcr.ini 详细参数配置说明参见《DM8 共享存储集群》。

点击“下一步”。

6. DM.INIT.ini 参数配置

如下图所示：

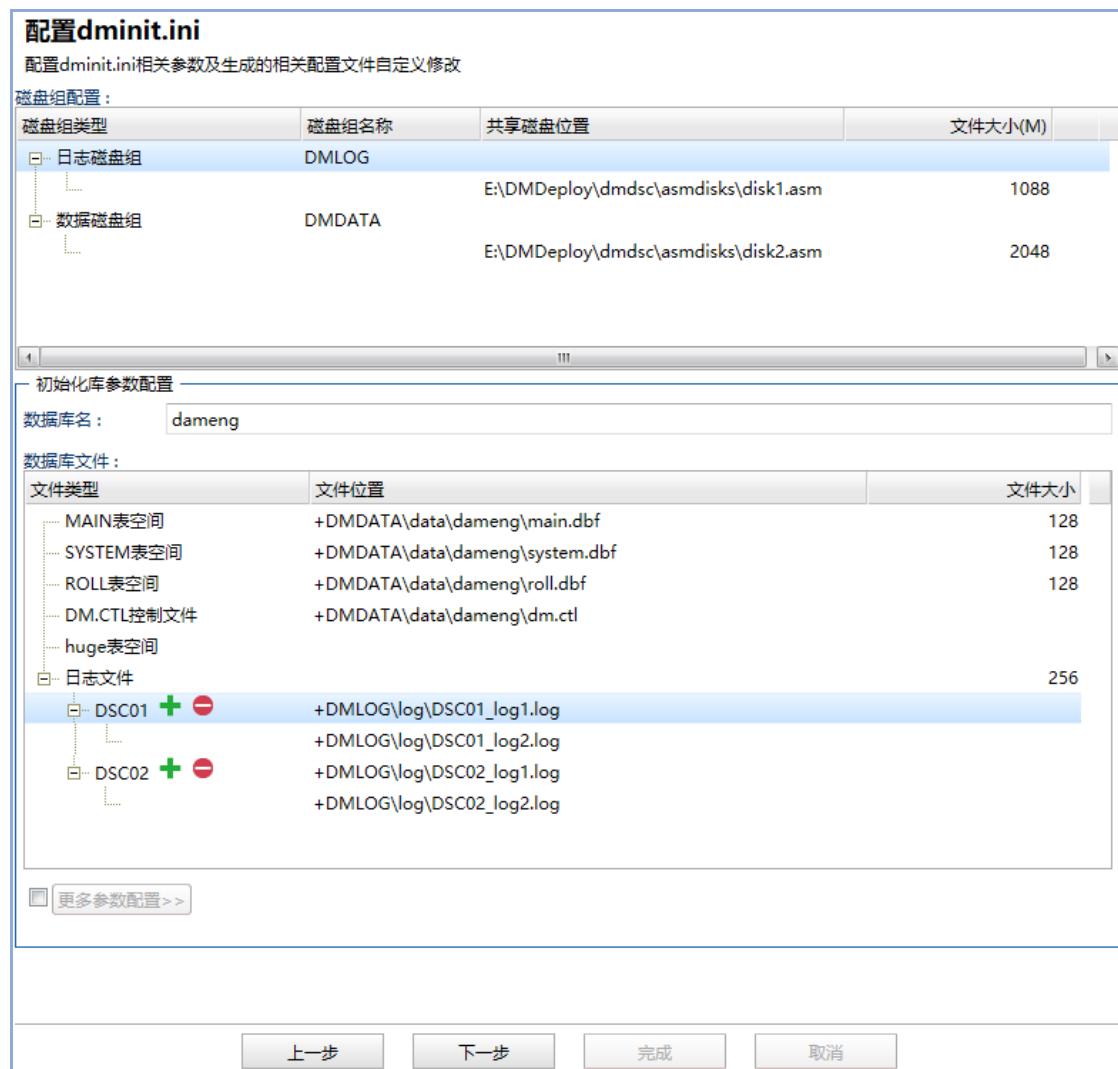


图 6.33 配置 DM.INIT.ini

图 6.34 上方部分配置磁盘组相关信息，下方部分配置初始化参数，除配置 DMDSC 文件路径信息外，也可以通过“更多参数配置”配置更多的初始化参数。

7. dmimal.ini 参数配置

参见章节 [6.4.3.1.1 4.dmimal.ini 参数配置](#)。

8. dmarch.ini 参数配置

参见章节 [6.4.3.1.1 5.dmarch.ini 参数配置](#)。

DMDSC 归档可选择是否配置，且可以选择是否配置“远程归档”，具体的 DMDSC 归档配置规则参见《DM8 共享存储集群》。

9. 配置监视器 dmcssm

如下图所示：

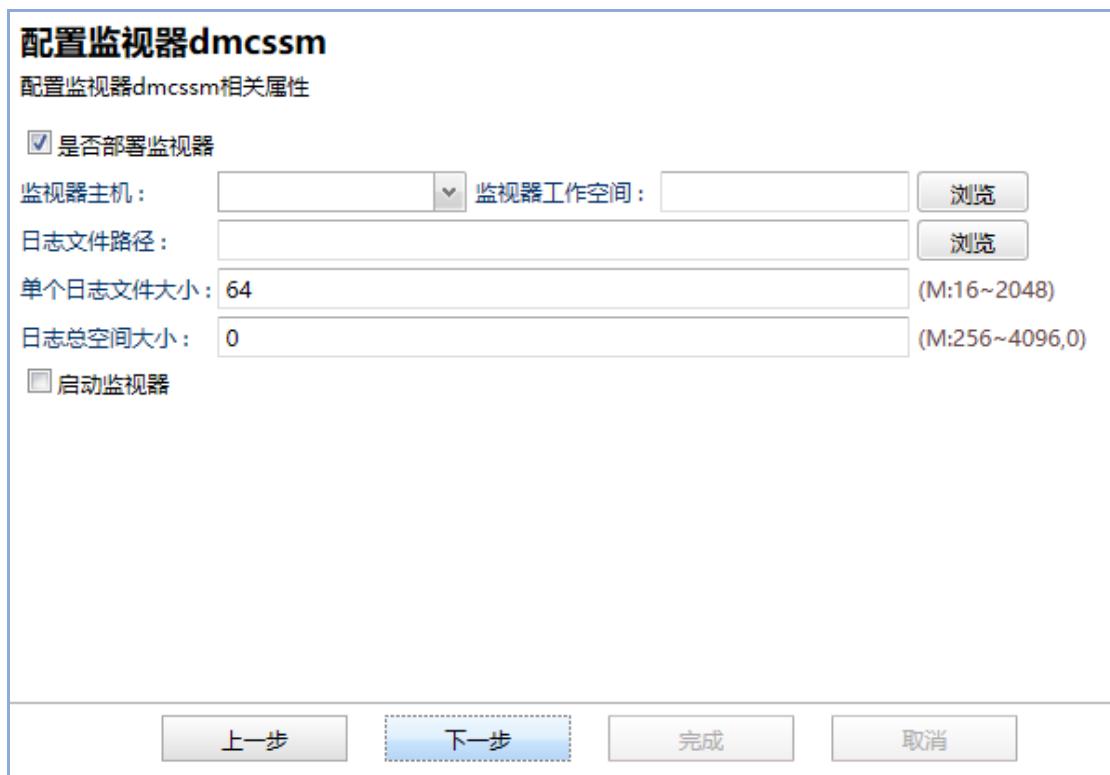


图 6.34 配置监视器 dmcssm

配置 dmcssm 相关参数，详细参见《DM8 共享存储集群》。

点击“下一步”

10. 上传服务器文件

参见章节 [6.4.3.1.1 6.上传服务器文件](#)。

11. 部署信息预览

参见章节 [6.4.3.1.1 7.部署信息预览](#)。

12. 执行部署

参见章节 [6.4.3.1.1 8.执行部署](#)。

6.4 数据迁移

DEM 数据迁移通过 Web 页面方式实现，功能同桌面版数据迁移工具 DTS。用户通过 Web 页面配置数据源信息、迁移对象以及迁移策略，DEM 服务器根据这些配置信息将源数据库中的待迁移对象迁移至目标数据库，使在完成数据库迁移的过程中不需要安装客户端，仅通过打开的一个 Web 页面进行配置便可方便的完成数据库的迁移。

6.5.1 系统结构

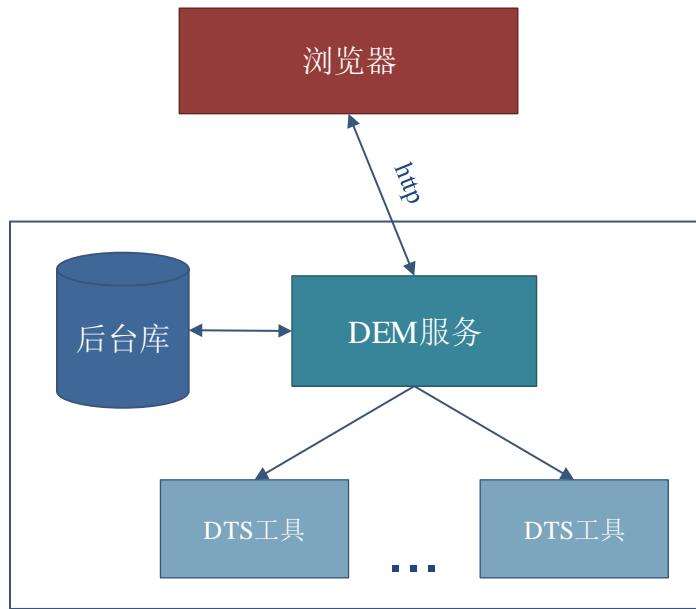


图 6.35 数据迁移系统结构图

6.5.2 实现原理

Web 版 DTS 通过 DEM 调用 DTS 工具（命令行版 DTS 工具）实现。

用户通过前端配置迁移数据源、迁移对象和迁移策略信息，配置完成后 DEM 将配置信息存储在 DTS 配置文件中（xml 文件，桌面版 DTS 工具本身用于保存迁移信息，且桌面版 DTS 调度执行方式和命令行版 DTS 工具都通过解析该配置文件执行迁移），DEM 启动 DTS 工具，将配置文件传给 DTS 工具，DTS 工具解析配置文件，根据解析信息执行迁移操作，并将迁移结果发给 DEM，DEM 将迁移结果返回给前端，并保存在 DEM 后台库中。

6.5.3 功能介绍

数据迁移工具提供的功能有以下几个：

- 迁移管理
- 迁移配置
- 迁移执行

6.5.3.1 迁移管理

桌面版 DTS 在 DTS 工具退出后，迁移执行信息丢失，但 Web DTS 在浏览器关闭时，由于其通过 DEM 服务执行，因此只要 DEM 服务一直在，DTS 便会在后台一直执行到结束，且由于 DEM 会将执行结果保存在 DEM 后台库中，用户重新打开浏览器时，也可查看 DTS 执行信息，因此 Web 版 DTS 多了一个最近执行迁移结果的管理功能。

如下用图标来表示上一次执行结果：

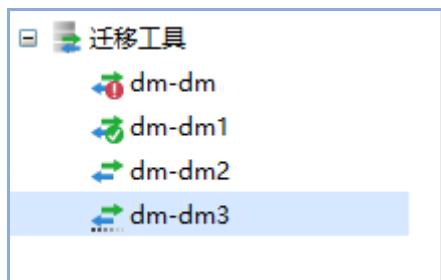


图 6.36 迁移管理

其中 表示执行失败， 表示执行成功， 表示原始的状态， 表示正在执行。

打开迁移可以查看最近一次执行的具体信息；右键菜单可以查看最近执行迁移结果；“新建迁移”或“修改迁移”可以设置查看最近执行结果的次数。

“修改迁移”如下图所示：

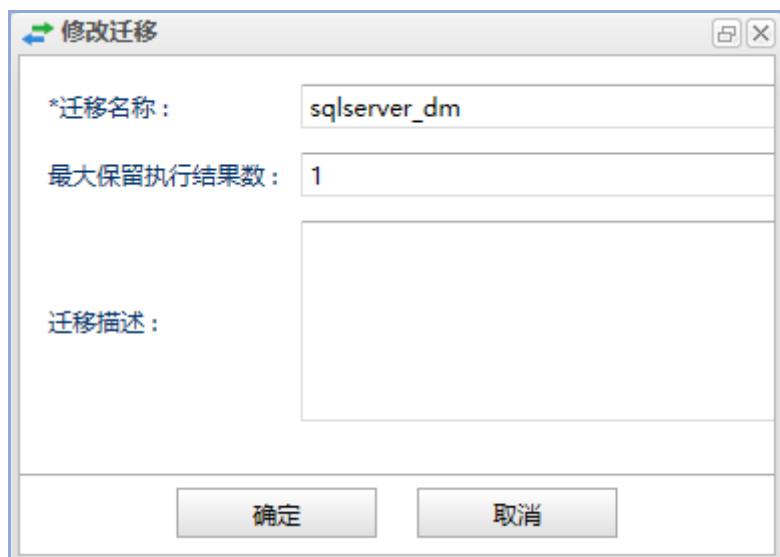


图 6.37 修改迁移

设置“最大保留执行结果数”。

其他如新建迁移、修改迁移和删除迁移功能与桌面版 DTS 一致。

6.5.3.2 迁移配置

迁移配置功能与桌面版 DTS 一致，详细请参见桌面版 DTS 联机帮助手册。

6.5.3.3 迁移执行

迁移执行在用户使用上与桌面版 DTS 一致，但在执行方式上 Web 版 DTS 实际是调用命令行版 DTS，且在结果反馈上，是通过消息推送方式反馈到前端的，有时会因网络等问题，不能推送到前端，但由于迁移结果保存到了 DEM 后台库，故可通过刷新手动获取迁移结果。

6.6 PROXY 管理

参见《DM8 集群代理使用手册》。

6.7 批量执行

DEM 的批量执行功能包括批量执行 SHELL 脚本和批量执行 SQL 语句，它是 dmagent 的 command 功能和 sql 功能的客户端实现方式。

6.7.1 系统结构

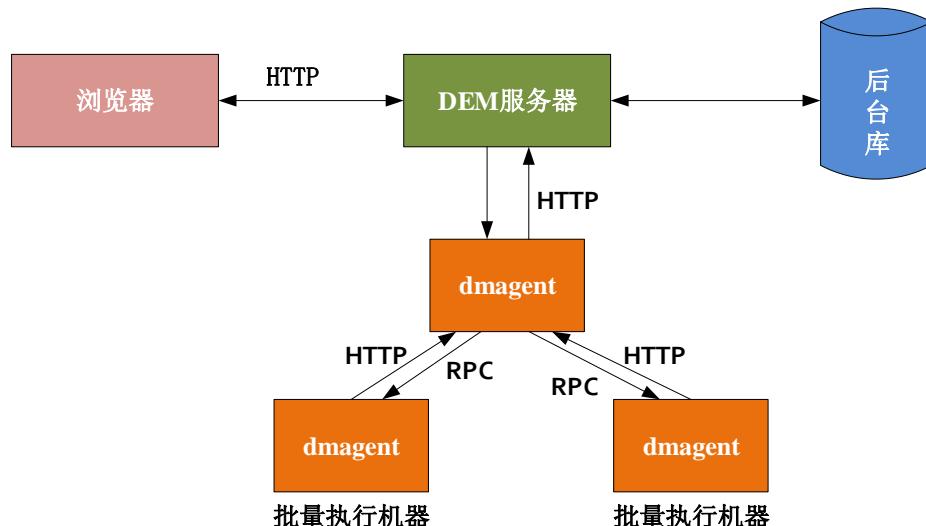


图 6.38 批量执行系统结构图

6.7.2 实现原理

批量执行分为四层：前端浏览器、DEM 服务、DEM 拉起的 dmagent 以及要批量执行机器的 dmagent。

用户通过前端浏览器配置批量执行的相关配置后，DEM 服务按配置信息，创建批量执行任务，并拉起 DEM 端 dmagent，将批量执行任务分发到各机器上执行，并收集执行日志信息发送到 DEM 服务，最终保存到 DEM 后台库。

批量执行采用先配置后执行的方式，以向导式方式引导用户配置批量执行的相关配置内容，然后 DEM 拉起中心 dmagent，其通过 RMI 方式访问远程机器的 dmagent，将具体的批量执行任务例如用户名、密码、超时、执行 SHELL 脚本或 SQL 语句等分发给 dmagent 执行，这样既能实现集中式配置、管理、执行，实现只需在 DEM 上操作就可以完成所有批量任务的执行，使得在没有依赖关系前提下，多台机器可以并发执行批量任务。

6.7.3 功能介绍

参见 [6.7 批量执行](#)。

6.8 升级管理

DEM 的升级管理功能包括升级服务器、KEY 和文件功能，它是 dmagent 的 REPLACE 功能的客户端实现方式。

6.8.1 系统结构

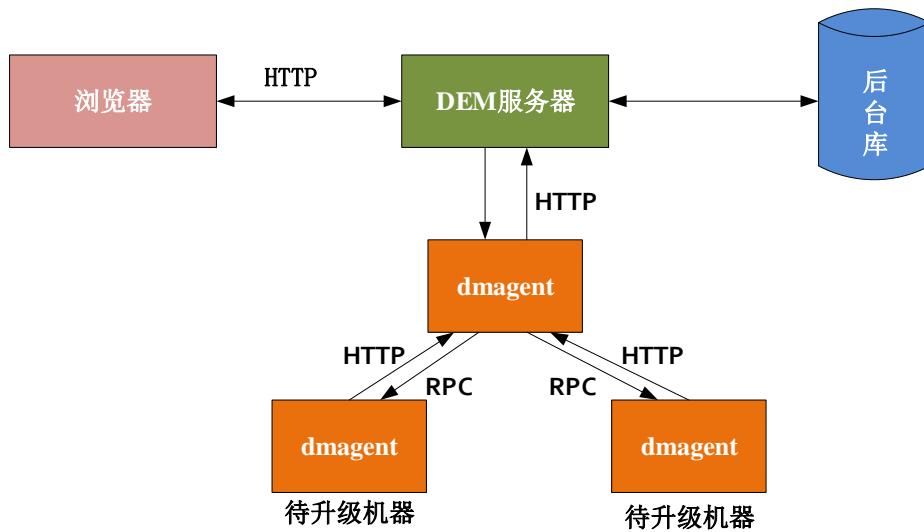


图 6.39 升级管理系统结构图

6.8.2 实现原理

升级管理分为四层：前端浏览器、DEM 服务、DEM 拉起的 dmagent 以及要待升级主机的 dmagent。用户通过前端浏览器配置 REPLACE 功能的相关配置后，DEM 服务按配置信息，创建 REPLACE 任务，并拉起 DEM 端 dmagent，将升级任务分发到各机器上执行，并收集执行日志信息发送到 DEM 服务，最终保存到 DEM 后台库。

升级管理采用先配置后执行的方式，以向导方式引导用户配置升级管理的相关配置内容，然后 DEM 拉起中心 dmagent，其通过 RMI 方式访问远程机器的 dmagent，将具体的升级管理任务例如目录，所有者，组，权限等分发给 dmagent 执行，这样既能实现集中式配置、管理、执行，实现只需在 DEM 上操作就可以完成所有升级任务的执行，使得在没有依赖关系前提下，多台机器可以并发执行升级任务。

6.8.3 功能介绍

6.8.3.1 升级管理

升级管理根节点下是已创建的升级管理任务。升级管理节点的图标可表示当前升级管理的状态，如下表所示：

表 6.1 升级管理节点图标与状态对应表

| 图标 | 状态 |
|----|-------|
| | 配置进行中 |
| | 升级进行中 |
| | 升级完成 |
| | 升级失败 |

双击升级管理节点，可打开当前升级状态对应的升级面板。

在升级管理根节点和升级管理节点右键点击，在弹出的菜单中进行创建、修改、查看、和除等功能。

6.8.3.2 配置并执行升级

1. 新建

升级管理根节点的右键菜单，点击新建升级管理，弹出“新建升级管理”对话框

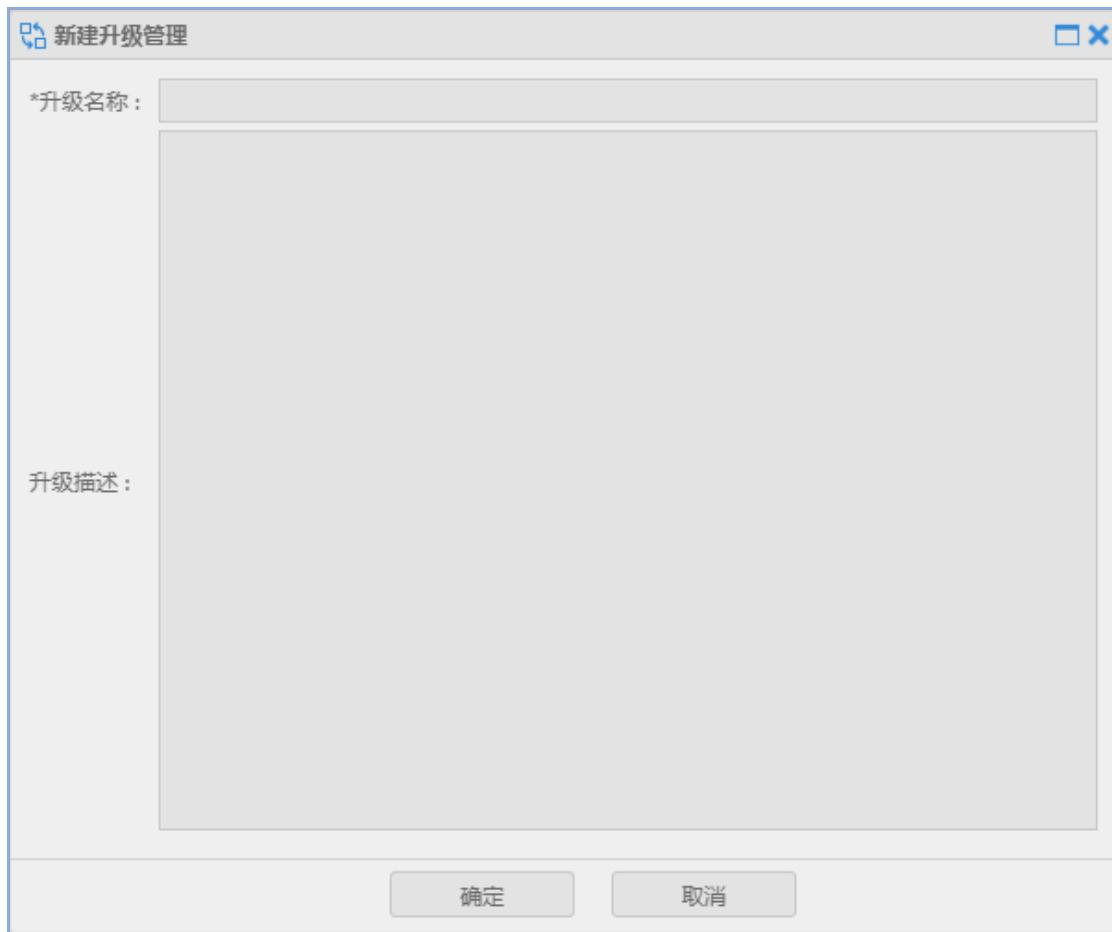


图 6.40 创建升级管理对话框

输入升级名称和升级描述，点击确定，将创建升级管理节点并打开升级面板。

2. 选择升级类型

可选择升级类型：数据库，数据库 KEY，文件。如下图所示：



图 6.41 升级类型选择界面

选择升级类型后，点击“下一页”进入“升级”配置页。

3. 配置

用户可在当前配置页面配置升级的相关参数，配置页面如下所示：

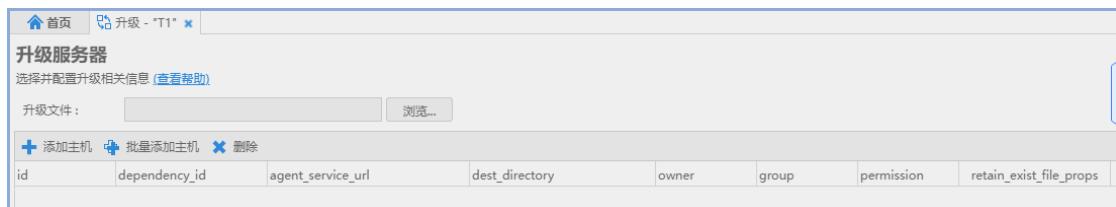


图 6.42 升级管理配置界面

配置页面功能说明：

- 1) 升级文件：待升级文件。上传升级文件，目前只支持 zip 压缩文件，压缩文件内部文件路径必须与待升级文件的相对路径（相对与升级路径）一致。
- 2) 添加主机：增加一条主机的升级配置。
- 3) 批量添加主机：通过资源选择对话框，批量添加待升级主机。如下图所示：

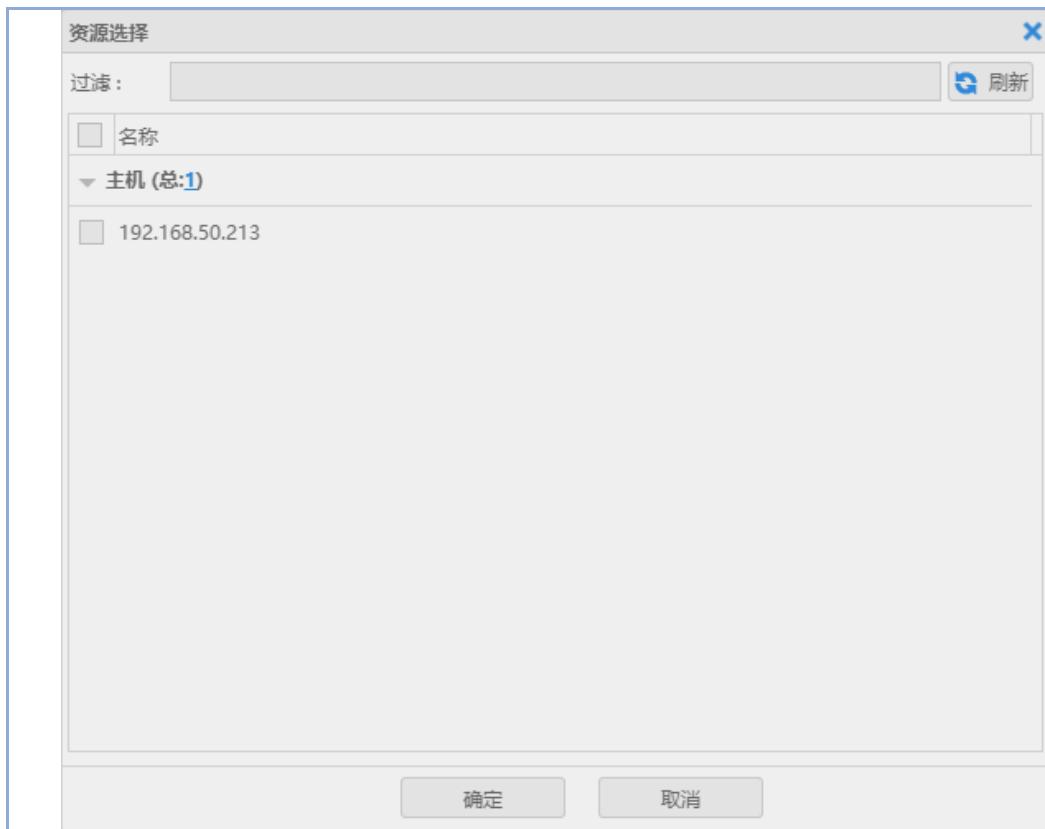


图 6.43 升级管理资源选择对话框

- 4) 删除：删除一条主机的升级配置。

- 5) 升级配置参数说明，如下表所示：

表 6.2 升级管理参数配置说明

| 参数 | 描述 |
|---------------|--------------------------------------|
| id | 执行 id，不可重复(可为空) |
| dependency_id | 此执行依赖的其他执行 id 集合，格式:id1,id2,...(可为空) |

| | |
|-------------------------|---|
| agent_service_url | 目标主机的 dmagent 服务地址 |
| dest_directory | 目标路径(服务器所在路径) |
| owner | 文件所有者(Windows 忽略此参数) |
| group | 文件所属组(Windows 忽略此参数) |
| permission | 文件权限, 默认值 755(Windows 忽略此参数) |
| retain_exist_file_props | 是否保留已存在文件的属性(所有者、组、权限), 默认保留(Windows 忽略此参数) |

配置完成后, 点击“下一页”进入详情总览。

4. 详情总览

当前页面显示升级文件和升级目标的相关信息, 如下图所示:

The screenshot shows the 'Upgrade Summary' page. At the top, there are tabs for 'Home' and 'Upgrade - "T1"'. The main content area has a title 'Upgrade Summary' and a sub-section 'Display upgrade resource related information ([Export Upgrade Summary](#))'. Below this, there are two sections: 'Upgrade File Information' and 'Upgrade Target Information'. The 'Upgrade File Information' section shows a file named 'bin.zip' (48.94MB, SHA256: 4BC76B69CAD0CE196ECF34B650BDE05443F1BDC48B5121CB2A2CDD6AB5D13352). The 'Upgrade Target Information' section shows one target with ID: 'id: dependency_id: agent_service_url: rpc://192.168.50.213:6364 dest_directory: D:\dmdbms\bin2'.

图 6.44 升级详情总览界面

点击“升级”按钮, 开始升级。

5. 升级

当进入升级页面后, 升级日志将显示升级过程中产生的日志。如下图所示:

The screenshot shows the 'Execute Upgrade Task' page. At the top, there are tabs for 'Home' and 'Upgrade - "T1"'. The main content area has a title 'Execute Upgrade Task' and a sub-section 'Execute upgrade task completed'. Below this, there is a section for 'Upgrade Log' containing the following log entries:

```

2022-03-24 14:27:18 - INFO
start check configuration...
2022-03-24 14:27:18 - INFO
start progress echo service on 1234
2022-03-24 14:27:18 - INFO
start replace tasks...
2022-03-24 14:27:18 - INFO - [0] - rpc://192.168.50.213:6364
(1/5) start uploading file...

```

图 6.45 执行升级任务界面

升级日志界面提供了相关功能按钮, 如下所示:

表 6.3 升级日志页面功能按钮

| 名称 | 描述 |
|------|---|
| 停止 | 点击按钮  ，停止升级功能。 |
| 导出日志 | 点击按钮  ，导出升级日志。 |
| 日志级别 | 通过下拉列表选择日志级别，对日志进行过滤。 |
| 任务编号 | 通过下拉列表选择任务编号，对日志进行过滤。 |

6.9 迁移评估

迁移评估帮助用户评估迁移数据库之间的兼容性，迁移数据的规格，迁移风险。评估内容包括但不限于：源库信息采集、目的库配置推荐、对象兼容情况、SQL 语句兼容情况和程序兼容情况。用户可以根据评估结果对源库对象的兼容度、改造工作量以及迁移成本拥有直观的了解。

6.9.1 系统结构

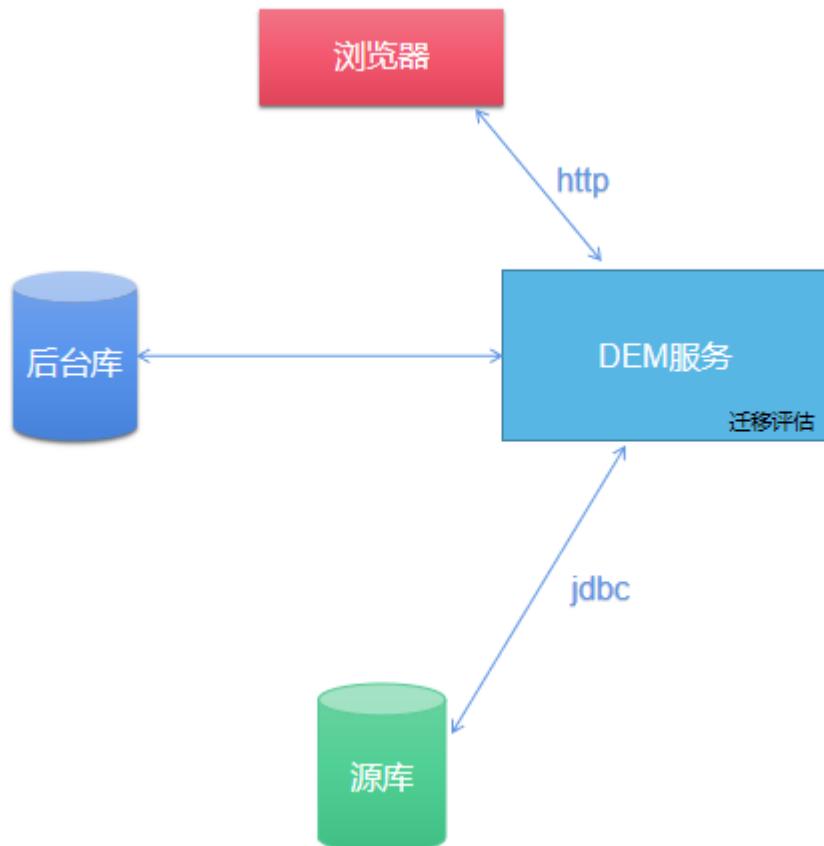


图 6.46 迁移评估系统结构图

6.9.2 实现原理

1. 用户通过浏览器登录 DEM 服务创建迁移评估。
2. 用户填写源库地址、用户名、密码等通过 JDBC 驱动连接源库并选择迁移对象，同时 DEM 服务将迁移评估相应配置信息储存在后台库。
3. 对用户选中对象创建评估任务并执行。任务具体分为：
 - 1) 获取源库信息。包括不限于源库版本，大小写敏感，页大小，编码。
 - 2) 对对象进行兼容性评估。具体流程：从源库获取对象 DDL 语句，用 DM 语法分析器解析源 DDL；如果解析通过则对象完全兼容；解析不通过，将源 DDL 转化成目的 DDL，DM 语法解析解析目的 DDL，如果解析通过则对象转化后兼容，间隙不通过则对象不兼容。
 - 3) 对 SQL 语句进行兼容性评估。SQL 语句可以从源库中获取历史执行 SQL，也可以通过上传的文件中获取，用 DM 语法分析器解析。如果解析通过，则 SQL 语句完全兼容；解析不通过则 SQL 语句不兼容；若出现 SQL 语句不完整等特殊情况则显示兼容状态未知。
 - 4) 对程序进行兼容性评估。通过扫描程序文件，分析程序中所使用的源库的接口信息，对比目的库，分析兼容性。
- 评估任务信息和执行结果都将储存在后台库。
4. DEM 服务根据后台库数据生成评估报告显示在浏览器上。

6.9.3 功能介绍

迁移评估功能包括了达梦数据库之间的迁移评估和不同数据库到达梦数据库的迁移评估。用户可以根据自己需求指定源库和源库中具体对象进行评估。评估内容包括但不限于：源库信息采集，目的库配置推荐，对象兼容情况、SQL 语句兼容情况和程序兼容情况。

评估结束后可在线查看报告，也可下载 HTML 格式的报告文件保存在本地查看。在线查看报告时可以对不兼容的历史执行 SQL 和目的 DDL 在线修改并分析修改后语句的兼容性；同时支持导出 DDL 语句和 SQL 语句文件。

6.9.3.1 迁移评估管理

迁移评估根节点下是已创建的迁移评估节点。迁移评估节点的图标可表示当前迁移评估任务的状态，如下表所示：

表 6.4 迁移评估节点图标与状态对应表

| 图标 | 状态 |
|----|-------|
| ① | 配置进行中 |
| ② | 评估进行中 |
| ③ | 评估完成 |
| ④ | 评估失败 |

右键点击迁移评估根节点和迁移评估节点，可选择新建、修改、删除和刷新等功能。

双击迁移评估节点，可打开当前迁移评估对应状态的面板。

6.9.3.2 迁移评估配置与执行

1. 新建

右键点击迁移评估根节点，选择新建迁移评估，弹出“新建迁移评估”对话框，如下图所示：

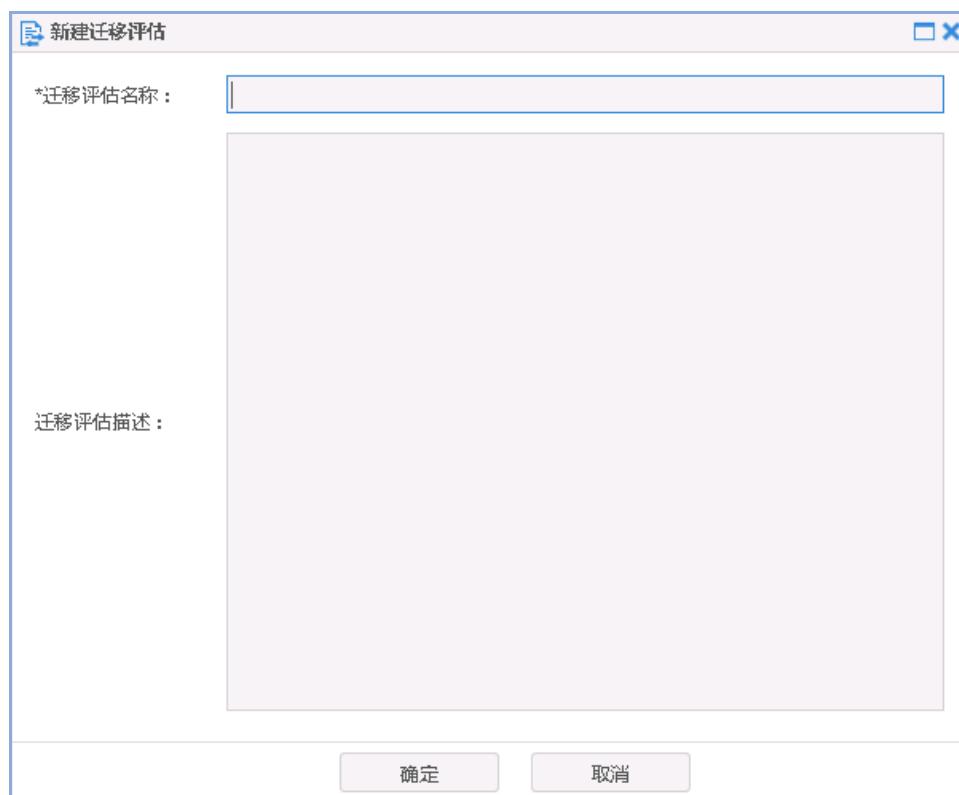


图 6.47 新建迁移评估对话框

输入迁移评估名称和迁移评估描述，点击确认，创建迁移评估并打开“选择迁移评估方式”界面。

2. 选择迁移评估向导

如下图所示，当前页面展示了 DEM 当前支持的所有迁移评估类型。

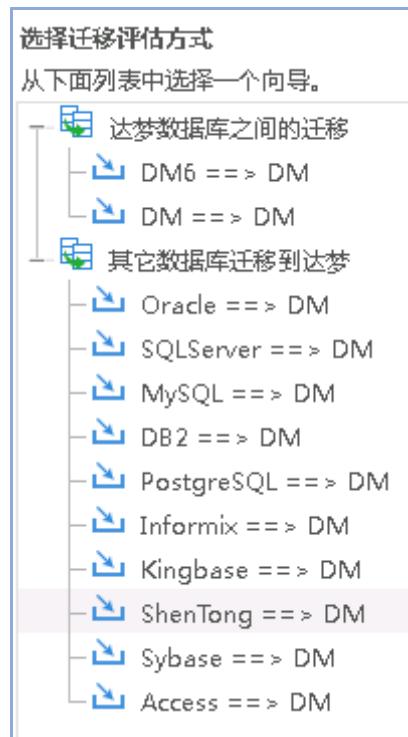


图 6.48 迁移评估类型选择界面

从树列表选择一个迁移评估向导，向导会一步步地引导用户完成操作。

点击下一步，进入“选择评估内容”界面。

3. 选择评估内容

在当前页面选择要评估的内容，可以选择评估对象兼容性，评估 SQL 兼容性，评估程序兼容性。不同的数据源支持的评估内容不同。具体如下图：

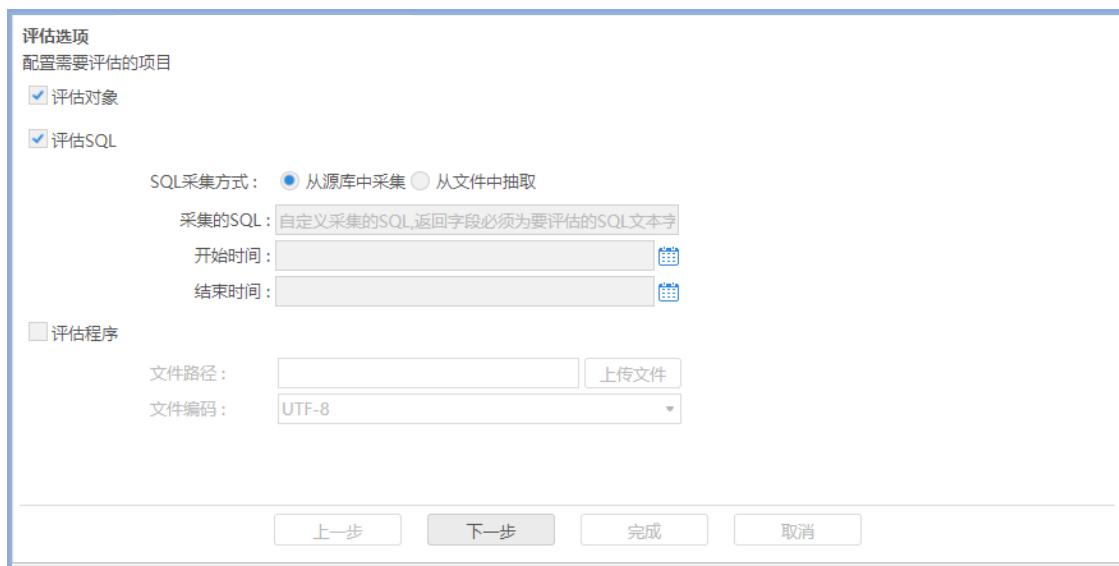


图 6.49 评估选项

1. 评估对象

从源端数据库获取数据库对象信息，评估源端数据库对象对应 DM 对应版本的兼容性。

源端数据库对象包括表，视图，序列，存储过程/函数，包，类，自定义类型，物化视图，触发器，同义词等，根据数据源不同，支持的对象类型不同。

2. 评估 SQL

获取 SQL，评估 SQL 与对应 DM 对应版本的兼容性。

SQL 来源包括以下几种方式：

1) 从源库中采集

从源端数据源直接通过系统表或视图获取 SQL 执行历史进行评估

2) 从文件中抽取

支持从以下几种文件格式中抽取 SQL：

(1) 支持从文本文件抽取 SQL 进行评估

文本文件中 SQL 分隔符可以自定义，默认按 (\$\$) 分隔。例如：

```
Select 1 from dual;
$$
Select 2 from dual;
```

(2) 支持从应用配置的 MYBATITS 的 SQLMAP 配置文件抽取 SQL 进行评估

如果多个 SQLMAP 配置文件之间存在相互引用关系，只需同时上传这些文件即可。

(3) 支持从应用配置的 IBATITS 的 SQLMAP 配置文件抽取 SQL 进行评估

3. 评估程序

评估是否支持业务代码中使用的 JAVA 接口, OCI 接口和 Pro*C 接口。

4. 输入数据源连接信息

在当前页面填写迁移评估的源数据库信息, DEM 根据填写的信息连接源数据库。

不同数据库连接需要的数据库信息不完全一致, 所以根据源数据库的不同, 打开的“数据源”界面也有所不同。不同“数据源”界面介绍如下:

1) DM6



图 6.50 DM6 数据源界面

根据源数据库信息配置好相关参数。点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。如果选择<默认>，则连接到 SYSTEM 库上。使用默认驱动，使用 DM 安装程序自带的驱动连接；指定驱动，打开指定驱动对话框，使用用户指定的驱动连接数据库服务器。如下图所示：

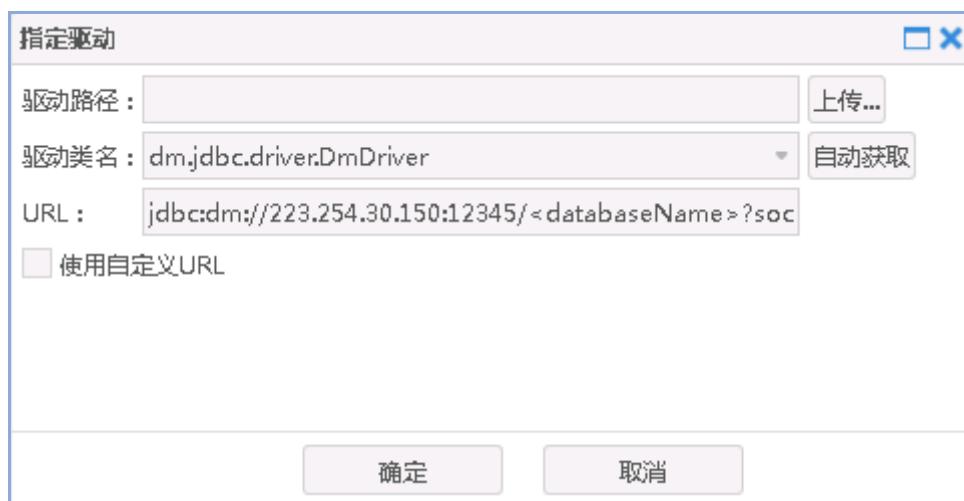


图 6.51 DM 指定驱动

在指定驱动对话框中输入正确的驱动信息。驱动路径，输入 JDBC 驱动文件路径，或点击上传按钮上传文件；驱动类名，输入 JDBC 驱动文件的 java 类名，点击自动获取按钮，会自动在下拉框中列出可以供用户选择的类名；URL，输入 JDBC 连接字符串；使用自定义 URL，选中该选项后，用户可对 URL 的输入框内容进行自定义修改。

2) DM8

DM 数据源界面提供了常规与高级两种数据库连接方式。

常规模式。常规模式需要的数据库信息与 DM6 的大体类似，与之相比，不用填写数据库名，且新增了使用数据压缩模式的选择。如下图所示：

常规

高级

主机名： localhost

端口： 5236

验证方式： 达梦服务器验证

用户名： SYSDBA

口令： *****

使用数据压缩模式

使用默认驱动 指定驱动...

(注：给定的默认驱动版本为8.1,建议通过“指定驱动”修改为所连接数据库自带的驱动)

图 6.52 DM 数据源常规模式界面

高级模式。需要填写的信息如下图所示，其中，全局 MPP——建立的会话为全局会话；局部 MPP——建立的会话为局部会话

常规

高级

SSL密钥文件： 浏览

SSL验证密码：

MPP登录 全局MPP 局部MPP

图 6.53 DM 数据源高级模式界面

3) Oracle



根据源数据库信息配置好相关参数。

4) SQLServer



根据源数据库信息配置好相关参数。点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。如果选择<默认>，则连接到 master 库上。

5) MySql

MySql 数据源界面需要填写的数据库信息与 SQLServer 的一致。其中点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。如果选择<默认>，则连接到 mysql 库上。

6) DB2



图 6.56 DB2 数据源界面

根据源数据库信息配置好相关参数。

7) PostgreSQL

PostgreSQL 数据源界面需要填写的数据库信息与 SQLServer 的一致。其中点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。如果选择<默认>，则连接到 tpcw 库上。

8) Informix



图 6.57 Informix 数据源界面

根据源数据库信息配置好相关参数。

9) Kingbase

Kingbase 数据源界面需要填写的数据库信息与 SQLServer 的一致。其中点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。

10) Shentong

Shentong 数据源界面需要填写的数据库信息与 SQLServer 的一致。其中点击刷新按钮获取已有的数据库，在数据库下拉列表中选择需要连接的库。

11) Sybase

数据源
请输入 Sybase 数据库信息。

| | |
|-------|-----------|
| 主机名： | localhost |
| 端口： | 5000 |
| 用户名： | sa |
| 口令： | |
| 数据库名： | <默认> |
| 字符集： | <默认> |

刷新 使用默认驱动 指定驱动...

图 6.58 Sybase 数据源界面

根据源数据库信息配置好相关参数。字符集——指定字符集来连接数据库

12) Access

数据源
请输入 Access 数据库信息。

| | | |
|-------|--|--------------------------------------|
| 数据库： | <input type="text"/> | <input type="button" value="浏览..."/> |
| 用户名： | <input type="text"/> | |
| 口令： | <input type="text"/> | |
| URL： | jdbc:odbc:driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ= | <input type="checkbox"/> 使用自定义URL |
| 文件编码： | GBK | |

图 6.59 Access 数据源界面

根据源数据库信息配置好相关参数。数据库——输入 Access 数据库文件；使用自定义 URL，选中该选项后，用户可对 URL 的输入框内容进行自定义修改。

源数据库信息填写正确，数据库连接成功后，点击下一步，进入“指定要评估的对象类型”界面。

5. 指定评估对象类型

在当前页面选择要评估的对象类型，如下图所示：

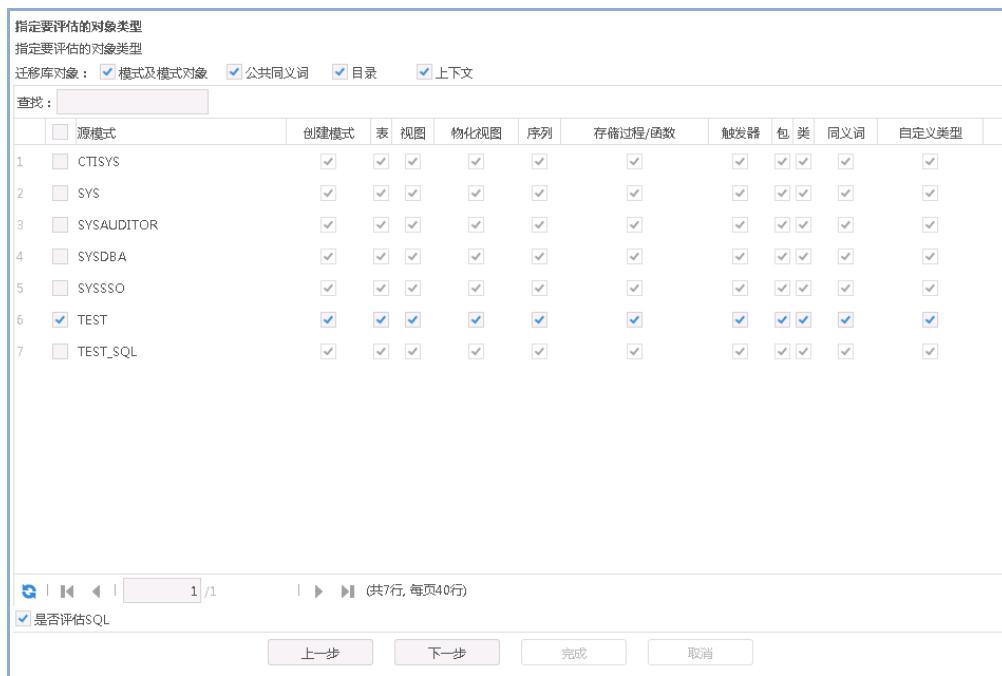


图 6.60 DM 指定要评估的对象类型界面

迁移库对象包括模式及模式对象、公共同义词、目录和上下文等，根据不同数据源，支持评估的对象类型不同。如上图为 DM 数据源支持评估的对象类型，包括模式及模式对象、公共同义词、目录和上下文；而下图为 SQLServer 数据源支持评估的对象类型，仅支持评估模式及模式对象。

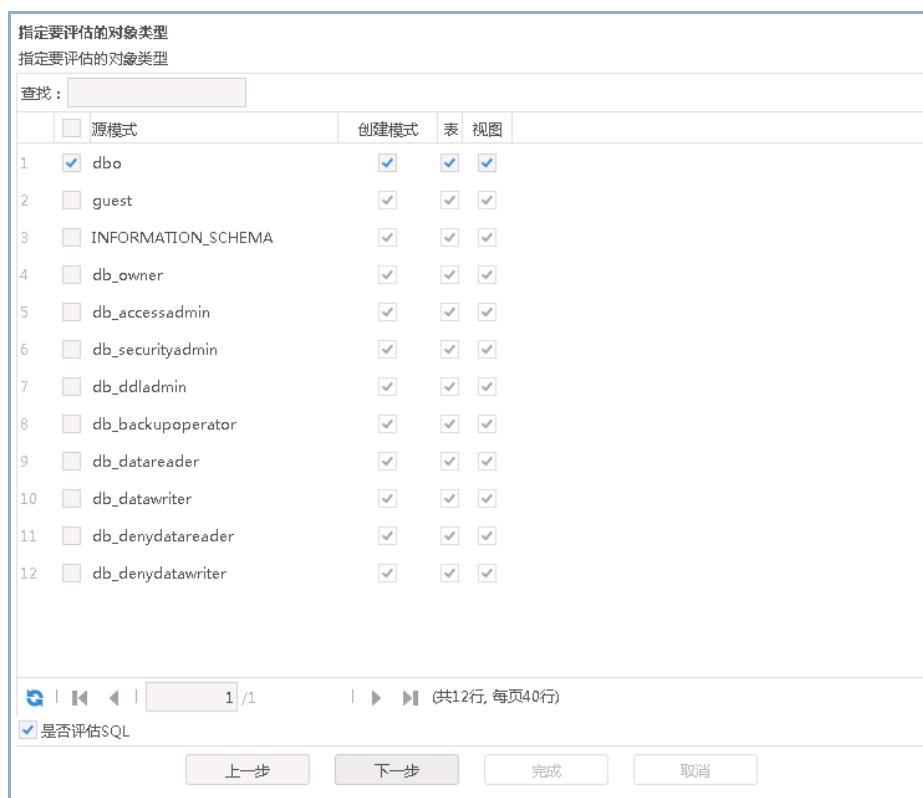


图 6.61 SQLServer 指定要评估的对象类型界面

模式及模式对象，包括模式、表、视图、物化视图、序列、存储过程/函数、触发器、包、类、同义词以及自定义类型。根据数据源不同，支持的模式及模式对象也不同。如上两图所示，以 DM 为源数据库时模式对象包括几乎所有对象，而以 SQLServer 为源数据库时，仅支持表和视图两模式对象。另外，如果源数据库为 MySQL，则不会显示“源模式”列；如果源数据库不支持序列，则不会显示“序列”列；只有源数据库是达梦时才会显示“存储过程/函数”列。

如果选择模式及模式对象，则在下面的表格中选择源模式、是否创建模式以及需要评估的迁移对象类型。

公共同义词，选择是否评估迁移公共同义词。

目录，选择是否评估迁移目录。

上下文，选择是否评估迁移上下文。

选定评估对象类型后，点击下一步，进入“指定要评估的对象”界面。

6. 指定评估对象

如下图，勾选要评估的对象。可点击操作查看被选对象的定义语言。

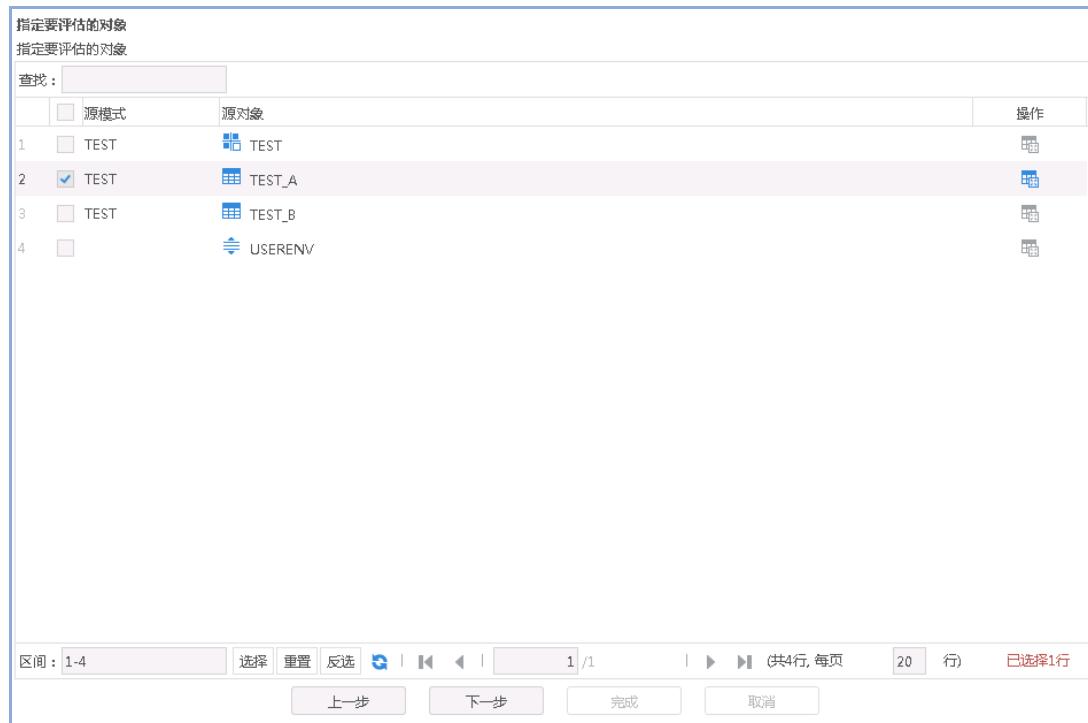


图 6.62 指定要评估的对象界面

指定要评估对象后，点击下一步，进入“详情总览”界面。

7. 详情总览

当前页面显示迁移评估对象的详细信息，如下图所示：

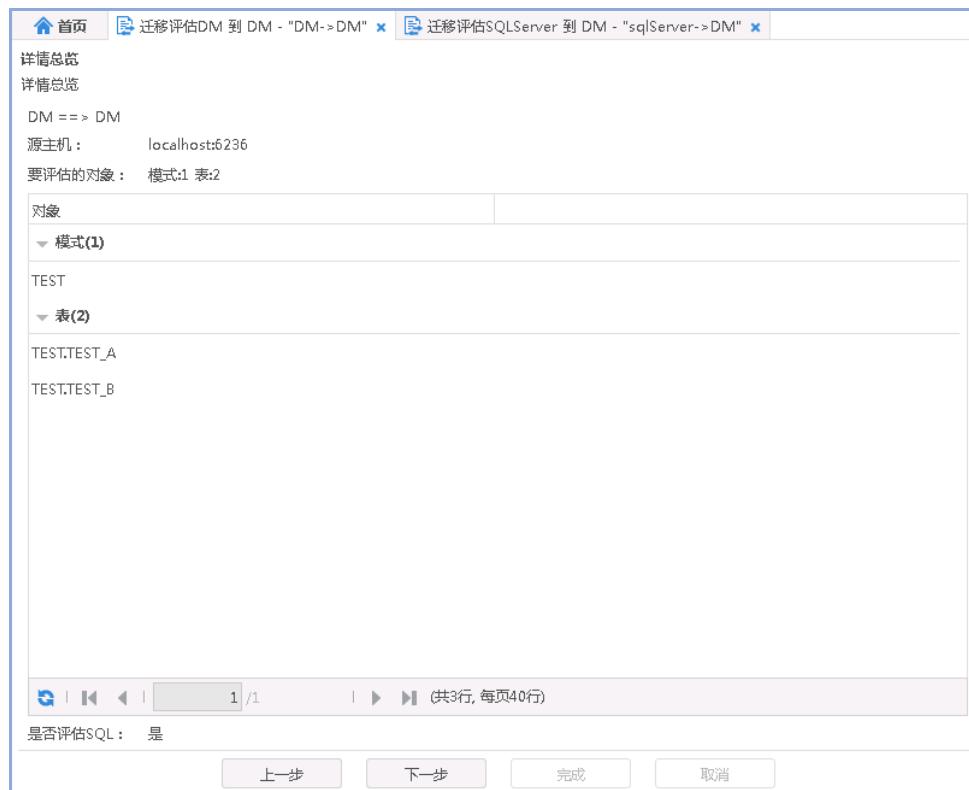


图 6.63 迁移评估详情总览界面

仔细阅读详情总览信息，确认配置无误后，点击下一步，开始迁移评估，并进入“达梦评估报告”界面。

8. 执行评估任务并获取评估报告

达梦评估报告页面包括源库概要、目的库推荐、对象兼容和 SQL 兼容 4 个部分，如下图所示：

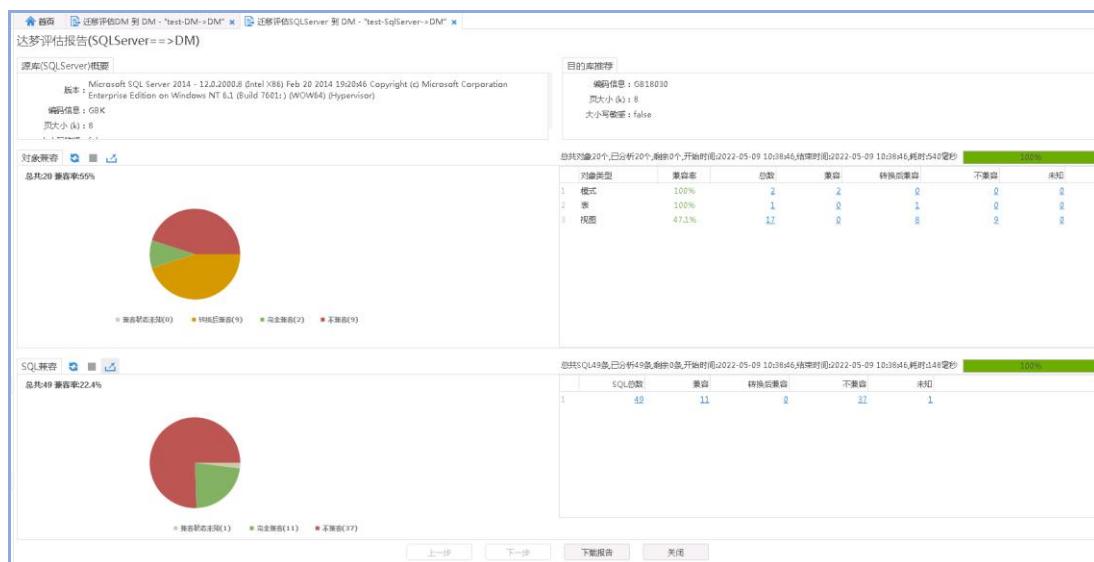


图 6.64 迁移评估报告界面

“源库概要”显示源数据库版本信息、编码信息、页大小和大小写敏感信息；“目的库推荐”显示根据评估推荐的目的库编码信息、页大小和大小写敏感信息。

“对象兼容”和“SQL 兼容”通过进度条展示评估进度。在评估未完成时可以通过暂停按钮停止评估对象或停止评估 SQL，可通过刷新按钮刷新对象或 SQL 的评估结果，在评估结束后可以通过对象兼容和 SQL 兼容旁的导出按钮分别导出对象 DDL 和 SQL，同时也可点击“下载报告”按钮下载评估报告。

数据库评估兼容结果包括兼容、转换后兼容、不兼容和未知四种兼容情况。“对象兼容”和“SQL 兼容”评估内容由兼容概要和兼容详情两部分组成。兼容概要，以扇形统计图简单展示各种兼容结果的占比情况。兼容详情，通过表格详细展示数据库各种对象类型对应的兼容结果。点击兼容详情表格中的一个兼容评估结果数字链接可打开评估详细对话框，如下图所示：

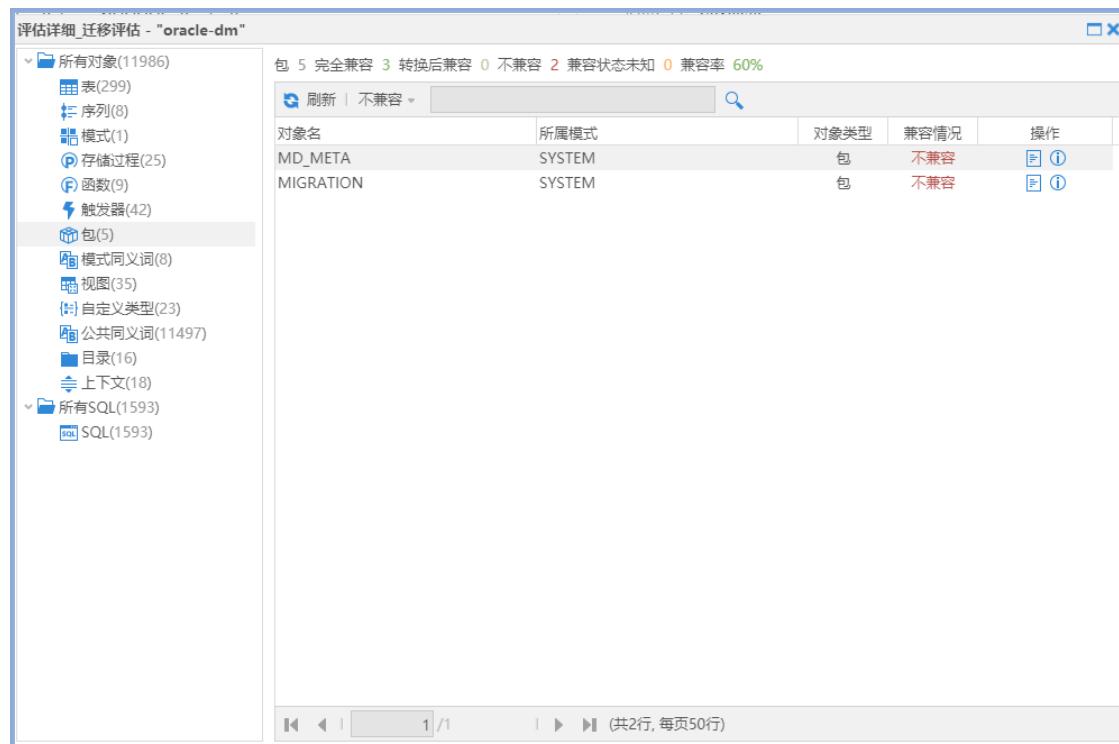


图 6.65 评估详细对话框

评估详细对话框记录了所有评估对象的评估结果与兼容详情。可以通过对象选择、兼容情况选择和筛选框查看特定对象的兼容评估情况。

可以通过“查看详细”操作，查看详细的评估详细信息，具体如下图：

评估对象DDL_包_MD_META

对像类型:包 对象名:MD_META 所属模式:SYSTEM 兼容情况:不兼容

原DDL 转换后DDL

```

1 CREATE PACKAGE "SYSTEM"."MD_META"
2 AS
3   FUNCTION get_next_id RETURN NUMBER;
4   -- Following code taken directly from www_flow_random from APEX
5   --
6   -- seed random function
7   procedure srand( new_seed in number );
8
9   function rand return number;
10  pragma restrict_references( rand, WNDS );
11
12  procedure get_rand( r OUT number );
13
14  function rand_max( n IN number ) return number;
15  pragma restrict_references( rand_max, WNDS );
16
17  procedure get_rand_max( r OUT number, n IN number );
18
19  function quote(catalog IN VARCHAR2, schema IN VARCHAR2, object IN VARCHAR2, connid LONG) RETURN
20  VARCHAR2;
21  END;
22  CREATE PACKAGE BODY "SYSTEM"."MD_META" AS
23
24    multiplier constant number := 22695477;
25    increment constant number := 1;
  "2^32"  constant number := 2 ** 32;
  "2^32"  constant number := 2 ** 32;

```

转换后不兼容原因:尝试转换失败(第19行,第36列[schema]附近有语法错误)

修改 校验 保存

图 6.66 评估对象

包括源对象 DDL/源 SQL，源与目的不兼容的原因。转换后对象 DDL/转换后 SQL，以及转换后还不兼容的原因，可以手动编辑转换后的 SQL，然后校验编辑后的 SQL 是否兼容，并可以保存编辑的 SQL。

也可以通过“修改建议”操作，查看不兼容对象/SQL/程序的修改建议。具体如下：

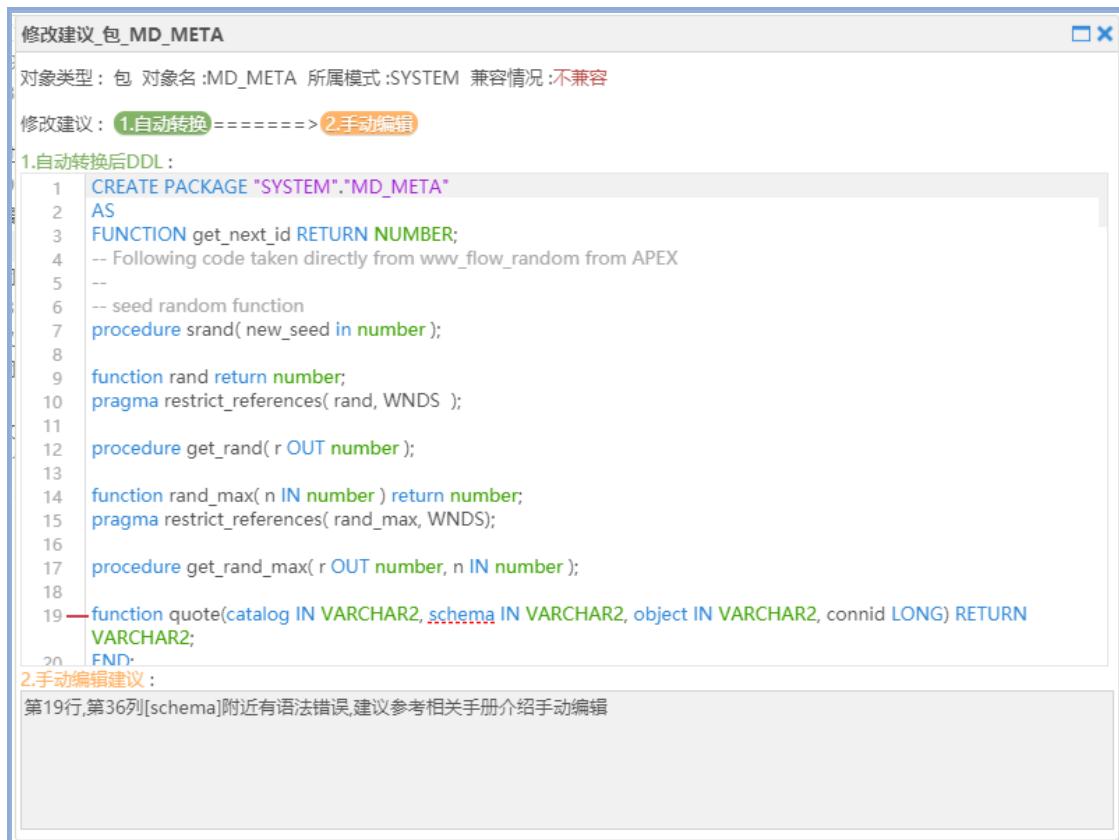


图 6.67 修改建议

通常建议先按自动方式转换对象 DDL/SQL，再对具体指出的转换后的对象 DDL/SQL 不兼容的具体位置进行针对性修改。

6.10 联机巡检

联机巡检在用户完成巡检配置后，立刻执行巡检任务，生成巡检报告以展示当前数据库运行情况。巡检的内容包括但不限于：数据库健康度、主机概况、数据库概况、性能概况、表空间、REDO 日志、归档日志、备份、用户与会话、运行日志、INI 配置和自定义任务等。可通过巡检报告进行数据分析与异常检查。

6.10.1 系统结构

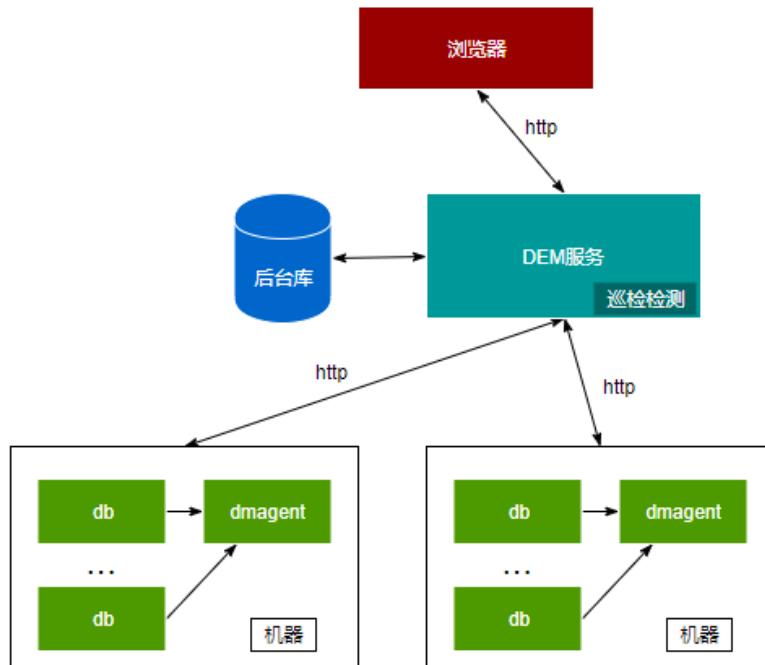


图 6.68 DEM 巡检系统结构图

6.10.2 实现原理

联机巡检通过两种方式获得相关信息。一 远程调用巡检目标数据库所在主机的 dmagent 的 service 模块中的远程方法获取主机信息；二 JDBC 数据库驱动直连巡检目标数据库，调用系统函数，查询系统表和动态视图等获取数据库信息。

可能涉及的 SQL 语句如下：

```

SELECT * FROM V$PROCESS;
SELECT * FROM V$SYSSTAT;
SELECT * FROM V$THREADS;
SELECT * FROM V$SESSIONS;
SELECT * FROM V$TASK_QUEUE;
SELECT * FROM V$ARCH_QUEUE;
SELECT * FROM V$TABLESPACE;
SELECT * FROM V$DATAFILE;
    
```

```

SELECT * FROM V$RLOG;

SELECT * FROM V$RLOGFILE;

SELECT * FROM V$ARCH_FILE;

SELECT * FROM V$DM_INI;

SF_BAKSET_BACKUP_DIR_ADD('DISK', ?);

SELECT * FROM V$BACKUPSET;

SELECT * FROM V$BACKUPSET_BKP;

SELECT SF_BAKSET_CHECK(?, ?);

SELECT * FROM SYSOBJECTS;

SELECT * FROM SYSUSERS;

SELECT * FROM V$SESSIONS;

SELECT * FROM V$SESSION_STAT;

SELECT * FROM V$LICENSE;

SELECT * FROM V$DEADLOCK_HISTORY;

SELECT * FROM V$SESSION_HISTORY;

SELECT * FROM V$instance_LOG_HISTORY;

SELECT * FROM V$DM_INI;

```

6.10.3 功能介绍

联机巡检客户端工具提供了对数据库及相关服务器运行时状况的快速巡检功能。

巡检的内容包括但不限于：一 数据库所在主机信息。操作系统、CPU、内存、磁盘、网络等信息；二 数据库信息。达梦数据库名称、版本、实例信息、数据文件、控制文件、日志、归档、备份、用户情况、集群状态等信息。除以上内容外，用户可以自定义巡检项目（需要提供有结果集的 SQL 语句）。

巡检报告分为四个维度：简报、详情-数据库维度、详情-巡检项维度、DM.INI 全部配置。除了在线查看巡检报告，还提供了 HTML 格式的巡检报告文件下载功能，方便本地保存和传送。

联机巡检还会对收集到的巡检内容做自动分析与异常检查，根据一些规则为数据库进行健康度评分，具体的评分规则如下表所示：

表 6.5 健康度评分规则

| 扣分项 | 扣分值 |
|-------------------------------|-----|
| 未配置归档 | 5 |
| 归档未设置磁盘空间限制 | 2 |
| 未检测到备份 | 10 |
| 有效备份文件超过一个月 | 10 |
| 有效备份文件超过一周 | 5 |
| 存在校验失败的备份 | 5 |
| 授权还有 30 天过期 | 60 |
| 授权还有 60 天过期 | 40 |
| 授权还有 90 天过期 | 20 |
| 用户口令已过期 | 5 |
| 用户口令还有 15 天过期 | 3 |
| 死锁个数大于 5 | 5 |
| 死锁个数大于 0 | 2 |
| 主机磁盘使用率 $\geq 90\%$ | 5 |
| 主机磁盘使用率 $\geq 80\%$ | 2 |
| CPU 核心数与 WORKER_THREADS 配置不合理 | 2 |
| 主机 cpu 使用率 $\geq 90\%$ | 5 |
| 主机 cpu 使用率 $\geq 80\%$ | 2 |
| 主机内存使用率 $\geq 90\%$ | 5 |
| 主机内存使用率 $\geq 80\%$ | 2 |
| 非自动扩展表空间使用率 $\geq 90\%$ | 10 |
| 非自动扩展表空间使用率 $\geq 80\%$ | 5 |
| BUFFER 配置不合理 | 2 |
| BUFFER 配置太小 | 2 |
| SORT_BUF_SIZE 小于 5M | 2 |

| | |
|-----------------------|----|
| SORT_BUF_SIZE 大于 100M | 3 |
| SORT_BUF_SIZEd 大于 50M | 2 |
| 数据库临时表空间未设置上限 | 1 |
| 临时表空间超过 10G | 5 |
| 临时表空间超过 5G | 3 |
| 数据库处于 MOUNT 状态 | 20 |
| 数据库处于 SUSPEND 状态 | 10 |

6.10.3.1 联机巡检管理

联机巡检根节点下已创建的联机巡检节点。联机巡检节点的图标可表示当前联机巡检任务的状态，如下表所示：

表 6.6 联机巡检节点图标与状态对应表

| 图标 | 状态 |
|----|-------|
| | 配置进行中 |
| | 巡检进行中 |
| | 巡检完成 |
| | 巡检失败 |

右键点击联机巡检根节点和联机巡检节点，可选择新建、修改、删除和刷新等功能。

双击联机巡检节点，可打开当前联机巡检对应状态的面板。

6.10.3.2 联机巡检配置与执行

1. 新建

右键点击联机巡检根节点，选择新建巡检，弹出“新建数据库巡检”对话框，如下图所示：

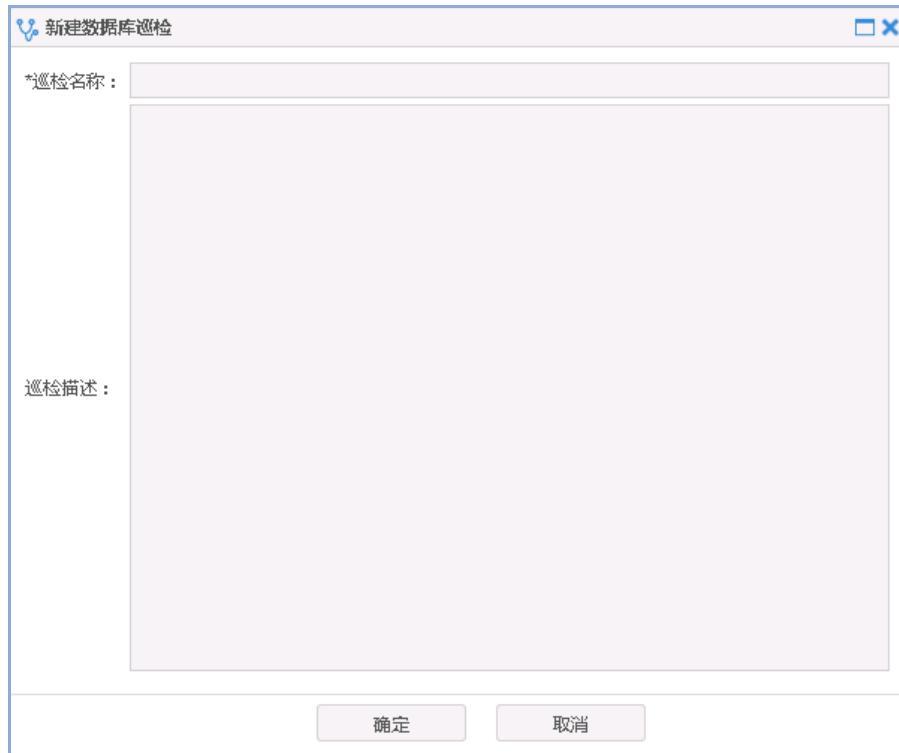


图 6.69 新建联机巡检对话框

输入联机巡检名称和联机巡检描述，点击确认，创建联机巡检并打开“配置巡检目标及巡检任务”界面。

4. 配置

用户可在当前配置页面配置巡检目标数据库和巡检目标对应的巡检任务。如下图所示：



图 6.70 联机巡检配置界面

联机巡检可对多目标进行多任务巡检。联机巡检任务分为内置任务和自定义任务。内置任务包括性能概况、表空间、Redo 日志、归档日志、备份、用户与会话、运行日志和 DM.INI 配置。添加目标数据库的同时，目标数据库自动添加这 8 个内置任务。自定义任务为用户根据特定需求设置添加的任务。

配置说明：

1) 添加目标数据库

点击“添加数据库”，添加一个默认数据库

点击“批量添加数据库”，通过“资源选择”对话框批量选择可巡检数据库，如下图所示：

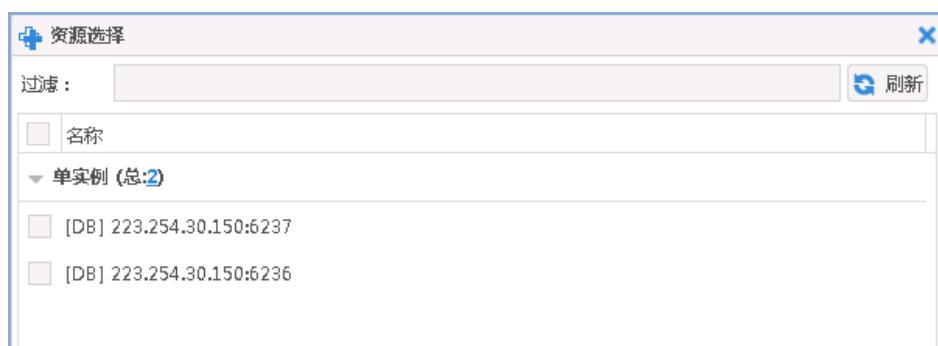


图 6.71 巡检资源选择对话框

2) 编辑数据库

点击数据库操作栏下拉按钮“编辑数据库”，打开“编辑巡检目标数据库”对话框，如下图所示，根据需求合理设置各参数。



图 6.72 编辑巡检目标数据库对话框

3) 删除数据库

点击数据库操作栏下拉按钮“删除数据库”删除指定数据库。

4) 添加自定义任务

点击数据库操作栏下拉按钮“添加自定义任务”，打开“编辑巡检目标数据库”对话框，通过 SQL 语句设置自定义任务，如下图所示：

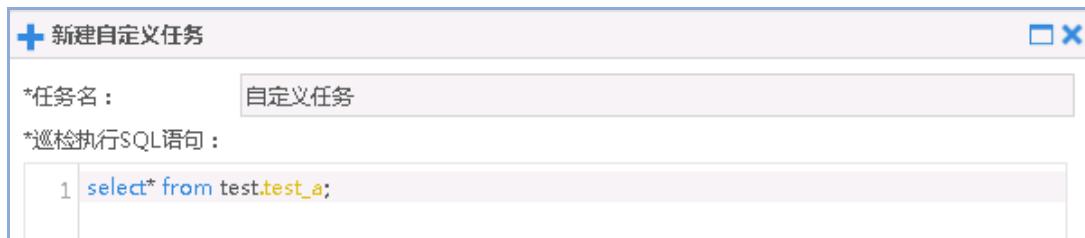


图 6.73 编辑巡检目标数据库对话框

5) 编辑任务

仅自定义任务可编辑，内置任务只可查看不可编辑。

双击自定义任务，或点击自定义任务操作栏下拉按钮“查看/修改任务”，打开任务对话框可编辑修改自定义任务。

6) 删除任务

点击任务操作栏下拉按钮“删除任务”删除指定任务。

7) 配置应用到所有数据库

点击数据库操作栏下拉按钮“配置应用到所有数据库”，打开“应用配置到所有数据库”对话框，选择需要应用到其它数据库的配置，点击确认将被勾选配置应用到其它所有数据库。如下图所示：



图 6.74 巡检配置应用到所有数据库对话框

巡检目标和巡检任务都配置好后，点击下一步，进入“详情总览”界面。

5. 详情总览

当前页面显示联机巡检详细信息，如下图所示，可点击“导出详情总览”下载总览信息。

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---------|----|---------|----|-----------|----|------|---|-----|---|--------|---|------|------------------------------|----|---|-------|--|----|---|------|--|--------|---------------------------|-------|----------------------------|
| 详细总览 | 显示数据库巡检的完整配置信息 (显示详细总览) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 取机巡检信息: | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 名称: | test | 描述: | | | | | | | | | | | | | | | | | | | | | | | | | |
| 巡检数据源简介: | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 巡检数据源列表: 223.254.30.150&237 223.254.30.150&236 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 巡检数据源详细列表: | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 数据库: | 223.254.30.150&237 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JDBC连接串: | jdbc:dm://223.254.30.150&237 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 用户名: | SYSDBA | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 密码: | ***** | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 备份路径: | d:\agent\服务地址: rpc://223.254.30.150&6364 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 归档查看天数: | 30 | 备份查看天数: | 30 | 死物查看天数: | 30 | 运行日志查看天数: | 30 | | | | | | | | | | | | | | | | | | | | |
| 巡检任务列表: | <table border="1"> <tr> <td>性能概况</td> <td>[SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE]</td> </tr> <tr> <td>表空间</td> <td>[SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE;</td> </tr> <tr> <td>Redo日志</td> <td>[SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE]</td> </tr> <tr> <td>归档日志</td> <td>[SELECT * FROM V\$ARCH_FILE]</td> </tr> <tr> <td>备份</td> <td>[SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)]</td> </tr> <tr> <td>用户与会话</td> <td>[SELECT * FROM SYSOBJECTS; SELECT * FROM SYSUSERS]; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$SESSION_STAT; SELECT * FROM V\$LICENSE;]</td> </tr> <tr> <td>死物</td> <td>[SELECT * FROM V\$DEADLOCK_HISTORY; SELECT * FROM V\$SESSION_HISTORY]</td> </tr> <tr> <td>运行日志</td> <td>[SELECT * FROM V\$SESSION_LOG_HISTORY]</td> </tr> <tr> <td>dimn配置</td> <td>[SELECT * FROM V\$DMJN1;]</td> </tr> <tr> <td>自定义任务</td> <td>[select * from testtest_a]</td> </tr> </table> | | | | | | | 性能概况 | [SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE] | 表空间 | [SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE; | Redo日志 | [SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE] | 归档日志 | [SELECT * FROM V\$ARCH_FILE] | 备份 | [SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)] | 用户与会话 | [SELECT * FROM SYSOBJECTS; SELECT * FROM SYSUSERS]; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$SESSION_STAT; SELECT * FROM V\$LICENSE;] | 死物 | [SELECT * FROM V\$DEADLOCK_HISTORY; SELECT * FROM V\$SESSION_HISTORY] | 运行日志 | [SELECT * FROM V\$SESSION_LOG_HISTORY] | dimn配置 | [SELECT * FROM V\$DMJN1;] | 自定义任务 | [select * from testtest_a] |
| 性能概况 | [SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 表空间 | [SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE; | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Redo日志 | [SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 归档日志 | [SELECT * FROM V\$ARCH_FILE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 备份 | [SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 用户与会话 | [SELECT * FROM SYSOBJECTS; SELECT * FROM SYSUSERS]; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$SESSION_STAT; SELECT * FROM V\$LICENSE;] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 死物 | [SELECT * FROM V\$DEADLOCK_HISTORY; SELECT * FROM V\$SESSION_HISTORY] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 运行日志 | [SELECT * FROM V\$SESSION_LOG_HISTORY] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dimn配置 | [SELECT * FROM V\$DMJN1;] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 自定义任务 | [select * from testtest_a] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 数据库: | 223.254.30.150&236 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| JDBC连接串: | jdbc:dm://223.254.30.150&236 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 用户名: | SYSDBA | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 密码: | ***** | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 备份路径: | d:\agent\服务地址: rpc://223.254.30.150&6364 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 归档查看天数: | 30 | 备份查看天数: | 30 | 死物查看天数: | 30 | 运行日志查看天数: | 30 | | | | | | | | | | | | | | | | | | | | |
| 巡检任务列表: | <table border="1"> <tr> <td>性能概况</td> <td>[SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE]</td> </tr> <tr> <td>表空间</td> <td>[SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE;</td> </tr> <tr> <td>Redo日志</td> <td>[SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE]</td> </tr> <tr> <td>归档日志</td> <td>[SELECT * FROM V\$ARCH_FILE]</td> </tr> <tr> <td>备份</td> <td>[SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)]</td> </tr> </table> | | | | | | | 性能概况 | [SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE] | 表空间 | [SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE; | Redo日志 | [SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE] | 归档日志 | [SELECT * FROM V\$ARCH_FILE] | 备份 | [SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)] | | | | | | | | | | |
| 性能概况 | [SELECT * FROM V\$PROCESS]; SELECT * FROM V\$SYSTAT; SELECT * FROM V\$THREADS; SELECT * FROM V\$SESSIONS; SELECT * FROM V\$TASK_QUEUE; SELECT * FROM V\$ARCH_QUEUE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 表空间 | [SELECT * FROM V\$TABLESPACE]; SELECT * FROM V\$DATAFILE; | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Redo日志 | [SELECT * FROM V\$LOG]; SELECT * FROM V\$LOGFILE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 归档日志 | [SELECT * FROM V\$ARCH_FILE] | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 备份 | [SELECT * FROM V\$DMJN1; SF_BAKSET_BACKUP_DIR_ADD(DISK, ?); SELECT * FROM V\$BACKUPSET; SELECT * FROM V\$BACKUPSET_BKP; SELECT SF_BAKSET_CHECK(?, ?)] | | | | | | | | | | | | | | | | | | | | | | | | | | |

图 6.75 联机巡检详情总览界面

仔细阅读详情总览信息，确认配置无误后，点击保存并执行，开始执行巡检任务，并进入“执行巡检任务”界面。

6. 执行巡检任务

该页面实时显示所有目标所有任务的巡检进度，如下图所示：

| 执行概览 | | | |
|----------------------------|---------|---------|--------------------------|
| 正在执行的任务数 | | | |
| 操作 | 状态 | 耗时 | 消息 |
| 调度器状态(223.254.30.150#6238) | 已完成 | 取消 | 开始时间:2022-05-09 17:44:33 |
| 任务 | | | |
| [223.254.30.150#6238] | | | |
| 主机概况 | 成功 | 3秒215毫秒 | |
| 数据库概况 | 成功 | 401毫秒 | |
| 性能概况 | 成功 | 4秒50毫秒 | |
| 索引空间 | 成功 | 53毫秒 | |
| redo日志 | 成功 | 16毫秒 | |
| 归档日志 | 成功 | 56毫秒 | |
| 备份 | 正在执行... | | |
| 用户与会话 | 等待执行... | | |
| 策略 | 等待执行... | | |
| 运行日志 | 等待执行... | | |
| dnbin直联 | 等待执行... | | |
| 自定义任务 | 等待执行... | | |
| 健康度评估 | 等待执行... | | |
| [223.254.30.150#6237] | | | |
| 主机概况 | 成功 | 3秒215毫秒 | |
| 数据库概况 | 成功 | 401毫秒 | |
| 性能概况 | 成功 | 5秒555毫秒 | |
| 索引空间 | 成功 | 41毫秒 | |
| redo日志 | 成功 | 41毫秒 | |
| 归档日志 | 成功 | 23毫秒 | |
| 备份 | 成功 | 9毫秒 | |
| ■ 停止执行 | | | |

图 6.76 联机巡检执行界面

可点击右下角的暂停按钮停止巡检。

所有巡检任务完成后，点击下一步，进入“巡检报告预览和下载界面”

7. 巡检报告预览和下载

如下图所示，联机巡检提供了简报、详情（数据库维度）、详情（巡检项维度）和 DM.INI 全部配置 4 个巡检报告。可通过右上角的下载巡检报告按钮下载所有 4 个巡检报告。

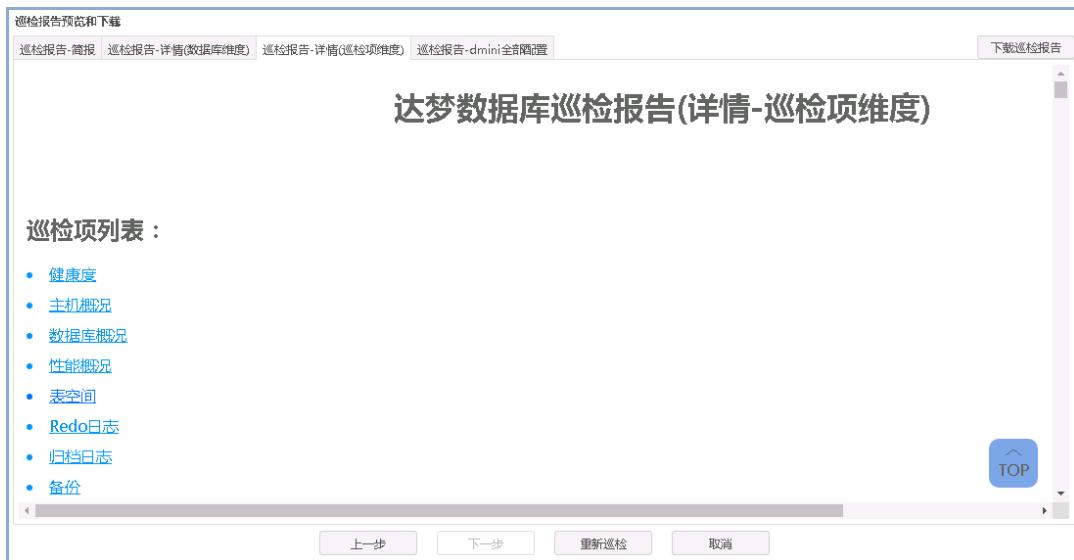


图 6.77 联机巡检报告界面

四个巡检报告简单介绍如下：

简报，显示数据库健康度、扣分项信息和提分建议。

详情（数据库维度），顺序显示每个数据库的巡检项目内容。

详情（巡检项维度），顺序显示每个巡检项目巡检结果，每个巡检中依次显示各数据库在这个巡检项目中的结果。

DM.INI 全部配置，依次显示各数据库的全部 DM.INI 配置参数信息。

其中内置项目与自动巡检项目相同，详情见 [7.1.4.6.7 巡检报告](#)。

6.11 联机大盘

联机大盘实时监测主机或数据库信息。

6.11.1 系统结构

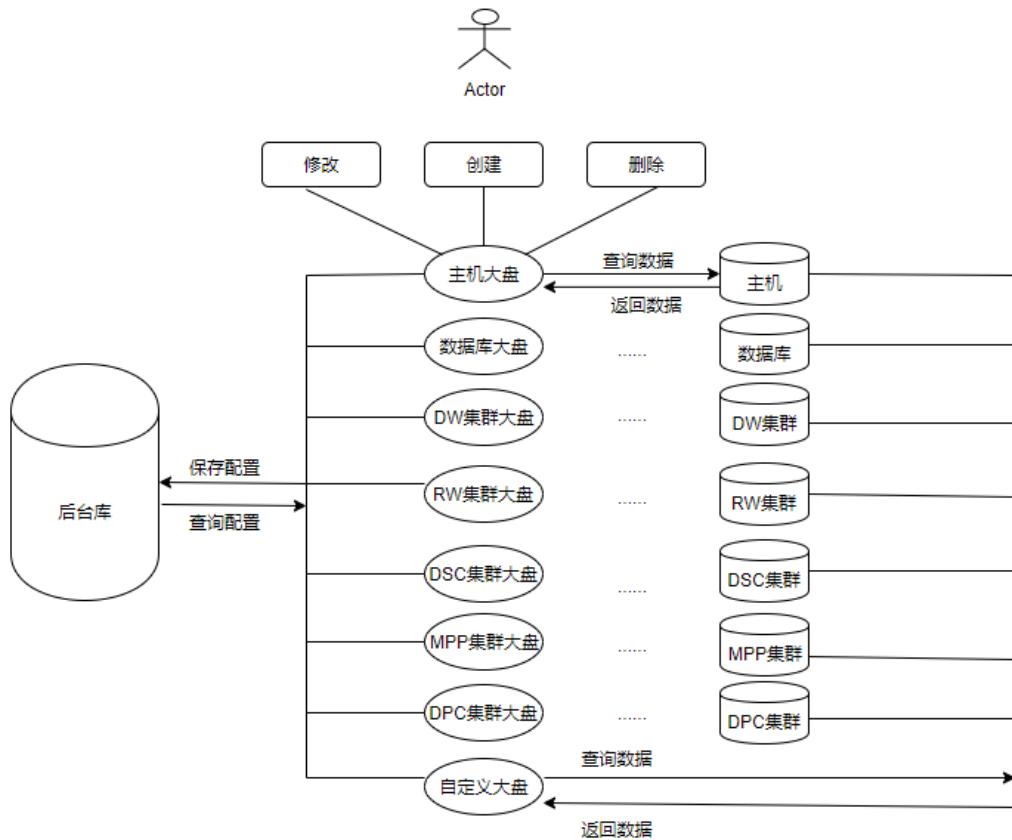


图 6.78 联机大盘系统结构图

6.11.2 实现原理

1. 联机大盘配置：按照创建配置的向导页创建配置，然后持久化到后台库中。提供对已创建大盘配置的修改、删除。
2. 联机大盘：根据大盘配置创建出联机大盘页面，然后获取数据并显示。
3. 数据获取：
 - 1) 连接

数据库和数据库集群联机大盘，通过大盘配置中数据库的连接信息建立一个持久的数据连接，根据前端请求获取数据。数据库连接的断开由前端大盘页面关闭触发，同时设置数据库连接如果长时间未使用则自动断开作为保底策略。提供对数据库连接断开后的重连，由前端重连请求触发。

主机和自定义联机大盘，则是通过连接信息建立一个短连接，根据前端请求获取数据后，断开连接。

2) 请求数据

前端定时发送请求获取数据，且对同一类型的请求是串行发送，即上次的请求获取到结果后，再发送当前请求，否则不发送，以避免出现堆积，耗尽线程资源的问题。

DEM 系统配置参数 `dashboard_refresh_freq` 值为前端发送请求获取数据的频率。

DEM 系统配置参数 `dashboard_points` 值为折线图中单条线数据点的最大个数，超过最大个数则会淘汰最早的数据点。

3) 获取数据

设计对不同类型请求的数据的查询 SQL 和执行策略，需要同时保证可靠性和高效性。对于单库查询 SQL 就可满足的数据，查询返回即可；不满足的数据需要在多个库上并行执行 SQL，然后对整合数据后返回。

6.11.3 功能介绍

联机大盘能够提供主机、单机数据库、DW 集群数据库、RW 集群数据库、DSC 集群数据库、MPP 集群数据库、DPC 集群数据库、自定义 SQL 的实时指标。实时指标分为普适指标和特定指标。普适指标包括 CPU 使用率、内存使用率、磁盘读写速率、网络读写速率。特定指标包括会话数、TPS、QPS、会话信息、任务信息、实例信息、磁盘空间使用情况。

6.11.3.1 联机大盘管理

联机大盘根节点下是已创建的联机大盘节点。右键点击联机大盘根节点和联机大盘节点，可选择打开、新建、修改、删除和刷新等功能。双击联机大盘节点，可打开当前联机大盘的面板。

6.11.3.2 联机大盘配置

1. 新建

右键点击联机大盘根节点，选择新建联机大盘，弹出“新建联机大盘”对话框，如下图所示：



图 6.79 新建联机大盤对话框

输入联机大盘名称和联机大盘描述，点击下一步，进入联机大盘类型选择页面。

2. 选择联机大盘类型

如下图所示，当前页面展示了 DEM 当前支持的所有联机大盘类型。



图 6.80 联机大盘类型选择界面

选择联机大盘类型，点击下一步，进入数据源配置页面。

3. 配置数据源

根据联机大盘类型的不同选择，数据源类型也有不同，大致可分为主机、单机和集群三种类型。不同类型数据源对应的数据源配置页不同，几种数据源配置页简单介绍如下：

1) 主机配置



图 6.81 联机大盘主机配置页

如上图所示，点击添加数据源按钮添加主机，点击删除数据源按钮删除被选主机。点击添加的数据源，可根据实际需求修改主机信息。

2) 单机配置



图 6.82 联机大盘单机配置页

如上图所示，点击添加数据源按钮添加数据库，点击删除数据源按钮删除被选数据库。点击添加的数据源，可根据实际需求修改数据库信息。选择一个数据库，点击测试数据源，根据该数据库连接测试结果判断填写的数据库信息是否准确。

3) 集群配置



图 6.83 联机大盘集群配置页

如上图所示，与单机数据源配置页相比，集群配置页仅多一个批量添加数据源按钮，其它按钮功能与单机数据源配置页一致。

批量添加数据源有连接数据源添加和连接后台库添加两种添加方法。

连接数据源添加，通过添加集群关键节点数据库来进行批量添加，如下图所示：



图 6.84 联机大盘集群添加数据库对话框

连接后台库添加，即在资源选择对话框列举的 DEM 后台库记录的集群中选择数据源。

如下图所示：

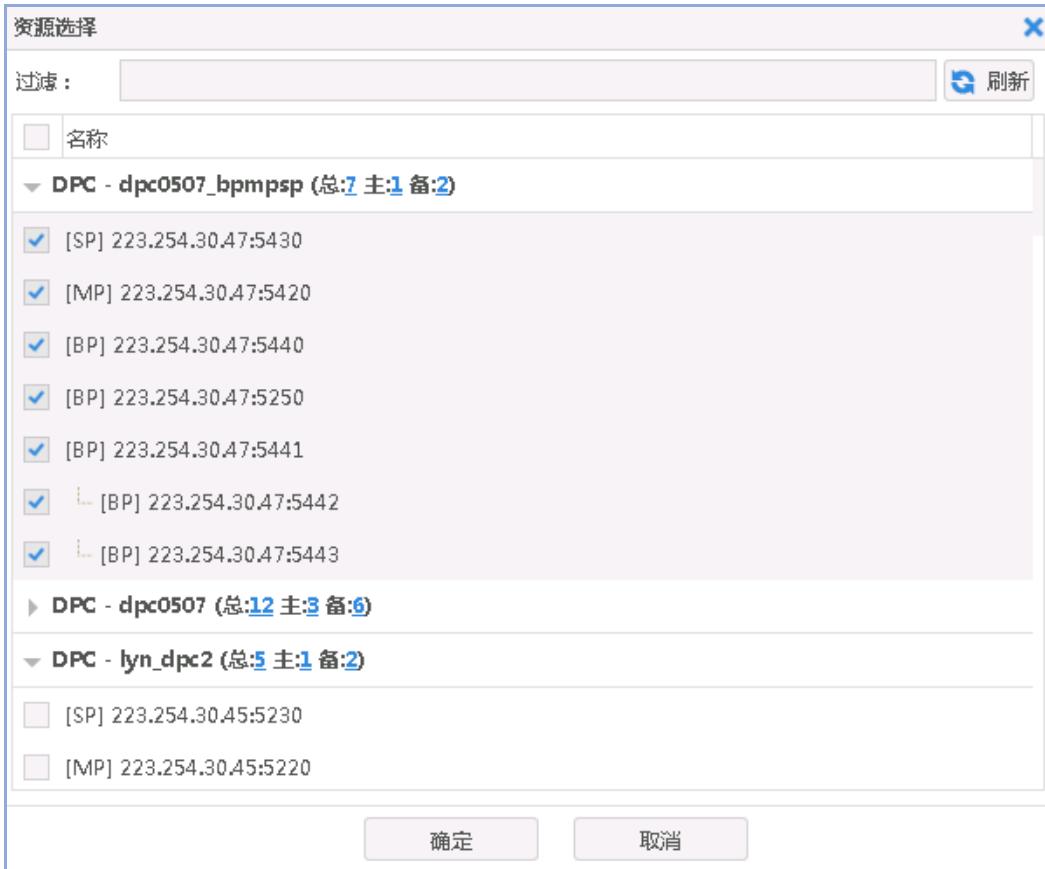


图 6.85 联机大盘集群选择对话框

配置好数据源后，点击下一步，进入配置总览页面。

4. 配置总览与完成配置

当前页面显示联机巡检配置相关信息，如下图所示



图 6.86 联机大盘配置总览页

仔细阅读配置总览信息，确认配置无误后点击完成按钮完成联机大盘配置，联机大盘创建成功。

6.11.3.3 联机大盘查看

联机大盘用于查看设置对象的实时监控信息。联机大盘每 5s 刷新一次监控信息（5s/次为默认频率，可通过参数“`dashborad_refresh_freq`”设置），联机大盘页面一次展示最近的 30 个监测结果（30 为默认值，可通过参数“`dashboard_points`”设置）。

联机大盘类型不同，对应监测内容也有所不同。各类型联机大盘页面介绍如下：

6.11.3.3.1 主机大盘

主机大盘信息包括主机名、操作系统信息、CPU 信息、CPU 使用率、内存使用率、交换区使用率、磁盘读写速率、网络读写速率、磁盘使用率、磁盘总大小和磁盘空闲大小。

其中，CPU 使用率、内存使用率、交换区使用率、磁盘读写速率和网络读写速率通过折线图实时展示最近的 30 次监控结果。鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

如下图所示：

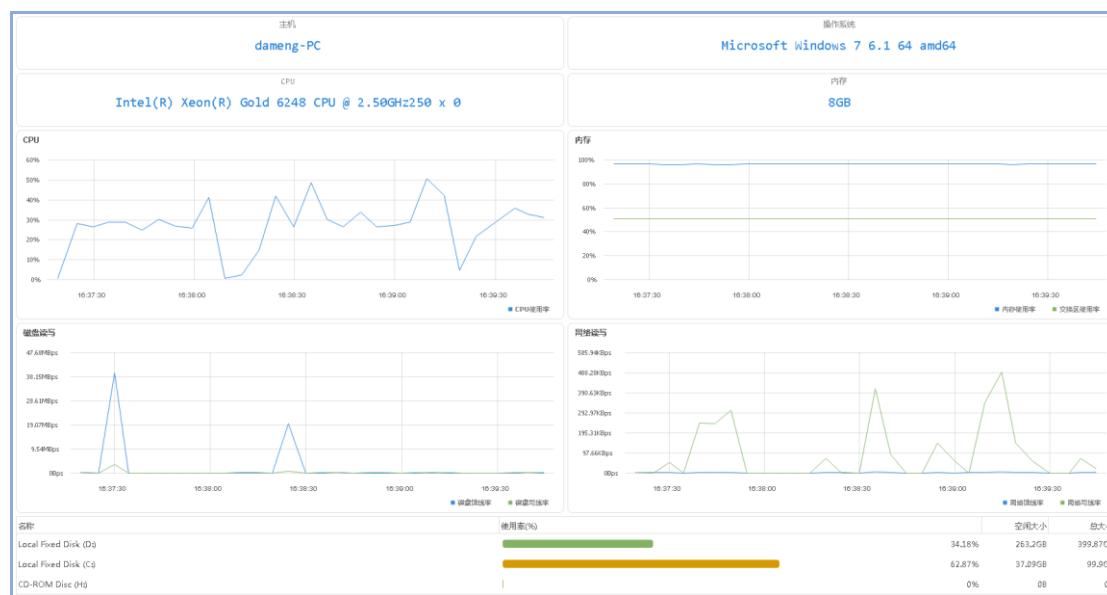


图 6.87 主机大盘界面

6.11.3.3.2 数据库大盘

数据库大盘包括总览、会话和配置三个页面。

1. 总览

数据库大盘总览页面信息包括数据库名、数据库启动时间、数据库版本、数据库状态和模式、数据库 LSN 信息、SESS（会话数）、TPS（每秒事务数）/QPS（每秒请求数）、主机和数据库 CPU 使用率、主机和数据库内存使用率、磁盘读写速率和网络读写速率。

其中，SESS（会话数）、TPS（每秒事务数）/QPS（每秒请求数）、主机和数据库 CPU 使用率、主机和数据库内存使用率、磁盘读写速率和网络读写速率通过折线图实时展示最近的 30 次监控结果。鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

如下图所示：



图 6.88 数据库大盘总览页面

2. 会话

会话页面以列表形式显示数据库的会话实时信息。如下图所示：

| 过滤：所有表列 | | | | | | |
|------------|--------|--------|----------------------------|----------------|-------|--|
| 会话ID | 会话状态 | 当前用户 | 会话创建时间 | 客户端IP | 客户端类型 | 操作 |
| 158794048 | IDLE | SYSDBA | 2022-05-09 10:38:32.000000 | 223.254.30.35 | | JC X 关闭会话 JC ■ 终止会话操作 [详细信息] |
| 1872757056 | ACTIVE | SYSDBA | 2022-05-10 15:52:12.000000 | 223.254.30.150 | | |

图 6.89 数据库大盘会话页面

通过会话操作栏下拉列表的功能按钮可以关闭会话、终止会话操作以及查看会话详细信息。会话查看示例如下图所示：



图 6.90 数据库大盘会话查看对话框

3. 配置

配置页面显示数据库的 dmini 参数信息。

可通过条件过滤快速查找参数。过滤条件支持参数名、参数类型、参数值和参数描述过滤。

双击和点击操作栏下拉列表的查看参数按钮可查看参数。点击操作栏下拉列表的修改参数值按钮可修改参数的参数值。

6.11.3.3.3 DPC 集群大盘

DPC 集群大盘包括总览、任务、会话和实例四个页面。

1. 总览

DPC 集群大盘总览页面信息包括总任务数、执行任务数、等待任务数、SP 实例数、BP 实例数、MP 实例数、RAFT 组数、CPU 使用率（最大值）、内存使用率（最大值）、磁盘

读写速率（最大值）、网络读写速率（最大值）、SESS（会话数）、TPS（每秒事务数）/QPS（每秒请求数）。

其中，SESS（会话数）和 TPS（每秒事务数）/QPS（每秒请求数）通过折线图实时展示最近的 30 次监控结果。鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

如下图所示：



图 6.91 DPC 集群大盘总览页面

2. 任务

DPC 集群大盘任务页面显示集群当前所有执行中的任务。

DPC 集群大盘任务页面展示了当前正在执行的任务列表。任务信息包括会话ID、状态、执行耗时、执行进度、内存占用、SQL 语句以及操作栏。操作栏提供了终止任务和属性查看功能。

| 过滤: 会话ID/SQL | | ■ 终止任务 | | | | |
|-----------------|---------|--------|------|------------|---------------------------------------|--|
| 会话ID | 状态 | 执行耗时 | 执行进度 | 内存占用 | SQL | 操作 |
| 140190172737504 | 运行 < 1s | | | 99% 9.13MB | /**DEM***/select /*+ enable_monitor*/ | ■ 终止任务 ■ 属性 |

图 6.92 DPC 集群大盘任务页面

可通过点击工具栏或任务操作栏下拉列表的终止任务按钮终止任务。可点击任务操作栏下拉列表的属性按钮查看任务的会话序号、用户名、客户端 IP 和客户端主机名等信息。

3. 会话

DPC 集群大盘会话页面与数据库大盘会话页面大致一致，唯一不同处在于，DPC 集群有多个数据库，所以会话将按所属数据库分组显示。

| 过滤: 所有表列 | | | | | | |
|--------------------------------|--------|--------|-------------------------|---------------------|-------|----|
| 会话ID | 会话状态 | 当前用户 | 会话创建时间 | 客户端IP | 客户端类型 | 操作 |
| ▼ [SP]192.168.17.48:5230会话数: 1 | | | | | | |
| 140190172737504 | ACTIVE | SYSDBA | 2022-05-10 17:49:41.... | :ffff:223.254.30.35 | JDBC | ▼ |
| ▼ [SP]192.168.17.48:5232会话数: 1 | | | | | | |
| 140070816613184 | ACTIVE | SYSDBA | 2022-05-10 18:08:01.... | :ffff:192.168.17.48 | JDBC | ▼ |

图 6.93 DPC 集群大盘会话页面

4. 实例

DPC 集群大盘实例页面按数据库模式类型分组显示所有数据库实例信息。如下图所示：

| 过滤: 数据库 | | | | | | | | | | | |
|--------------------|-----------------|------|-----|------|-----|----|------|------|----------|-----------|----|
| 数据库 | 连接 | 系统状态 | 会话 | 每秒事务 | CPU | 内存 | 磁盘读 | 磁盘写 | 网路读 | 网路写 | 操作 |
| ▼ SP 实例数: 3 | | | | | | | | | | | |
| 192.168.17.48:5232 | 成功 | OPEN | 0/2 | 31 | 0% | 0% | 0Bps | 0Bps | 3.16KBps | 4.35KBps | ▼ |
| 192.168.17.48:5231 | 成功 | OPEN | 0/2 | 42 | 0% | 0% | 0Bps | 0Bps | 4.33KBps | 41.57KBps | ▼ |
| 192.168.17.48:5230 | 成功 | OPEN | 1/2 | 9 | 0% | 0% | 0Bps | 0Bps | 410Bps | 2.35KBps | ▼ |
| ▼ BP 实例数: 2 | | | | | | | | | | | |
| 192.168.17.48:3243 | 成功 | OPEN | 0/0 | 31 | 0% | 0% | 0Bps | 5Bps | 0Bps | 0Bps | ▼ |
| 192.168.17.48:3240 | 成功 | OPEN | 0/0 | 79 | 0% | 0% | 0Bps | 5Bps | 9.02KBps | 83.34KBps | ▼ |
| ▼ MP 实例数: 1 | | | | | | | | | | | |
| 192.168.17.48:5220 | 成功 | OPEN | 0/0 | 79 | 0% | 0% | 0Bps | 5Bps | 9.01KBps | 83.31KBps | ▼ |

图 6.94 DPC 集群大盘实例页面

可点击实例操作栏下拉列表的属性按钮查看数据库实例的实例名、启动时间、RAFT 组、服务器版本和数据库版本等信息。

6.11.3.3.4 MPP 集群大盘

MPP 集群大盘包括总览、会话和实例三个页面。

1. 总览

MPP 集群大盘总览页面信息包括 CPU 使用率（最大值）、内存使用率（最大值）、磁盘读写速率（最大值）、网络读写速率（最大值）、SESS（会话数）、TPS（每秒事务数）/QPS（每秒请求数）。

其中，SESS（会话数）和 TPS（每秒事务数）/QPS（每秒请求数）通过折线图实时展示最近的 30 次监控结果。鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

如下图所示：

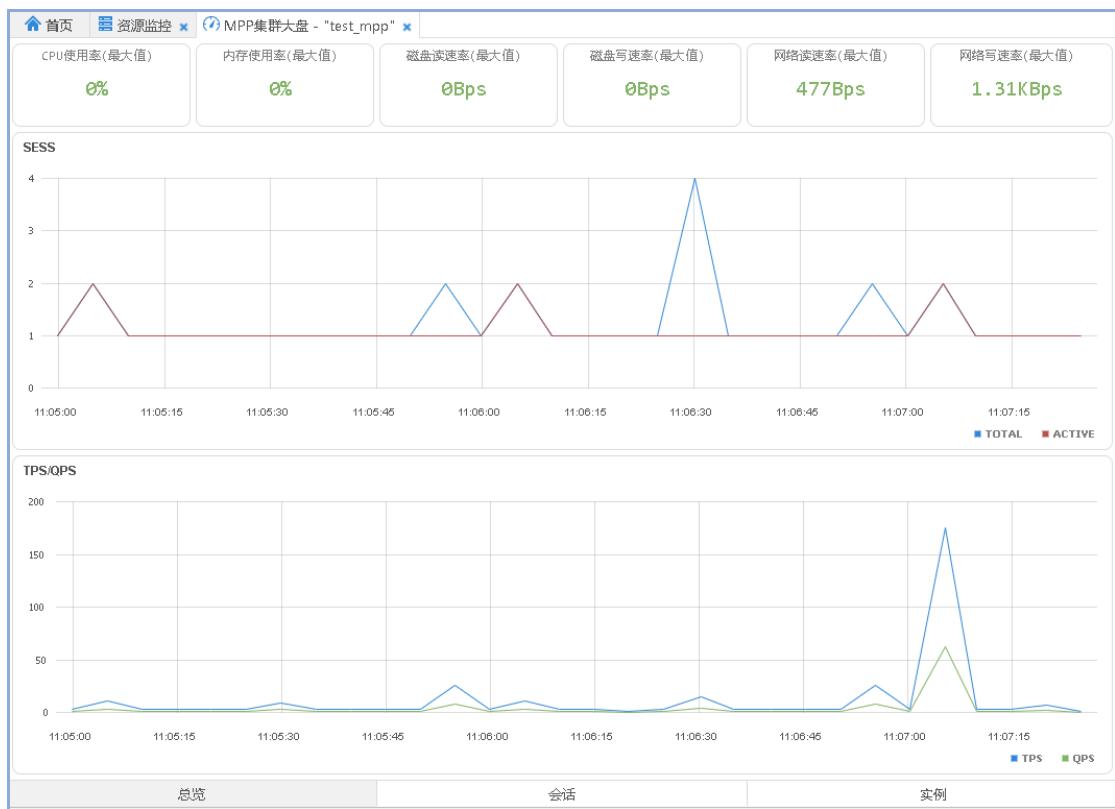


图 6.95 MPP 集群大盘总览页面

2. 会话

MPP 集群大盘会话页面与数据库大盘会话页面大致一致，唯一不同处在于，MPP 集群有多个数据库，所以会话将按所属数据库分组显示。

| MPP 集群大盘 - "test_mpp" - DSC 集群大盘 - "test_dsc" | | | | | | |
|---|--------|--------|------------------------|---------------------|-------|----|
| 过滤: 所有表列 | | | | | | |
| 会话ID | 会话状态 | 当前用户 | 会话创建时间 | 客户端IP | 客户端类型 | 操作 |
| 192.168.100.120:47236 | 会话数: 2 | | | | | |
| 1094439648 | ACTIVE | SYSDBA | 2022-05-27 14:53:48... | :ffff:192.168.122.1 | JDBC | ▼ |

图 6.96 MPP 集群大盘会话页面

3. 实例

MPP 集群大盘实例页面依次显示集群的所有数据库实例信息。如下图所示：

| MPP 集群大盘 - "test_mpp" | | | | | | | | | | | |
|-----------------------|----|------|----|------|-----|----|-----|-----|-----|-----|----|
| 过滤: 数据库 | | | | | | | | | | | |
| 数据库 | 连接 | 系统状态 | 会话 | 每秒事务 | CPU | 内存 | 磁盘读 | 磁盘写 | 网路读 | 网路写 | 操作 |
| 192.168.100.120:47236 | 失败 | OPEN | -- | -- | -- | -- | -- | -- | -- | -- | ▼ |
| 192.168.100.120:47237 | 失败 | OPEN | -- | -- | -- | -- | -- | -- | -- | -- | ▼ |
| 192.168.100.120:47238 | 失败 | OPEN | -- | -- | -- | -- | -- | -- | -- | -- | ▼ |

图 6.97 MPP 集群大盘实例页面

可点击实例操作栏下拉列表的属性按钮查看数据库实例的实例名、启动时间、服务器版本和数据库版本等信息。

6.11.3.3.5 DSC 集群大盘

DW 集群大盘与 MPP 集群大盘内容相同，参见章节 [6.11.3.3.4 MPP 集群大盘](#)。

6.11.3.3.6 RW 集群大盘

DW 集群大盘与 MPP 集群大盘内容相同，参见章节 [6.11.3.3.4 MPP 集群大盘](#)。

6.11.3.3.7 DW 集群大盘

DW 集群大盘与 MPP 集群大盘内容相同，参见章节 [6.11.3.3.4 MPP 集群大盘](#)。

7 智能运维

DEM 提供主机监控、数据库监控以及告警功能。用户可以通过 DEM 监控诊断主机和数据库的性能问题，其中数据库包括单实例以及各种集群。

7.1 DEM 监控

7.1.1 系统结构

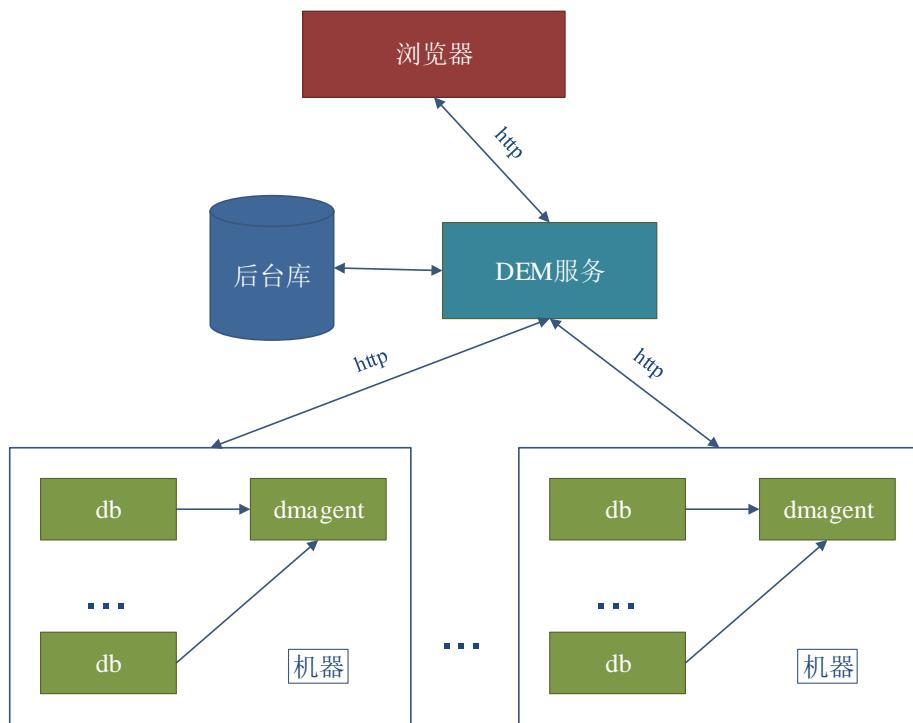


图 7.1 DEM 监控系统结构图

7.1.2 实现原理

DEM 监控主要由 dmagent 收集数据库或主机信息，发送给 DEM 服务，DEM 服务将监控信息存储在后台库中，通过检索 DEM 后台库，将监控信息显示到前端。

其中，dmagent 会分别启动 gather 线程去收集各监控信息，如果 dmagent 发送信息失败，即 dmagent 与 DEM 掉线的情况下，gather 线程会将消息保存至本地（dmagent

安装目录的 data 目录下），dmagent 有一个 flush 线程用于刷盘，还有一个 send 线程用于发送离线信息。当有信息发送失败时，失败信息将被放到消息队列中，刷盘线程从消息队列取出信息，写到文件。Send 线程每隔一段时间（默认 5 分钟）读取文件信息发送给 DEM。



图 7.2 DEM 监控实现原理图

7.1.3 主机监控

DEM 在主机监控页面列出监控主机的状态、CPU、内存、磁盘以及网络等信息，并可以通过动作完成统计、分析、配置以及升级等操作。

主机监控页面每隔 1 分钟（该值可以通过参数 “dem_page_refresh_freq” 配置）会自动刷新主机监控信息。在主机监控页面的最右侧会显示最新监测时间，表示所有主机中收到最新的主机信息的时间。

7.1.3.1 添加主机监控

以 monitor 模式启动 dmagent，自动添加主机信息到 DEM 主机监控页面。

启动 dmagent 后（详细参见 [3.2 dmagent 部署](#)），如果以 monitor 模式启动，则 dmagent 会收集主机信息发送给 DEM，DEM 将主机监控信息存储在 DEM 后台库，DEM 主机监控页面每隔 1 分钟（管理员或超级管理员可以配置 “dem_page_refresh_freq” 来修改该值）会从 DEM 后台库中拉取主机信息，展示到 DEM 主机监控页面。

7.1.3.2 操作工具栏

主机监控页面最上面的操作工具栏为主机监控提供了刷新、删除、导出等众多操作功能，如下图所示：



图 7.3 主机监控操作工具栏

各功能介绍如下：

1. 刷新

点击刷新按钮，获取最新的主机监控信息列表。

2. 删除

点击删除按钮，删除主机监控信息列表中被选中的一行或多行主机监控信息。

3. 导出

点击导出按钮，导出整个主机监控信息页面，导出格式为 PDF。

4. 配置网络

若主机有多个网卡，需要用户手动配置网络环境。所谓的网络环境是指数据库代理连接数据库收集监控信息的 IP 地址（内网 IP）以及 DEM 中心与数据库代理进行通信的 IP 地址（外网 IP）。

点击配置网络按钮，打开配置网络对话框，配置主机监控信息列表中的主机内外网 IP 信息。如下图所示：

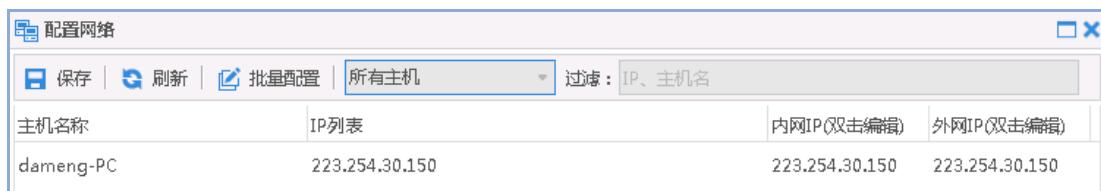


图 7.4 主机配置网络对话框

5. 下载代理

点击下载代理按钮，下载最新 dmagent 到本地。

6. 忽略全部告警

点击忽略全部告警按钮，将所有主机产生的告警都标记为已处理。

7. 搜索

根据输入的 IP/主机名/MAC 地址信息在主机监控信息列表中搜索筛选对应主机。

8. 暂停/继续

操作工具栏右边的倒计时是关于页面刷新的倒计时，点击它可暂停或继续页面刷新。

9. 隐藏/显示摘要

点击操作工具栏最右边的矩形框，隐藏或显示监控信息汇总栏。

7.1.3.3 监控信息汇总栏

7.1.3.3.1 摘要

主机监控摘要部分统计了所有主机中不同连接状态的主机个数，即成功状态的主机个数和未知状态的主机个数；统计了所有主机中不同大类操作系统的主机个数，即 WIN 系列操作系统的主机个数、LINUX 系列操作系统的主机个数、MAC 系列操作系统的主机个数和其他操作系统的主机个数。

如下图所示：

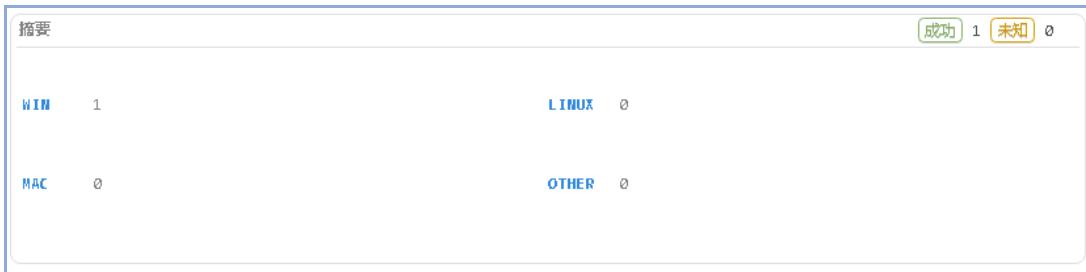


图 7.5 主机监控摘要

7.1.3.3.2 告警

主机监控告警部分统计了所有主机产生的未处理告警中不同告警级别的告警个数，即恢复的告警个数、轻微的告警个数、一般的告警个数、严重的告警个数和致命的告警个数；统计了所有主机产生的未处理告警中告警数量最多的三个主机信息；统计了所有主机产生的未处理告警中时间最近的三条告警信息。

如下图所示：



图 7.6 主机监控告警

7.1.3.4 主机监控项

下表详细介绍当前收集到的各主机最新的监控信息：

表 7.1 主机监控信息说明

| 监控项 | 说明 |
|--------|---|
| 主机 | <p>显示主机地址和主机的状态。 鼠标停留在主机列，可以显示主机的其他一些信息，如下图所示：</p>  <p>其中监测时间为最近一次收到该主机信息的时间。 当主机名前出现 ⓘ 图标时，表示主机可能存在以下异常情况： 1. 主机未进行网络配置； 2. 主机上的数据库代理版本过低；</p> |
| 连接 | 该主机的连接状态，为成功或者未知。成功指 DEM 服务器与该主机网络通信成功。未知指 DEM 服务器与该主机网络通信失败。 |
| 健康度 | <p>该主机的健康度分数，满分 100 分。 鼠标悬浮在健康度上时将显示各扣分项内容，如下图所示：</p>  <p>可根据提示信息去提升健康度。</p> |
| CPU | cpu 使用率。 |
| CPU 等待 | cpu 等待时间占比。 |
| 内存 | 内存使用率。 |
| 交换区 | 交换区使用率。 |
| 磁盘读 | 磁盘读取速率。 |
| 磁盘写 | 磁盘写入速率。 |
| 网络读 | 网络接收速率。 |
| 网络写 | 网络发送速率。 |
| 告警 | 主机产生的未处理的告警数量。点击打开告警历史面板。 |
| 操作 | 点击 ▾ 将弹出操作菜单，详细介绍请参见下面章节 5 主机监控操作。 |

7.1.3.5 主机监控操作

在 DEM 主机监控页面，可以点击主机最右侧“操作”栏的下拉按钮查看对每个主机监控的具体操作。

7.1.3.5.1 信息大盘

点击信息大盘按钮，打开主机大盘页面。主机大盘显示主机最近的监控汇总信息，主机大盘以模块形式布局显示各信息，用户可点击布局按钮按照个人需求与喜好对各模块布局。点击导出按钮可导出整个主机大盘页面图表，导出格式为 PDF。主机大盘页面右上角监测时间为上次页面刷新时间。模块右上角的下拉菜单为模块提供了一些功能按钮。不同模块提供的功能按钮不尽相同，每个模块仅提供与该模块相关的功能按钮。

主机大盘各模块简单介绍如下：

1. 基本信息

显示该主机连接状态、主机名、主机 IP、CPU、编号、内存、代理版本、操作系统和上一次监测时间等信息。

2. 最近告警 (TOP3)

显示该主机产生的未处理告警中不同告警级别的告警个数和时间最近的三个告警信息。

3. 健康度

显示主机健康度和对应的扣分项与提分建议。

4. 统计

显示该主机的自定义监控个数和进程监控个数。

5. CPU 使用率

显示该主机最近 10 次监测的 CPU 使用率。

6. 交换区使用率

显示该主机最近 10 次监测的交换区使用率。

7. FIO

显示该主机最近 10 次监测的磁盘读写速率。

8. NIO

显示该主机最近 10 次监测的网络读写速率。

9. 内存使用率 (TOP3)

显示该主机内存的总量、使用率大小、空闲大小和使用率。

10. 磁盘使用率 (TOP3)

显示该主机上磁盘使用率最高的三个磁盘的总量、使用大小、空闲大小和使用率。

鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

路径：智能运维->资源监控->主机->列表“操作栏” ->下拉按钮->信息大盘。

如下图所示：

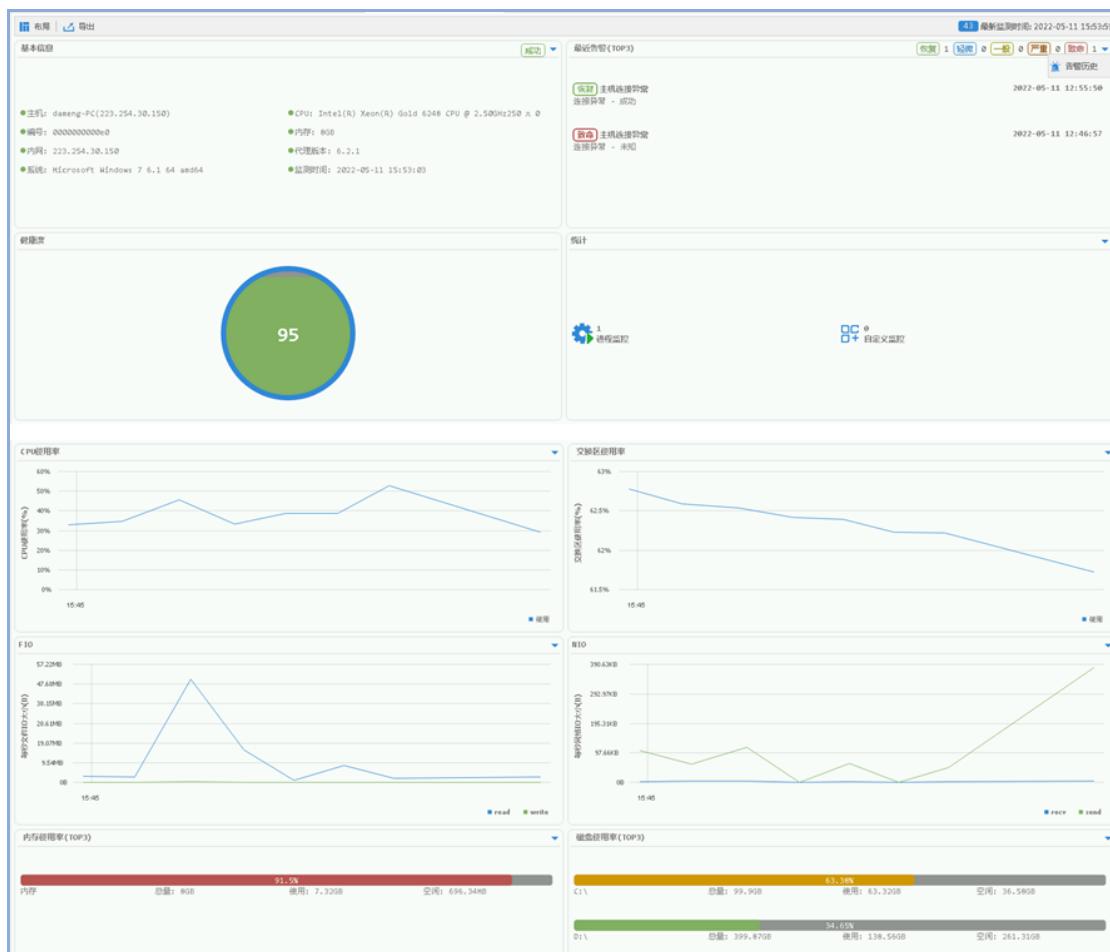


图 7.7 主机大盘

7.1.3.5.2 负载分析

统计主机负载的历史信息，可以查看不同时间段内的主机内存、CPU、磁盘 IO 和网络 IO 信息，通过这些负载信息，在数据库实例变慢时，可通过该统计信息展示数据库实例所在主机资源的使用情况，从而帮助确认性能问题的根本原因。点击导出按钮可导出整个负载分析页面的图表，导出格式为 PDF。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->负载统计。

如下图所示：

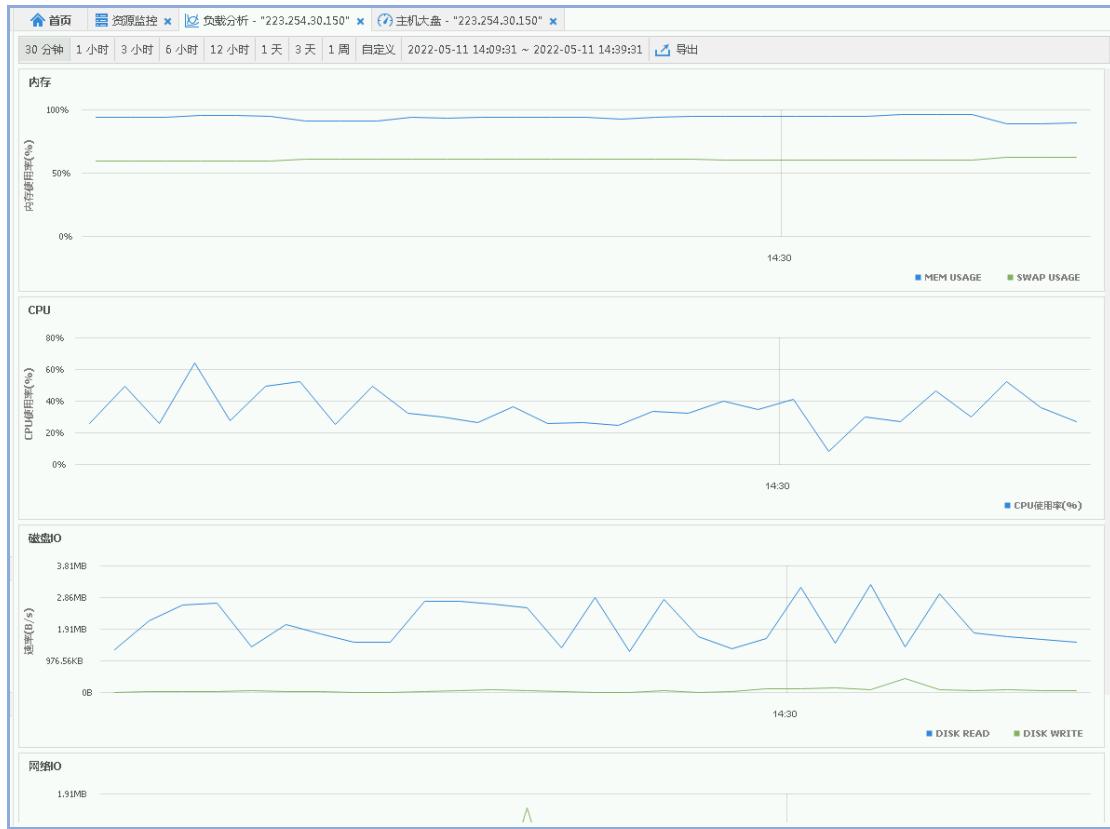


图 7.8 主机负载统计图

7.1.3.5.3 磁盘分析

统计磁盘使用的历史信息，可以查看不同时间段内的磁盘使用情况，包括每个分区的磁盘使用率、空闲大小以及磁盘总大小等。点击导出按钮可导出整个磁盘分析页面的图表，导出格式为 PDF。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->磁盘分析。

如下图所示：

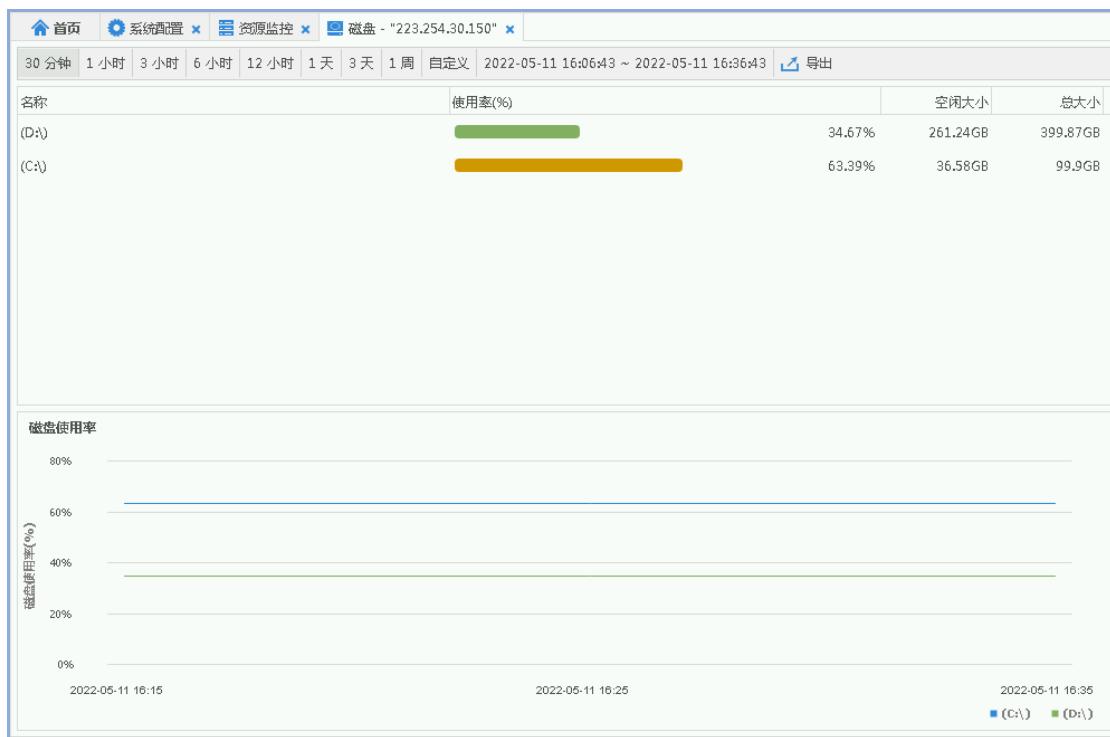


图 7.9 磁盘分析图

7.1.3.5.4 自定义监控

除了数据库和主机监控之外，还需监控其他程序或脚本的执行情况（执行结果，成功/失败），但往往与实际应用需求相关，无法详列，故只能自定义，根据实际情况指定要监控的程序或脚本，该方法可适用更多的监控场景。dmagent 会按一定频率（添加自定义监控时设置的监控频率）执行自定义的程序或脚本，并将结果发送给 DEM，DEM 显示自定义的程序或脚本的执行情况。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->自定义监控。

1. 添加自定义监控

在自定义监控面板点击“添加”，打开添加自定义监控对话框，如下所示：



图 7.10 添加自定义监控界面

填写监控名称、指定程序或脚本路径、配置好监控频率以及超时时间后，点击“确定”即可。

2. 自定义监控详情

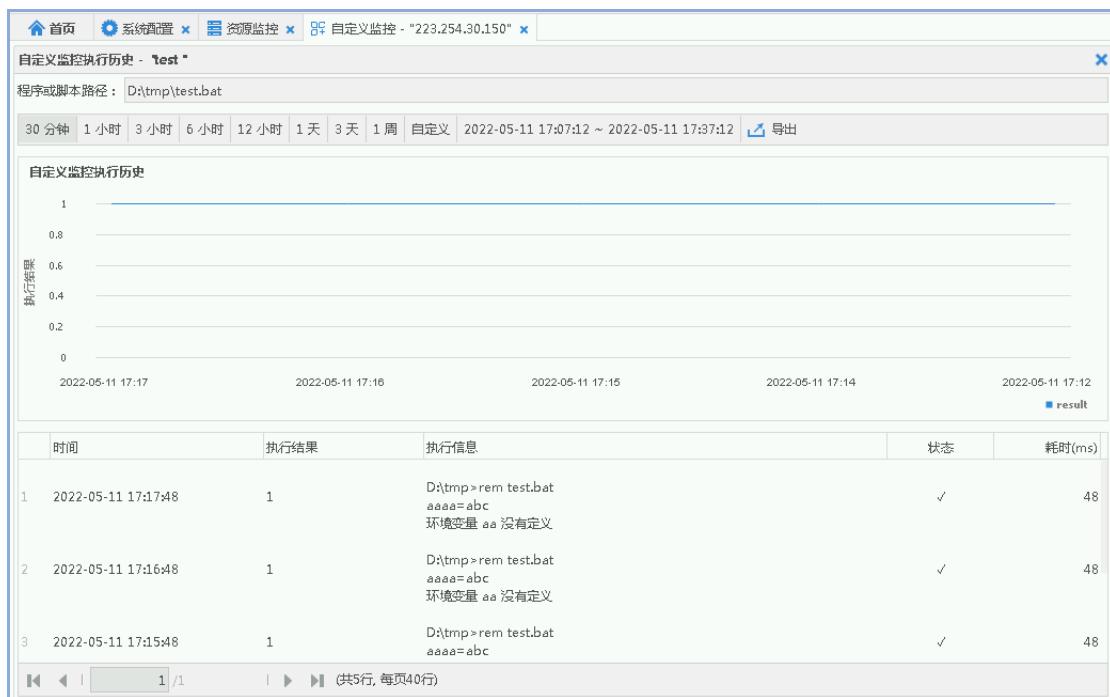
可在自定义监控页面查看每个自定义监控项的最近执行时间、最近执行状态、最近执行结果、最近执行信息以及最近执行耗时等，点击导出按钮可导出整个自定义监控页面的监控信息，导出格式为 PDF。

如下图所示：

| 自定义监控 - "223.254.30.150" | | | | | | | |
|--------------------------|------|-------------------------------------|-----------------|---------------------|--------|--------|--|
| | 名称 | 启用 | 路径 | 最近执行时间 | 最近执行状态 | 最近执行结果 | 最近执行信息 |
| 1 | test | <input checked="" type="checkbox"/> | D:\tmp\test.bat | 2022-05-11 17:12:48 | √ | 1 | D:\tmp>rem test.bat aaaa=abc 环境变量 aa 没有定义 48 |

图 7.11 自定义监控详情界面

点击详情栏按钮，打开自定义监控执行历史页面，可查看程序或脚本不同时间段的执行历史信息。如下图所示：



3. 自定义监控启用和禁用

通过启用和禁用来开启或停止自定义的监控项。

启用和禁用方式：

- 1) 在添加自定义监控时，选择是否启用；
- 2) 在自定义监控面板的监控列表中，点击“启用”栏的启用/禁用按钮。

7.1.3.5.5 进程监控

除了数据库和主机监控之外，还需监控其他进程的资源使用情况（例如监控 dmagent 的资源占用情况），但往往与实际应用需求相关，无法详列，故只能自定义，根据实际情况指定要监控的进程，该方式可以适用更多的监控场景。dmagent 会按一定频率（配置参数 “gather_mf_process_freq”）收集相关进程的信息，并将结果发送给 DEM，DEM 显示进程的状态和资源使用情况。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->进程监控。

1. 添加进程监控

在自定义监控面板点击“添加”，打开添加进程监控对话框，如下所示：

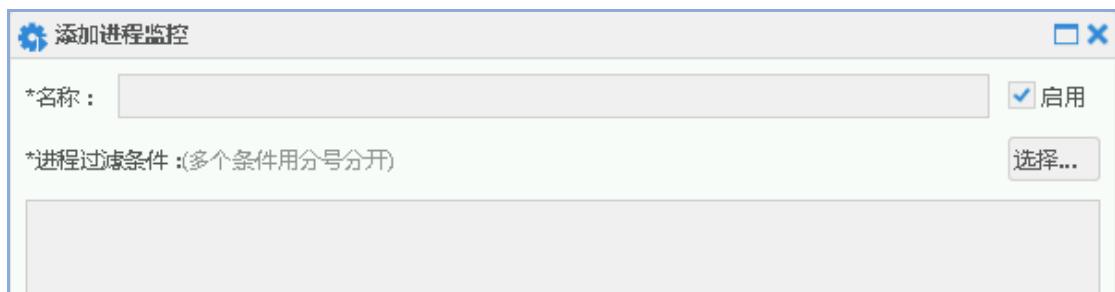


图 7.13 添加进程监控界面

指定名称和进程过滤条件，点击“确定”即可，其中名称为自定义监控的标识，进程过滤条件为 dmagent 搜索要监控进程的过滤条件，当进程启动命令中包含过滤条件时，则认为是需要监控的进程，否则不是。

可以通过选择->选择进程，快速填写过滤条件。

例如：机器上启动了 2 个 dmserver，启动命令分别为 dmserver

d:/data/dameng/DM.INI 和 dmserver d:/data1/dameng/DM.INI。

若要监控 dmserver，则可指定名称“dmserver”（名称只起标识作用），过滤条件“dmserver”，此时会监控这 2 个 dmserver 的进程，如果添加监控后，又启动了一个 dmserver，则也会监控新启动的 dmserver 进程。即会监控该台电脑上启动的所有启动命令中包含“dmserver”的进程。

如果只需监控 dmserver d:/data/dameng/DM.INI 这一个进程，则进程过滤条件指定为进程启动命令“dmserver d:/data/dameng/DM.INI”。



如果需要唯一监控某一个进程，则过滤条件应该作严格限制，给定多个过滤条件

注意：件，或者过滤条件指定为进程的启动命令。

2. 进程监控信息

可以在进程监控页面查看添加的进程监控信息，如下图所示：

| 32 最新监测时间: 2022-05-11 17:57:39 | | | | | | | | |
|--------------------------------|--------|------|------|-------|--------|--------|----------|---|
| 名称 | 进程ID | 进程状态 | 运行时间 | CPU | 内存 | VSZ | RSS | 磁盘I/O 启动进程命令 |
| test_dmserver(3) | | | | | | | | |
| dmserver | 70936 | 成功 | 15天 | 0% | 1.7% | 2.89GB | 139.59MB | 6.89KBps/279.98Bps dmserver D:\dmdbms\data\TEST_DTS\dm.ini |
| dmserver | 213092 | 成功 | 4天 | 0.03% | 16.63% | 1.62GB | 1.33GB | 12.19MBps/31.53KBps dmserver D:\dmdbms\data\DMENG\dm.ini |
| dmserver | 19136 | 成功 | 17天 | 0.01% | 5.32% | 3.79GB | 435.69MB | 0Bps/13.92KBps dmserver D:\web\dem\dmdbms\data\DMENG\dm.ini |

图 7.14 进程监控信息

此页面显示搜索到进程的最新信息，包括进程 ID、进程状态、运行时间、CPU、内存、VSZ、RSS、磁盘 I/O 以及启动进程命令等。按添加的进程监控信息分组显示，组下方为从

主机收集到的进程具体信息。点击导出按钮可导出整个进程监控页面的监控信息，导出格式为 PDF。

可以对进程监控进行编辑、删除、启用和禁用操作以及查看历史信息操作，也可查看具体进程详细信息。

如下：

1) 进程监控编辑

在进程监控面板的监控列表中，点击最右侧“操作栏”的进程监控的下拉列表，选择“编辑”按钮，打开“编辑进程监控”对话框，可以对进程监控名称和过滤条件进行重新编辑。

2) 进程监控删除

在进程监控面板的监控列表中，点击最右侧“操作栏”的进程监控的下拉列表，选择“删除”按钮，删除进程监控，不再对相应的进程进行监控。

3) 进程监控启用和禁用

通过启用和禁用来开启或停止进程监控。

启用和禁用方式：

(1) 在添加进程监控时，选择是否启用；

(2) 在进程监控面板的监控列表中，点击最右侧“操作栏”的进程监控的下拉列表，选择“启用/禁用”按钮。

4) 进程监控历史记录

在进程监控面板的监控列表中，点击最右侧“操作栏”的进程监控的下拉列表，选择“历史记录”按钮，打开查看进程监控历史记录的对话框，如下图所示：



图 7.15 进程监控历史记录

可以查看该进程监控任意时间段内的历史信息。

进程监控历史信息包括进程运行情况和时间段内所有搜索到的进程列表。点击导出按钮可导出该进程监控当前时段的监控历史信息，导出格式为 PDF。

其中进程个数变化曲线图显示所选时间段的进程个数变化情况；进程运行时间图显示每个进程的运行时间周期；进程列表显示所选时间段搜索到的进程列表。

选中进程个数变化曲线图中的点，对应的进程列表中会选中对应的进程，以显示该点中存在的是哪几个进程。

选中进程运行时间图中的点，对应的进程列表中会选中对应的进程，以显示对应进程号的进程的具体信息。

在进程列表中最右侧的“操作”栏中点击“详细信息”按钮，可以查看具体进程的详细信息。

5) 具体进程详细信息

在进程监控面板的监控列表中，点击最右侧“操作栏”的具体进程的“详细信息”按钮，打开具体进程的详细信息面板，可以查看具体进程的资源使用历史。如下图所示：

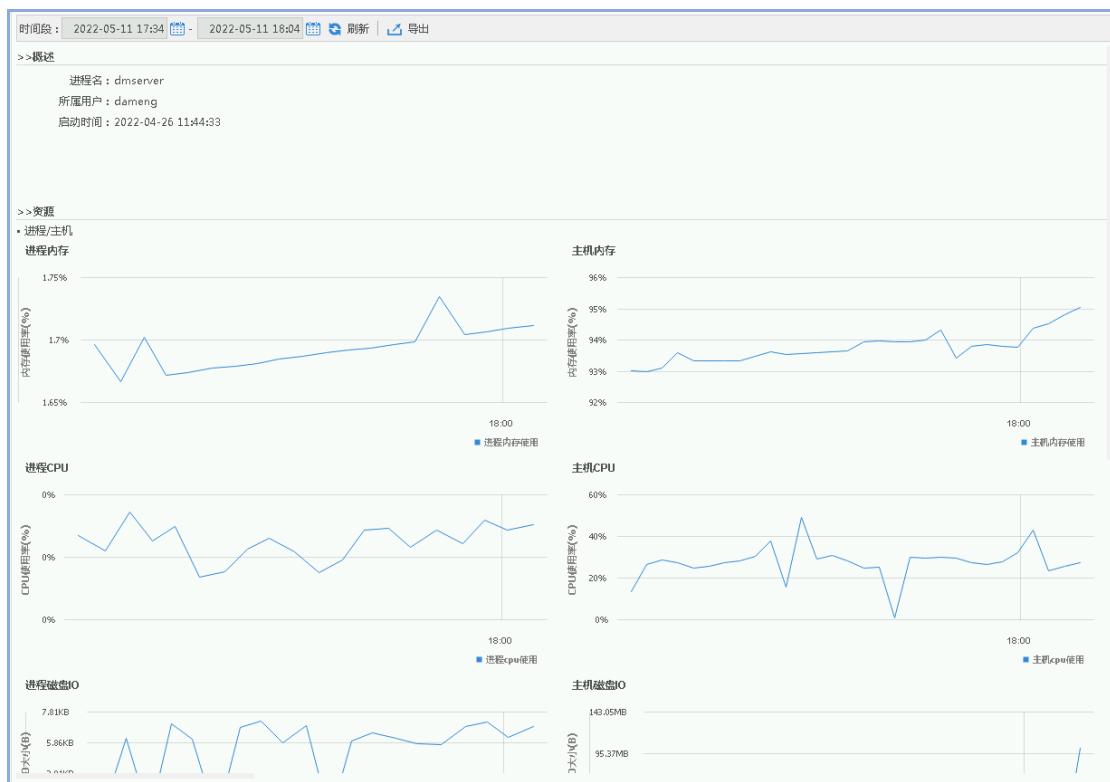


图 7.16 具体进程详细信息

显示选中时间段的进程详细信息，默认时间间隔为半小时，最长时间间隔为 7 天。

显示内容为概述和资源，其中概述包括进程名、所属用户、启动时间以及所在主机的主机名、系统、CPU 和内存等信息。资源包括进程和主机内存使用情况、进程和主机 CPU 使

用情况、进程和主机磁盘 IO、进程打开文件数、进程 vsz、进程 RSS 以及进程的线程数等信息。点击导出按钮可导出该进程当前时段的详细信息，导出格式为 PDF。

7.1.3.5.6 Core 监控

除了数据库和主机监控之外，还需监控进程崩溃情况（例如 DSC 使用了故障自动拉起功能，因此有时单台数据库在 core 之后，又迅速被拉起，此时产生了 core 文件，但运维人员不主动去主机上检查，无法及时发现 core 信息），但往往与实际应用需求相关，无法详列，故只能自定义，根据实际情况指定要监控的应用，该方式可以适用更多的监控场景。dmagent 会按一定频率（配置参数“gather_mf_core_freq”）收集指定文件夹中相关进程的 Core 文件信息，并将结果发送给 DEM，DEM 显示 Core 文件生成情况。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->Core 监控。

1. 添加 Core 监控

在 Core 监控面板点击“添加”，打开添加 Core 监控对话框。如下所示：

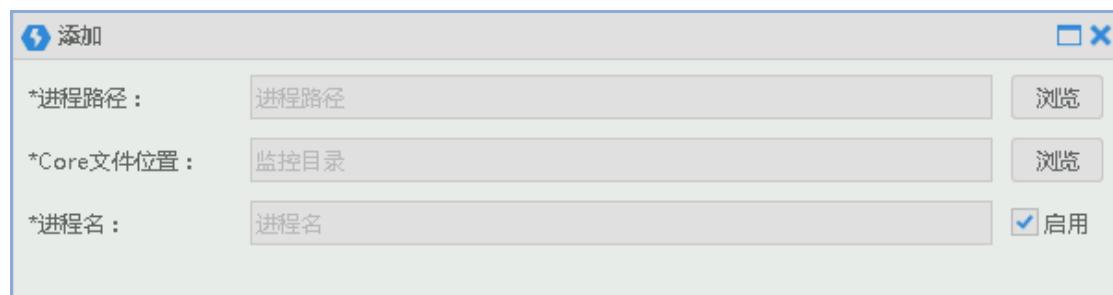


图 7.17 添加 Core 监控界面

指定进程路径和 Core 文件位置，填好进程名，点击“确定”即可。

2. Core 监控详情

可在 Core 监控页面查看每个 Core 监控项的进程名、进程路径、监控目录、创建日期以及最近一次监控时间等，点击导出按钮可导出整个 Core 监控页面的监控信息，导出格式为 PDF。

如下图所示：

| 进程名 | 启用 | 进程路径 | 监控目录 | 创建日期 | 最近一次监控 | 操作 |
|---------------------------------|-------------------------------------|---|---|---------------------|---------------------|----|
| start_client_127.0.0.1_57870.sh | <input checked="" type="checkbox"/> | /home/ATS/ats_root/smoke_client/start_client... | /home/test/smoketest/smoke_57870/dpc/build... | 2022-05-11 18:35:20 | 2022-05-11 18:34:46 | |

图 7.18 Core 监控详情界面

可以对 Core 监控进行编辑、删除以及查看历史信息操作。

相关操作如下：

1) Core 监控编辑

在 Core 监控面板的监控列表中，点击最右侧“操作栏”的 Core 监控的下拉列表，选择“编辑”按钮，打开“编辑”对话框，可以对进程路径、Core 文件位置和进程名进行重新编辑。

2) Core 监控删除

在 Core 监控面板的监控列表中，点击最右侧“操作栏”的 Core 监控的下拉列表，选择“删除”按钮，删除 Core 监控，不再对相应的文件夹中指定程序生成的 core 文件进行监控。

3) Core 监控历史查看

点击详情栏按钮，打开 Core 监控历史页面，可查看生成 Core 文件的历史信息。如下图所示：

| 文件名 | 进程名 | 进程启动时间 | 文件路径 | 操作系统 | Core来源 |
|------------------------------|-------------------------------------|------------------------------------|---------|------|--------|
| dservr.exe.515... dservr.exe | 2022-05-27 14:07... Dcore_alertDump | D:\git\trunk\code\Debug\dservr.exe | Windows | | |
| dservr.exe.516... dservr.exe | 2022-05-27 14:09... Dcore_alertDump | D:\git\trunk\code\Debug\dservr.exe | Windows | | |

图 7.19 Core 监控历史界面

3. Core 监控启用和禁用

通过启用和禁用来开启或停止 Core 的监控项。

启用和禁用方式：

- 1) 在添加 Core 监控时，选择是否启用；
- 2) 在 Core 监控面板的监控列表中，点击“启用”栏的启用/禁用按钮。

7.1.3.5.7 配置网络

配置主机的内外网 IP 信息。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->配置网络。

配置网络对话框如下所示：

| 配置网络 - 223.254.30.150 | | | |
|-----------------------|----------------|----------------|----------------|
| 主机名称 | IP列表 | 内网IP(双击编辑) | 外网IP(双击编辑) |
| dameng-PC | 223.254.30.150 | 223.254.30.150 | 223.254.30.150 |

图 7.20 配置网络界面

配置完成，点击“保存”按钮，保存配置信息。

7.1.3.5.8 升级代理

主机上的数据库代理版本过低可能会影响部分功能的使用，例如监控信息的收集以及部署的功能等。因此当主机上的数据库代理版本过低时，建议升级至与 DEM 版本一致。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->升级代理。

升级代理对话框如下图所示：

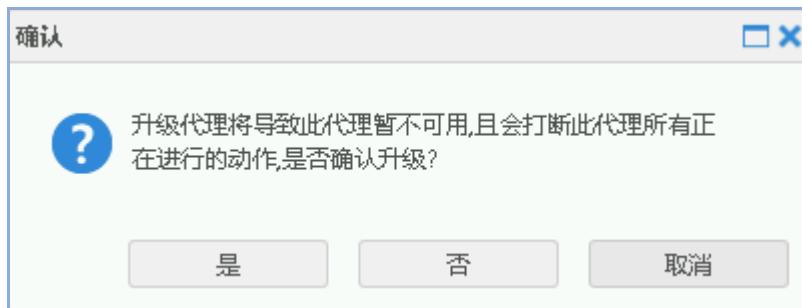


图 7.21 升级代理确认对话框

点击是即可开始升级代理，代理升级完成后会弹出升级成功的提示，“状态”栏会显示升级成功或失败。

在升级过程中，也可以点击“停止升级”按钮，停止升级，但在升级过程中慎用该按钮，因为中途停止可能导致数据库代理无法继续使用。

7.1.3.5.9 删除

删除该主机的监控信息。不可删除存活及连接状态为成功的主机。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->删除。

7.1.3.5.10 属性

查看该主机的基本信息。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->属性。

主机属性介绍如下图所示：



图 7.22 主机属性对话框

7.1.3.6 应用问题分析

本章节主要针对三个常见问题进行分析：无法检查到主机问题、内外网问题以及版本问题。

7.1.3.6.1 无法检查到主机问题

若在 dmagent 启动时，DEM 上未显示对应主机或主机显示连接失败，此时需做如下检查：

1. 检查 dmagent 所在机器防火墙，与 DEM 服务器是否可以 ping 通；
2. 检查 dmagent 所在机器时间和 DEM 服务器时间是否一致，最多相差 5min（主机失效时间，该值通过参数“mf_invalid_time”配置，详见[第 4 章 配置参数说明](#)）。

7.1.3.6.2 内外网问题

DEM 连接 dmagent 用外网，dmagent 收集数据库实例监控信息或集群部署时配置集群内部通讯用内网。

建议配置内外网信息，若不配置，主机最左侧会有警告提醒。

7.1.3.6.3 版本问题

dmagent 版本最好与 DEM 版本一致，建议从 DEM 下载安装，或通过 DEM 升级到一致，否则会导致部分功能出现问题。

7.1.4 数据库监控

DEM 在数据库监控页面列出监控数据库实例的状态、运行时间、CPU、内存、交换区、会话、事务、等待任务数、磁盘以及网络等信息，并可以通过数据库实例各种分析、配置以及启停等操作。

数据库监控页面每隔 1 分钟（该值可以通过配置参数“`dem_page_refresh_freq`”配置）会自动刷新数据库监控信息。数据库监控页面的最右侧会显示最新监测时间，表示所有数据库中收到最新的数据库信息的时间。

7.1.4.1 数据库组

数据库组将多个数据库实例监控信息显示在一起，并可统一操作整个组。

对于数据库集群，集群中包含的组件自然的分到一个组中（强烈建议分到一个组），该方式方便提供一些集群特有操作。

同时，对于其他数据库实例，为在数据库监控页面显示在一起，并能一键做相同的操作，可将这些数据库分为一个组。

7.1.4.2 添加数据库监控

在数据库监控页面的工具栏点击添加按钮添加数据库监控，可以选择添加“单机”或“集群”。

如下添加单机：

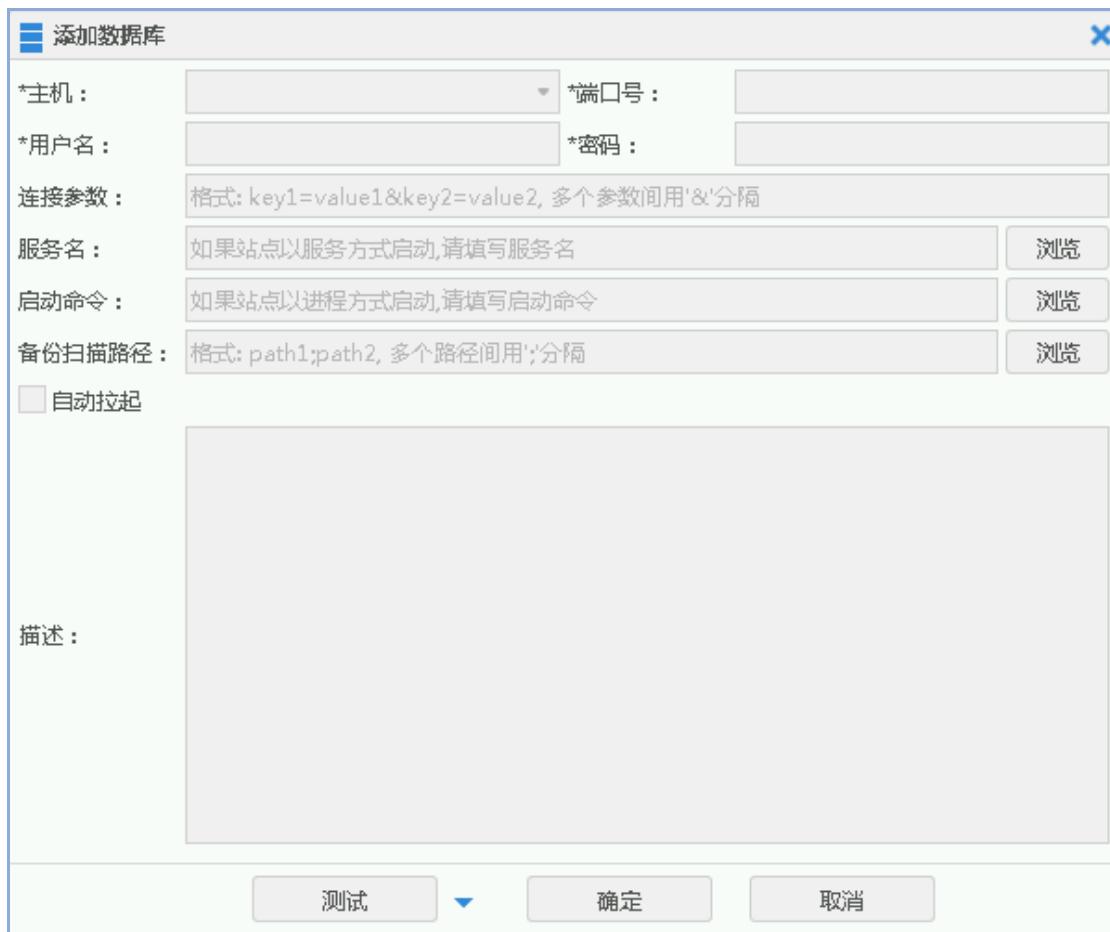


图 7.23 数据库添加界面

填写数据库连接信息，点击“确定”。

如下添加集群：



图 7.24 集群添加界面

填写集群名称添加站点（数据库实例），站点的添加可以通过站点扫描的方式添加，点击“站点扫描”，填写数据库集群的主站点连接信息，DEM 会自动扫描出集群的其他站点信息。点击“确定”，添加集群监控到数据库监控页面。

若需要将一些非属同一集群的数据库分为一组，则可选择集群类型“OTHER”，通过点击“添加”按钮将数据库逐个进行添加。点击“确认”，这些手动添加的数据库在数据库监控页面将以同一个数据库组显示。

7.1.4.3 操作工具栏

主机监控页面最上面的操作工具栏为主机监控提供了刷新、删除、导出等众多操作功能，如下图所示：



图 7.25 数据库监控操作工具栏

各功能介绍如下：

1. 刷新

点击刷新按钮，获取最新的数据库监控信息列表。

2. 添加

点击添加按钮，添加要监控数据库的连接配置信息，或者批量添加数据库集群中所有数据库的连接配置信息。

3. 编辑

点击编辑按钮，修改选中数据库的连接配置信息。

4. 删除

点击删除按钮，从数据库监控信息列表中选中删除一行或多行数据库监控信息。

5. 导出

点击导出按钮，导出整个数据库监控信息页面，导出格式为 PDF。

6. 忽略全部告警

点击忽略全部告警按钮，将所有数据库产生的告警都标记为已处理。

7. 集群显示

点击集群显示按钮，对数据库集群进行过滤、排序后再显示。

8. 搜索

根据输入的 IP/PORT/名称信息在数据库监控信息列表中搜索筛选对应数据库。

9. 暂停/继续

操作工具栏右边的倒计时是关于页面刷新的倒计时，点击它可暂停或继续页面刷新。

10. 隐藏/显示摘要

点击操作工具栏最右边的矩形框，隐藏或显示监控信息汇总栏。

7.1.4.4 监控信息汇总栏

7.1.4.4.1 摘要

数据库摘要部分统计了所有数据库中不同连接状态的数据库个数，即成功状态的数据库个数、失败状态的数据库个数和未知状态的数据库个数；统计了所有数据库中不同集群类型的集群个数，即单机数据库个数、MPP 集群个数、DSC 集群个数、RW 集群个数、DW 集群个数、DPC 集群个数和 MCOPY 集群个数。

如下图所示：



图 7.26 数据库监控摘要

7.1.4.4.2 告警

数据库监控告警部分统计了所有数据库产生的未处理告警中不同告警级别的告警个数，即恢复的告警个数、轻微的告警个数、一般的告警个数、严重的告警个数和致命的告警个数；统计了所有数据库产生的未处理告警中告警数量最多的三个数据库信息；统计了所有数据库产生的未处理告警中时间最近的三条告警信息。

如下图所示：

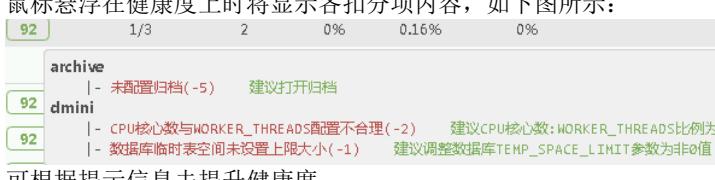


图 7.27 数据库监控告警

7.1.4.5 数据库监控项

下表详细介绍当前收集到的各数据库最新的监控信息：

表 7.2 数据库监控信息说明

| 监控项 | 说明 |
|------|---|
| 数据库 | <p>显示数据库的 IP 地址和端口。 鼠标停留在数据库列，可以显示主机的其他一些信息，如下图所示：</p>  <p>监测时间: 2022-05-12 13:02:26 注册时间: 2022-04-25 14:19:17 编号: 1650867628476 内网: 223.254.30.150 实例名: DMSERVER 描述: 进程号: 213092 状态/模式: OPEN / NORMAL 服务器版本: DM Database Server x64 V8 (--03134283890-20220304-158322-10045) 数据库版本: DB Version: 0x7000c 归档状态: LOCAL/VALID LSN: CKPT(667972), FILE(668090), FLUSH(668090), CUR(668091) KEY: DEVELOP USER ~ 2023-03-04 启动/创建时间: 2022-05-07 09:47:53 / 2022-04-24 14:03:54 主机: dameng-PC(223.254.30.150) 主机编号: 0000000000e0 系统: Microsoft Windows 7 6.1 64 amd64 CPU: Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz250 x 0 内存: 8GB</p> <p>其中监测时间为最近一次收到该主机信息的时间。</p> |
| 连接 | 该数据库的连接状态，为成功、失败或者未知。成功指该数据库可连接且 DEM 服务器与该数据库所在主机网络通信成功。未知指 DEM 服务器与该数据库所在主机的网络通信失败。失败指该数据库不可连接且 DEM 服务器与该数据库所在主机网络通信成功。 |
| 健康度 | <p>该数据库的健康度分数，满分 100 分。 鼠标悬浮在健康度上时将显示各扣分项内容，如下图所示：</p>  <p>根据提示信息去提升健康度。</p> |
| 会话 | 数据库的活动会话和总会话 |
| 每秒事务 | 数据库的每秒事务数 |
| CPU | CPU 使用率。 |
| 内存 | 内存使用率。 |
| 交换区 | 交换区使用率。 |
| 磁盘读 | 磁盘读取速率。 |
| 磁盘写 | 磁盘写入速率。 |
| 网络读 | 网络接收速率。 |
| 网络写 | 网络发送速率。 |
| 告警 | 数据库未处理的告警数量。点击打开告警历史面板。 |
| 操作 | 点击 ▾ 将弹出操作菜单，详细介绍请参见下面章节 6 主机监控操作。 |
| 站点数 | 该数据库组站点总数 |

| | |
|--------|------------------|
| 会话数 | 该数据库组会话获取数和会话总数 |
| 重启次数 | 该数据库组数据库实例重启次数 |
| 故障切换次数 | 该数据库组数据库实例故障切换次数 |

7.1.4.6 数据库监控操作

数据库监控操作包含对每个数据库实例的操作，也有对数据库组的操作，下面将详细展开讲解。

7.1.4.6.1 信息大盘

点击信息大盘按钮，打开数据库大盘页面。数据库大盘显示数据库实例最近的监控汇总信息。

数据库大盘以模块形式布局显示各信息，用户可点击布局按钮按照个人需求与喜好对各模块布局。点击导出按钮可导出整个数据库大盘页面图表，导出格式为 PDF。数据库大盘页面右上角监测时间为上次页面刷新时间。模块右上角的下拉菜单提供与数据库监测页面数据库操作栏下拉菜单中功能相同的操作按钮。不同模块提供的操作按钮不尽相同，每个模块仅提供与该模块相关的数据库操作按钮。

数据库大盘各模块简单介绍如下：

1. 基本信息

显示该数据库的连接状态、库名、归档状态、编号、KEY、实例名、启动时间、运行时长、状态、模式、主机名、主机 IP、服务器版本、操作系统、数据库版本、CPU、上一次监测时间和内存等信息。

2. 告警 | 日志 (TOP3)

告警，显示该数据库产生的未处理告警中不同告警级别的告警个数和时间最近的三个告警信息；日志，显示该数据库时间最近的三条日志信息。

3. 健康度

显示数据库健康度和对应的扣分项与提分建议。

4. CPU&MEM

显示 CPU 使用率和 MEM 使用率。

5. 运维事项

显示关于该数据库的巡检信息和备份信息。

6. 高频 | 最慢 (TOP3)

高频，显示时间最近的三条 SQL 信息；最慢，显示时间最近的三条 SQL 信息。

7. FIO

显示该数据库最近 10 次监测的磁盘读写速率。

8. NIO

显示该数据库最近 10 次监测的网络读写速率。

9. TPS&QPS

显示该数据库最近 10 次监测的 TPS（每秒事务数）和 QPS（每秒请求数）。

10. 会话

显示该数据库最近 10 次监测的会话总数和活跃会话数。

11. DML

显示该数据库最近 10 次监测的每秒 DML 个数。

12. DDL

显示该数据库最近 10 次监测的每秒 DDL 个数。

13. 表空间使用 | 表空间使用率 (TOP3)

表空间使用，以扇形统计图显示所有表空间的使用大小；表空间使用率 (TOP3)；显示使用率最大的三个表空间的总量、空闲大小、最大大小和文件数。

14. 线程

显示该数据库最近 10 次监测的线程数。

15. 等待任务

显示该数据库最近 10 次监测的等待任务数。

16. 等待时间

显示该数据库最近 10 次监测的任务等待时间。

鼠标移动到曲线图上，可以查看具体每个时间点的监测结果。

路径：智能运维->资源监控->主机->列表“操作栏”->下拉按钮->信息大盘。

如下图所示：

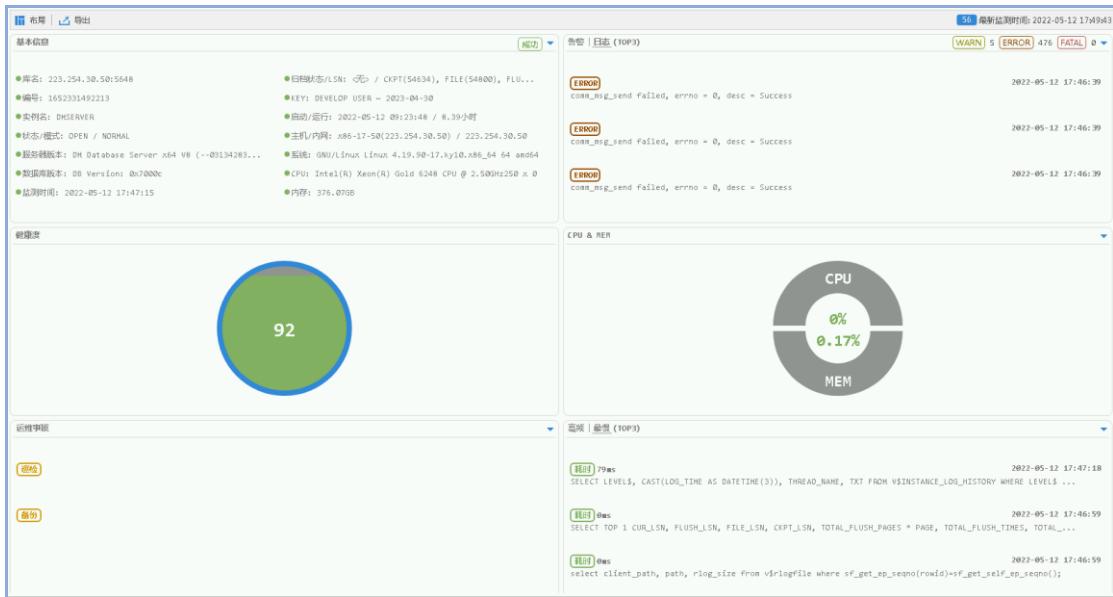


图 7.28 数据库大盘

7.1.4.6.2 拓扑结构

打开数据库或组的拓扑结构图，可以直观的看到数据库实例或组的拓扑关系，特别是数据库集群组，可以直观详尽的显示数据库集群的拓扑关系图。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->拓扑结构。

7.1.4.6.3 辅助组件管理

数据库或组的运行通常会有些辅助组件来完成一些功能。

例如：

数据守护环境：对应数据库实例利用 dmwatcher 组件辅助拉起数据库。

DMSC 环境：对应数据库实例包含 DMASM 和 DMCSS 组件。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->辅助组件管理。

打开辅助组件管理，可以对数据库或组相关辅助组件进行管理。

7.1.4.6.4 AWR 报告

用于查看当前数据库实例的 AWR 报告。报告中包括以下几个部分：

1. 概述

数据库及所在主机的基本信息。

2. 资源

数据库及其所在主机的资源使用情况曲线图。包括内存、CPU、网络 IO 和文件 IO。

鼠标移动到曲线图上，可以查看具体每个时间点的资源使用情况。

3. 性能

反映数据库性能指标的曲线图。包括 TPS（每秒事务数）、QPS（每秒请求数）、DML（每秒操作数，包括插入、更新和删除）、DDL（每秒 DDL 操作数）、DB 等待任务数、DB 任务平均等待时间、归档等待任务数和归档任务平均等待时间。

鼠标移动到曲线图上，可以查看具体每个时间点的性能指标情况。

4. 会话

数据库会话数和线程数曲线图。包括会话数（总会话和活动会话）以及 DB 线程数。

鼠标移动到曲线图上，可以查看具体每个时间点的会话情况或 DB 线程数。

5. SQL

数据库执行 SQL 的统计信息，包括前 20 条慢 SQL 和前 20 条高频 SQL。

6. 死锁

数据库最近 20 次死锁信息。

7. 登录异常

数据库最近 20 次用户异常登录信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->AWR 报告。

如下图所示：

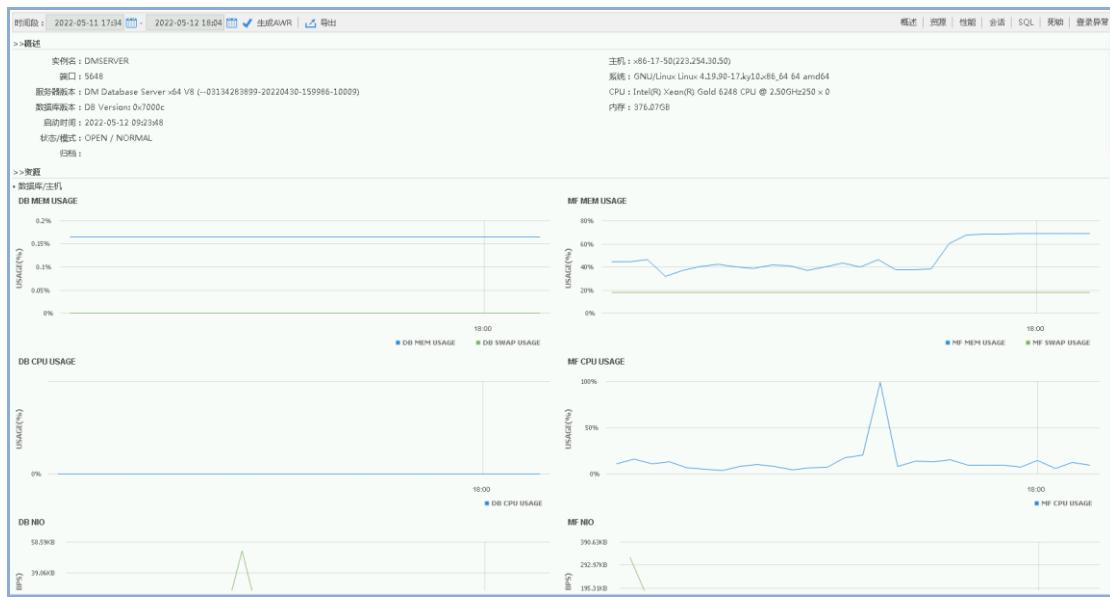


图 7.29 AWR 报告

7.1.4.6.5 巡检报告

用于查看当前数据库实例的所有自动巡检报告。可通过下载按钮下载指定巡检报告，通过删除按钮删除指定报告。报告包括数据库健康度和巡检项两部分。数据库健康度部分为巡检结果简报，显示数据库健康度、扣分项、扣分值和提分建议；巡检项部分为巡检内容详情介绍，巡检项包括以下几个部分：

1. 主机概括

数据库所在主机的基本信息。包括主机 IP、操作系统、内存大小、CPU 核心数、磁盘使用情况。

2. 数据库概括

数据库基本信息。包括数据库名、数据库状态、数据库启动时间、数据库版本号、实例名称、数据库模式、所属集群类型、字符集、页大小、簇大小和大小写敏感。

3. 性能概括

数据库及其所在主机的 CPU 使用率、内存使用率、交换区使用率、网络收发速率、磁盘读写速率。

4. 表空间

数据库表空间信息。包括表空间数据文件总个数、表空间总量和表空间总量增量，以及各个表空间的表空间大小、总大小、总大小增量、空闲大小、使用率、数据文件名、数据文件路径和是否为自动扩展。

5. Redo 日志

数据库 Redo 日志信息。包括日志总量、日志空闲量、空闲率、以刷盘 LSN 和当前 LSN，以及 Redo 日志文件的名称、路径和总大小。

6. 归档日志 (最近 N 天)

数据库最近 N 天的归档信息。“N”由创建自动巡检时的参数“归档文件查看天数”配置。包括归档名称、归档方式、归档目标、归档状态和归档文件空间限制，以及各归档日志文件的路径、大小和关闭时间。

7. 备份 (最近 N 天并校验最近 0 天)

数据库最近 N 天的备份信息。“N”由创建自动巡检时的参数“备份文件查看天数”配置。包括备份级别、备份文件总数、备份文件总量，以及每次备份的备份路径、备份名称、备份时间、备份有效性、备份类型、备份级别、备份大小和备份文件列表。

8. 用户与会话

数据库用户信息与会话信息。用户信息，包括序列号、服务器类型、有效期、授权用户名和授权用户数，以及各用户的用户名、当前会话数、上限会话数、口令过期日期、表数量和表数增量；会话信息，包括总会话数、活跃会话数，以及所有会话的会话状态、用户名、会话创建时间、客户端 IP 和客户端类型。

9. 死锁 (最近 N 天)

数据库最近 N 天的死锁情况。“N”由创建自动巡检时的参数“死锁历史查看天数”配置。包括死锁时的序列号、会话 ID、事物 ID、SQL 语句、发生时间、用户名、会话 IP 和应用名称。

10. 运行日志 (最近 N 天)

数据库最近 N 天的运行日志信息。“N”由创建自动巡检时的参数“运行日志查看天数”配置。包括每条日志的发生时间、日志级别和日志信息。

11. DmINI 配置 (非默认值)

该数据库的 dm.ini 参数信息。点击 dmINI 配置旁的矩形框可隐藏/显示参数信息。

通过点击右下角的 TOP 按钮可以回到报告首页。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->巡检报告。

如下图所示：

| 扣分项 | 扣分值 | 建议 |
|------------------------------|-----|-------------------------------|
| 1 CPU核心数与WORKER_THREADS配置不合理 | 2 | 建议CPU核心数:WORKER_THREADS比例为2:1 |
| 2 内存使用率超过90% | 5 | 建议观察系统内存 |
| 3 数据库临时表空间未设置上限大小 | 1 | 建议调整数据库TEMP_SPACE_LIMIT参数为非0值 |

图 7.30 巡检报告

7.1.4.6.6 表空间分析

用于查看最近一段时间内数据库实例各个表空间的使用情况。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->表空间分析。

表空间分析页面如下：

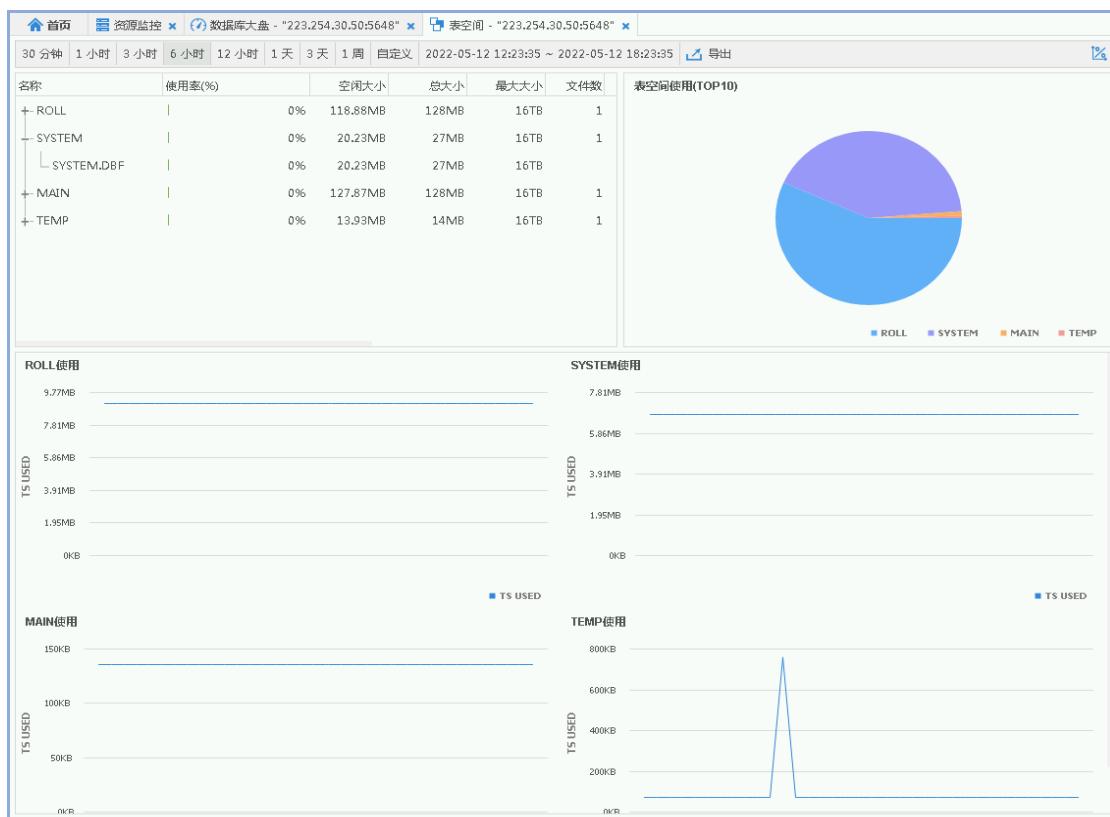


图 7.31 表空间分析

上方表格和扇形统计图为收集到的最新表空间监控信息。

下方曲线图为各时间段每个表空间的使用情况，通过右上角的按钮可以选择切换查看表空间使用大小或表空间使用率。通过最上面的时间按钮，切换查看不同时间段内的表空间使用情况。

点击导出按钮可导出整个表空间页面的图表，导出格式为 PDF。

7.1.4.6.7 SQL 分析

用于查看最近一段时间内数据库实例执行的 SQL 语句。通过右上角的下拉按钮可以切换分析的 SQL 数据源：动态视图或日志文件。点击导出按钮可导出整个 SQL 分析页面的图表，导出格式为 PDF。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->SQL 分析。

如下图所示：

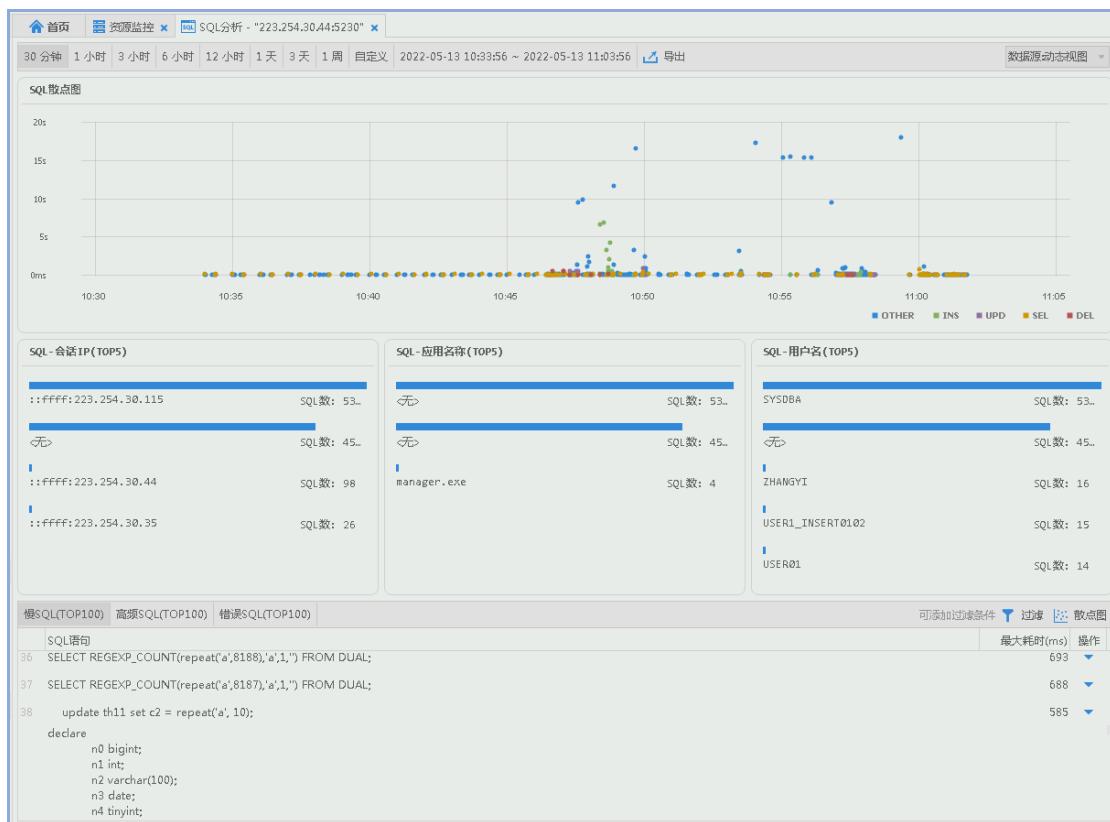


图 7.32 SQL 分析

上方散点图为收集到的指定时间段的 SQL 信息，包括 SQL 开始时间、执行耗时和 SQL 类型。

中间的柱状图显示指定时间段内 SQL 数最多的 5 个会话 IP、应用名称和用户名。

下方列表可以查看指定时间段的慢 SQL、高频 SQL 和错误 SQL 信息。先通过最小执行耗时参数“gather_db_sqllog_file_execetime”或“gather_db_sqllog_view_execetime”筛选 DEM 收集的 SQL 语句，然后按照指定条件将收集的 SQL 语句排序显示。

其中慢 SQL、高频 SQL 和错误 SQL 的定义分别为：

1. 慢 SQL 为执行时间最长的前 100 条 SQL 语句；
2. 高频 SQL 为执行次数最多的前 100 条 SQL 语句；
3. 错误 SQL 即执行出错的时间最近的前 100 条 SQL 语句。

点击 SQL 操作栏“明细”按钮打开 SQL 详细信息对话框可查看更多 SQL 信息。通过过滤功能可筛选显示指定条件的 SQL。点击散点图按钮通过散点图来查看该类型 SQL 的执行情况。

7.1.4.6.8 登录锁定

用于查看最近一段时间内数据库用户登录被锁定的历史记录。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->登录锁定。

如下图所示：

图 7.33 登录锁定

通过切换时间按钮，可以查看不同时间段的登录锁定信息。点击导出按钮可导出整个登录锁定页面的锁定信息，导出格式为 PDF。

7.1.4.6.9 死锁分析

用于查看最近一段时间内数据库发生死锁的历史记录。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->死锁分析。

如下图所示：

图 7.34 死锁分析

通过切换时间按钮，可以查看不同时间段的数据库发生的死锁信息。点击导出按钮可导出整个死锁页面内容，导出格式为 PDF。

7.1.4.6.10 会话分析

用于查看最近一段时间内数据库实例或组会话数变化情况。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->会话分析。

如下图所示：

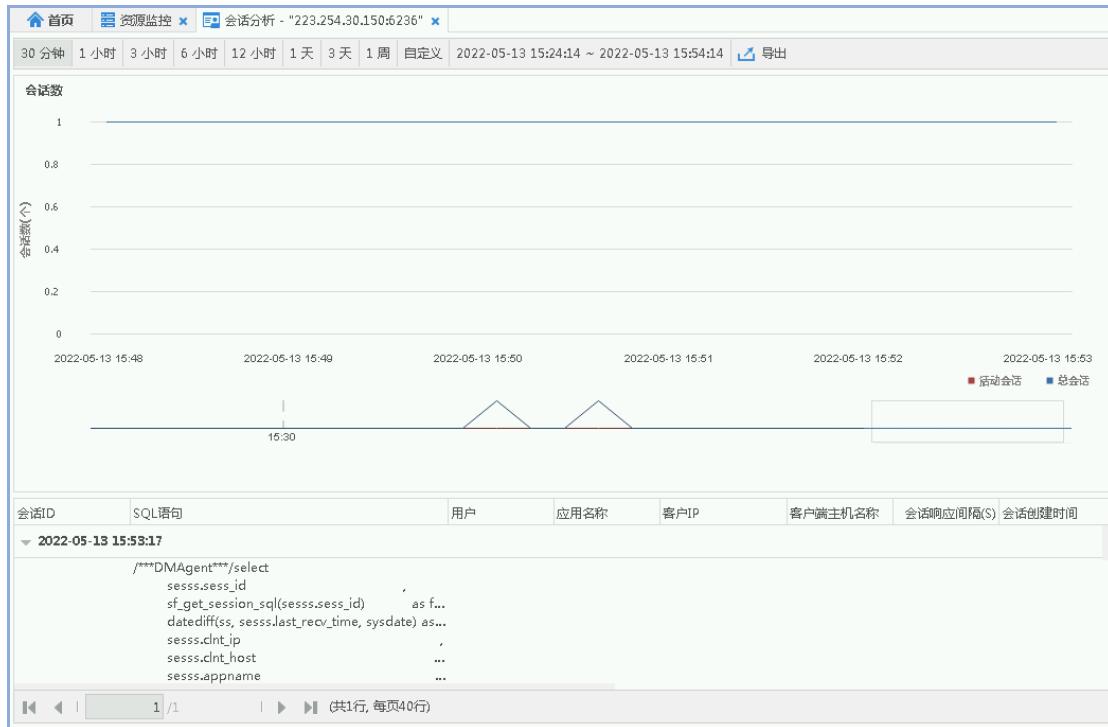


图 7.35 会话分析

切换时间按钮，可以查看不同时间段的会话信息。点击导出按钮可导出整个会话分析页面的图表，导出格式为 PDF。

会话数图表中显示的是总会话数和活动会话数变化曲线图。横坐标为时间，纵坐标为会话数。鼠标放到曲线图的某一点上可以查看该时刻的详细信息。当图表跨时间段较长不好定位到具体的时间点时，可以拖动图表下方的标尺，临时放大指定区间的曲线图。

下方会话信息列表中为当前所选时间段内的会话信息。列表中的记录按时间降序排列，当查看某个数据库实例时，按日期分组显示，当查看组的会话信息时，则按数据库实例分组显示。

7.1.4.6.11 事件分析

用于查看最近一段时间内数据库实例事件等待情况。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->事件分析。

如下图所示：

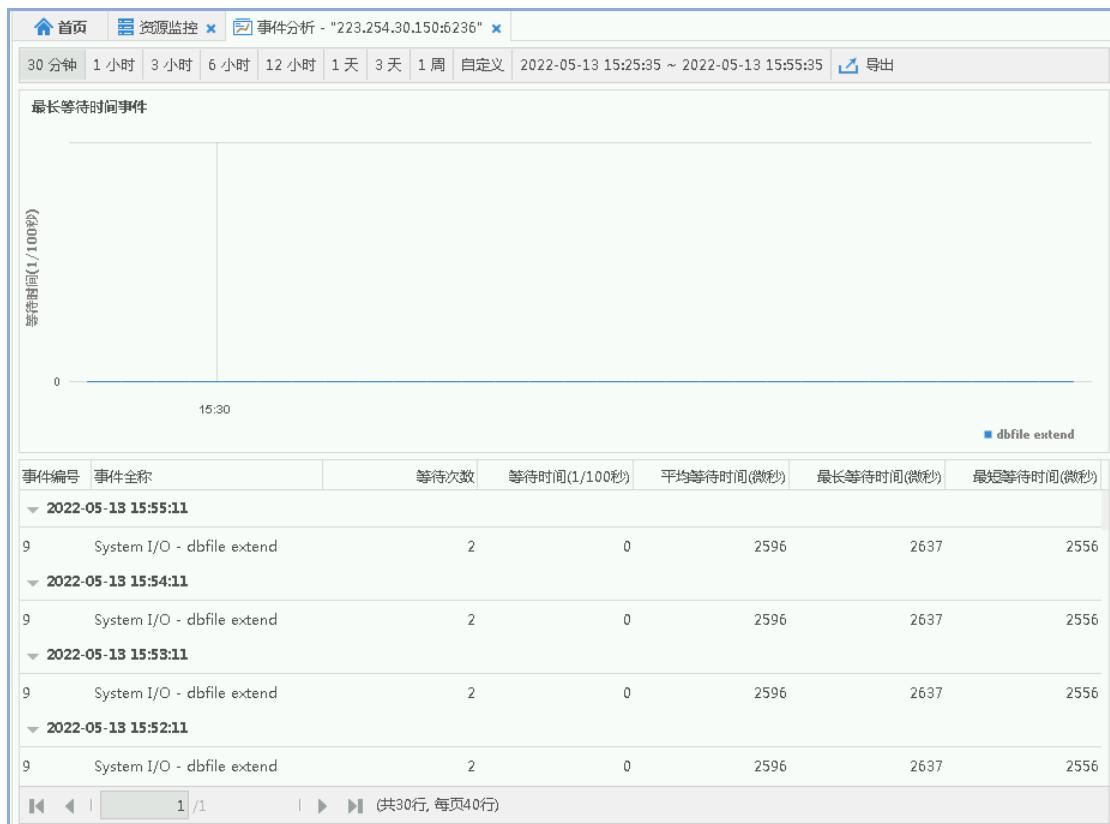


图 7.36 事件分析

切换时间按钮，可以查看不同时间段的事件等待情况。点击导出按钮可导出整个事件分析页面的图表，导出格式为 PDF。

上方为事件等待时间图表：图表显示最长等待时间排名前 5 的五种事件类型的曲线图。横坐标为时间，纵坐标为等待时间，单位 1/100 秒。

下方为事件等待信息列表：记录当前所选时间段内的事件等待信息。列表中的记录按日期分组显示。

7.1.4.6.12 自定义 SQL 监控

除了给出的会话、登录和表空间监控之外，还需监控数据库参数，但往往与实际应用需求相关，无法详列，故只能自定义，根据实际情况指定要监控的 SQL，该方式可以适用更多的监控场景。dmagent 会按一定频率（配置参数“添加自定义 SQL 监控时设置的监控频率”）执行自定义的 SQL，并将结果发给 DEM，DEM 显示自定义 SQL 的执行情况。

自定义 SQL 监控，必须先添加自定义 SQL，SQL 监控信息从添加后的一个收集频率后开始收集。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->自定义 SQL 监控。

4. 添加自定义 SQL 监控

在自定义 SQL 监控面板点击“添加”，打开添加自定义 SQL 监控对话框，如下所示：



图 7.37 添加自定义 SQL 监控界面

填写监控名称、配置监控频率和超时时间、自定义 SQL 语句后，点击“确定”即可。

5. 自定义 SQL 监控详情

可在自定义 SQL 监控页面查看每个自定义 SQL 监控项的自定义 SQL、最近执行时间、最近执行状态、最近执行结果、以及最近执行耗时等，点击导出按钮可导出整个自定义 SQL 监控页面的监控信息，导出格式为 PDF。

如下图所示：



图 7.38 自定义 SQL 监控详情界面

点击详情栏按钮，打开自定义 SQL 监控执行历史页面，可查看自定义 SQL 在不同时间段的执行历史信息。如下图所示：

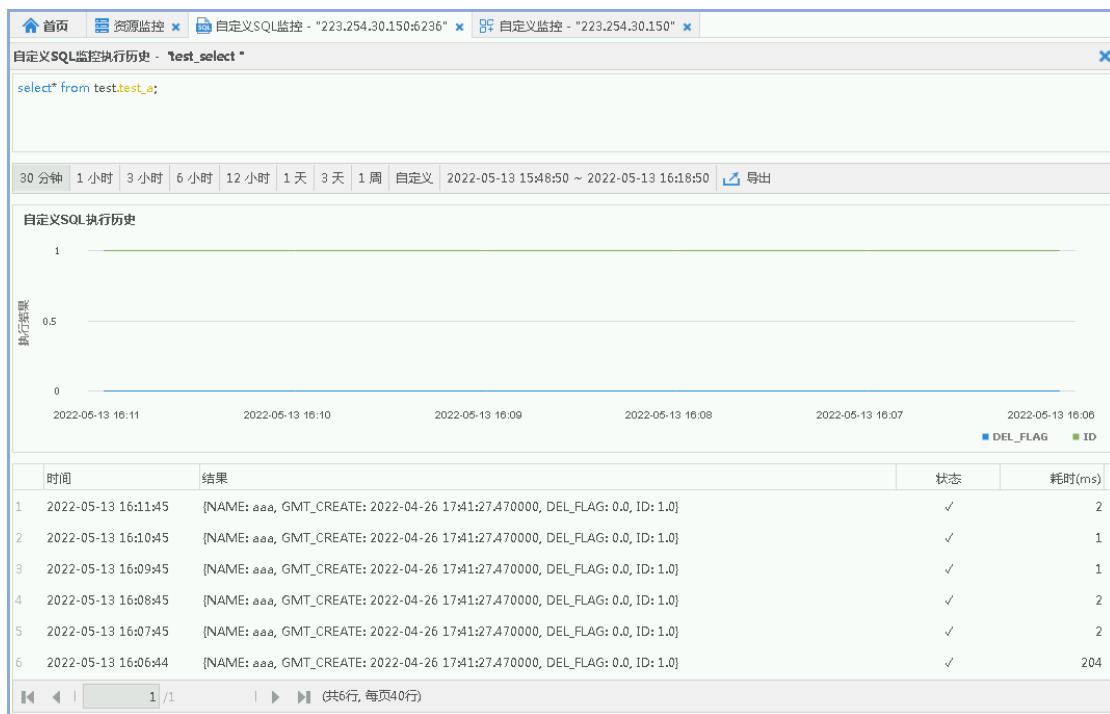


图 7.39 自定义 SQL 执行历史界面

6. 自定义 SQL 监控启用和禁用

通过启用和禁用来开启或停止自定义的监控项。

启用和禁用方式：

- 1) 在添加自定义监控时，选择是否启用；
- 2) 在自定义监控面板的监控列表中，点击“启用”栏的启用/禁用按钮。

7.1.4.6.13 表数据监控

用于查看数据库的表数据监控信息。

表数据监控，必须先添加表监控，监控信息从添加后的一个收集频率后开始收集。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->表数据监控。

1. 添加表数据监控

在表数据监控面板点击“添加”，打开添加表数据对话框，从下拉列表中选择被监控表的模式名和表名，点击确认，完成添加表数据监控。

2. 表数据监控详情

可在表数据监控页面查看每个表数据监控项的监控表名、最近获取时间和表数据行数信息。

如下图所示：

| 表名 | 最近获取时间 | 表数据行数 | 详情 |
|-----------------|---------------------|-------|----|
| "TEST"."TEST_A" | 2022-05-13 16:27:45 | 3 | |

图 7.40 表数据监控

点击详情栏按钮，查看表数据行数曲线图。如下图所示：



图 7.41 表数据行数曲线图

切换时间按钮，可以查看不同时间段的表数据行数。点击导出按钮可导出该表行数曲线图，导出格式为 PDF。

7.1.4.6.14 运行日志监控

用于查看数据库的运行日志信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->运行日志监控。

如下图所示：

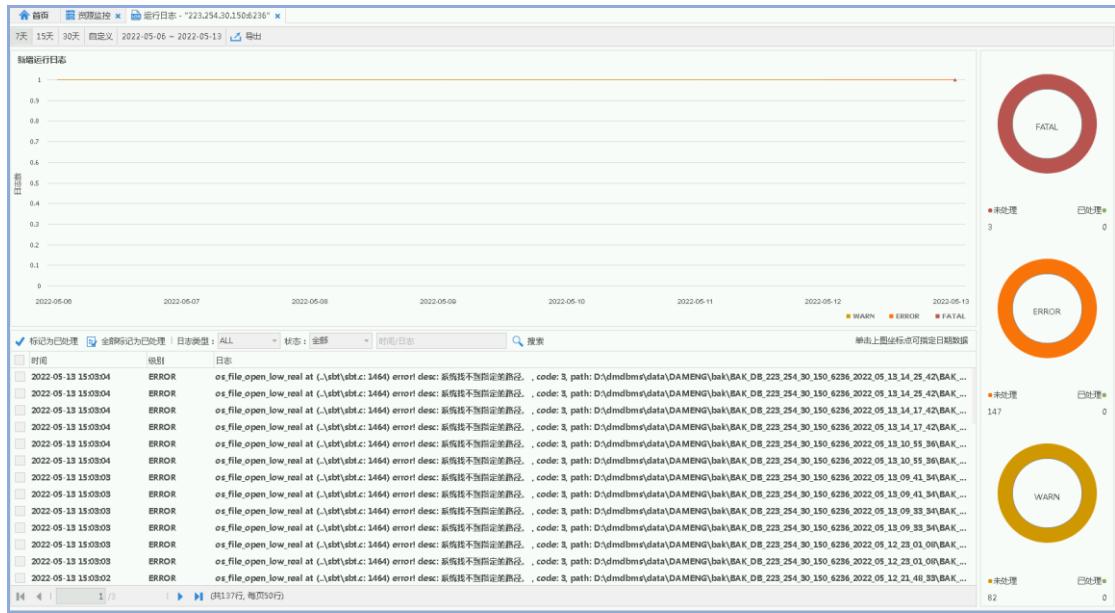


图 7.42 运行日志监控图

通过最上面的时间按钮，切换查看不同时间段内的数据库运行日志信息。点击导出按钮可导出整个运行日志页面的图表，导出格式为 PDF。

上方曲线图为时间段内数据库每日各类运行日志新增数变化曲线图。

下方表格记录时间段内数据库的所有运行日志信息。通过工具栏的类型按钮、状态按钮和搜索按钮，可以查看指定类型、指定状态、指定时间/日志的日志信息。通过工具栏的标记为已处理按钮将被选日志标记为已处理，通过全部标记为已处理按钮将所有未处理日志标记为已处理。

右方统计了 fatal、error 和 warn 三个级别的未处理日志数和已处理日志数。

7.1.4.6.15 事务日志监控

用于查看数据库的事务日志信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->事务日志监控。

如下图所示：

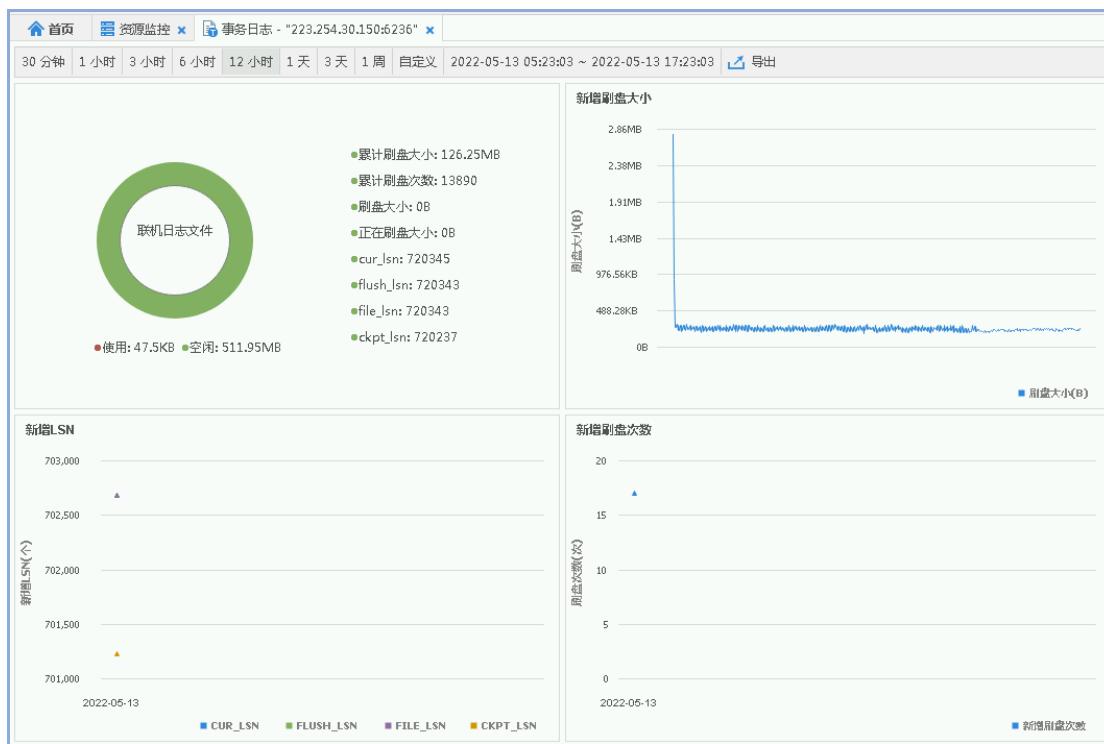


图 7.43 事务日志监控图

通过最上面的时间按钮，切换查看不同时间段内的数据库事务日志信息。点击导出按钮可导出整个事务日志页面的图表，导出格式为 PDF。

左上方为时间段内联机日志文件统计信息。包括联机日志文件使用大小、空闲大小、累计刷盘大小、累计刷盘次数、刷盘大小、正在刷盘大小和 4 类 LSN 值。

右上方曲线图记录时段内的实时新增刷盘大小。

左下方曲线图记录时段内 4 类 LSN 每日新增个数。

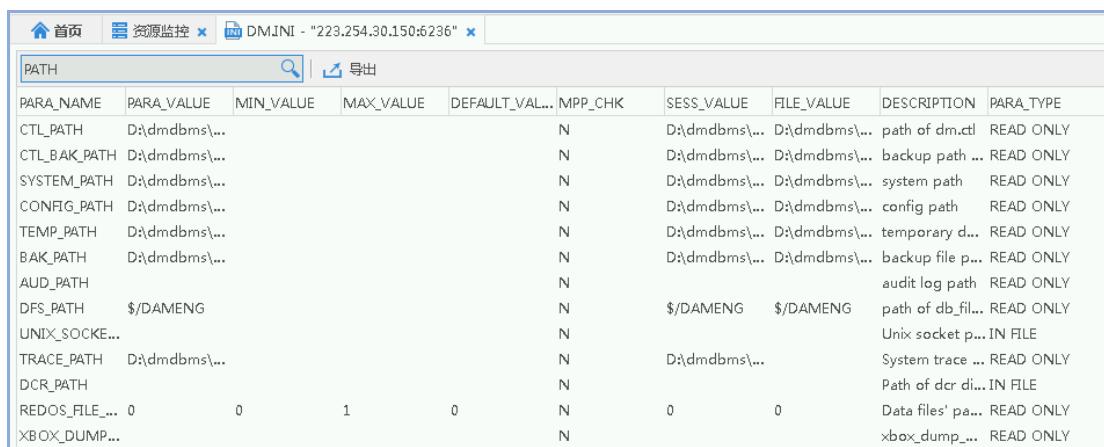
右下方曲线图记录时段内每日新增刷盘次数。

7.1.4.6.16 查看 DM.INI

用于查看数据库 DM.INI 配置文件。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->查看 DM.INI。

如下图所示：



The screenshot shows a table titled 'PATH' within the 'DM.INI - "223.254.30.150:6236"' tab. The table has columns: PARA_NAME, PARA_VALUE, MIN_VALUE, MAX_VALUE, DEFAULT_VAL..., MPP_CHK, SESS_VALUE, FILE_VALUE, DESCRIPTION, and PARA_TYPE. The data includes various system paths like CTL_PATH, BAK_PATH, SYSTEM_PATH, etc., with their respective values and descriptions.

| PARA_NAME | PARA_VALUE | MIN_VALUE | MAX_VALUE | DEFAULT_VAL... | MPP_CHK | SESS_VALUE | FILE_VALUE | DESCRIPTION | PARA_TYPE |
|----------------|---------------|-----------|-----------|----------------|---------|---------------|--------------------------------|-------------------|-----------|
| CTL_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... path of dmctl | READ ONLY | |
| CTL_BAK_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... backup path ... | READ ONLY | |
| SYSTEM_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... system path | READ ONLY | |
| CONFIG_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... config path | READ ONLY | |
| TEMP_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... temporary d... | READ ONLY | |
| BAK_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | D:\dmdbms\... backup file p... | READ ONLY | |
| AUD_PATH | | | | | N | | | audit log path | READ ONLY |
| DFS_PATH | \$/DAMENG | | | | N | \$/DAMENG | \$/DAMENG | path of db_file | READ ONLY |
| UNIX_SOCKET... | | | | | N | | | Unix socket p... | IN FILE |
| TRACE_PATH | D:\dmdbms\... | | | | N | D:\dmdbms\... | | System trace ... | READ ONLY |
| DCR_PATH | | | | | N | | | Path of dcr di... | IN FILE |
| REDOS_FILE_... | 0 | 0 | 1 | 0 | N | 0 | 0 | Data files' pa... | READ ONLY |
| XBOX_DUMP... | | | | | N | | | xbox_dump_... | READ ONLY |

图 7.44 查看 DM.INI

7.1.4.6.17 修改 DM.INI

用于修改数据库的 DM.INI 配置文件。

路径：智能运维->资源监控->数据库->列表“操作栏”->数据库实例下拉按钮->修改 DM.INI。

如下图所示：

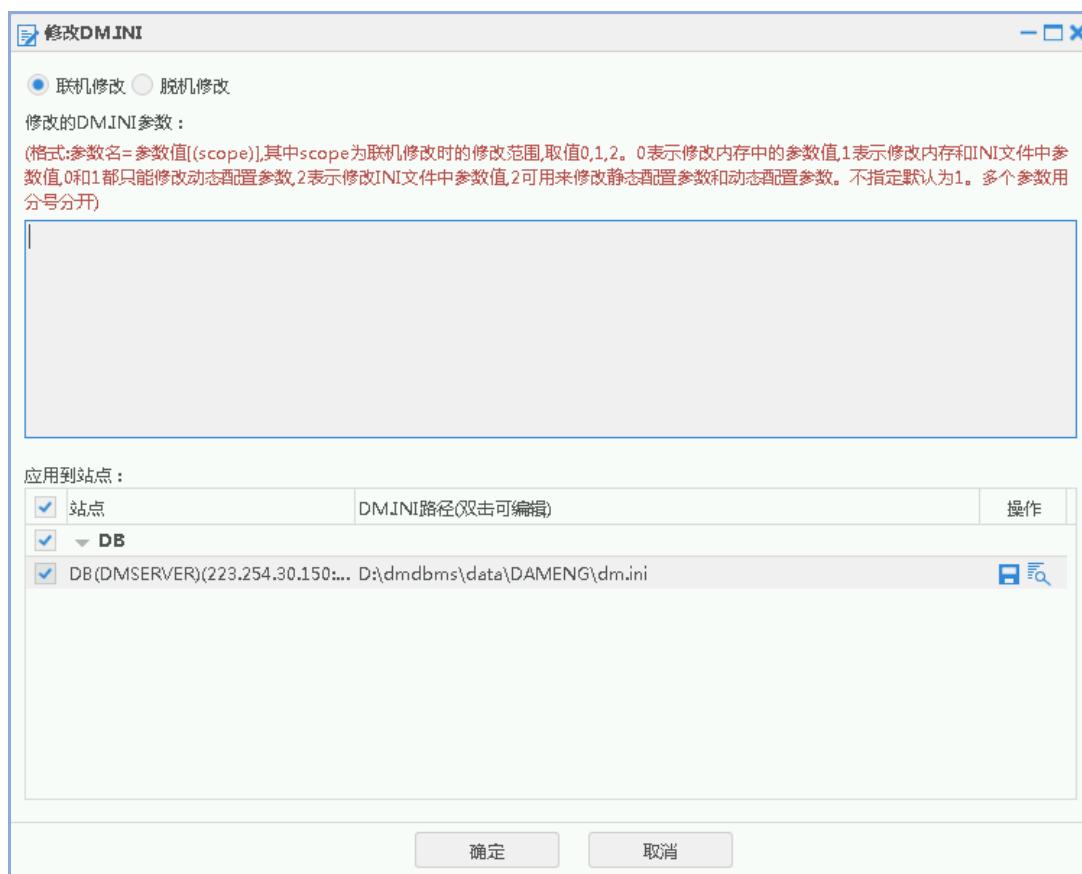


图 7.45 修改 DM.INI

有 2 种修改 DM.INI 的方式：联机修改 DM.INI 和脱机修改 DM.INI。

1. 联机修改 DM.INI

通过连接数据库，执行 SF_SET_SYSTEM_PARA_VALUE(参数名, 参数值, 0, scope)，因此需要给定参数名和参数值，scope 格式为参数名=参数值 (scope)。

2. 脱机修改 DM.INI

通过访问 DM.INI 文件，直接替换文件中对于 INI 参数的值，因此需要给定参数名和参数值，格式为参数名=参数值。

7.1.4.6.18 启动/停止

用于启动/停止数据库实例或组。

若为数据库集群组，则可以启动/停止整个数据库集群组，包括集群组的辅助组件。

可以单独对每个数据库实例或数据库组件启动/停止，也可以对列表中的所有数据库实例和数据库组件一起启动/停止（启动/停止集群）。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->启动/停止。

如下图显示：



图 7.46 启动/停止数据库实例或组

可以通过服务方式来启动/停止，也可以通过进程方式来启动/停止。

对启动，数据库或数据库组件可以选择以服务方式启动还是以进程方式启动，如果以服务方式启动，需先将各数据库和数据库组件注册成服务。



注意： 对停止，数据库或数据库组件须按其启动的方式来停止，即如果以服务方式来启动，则必须以服务方式停止，如果以进程方式启动，则以进程方式停止。

1. 以服务方式启动/停止

如图 7.46 所示，需要给定要启动数据库的服务名，通过“操作”栏“启动”或“停止”按钮来启动/停止数据库。

服务填写可以通过点击右上角的“配置服务名”按钮来填写符合安装默认规定的服务名，也可以直接双击编辑服务名。

当表格中的“状态”信息自动更新不准确时，可通过右上角的“刷新状态”按钮来刷新状态信息。

右上角的“启动”和“停止”按钮可启动/停止列表中的所有组件。

2. 以进程方式启动/停止

如下图所示：

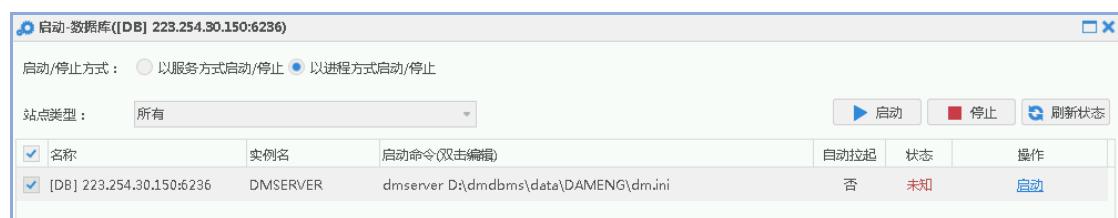


图 7.47 以进程方式启动/停止数据库实例或组

以进程方式启动需要给定全路径的数据库启动命令，通过“操作”栏“启动”或“停止”按钮来启动/停止数据库。

7.1.4.6.19 忽略全部告警

将该数据库组产生的告警都标记为已处理。

路径：智能运维->资源监控->数据库->列表“操作栏”->组下拉按钮->忽略全部告警。

7.1.4.6.20 记录清除

清除该数据库组的重启次数或故障切换次数记录。

路径：智能运维->资源监控->数据库->列表“操作栏”->组下拉按钮->记录清除。

7.1.4.6.21 编辑

用于编辑该组或数据库实例信息的连接配置信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->编辑

7.1.4.6.22 删除

用于删除该组或数据库实例监控。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->删除

7.1.4.6.23 属性

用于查看该组或数据库实例的基本信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->组或数据库实例下拉按钮->属性

数据库实例属性介绍如下图所示：



图 7.48 数据库实例属性对话框

数据库组属性介绍如下图所示：

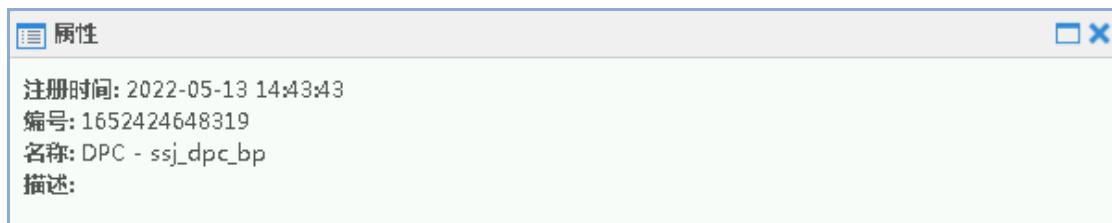


图 7.49 数据库组属性对话框

7.1.4.6.24 集群分析

用于查看集群的实时监控信息。报告中包括以下几个部分：

1. 概述

集群站点基本信息。包括实例、连接、状态、归档、服务器版本、数据库版本和启动时间等。

2. 会话

集群各站点的会话分布情况。

3. 表空间

集群各站点的表空间使用情况。

4. 表监控

表数据在各集群各站点的分布情况，如下图所示：



图 7.50 表监控

在表格上方的选择框中选择一个表进行查看。

5. 表锁

集群表锁信息，如下图所示：

| >>表锁 | | | | | |
|-------|--------|------|------|-------|--------|
| 模式名 : | CTISYS | 表名 : | | 客户端IP | 客户端主机名 |
| 实例 | | | 应用名称 | 表名 | 会话ID |
| | | | | | |

图 7.51 集群表锁信息

可在表格上方的选择框中选择一个表进行查看。

6. 死锁

集群各站点最近发生的 20 次死锁信息。

路径：智能运维->资源监控->数据库->列表“操作栏”->组下拉按钮->集群分析。

如下图所示：

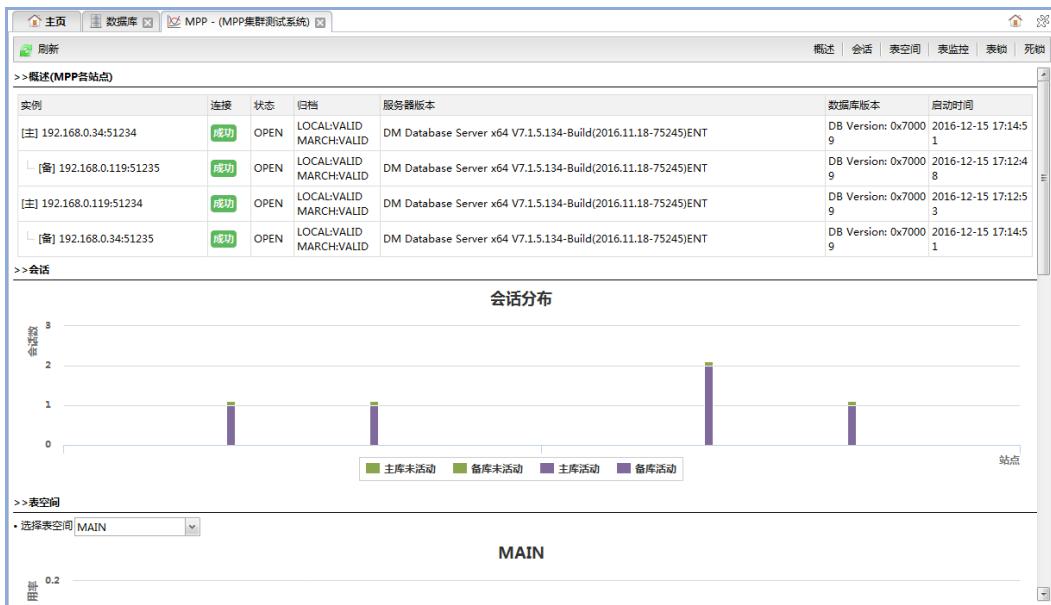


图 7.52 集群分析

7.1.4.6.25 集群监视器

用于启动集群监视器，与集群监视器命令行版功能一致，数据守护和读写分离集群的监视器为 dmmonitor，DMDSC 监视器为 DMCSSM。

路径：智能运维->资源监控->数据库->列表“操作栏”->组下拉按钮->集群监视器。

打开集群监视器时，首先弹出监视器 INI 配置文件信息，数据守护和读写分离集群的监视器的 INI 配置文件为 dmmonitor.ini，DMDSC 的监视器 INI 配置文件为 dmcssm.ini。

如下图所示：

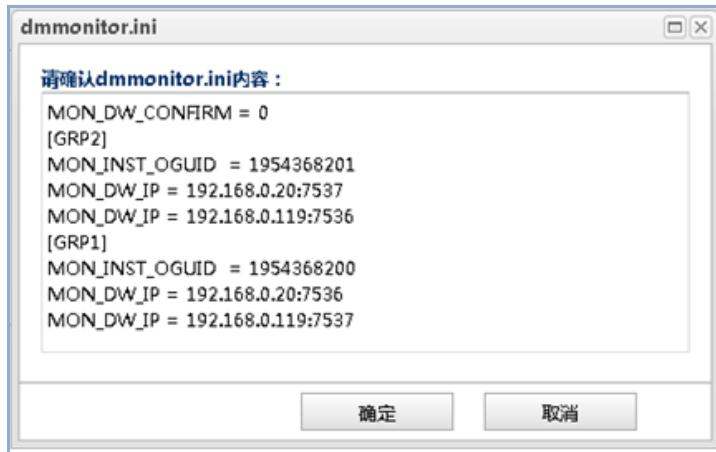


图 7.53 集群监视器 dmcaam.ini

该 INI 配置文件由系统根据收集的信息自动生成，但有可能因为信息收集不全，导致 INI 配置文件生成失败或有误，此时需要用户自己手动编辑。建议每次打开监视器时确认 INI 配置文件是否准确。

确认监视器 INI 配置文件信息无误后，点击确定，可打开集群监视器面板，该面板用于维护和管理集群。数据守护和读写分离集群的监视器功能同数据守护监视器（dmmonitor）一致，界面显示如下图所示：

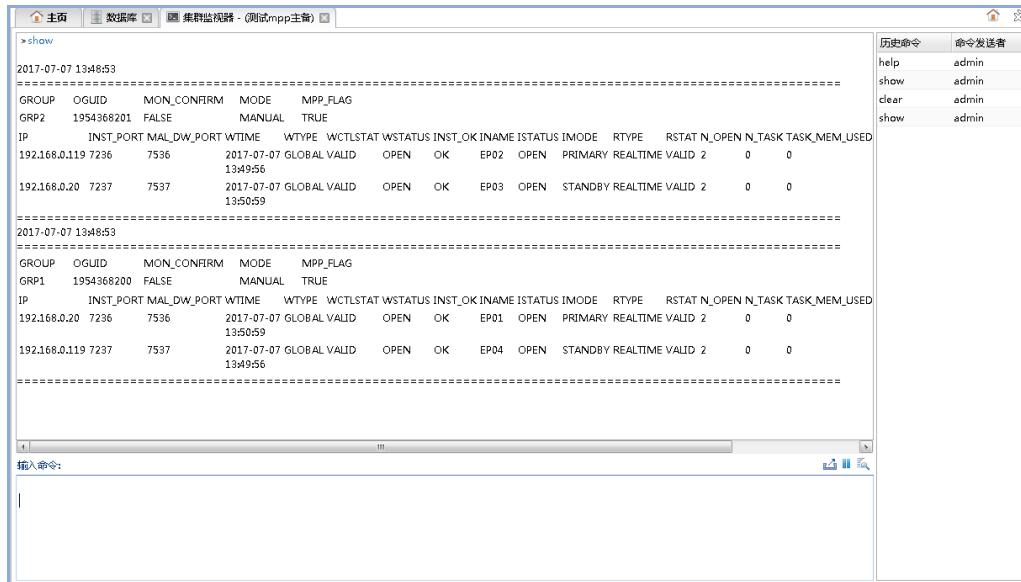


图 7.54 集群监视器面板

集群管理面板主要由以下几个部分组成：

1. 消息回显面板

用于显示执行的命令和命令执行的结果信息。

2. 命令输入面板

用于输入命令。

具体包含的命令以及命令的使用方式同监视器命令行版一致，使用时可以输入 help 查看，数据守护和读写分离集群的监视器详细信息请参见《DM8 数据守护与读写分离集群 v4.0》，DMDSC 监视器详细信息请参见《DM8 共享存储集群》。

3. 历史命令面板

用于查看最近执行的 10 条命令。

7.1.4.7 应用问题分析

数据库监控常见的问题主要为数据库监控列表中连接状态不正确。添加数据库后，数据库正常启动，但数据库监控列表中连接状态一直失败。

连接状态获取流程：

在添加数据库后的一个收集频率（配置参数“gather_db_info_freq”）才开始收集该数据库的信息，并且收集到的信息需要发送到 DEM 存储到后台数据库，然后按界面状态刷新频率（配置参数“dem_page_refresh_freq”），从 DEM 后台数据库中查询出数据库的信息。

此时需要检查：

1. 最新检测时间

鼠标移动到对应数据库监控行的第一列“数据库”，查看检测时间，可以查看最新收到的数据库信息的时间。

2. 数据库基本信息收集频率

参数配置中的“gather_db_info_freq”。

如果超过“gather_db_info_freq”设置的收集频率，则手动点击数据库监控页面的“刷新”按钮。

如果超过“gather_db_info_freq”设置的收集频率并且无法刷新出最新的数据库状态，则需检查数据库所在机器与 DEM 是否连通，检查数据库所在机器的 dmagent 是否正常运行。

7.2 DEM 告警

DEM 告警可以使用户主动管理数据库，通过设置各资源的阈值，使在问题严重化之前能及时通知用户，改善用户服务，减少一些不必要的麻烦。

DEM 内置一些常用的告警类型，此外用户也可以自定义 SQL 语句进行告警，详见
[7.1.4.6.14 自定义 SQL 监控。](#)

DEM 可以对数据库资源（包括相关组件）和主机资源设置告警。

7.2.1 系统结构

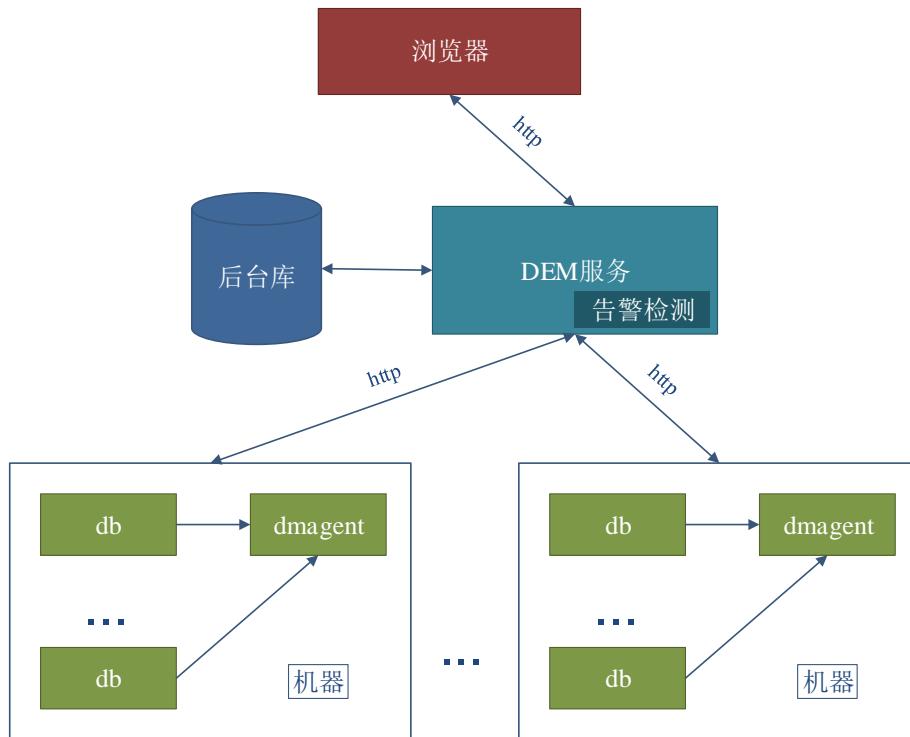


图 7.55 DEM 告警系统结构图

7.2.2 实现原理

告警实现有 2 种方式：

1. dmagent 收集信息发送给 DEM 时，DEM 会根据告警配置检测阈值；
2. DEM 启动专门的检测线程，定时检测。

其中第 2 种方式检测 2 项：

- 主机是否存活

主机存活判断依据为是否超过主机失效时间（配置参数 `mf_invalid_time`）没有收到主机信息（即当前时间 - 最新收到的主机信息时间 < 主机失效时间则认为存活）。

检测频率与数据库基本信息收集频率一致。

- 数据库是否存活

数据库存活判断依据为是否超过数据库失效时间（配置参数 `mf_invalid_time`）没有收到数据库信息（即当前时间-最新收到的数据库信息时间<数据库失效时间则认为存活）。

检测频率与数据库基本信息收集频率一致。

其他资源项都使用第 1 种检测方式，其流程如下图所示：

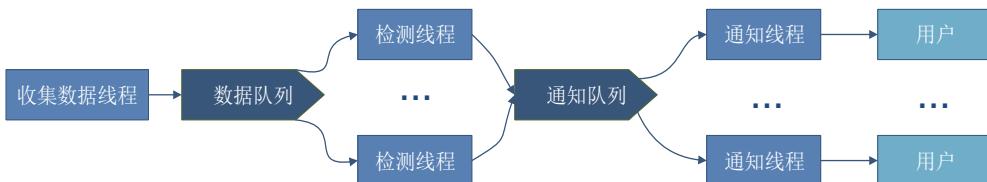


图 7.56 DEM 告警实现原理图

其中收集数据线程收集 dmagent 发送过来的数据，发送到数据队列，检测线程从数据队列中取出数据，根据告警配置，检测数据是否超过配置的阈值，如果超过则放到通知队列，通知线程从通知队列中取出通知，发送给配置的用户。

其中有几个地方可配置：

1. 告警检测线程数：默认 2，由配置参数 `alert_check_thread_count` 配置。
2. 告警通知线程数：默认 2，由配置参数 `alert_notify_thread_count` 配置。
3. 告警通知发送间隔：默认值 10 分钟，通知线程取出通知，并非每次都发送给用户，同一主机或数据库的同类告警通知在一定时间间隔内不会重复发送，该时间间隔由配置参数 `alert_his_re_notify_time` 配置。
4. 告警规则刷新频率：默认值 1 分钟，告警检测线程根据告警规则配置检测数据是否超过阈值，用户配置的告警规则存储在 DEM 后台库中，而告警检测线程要使用告警规则，会预先加载到内存中，为保证告警规则的时效性，会定时刷新内存中的告警规则。该值由配置参数 `alert_rule_reload_freq` 配置。

7.2.3 创建告警配置

创建告警可以选择智能运维->告警配置，在告警配置面板上点击功能按钮添加，打开告警配置对话框，根据提示完成每一页上的配置即可。向导框分为三页：

- 第一页：应用资源配置
- 第二页：巡检规则配置

- 第三页：通知用户配置

7.2.3.1 第一页：应用资源配置

如下图所示：

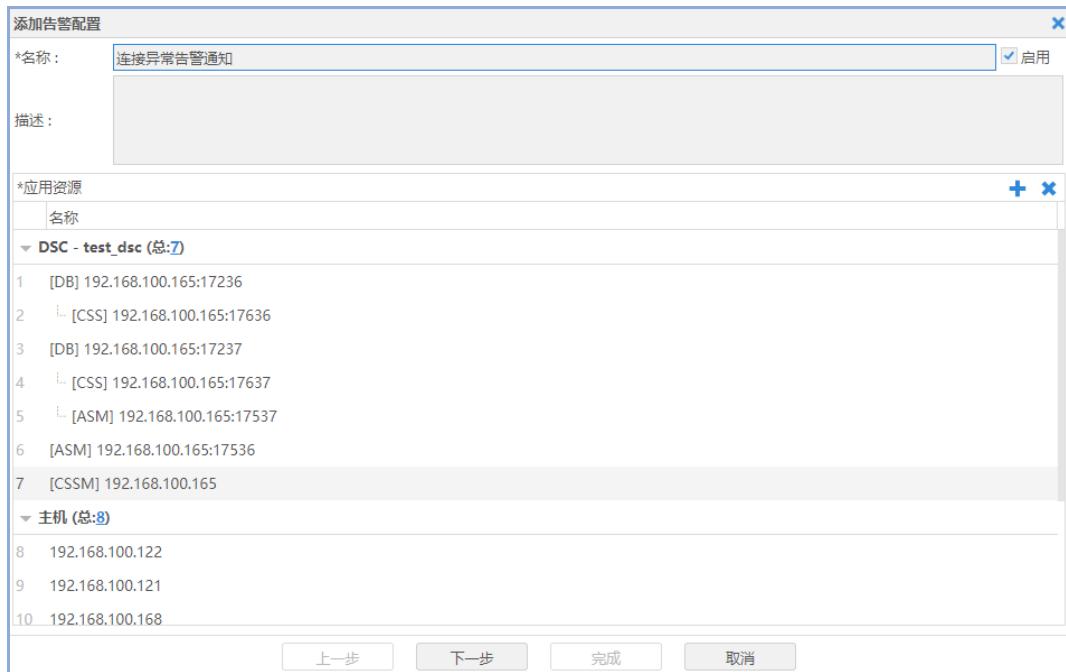


图 7.57 应用资源配置界面

给定告警名称，选择应用资源，即告警配置应用到哪些资源上。资源列表按所属集群分组显示，所有主机为一组，所有单机为一组，同时还保留了集群中站点的主备关系。

点击“下一步”。

7.2.3.2 第二页：告警规则配置

告警规则有数据库和主机两种类型，添加的告警规则会根据类型自动进行映射。例如：连接异常会映射到添加的所有数据库资源上；主机连接异常会映射到所有主机资源上。

如下图所示：

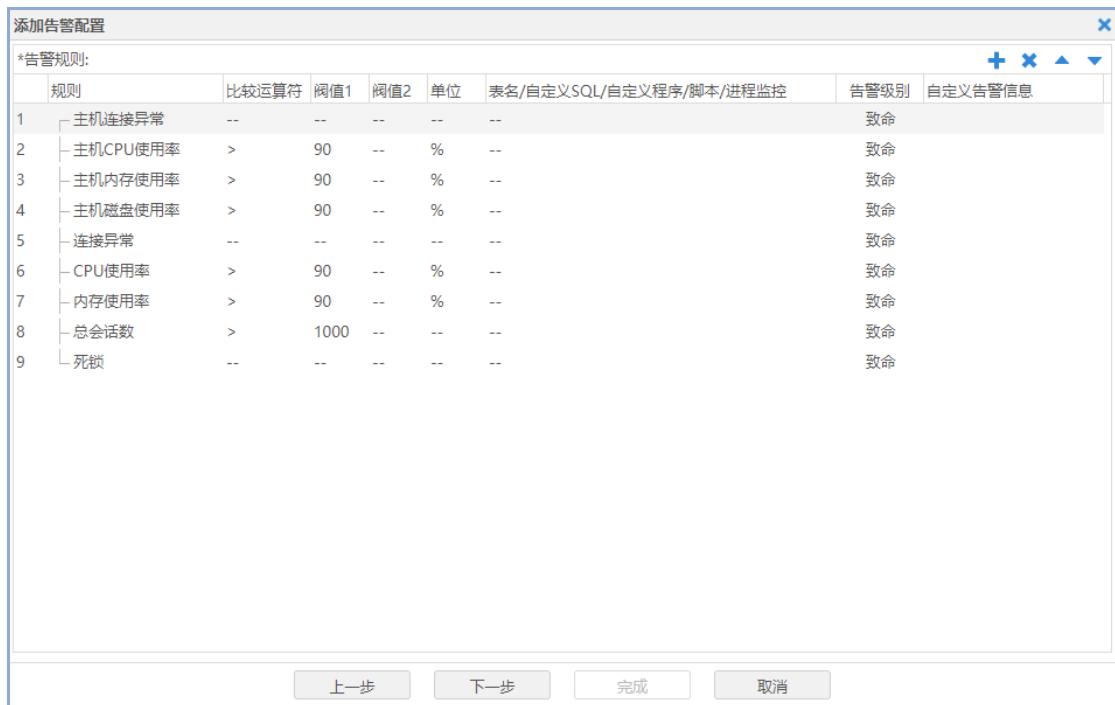


图 7.58 告警规则配置界面

其中，“--”符号表示当前配置项不可编辑，若没有“--”符号，则用户可双击编辑当前配置项。告警规则的默认告警级别为“致命”，用户可双击告警规则对应的告警级别进行更改，可选择的告警级别包括“致命”、“严重”、“一般”、“轻微”。

通过右上角的添加按钮，添加告警规则。添加对话框如下所示：

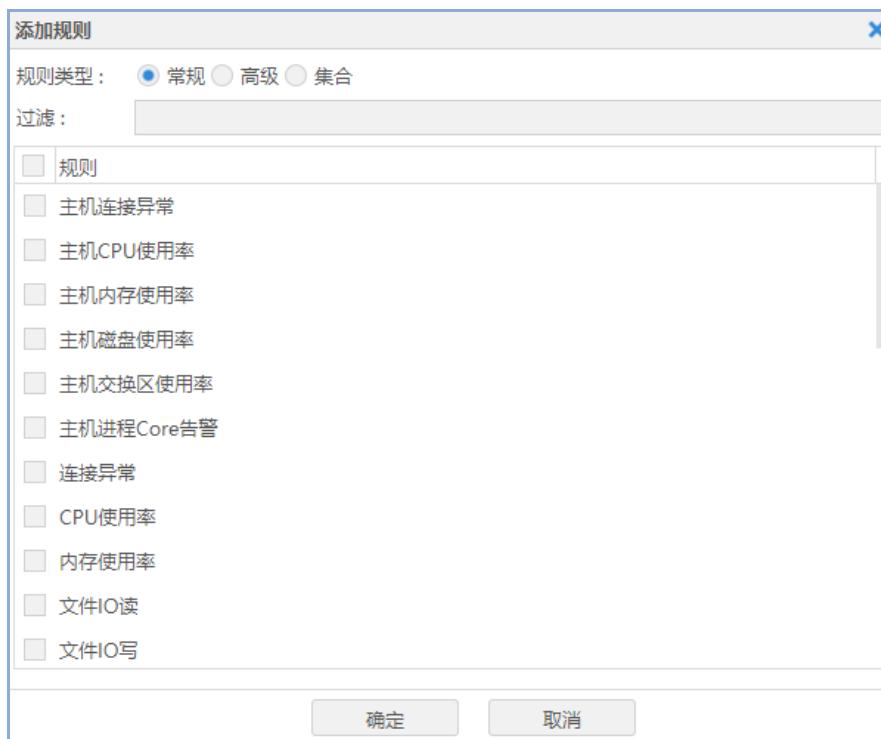


图 7.59 添加告警规则对话框

用户可根据自身需要选择添加常规规则、高级规则以及集合规则。

选择“确定”后，会在告警规则列表中添加相应行。

点击“下一步”。

7.2.3.3 第三页：通知用户配置

如下图所示：

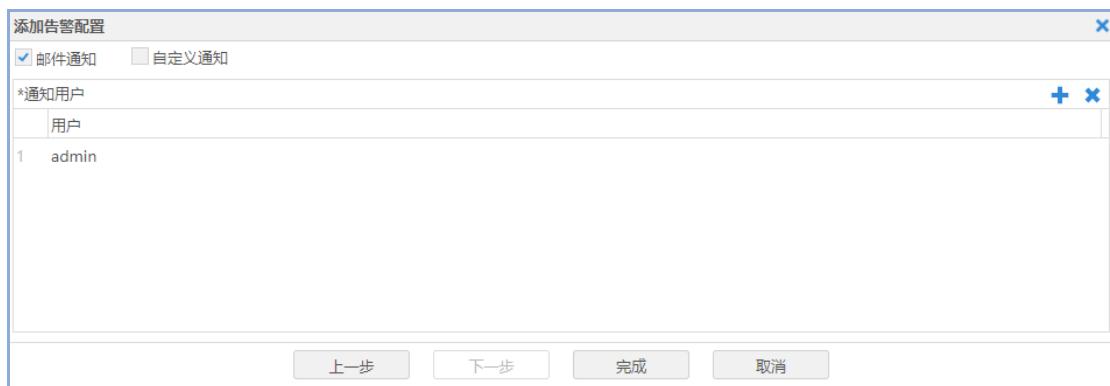


图 7.60 通知用户配置

指定通知给哪些用户，以及设置以什么方式通知给用户（邮件或自定义通知）。

点击“完成”，完成告警的配置。

7.2.4 查看告警状态和历史

告警信息记录会一直保留，用户可以通过告警历史对话框查看，入口在主机监控列表和数据库监控列表的告警字段，点击告警字段即可打开告警历史对话框查看对应监控资源的告警信息。如下图所示：

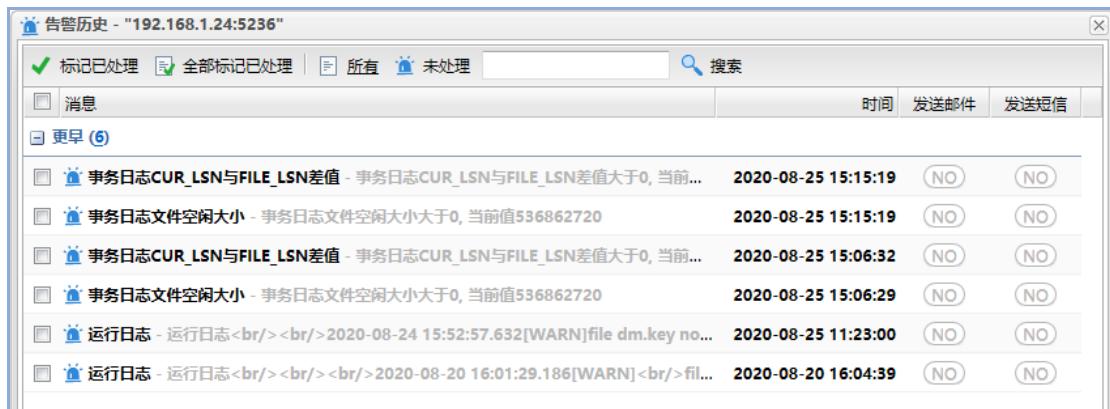


图 7.61 查看告警状态和历史

告警信息按时间先后降序排列，并按日期分组显示。

组标题后面统计了记录数，点击记录数可以快速选中该组的所有记录，如下图所示：

| 告警历史 - "192.168.1.24:5236" | | | | | |
|-------------------------------------|---|----------------------------------|---------------------|------|------|
| 标记已处理 | | 全部标记已处理 | | 所有 | 未处理 |
| 消息 | | | 时间 | 发送邮件 | 发送短信 |
| 更早 (6) | | | | | |
| <input checked="" type="checkbox"/> | 事务日志CUR_LSN与FILE_LSN差值 | 事务日志CUR_LSN与FILE_LSN差值大于0, 当前... | 2020-08-25 15:15:19 | (NO) | (NO) |
| <input checked="" type="checkbox"/> | 事务日志文件空闲大小 | 事务日志文件空闲大小大于0, 当前值536862720 | 2020-08-25 15:15:19 | (NO) | (NO) |
| <input checked="" type="checkbox"/> | 事务日志CUR_LSN与FILE_LSN差值 | 事务日志CUR_LSN与FILE_LSN差值大于0, 当前... | 2020-08-25 15:06:32 | (NO) | (NO) |
| <input checked="" type="checkbox"/> | 事务日志文件空闲大小 | 事务日志文件空闲大小大于0, 当前值536862720 | 2020-08-25 15:06:29 | (NO) | (NO) |
| <input checked="" type="checkbox"/> | 运行日志 - 运行日志 2020-08-24 15:52:57.632[WARN]file dm.key no... | 2020-08-25 11:23:00 | (NO) | (NO) | |
| <input checked="" type="checkbox"/> | 运行日志 - 运行日志 2020-08-20 16:01:29.186[WARN] fil... | 2020-08-20 16:04:39 | (NO) | (NO) | |

图 7.62 告警信息记录

7.2.5 告警通知

告警信息除了指定的邮件或手机通知之外，还会通知到用户的通知中心，用户登录 DEM 后，可以点击主页顶部的通知按钮打开通知面板，查看分发给自己处理的告警通知信息。

如下图所示：

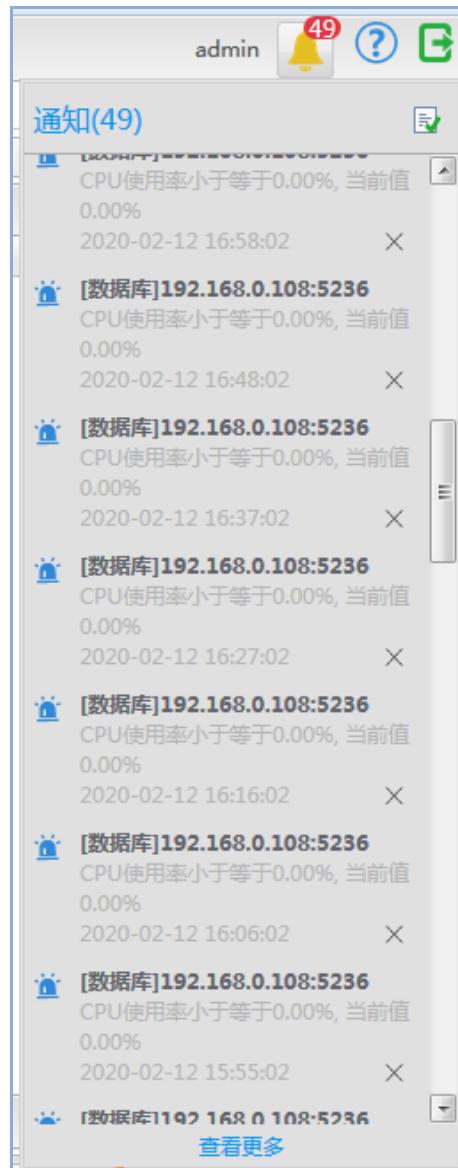


图 7.63 告警通知信息

此处显示未读信息，若要查看所有通知信息，点击“查看更多”，可以查看所有的通知信息，如下图所示：

| 目标 | 消息 | 时间 | 发送邮件 | 发送短信 |
|-----------------------------|---------------------------|---------------------|------|------|
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 16:27:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 16:16:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 16:06:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 15:55:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 15:45:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 15:35:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | 连接异常 | 2020-02-12 15:35:00 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 14:54:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 14:49:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 14:33:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | 连接异常 | 2020-02-12 14:32:15 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 14:19:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | 连接异常 | 2020-02-12 14:17:54 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 14:07:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | 连接异常 | 2020-02-12 13:49:32 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 13:25:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 13:15:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 13:05:02 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 12:50:41 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 12:43:54 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | 连接异常 | 2020-02-12 12:43:53 | NO | NO |
| [告警][数据库]192.168.0.108:5236 | CPU使用率小于等于0.00%, 当前值0.00% | 2020-02-12 11:23:59 | NO | NO |

图 7.64 更多警报通知信息

通知信息按时间先后降序排列，并按日期分组显示。

7.2.6 停止告警

监控及告警->告警配置，在告警配置页面的告警列表中，点击“启用”栏的启用/禁用按钮，可以切换告警的启用/禁用状态。

7.3 DEM 巡检

DEM 巡检可以在用户完成巡检配置后，周期性地生成巡检报告以展示数据库运行情况。巡检报告的内容包括但不限于：数据库健康度，主机概况，数据库概况，性能概况，表空间，Redo 日志，归档日志，备份，用户与会话，运行日志，DM.INI 配置等。此外，巡检报告还可以进行数据分析与异常检查。

7.3.1 系统结构

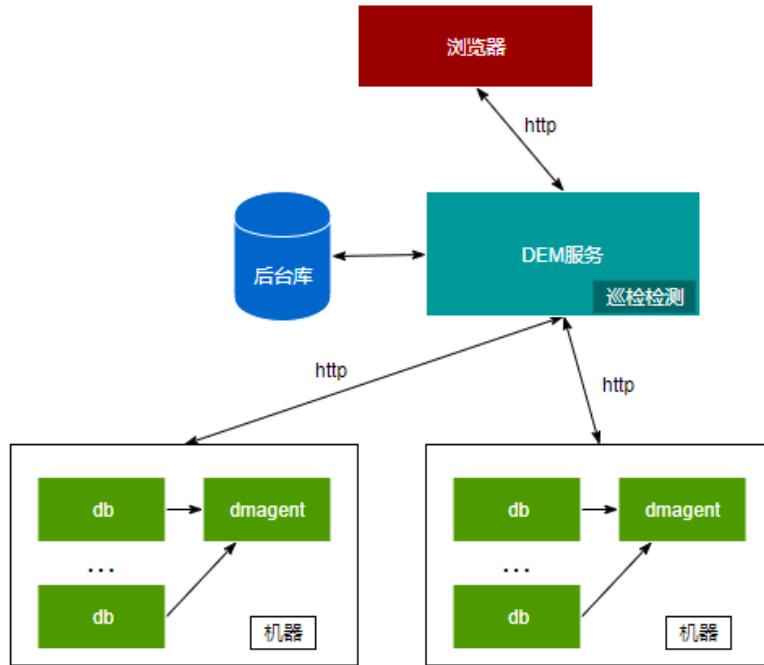


图 7.65 DEM 巡检系统结构图

7.3.2 实现原理

巡检的数据来源是代理 dmagent 最近一次从目标数据库收集的主机与数据库信息，因此巡检内容不是强实时性的，而是依赖 dmagent 收集频率的准实时。

巡检实现主要依赖巡检检测线程，每隔一定的时间(5s)，巡检检测线程从 DEM 后台库的巡检配置表中获取所有配置项，并判断是否满足条件（当前时间>起始时间 并且 当前时间>上次巡检时间+间隔），若满足则在项目根目录下的 download 目录生成巡检报告，并通过通知线程以邮件附件的形式通知用户，流程图如下所示：



图 7.66 DEM 告警实现原理图

7.3.3 创建巡检配置

创建巡检配置可以选择监控及告警->巡检配置，在巡检配置面板上点击功能按钮添加，打开巡检配置对话框，根据提示完成每一页上的配置即可。向导框分为三页：

- 第一页：应用资源配置
- 第二页：告警规则配置
- 第三页：通知用户配置

7.3.3.1 第一页：应用资源配置

如下图所示：

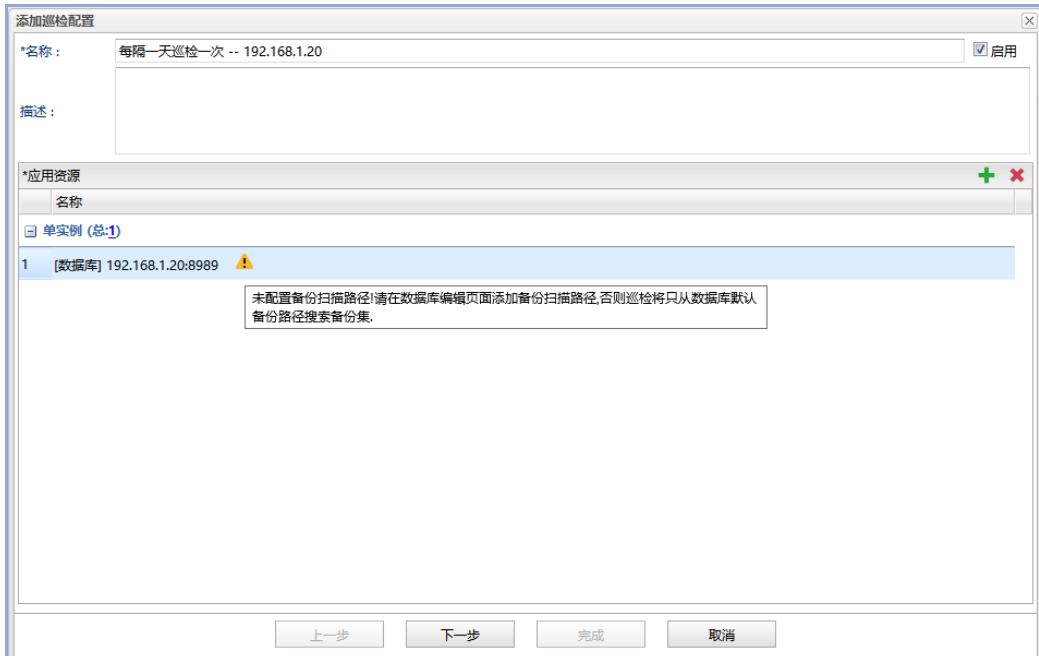


图 7.67 应用资源配置界面

给定巡检配置名称，选择应用资源，即巡检配置应用到哪些资源上。资源列表按所属集群分组显示，所有主机为一组，所有单机为一组，同时还保留了集群中站点的主备关系。如果某一资源后方出现了黄色感叹号警告，原因是未配置备份搜索目录，配置方法请参考

7.1.4.2 添加数据库监控。

点击“下一步”。

7.3.3.2 第二页：巡检规则配置

如下图所示：

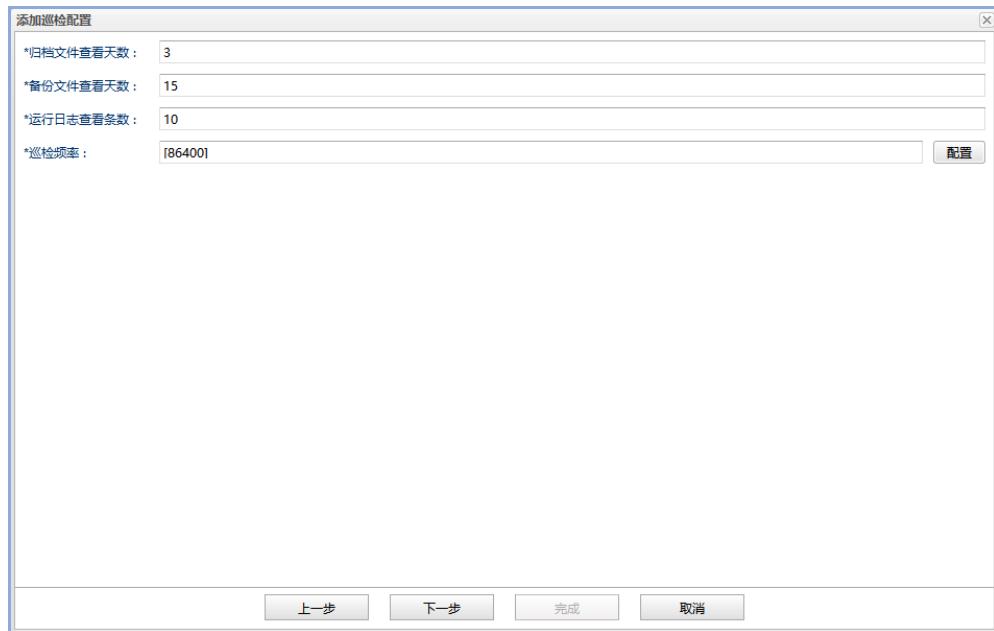


图 7.68 巡检规则配置界面

巡检规则配置项有：归档文件查看天数，默认最近 3 天；备份文件查看天数，默认最近 15 天；运行日志查看条数，默认最近 10 条；巡检频率，默认 86400 秒（一天）。

点击“下一步”。

7.3.3.3 第三页：通知用户配置

如下图所示：

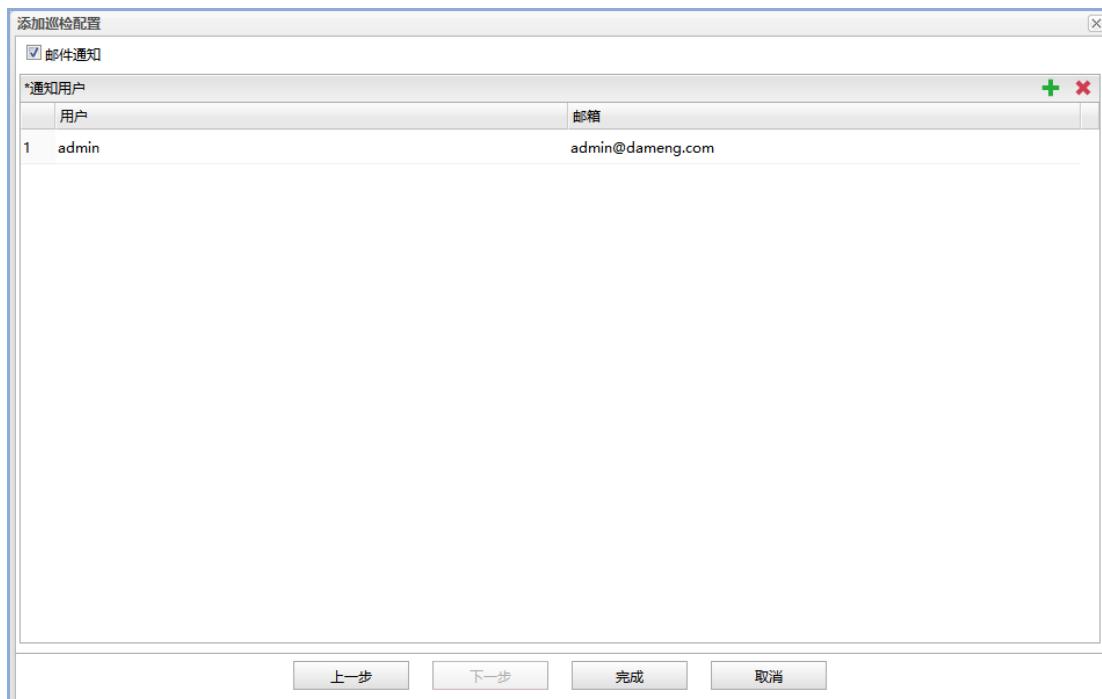


图 7.69 通知用户配置界面

指定用邮件通知的方式通知给哪些用户。

点击“完成”，完成巡检的配置。

7.3.4 查看巡检历史和巡检报告

共有两个查看巡检历史和巡检报告的入口：

(1) 巡检配置列表-操作-巡检历史，可以查看该配置下生成的所有巡检报告

| 巡检配置 | | | | | | |
|--------------------------|----|----|-------|---------------------|------|---------------------------------------|
| 名称 | 启用 | 描述 | 创建者 | 创建时间 | 邮件通知 | 操作 |
| 每隔一天巡检一次 -- 192.168.1.20 | 启用 | | admin | 2021-04-13 10:42:35 | ON | 巡检历史 |
| 每分钟一次 | 禁用 | 发 | admin | 2021-04-01 14:09:48 | | 编辑 删除 |

图 7.70 查看巡检历史

| 巡检报告 - "每隔一天巡检一次 -- 192.168.1.20" | |
|-----------------------------------|---------------------|
| 操作 | 巡检报告 |
| | 选择一行数据查看它的详细信息 |
| 1 | 2021-04-16 10:55:13 |
| 2 | 2021-04-15 10:52:44 |
| 3 | 2021-04-14 10:50:13 |

图 7.71 查看巡检历史

(2) 数据库列表-操作-巡检报告，可以查看所有应用了该数据库的巡检配置生成的所有巡检报告



图 7.72 查看巡检报告

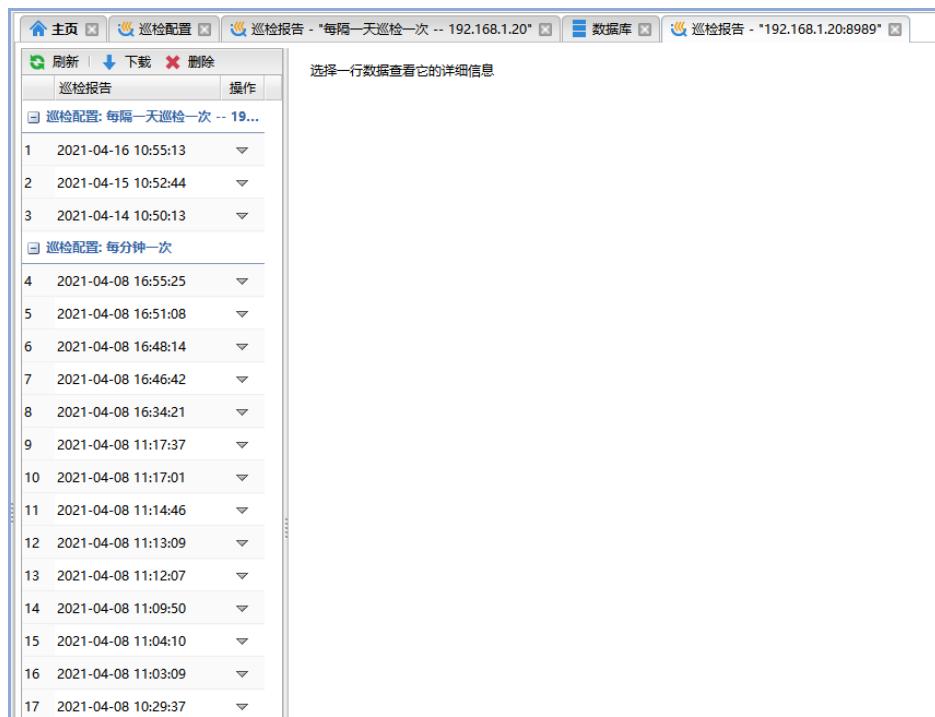


图 7.73 查看巡检报告

7.3.5 停止巡检

监控及告警-巡检配置，在巡检配置列表中，点击“启用”栏的启用/禁用按钮，可以切换告警的启用/禁用状态。

7.4 DEM 备份

DEM 备份可以定时周期性地对指定应用资源（数据库）进行完全备份或增量备份。用户可以根据自己的需要添加备份调度，指明具体的备份规则。备份调度可以灵活启停，备份异常还能及时通知到用户。

7.4.1 系统结构

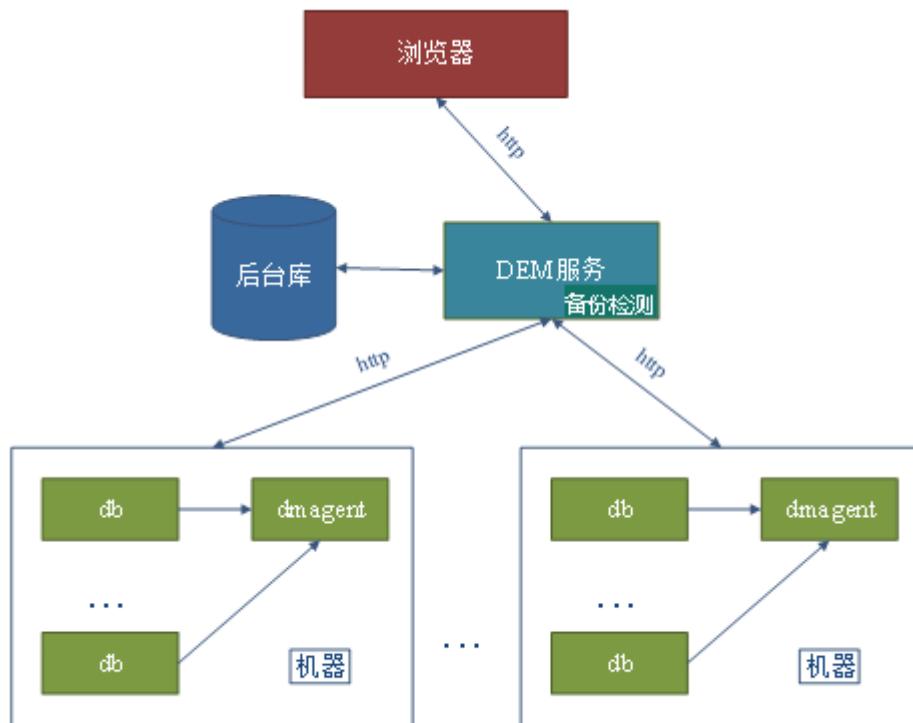


图 7.74 DEM 备份系统结构图

7.4.2 实现原理

如上图 7.79 所示：

1. 用户通过浏览器登录 DEM 服务配置备份调度。
2. DEM 服务将用户配置的备份调度存储于后台数据库中。
3. DEM 服务的备份工作线程一直不停的检测后台数据库中存储的所有备份调度，查找当前时间点需要执行的备份调度。

4. DEM 服务将需要执行的备份调度做任务拆分，分别下发到各个应用资源对应的 dmagent 上。

5. dmagent 收到任务后对关联的应用资源进行备份操作，并将备份结果反馈给 DEM 服务。

6. DEM 服务整理 dmagent 反馈的结果，记录到后台数据库中。

7. 之后用户通过浏览器连 DEM 服务（DEM 服务会从后台数据库拉取备份历史信息），即可查看到备份历史信息。

7.4.3 创建备份调度

创建备份调度可以选择智能运维->备份调度，在备份调度面板上点击功能按钮添加，打开备份配置对话框，根据提示完成每一页上的配置即可。向导框分为三页：

- 第一页：应用资源配置
- 第二页：备份规则配置
- 第三页：通知用户配置

7.4.3.1 第一页：应用资源配置

如下图所示：

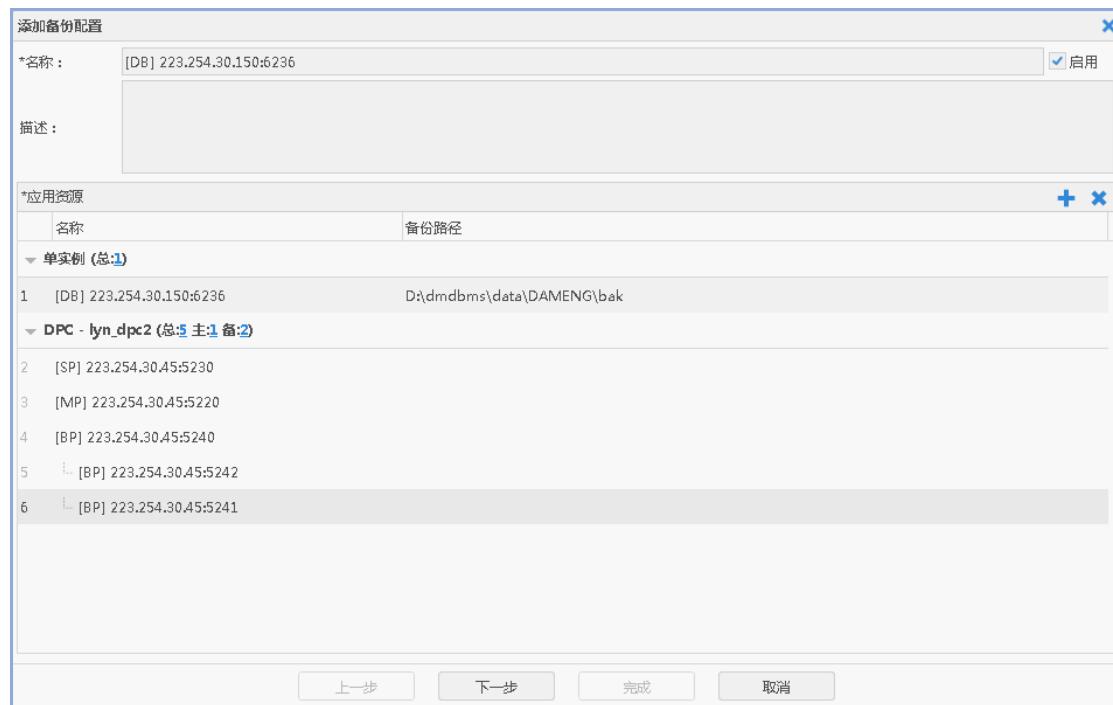


图 7.75 应用资源配置界面

给定备份名称，选择应用资源，即备份配置应用到哪些资源上，给每个应用资源设置备份路径。资源列表按所属集群分组显示，所有单机为一组，同时还保留了集群中站点的主备关系。

点击“下一步”。

7.4.3.2 第二页：备份规则配置

备份规则配置内容如下图所示：



图 7.76 备份规则配置界面

根据实际需求合理选择备份配置。其中，每次增量备份都是基于最近一次的完全备份进行的，所以使用增量备份时必须包含完全备份。

点击“下一步”。

7.4.3.3 第三页：通知用户配置

如下图所示：

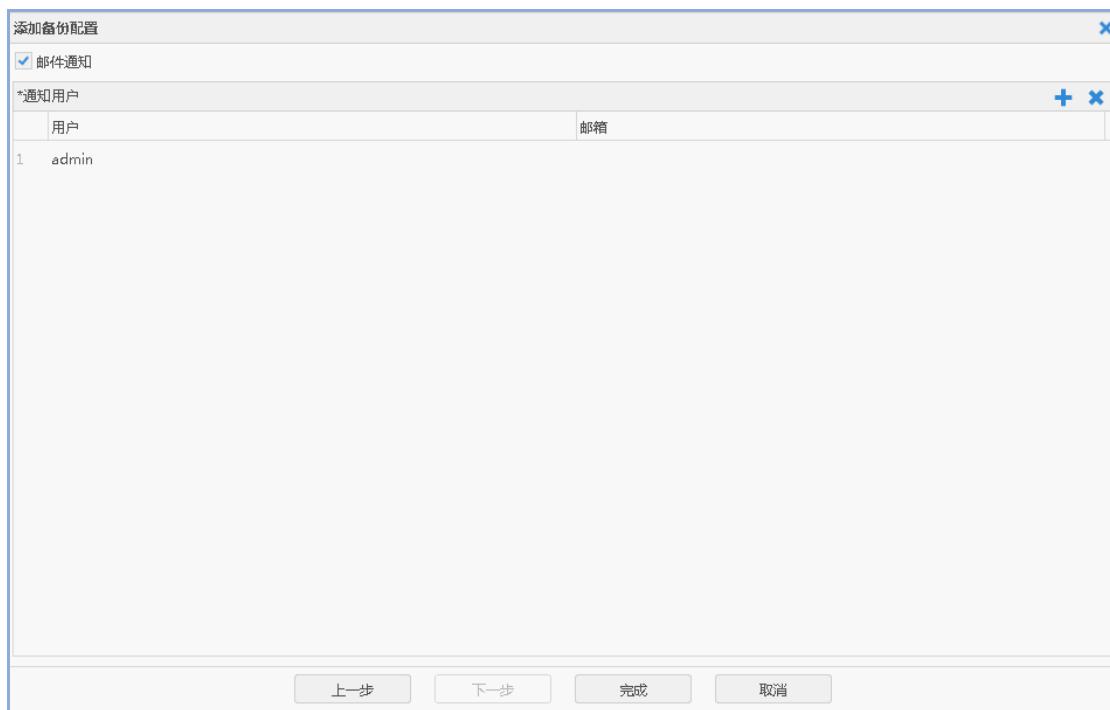


图 7.77 通知用户配置界面

指定用邮件通知的方式通知给哪些用户。当备份任务每次完成备份或备份失败都会通知到用户。

点击“完成”，完成备份的配置。

7.4.4 查看备份状态和历史

智能运维->备份调度，如下图，在备份调度列表中，选中查看目标，点击操作->备份历史，查看具体的备份历史信息

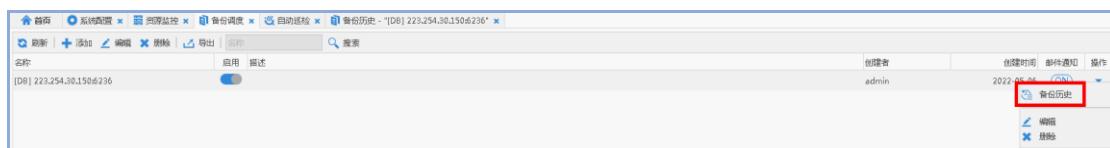


图 7.78 备份调度界面

如下图所示，在备份历史中可以查看当前备份调度中各应用资源的备份情况。备份历史按照应用资源分组显示，还可以指定时间范围、资源名称等进行过滤。

| 备份名 | 状态 | 开始时间 | 耗时 | 存储位置 |
|--|----|---------------------|-------|------------------------|
| BAK_DB_223_254_30_150_6236_2022_05_07_10_10_30 | 失败 | 2022-05-07 10:10:30 | 13毫秒 | D:\dmdbms\data\DEM_bak |
| BAK_DB_223_254_30_150_6236_2022_05_07_10_00_30 | 成功 | 2022-05-07 10:00:30 | 3.16秒 | D:\dmdbms\data\DEM_bak |

图 7.79 备份历史界面

7.4.5 停止备份

智能运维->备份调度，在备份调度列表中，点击“启用”栏的启用/禁用按钮，可以切换对应备份调度的启用/禁用状态。

8 代理 dmagent

达梦数据库代理（简称 dmagent）是运行在主机上的代理。通过代理，DEM 可以访问该主机，并监控该主机的相关信息。同时，代理还提供了一套辅助工具，比如 SQL 查询工具、TPCC 测试工具等。

8.1 安装部署

8.1.1 安装说明

- 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。
- dmagent 跨平台。
- 运行 dmagent 需要用户设置 JRE_HOME 或 JAVA_HOME 环境变量，dmagent 不自带 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。

8.1.2 安装步骤

参见 [3.2 dmagent 部署](#)。

8.2 目录结构

dmagent 安装后的目录结构如下图所示：

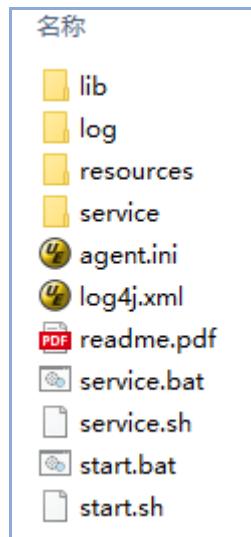


图 8.1 dmagent 安装后的目录结构图

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。

resources 目录: 存放资源文件或复杂的配置文件, 比如 tpcc 的 SQL 初始化脚本、冒烟测试的 xml 配置文件。

service 目录: 存放服务相关的执行文件。

log4j.xml: 日志配置文件。

readme.pdf: dmagent 使用说明文档。

agent.ini: dmagent 配置文件。

8.3 配置文件说明

配置文件在安装目录下 **agent.ini**, 包含参数如下:

表 8.1 agent.ini 参数说明

| 参数名 | 默认值 | 描述 |
|-------------------------|--------------------------------------|--|
| #general (常规配置项) | | |
| gen_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://192.1 68.2.10:8080 /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 |
| collect_command_data | false | 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 |
| #ap(外部函数功能) | | |
| 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 |
| #isql(SQL 命令行工具) | | |
| 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 |
| #build[达梦公司内部代码编译模块] | | |
| build_config_path | build.xml | 编译的配置文件路径 |
| #service(远程调用服务) | | |
| 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 |
| #gather(采集数据) | | |
| 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_age_nt | 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 |
| gather_offline_send_freq | 300 | 通信失败时数据再次发送频率 |
| 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_disk_freq | [60] | 收集主机磁盘信息的频率(秒) |
| gather_mf_exec_freq | [60] | 收集主机自定义可执行程序的频率(秒) |
| gather_mf_exec_timeout | 10 | 主机可执行程序超时时间(秒) |
| gather_mf_process_freq | [60] | 收集主机进程信息的频率(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_sqldslow_freq | [60] | 收集数据库慢 SQL 信息的频率(秒) |
| gather_db_sqldslow_time | 1000 | 监控中对于慢 SQL 的界定时间(毫秒) |
| gather_db_sqlcount_freq | [60] | 收集数据库高频 SQL 信息的频率(秒) |
| gather_db_sqLError_freq | [60] | 收集数据库执行 SQL 出错信息的频率(秒) |
| gather_db_event_freq | [60] | 收集数据库事件信息的频率(秒) |
| gather_db_session_freq | [60] | 收集数据库会话信息的频率(秒) |
| gather_db_tablespace_freq | [60] | 收集数据库表空间信息的频率(秒) |
| gather_db_user_freq | [60] | 收集数据库用户锁定信息的频率(秒) |
| gather_db_table_freq | [60] | 收集数据库表数据信息的频率(秒) |
| gather_db_udsql_freq | [60] | 收集数据库自定义 SQL 信息的频率(秒) |
| gather_db_udsql_timeout | 10 | 数据库自定义 SQL 执行超时时间(秒) |
| gather_db_audit_record_freq | [0] | 收集数据库审计记录的频率(秒) |

| | | |
|-----------------------------|---|--|
| gather_db_log_history_freq | [60] | 收集数据库运行日志记录的频率（秒） |
| gather_db_rlog_history_freq | [60] | 收集数据库事务日志记录的频率（秒） |
| gather_db_backup_freq | [60] | 收集数据库备份信息的频率（秒） |
| #upgrade (升级) | | |
| upgrade_enable | true | 是否启用 dmagent 自动升级功能。true: 启用; false: 关闭 |
| upgrade_center_servlet | /dem/dma_age_nt | 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 |
| #smoke [达梦公司内部冒烟测试模块] | | |
| 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 |
| #tpcc | | |
| 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 | 数据库密码 |
| tpcc_warehouses | 100 | TPCC 仓库数量 |

| | | |
|-------------------------------|-------|--|
| 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_terminal | 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_enable | 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_ssshaddr | | 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 |

8.4 功能介绍

dmagent 提供的功能有以下几个:

- Java 外部函数的代理功能

- 收集 DEM 监控数据功能
- 远程调用服务功能
- SQL 命令行工具
- TPCC 测试工具
- 编译打包工具（该功能仅供达梦公司内部使用）
- 冒烟测试代理功能（该功能仅供达梦公司内部使用）

8.4.1 Java 外部函数的代理功能

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。

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（ap_enable 详细参数说明参见 [8.3 配置文件说明](#) 章节）。

详细请参见《DM8_SQL 语言使用手册》第 10 章 10.2 java 外部函数。



启用 java 外部函数代理功能时，ap_port 必须与 DM.INI 中的

注意： EXTERNAL_JFUN_PORT 一致。

8.4.2 收集 DEM 监控数据功能

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

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，发送处理详见 [7.1.2 实现原理](#)。

可通过 gather_enable 参数启用或关闭该功能（gather_enable 详细参数说明参见 [8.3 配置文件说明](#) 章节）。

每个 gather 线程收集信息频率由配置参数控制，具体参见[第 4 章配置参数说明](#)中的

dmagent 监控频率配置。

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。

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!
```

8.4.3 远程调用服务功能

启用该功能需配置：

- service_enable =true
- service_port=任一没被占用的端口

该功能主要辅助内部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 service_enable 参数启用或关闭该功能（service_enable 详细参数说明参见 [8.3 配置文件说明](#) 章节）。

8.4.4 SQL 命令行工具

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:\dm8client_new\develop\tool\agent\build\agent>start.bat isql agent.ini
Agent-V5.0.1.5845-2021.03.04-DEV starting...
[SQL]> help
//          - last sql execute again
login      - login database, alias: "conn"
logout     - logout database
edit       - edit sql, alias: "ed"
exit       - exit this program
help       - show this usage message
[SQL]> /
```

图 8.2 agent isql 帮助信息

8.4.5 TPCC 测试工具

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
```

具体使用命令，执行 start.bat/start.sh tpcc help agent.ini，查看帮助信息。具体如下：

```
E:\dm8client_new\develop\tool\agent\build\agent>start.bat tpcc help agent.ini
Agent-V5.0.1.5845-2021.03.04-DEV

usage as: start [command] agent.ini
command
  -"smoke"      - run smoke test
  -"isql"        - start sql command
  -"tpcc"        - tpcc, run steps configed in agent.ini
  -"tpcc_table" - tpcc, run create table
  -"tpcc_load"  - tpcc, run load data
  -"tpcc_index" - tpcc, run create table index
  -"tpcc_run"   - tpcc, run test
```

图 8.3 agent tpcc 帮助信息

8.4.6 build 工具

代码编译模块，专门提供给达梦公司内部人员使用。

8.4.7 smoke 工具

冒烟测试模块，专门提供给达梦公司内部人员使用。

8.5 dmagent 升级

8.5.1 需要升级的情景

当 dmagent 满足下述情景中的任意一条，就表示 dmagent 需要升级了。

情景一，当前 dmagent 的版本号小于 DEM 中心(center_url 指定的 DEM)的 dmagent 的版本号；

情景二，对应 DEM 运行模式为 TEST，且当前 dmagent 的源码 svn 号小于 DEM 上的 dmagent 的源码 SVN 号。

8.5.2 升级准备

dmagent 自动完成升级，启用该功能需配置：

- upgrade_enable =true
- center.url=http://<dem_ip>:<dem_port>/dem
- upgrade_center_servlet=dem/dma_agent
- upgrade_version_check_freq=600

8.5.3 升级时机

dmagent 自动升级时机包括：

1. dmagent 启动时

dmagent 在启动时，如果 upgrade_enable=true 会根据 center_url 和 upgrade_center_servlet 配置执行升级。

2. 按 upgrade_version_check_freq 频率检查升级

如果 upgrade_enable=true 且 upgrade_version_check_freq>0 时，会按 upgrade_version_check_freq 频率根据 center_url 和 upgrade_center_servlet 配置执行升级。

8.5.4 升级步骤

升级操作步骤为：

1. 检查是否需要升级；

2. 下载新版 dmagent 升级安装包；

3. 准备升级 tmp 临时目录，解压下载好的 dmagent 升级包；

4. 创建升级脚本；

5. 保存旧配置；

6. 停止旧的 dmagent；

7. 执行升级脚本，具体的执行升级的操作包括：

7.1 备份旧 dmagent 到 tmp 目录（旧的 dmagent 文件保留在 tmp 目录，可以在 tmp 目录中查看，但日志文件太大则不会保留（如果日志文件超过 10m，则备份，否则

全部删除)) ;

7.2 删除 DMAGENT_HOME 下的 dmagent;

7.3 新的 dmagent 拷贝到 DMAGENT_HOME;

7.4 合并旧的 dmagent 和新的 dmagent 配置信息作为新的配置信息;

7.5 启动新的 dmagent (新运行 dmagent 除 agent.ini 会保留旧的配置信息, 其他都使用新的)。

9 接口

DEM 为用户提供多种访问 DEM 资源的接口。接口分为两大类：一类是基于 HTTP 协议开发的接口；另一类是基于 UDP 协议开发的接口。

9.1 基于 HTTP 协议开发的接口

DEM 为用户提供多种访问 DEM 资源的 HTTP 接口。按照 HTTP 接口的功能，将 HTTP 接口细分为 4 种类型：

1. 管理 DEM 的用户的接口
2. 管理数据库的连接信息（客户端工具->对象管理中注册的连接）的接口
3. 管理告警信息的接口
4. 管理 prometheus 监控数据的接口

HTTP 接口是基于 HTTP 协议开发的接口，给用户提供向指定的 URL 请求资源/提交数据的一种方式。即用户通过 URL 指定参数，然后获取返回的消息。DEM 提供的接口根据资源类型不同，访问的 URL 和返回的消息格式不同，具体如下：

1. prometheus 监控数据

访问 URL 为：

```
http://<dem_ip>:<dem_port>/dem/metrics
```

消息返回格式符合 prometheus 的监控数据格式，具体请查阅 prometheus 相关文档。

2. 其他资源（目前支持的 DEM 的用户，DEM 管理的数据库的连接信息，告警信息）

访问 URL 为：

```
http://<dem_ip>:<dem_port>/dem/api
```

通过请求参数 api_type 来指定资源类型，api_type 值可以为：

- 1) USER: DEM 的用户。
- 2) SERVER_CONN: DEM 管理的数据库的连接信息。
- 3) ALERT: DEM 的告警信息。

响应消息返回为 json 的数据格式，具体格式如下：

表 9.1 响应消息格式

| 名称 | 说明 |
|------|----------------|
| code | 返回码（0：成功，1：失败） |
| msg | 返回信息 |
| data | 返回数据 |

使用接口访问 **DEM** 资源时需要 **DEM** 登录用户的密码，密码需要加密后传入，**DEM** 用户密码加解密接口：

`com.dameng дем.server.util.IEncryptEngine`



注意：默认加解密实现类为：

`com.dameng дем.server.util.DefaultEncryptEngine`

也可以自定义实现，具体参见第 13 章的 13.1 `IEncryptEngine`

本章主要介绍每种资源的具体访问方式和使用示例。

9.1.1 管理 **DEM** 用户的接口

使用接口安全的连接到 **DEM** 服务器，可以查询并管理 **DEM** 的用户信息。



注意：由于相关的接口访问需要 **DEM** 登录用户密码信息，因此 **DEM** 需要部署在 **SSL** 环境，才能保证相关信息的安全性，关于 **DEM** 部署在 **SSL** 环境详见

第 1 章 **DEM** 部署相关介绍。

获取用户信息访问的 URL：

```
http://<dem_ip>:<dem_port>/dem/api
```

通过请求参数 `api_type=USER` 来指定资源类型为告警信息。

通过接口可以对 **DEM** 用户进行创建、删除、查询和修改的操作，操作类型通过传递 `action_type` 参数指定，`action_type` 的值具体定义如下：

表 9.2 `action_type` 参数说明

| 值 | 说明 |
|---|----|
| 1 | 创建 |
| 2 | 删除 |
| 3 | 查询 |
| 4 | 修改 |

不同的操作传入参数和返回消息不一样，具体如下介绍。

9.1.2.1 创建

请求消息参数说明如下：

表 9.3 创建用户参数

| 名称 | 说明 |
|-------------|--------------------------------|
| api_type | USER (指定获取用户信息) |
| action_type | 1 (创建) |
| dem_user | dem 的用户名 |
| dem_pwd | dem 的用户密码 |
| login_name | 新创建用户的用户名 |
| login_pwd | 新创建用户的密码 |
| role | 用户的角色类型 (1:普通用户, 2:高级用户 3:管理员) |
| enable | 是否可用 (true, false) |
| real_name | 用户真实姓名 |
| email | 可选, 邮箱 |
| phone | 可选, 电话 |
| sex | 可选, 性别 (男, 女) |

通常创建用户响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code 表示创建成功与否, msg 包括具体的信息 (成功提示或错误信息)。



DEM 管理员用户和超级管理员用户才有创建用户的权限。

注意:

9.1.2.2 删除

请求消息参数说明如下:

表 9.4 删除用户参数

| 名称 | 说明 |
|-------------|-----------------|
| api_type | USER (指定获取用户信息) |
| action_type | 2 (删除) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 删除用户的 ID |

通常删除用户响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code

表示删除成功与否，msg 包括具体的信息（成功提示或错误信息）。



注意：

1. DEM 管理员用户和超级管理员用户才拥有删除用户的权限。

2. DEM 用户无法删除自己。

3. DEM 管理员用户无权限删除管理员和超级管理员用户。

9.1.2.3 查询

请求消息参数说明如下：

表 9.5 查询用户参数

| 名称 | 说明 |
|-------------|----------------------------------|
| api_type | USER (指定获取用户信息) |
| action_type | 3 (查询) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 可选，不设置则查询所有用户信息，设置则查询对应 id 的用户信息 |

通常查询用户响应消息包括 code, msg 和 data 信息(详见表 9.1 响应消息格式说明)，code 表示查询操作成功与否，msg 包括具体的信息（成功提示或错误信息），data 返回具体的用户信息，返回用户信息 data 字段参数说明如下：

表 9.6 返回用户信息 data 字段数据格式

| 名称 | 说明 |
|---------|-------------------|
| count | 用户总数 |
| records | 用户对象数组 (Record[]) |

表 9.7 返回用户信息 Record 数据格式

| 名称 | 说明 |
|------------|--------------------------------|
| Id | 用户 ID |
| login_name | 用户的用户名 |
| role | 用户的角色类型 (1:普通用户, 2:高级用户 3:管理员) |
| enable | 是否可用 (true, false) |
| real_name | 用户真实姓名 |
| email | 邮箱 |
| phone | 电话 |
| sex | 性别 (男, 女) |

例如：

```
{
```

```

id: 1576124021889
login_name: "t2"
real_name: "1"
email: ""
phone: ""
sex: "男"
enable: true
role: 3
}

```



DEM 管理员用户和超级管理员用户才拥有获取用户信息的权限。

注意：

9.1.2.4 修改

请求消息参数说明如下：

表 9.8 修改用户参数

| 名称 | 说明 |
|---------------|--------------------------------|
| api_type | USER (指定获取用户信息) |
| action_type | 4 (修改) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 待修改用户的 ID |
| login_name | 待修改用户的用户名 (用户名不能修改) |
| old_login_pwd | 待修改用户的原密码 |
| new_login_pwd | 待修改用户的新密码 |
| role | 用户的角色类型 (1:普通用户, 2:高级用户 3:管理员) |
| enable | 是否可用 (true, false) |
| real_name | 用户真实姓名 |
| email | 可选, 邮箱 |
| phone | 可选, 电话 |
| sex | 可选, 性别 (男, 女) |

通常修改用户响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code 表示修改成功与否, msg 包括具体的信息 (成功提示或错误信息)。

1. DEM 管理员用户和超级管理员用户才拥有修改用户信息的权限。



2. DEM 超级管理员用户信息不能被修改。

注意：3. DEM 管理员用户不能修改其他管理员用户信息，只能修改自己的信息。

9.1.2.5 使用示例

通过接口查询用户列表，示例如下所示：

```
public class UserApiTest
{
    private final static String CHAR_SET = "utf-8";
    public static void main(String[] args)
    {
        // 请求用户信息访问的 URL
        HttpPost post = new HttpPost("http://192.168.2.10:8080/dem/api");
        post.setHeader(HTTP.CONTENT_TYPE,
"application/x-www-form-urlencoded; charset=" + CHAR_SET);
        post.setHeader(HTTP.CONN_DIRECTIVE, "close");
        try {
            // DEM 用户加解密引擎
            IEncryptEngine encryptEngine = new DefaultEncryptEngine();
            /***** 新建用户 *****/
            createUser(post, encryptEngine);
            /***** 查询用户 *****/
            queryUser(post, encryptEngine);
            /***** 修改用户 *****/
            modifyUser(post, encryptEngine);
            /***** 删除用户 *****/
            deleteUser(post, encryptEngine);
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
finally

    { try { post.abort();} catch (Exception ee) {} }

}

private static void createUser(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{

    // 新创建的用户信息

    BasicNameValuePair[] newUserParams = new BasicNameValuePair[] {

new BasicNameValuePair("login_name", "test"),
new BasicNameValuePair("login_pwd", encryptEngine.encrypt("888888")),
new BasicNameValuePair("role", "1"), new BasicNameValuePair("enable", "true"),
new BasicNameValuePair("real_name", "张三"), new BasicNameValuePair("email",
"test@dameng.com"), new BasicNameValuePair("phone", "8888888888"), new
BasicNameValuePair("sex", "男")};

    // 请求创建用户，并打印创建用户响应消息

    doPost(post, buildPostParams(encryptEngine, "1", newUserParams));

}

private static void queryUser(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{

    // 要查询的用户信息

    BasicNameValuePair[] queryUserParam = new BasicNameValuePair[] {/*new
BasicNameValuePair("id", "1111")*/};

    // 请求查询用户，并打印查询用户响应消息

    doPost(post, buildPostParams(encryptEngine, "3", queryUserParam));

}

private static void modifyUser(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{

    // 要修改的用户信息

    BasicNameValuePair[] modifyUserParams = new BasicNameValuePair[] {
new BasicNameValuePair("id", "1111"),new BasicNameValuePair("login_name",
"test"), new
```

```

BasicNameValuePair("old_login_pwd", encryptEngine.encrypt("888888")), new
BasicNameValuePair("new_login_pwd", encryptEngine.encrypt("888888")), new
BasicNameValuePair("role", "1"), new BasicNameValuePair("enable", "true"), new
BasicNameValuePair("real_name", "李四"), new BasicNameValuePair("email",
"test@dameng.com"), new BasicNameValuePair("phone", "8888888888"), new
BasicNameValuePair("sex", "男")};

        // 请求查询用户，并打印查询用户响应消息

        doPost(post, buildPostParams(encryptEngine, "4", modifyUserParams));

    }

private static void deleteUser(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{

        // 要删除的用户信息

        BasicNameValuePair[] delUserParam = new BasicNameValuePair[] {new
BasicNameValuePair("id", "1111")};

        // 请求删除用户，并打印删除用户响应消息

        doPost(post, buildPostParams(encryptEngine, "2", delUserParam));

    }

/*
 * 构造请求参数参数
 *
 * @param encryptEngine 密码加密引擎
 *
 * @param actionType 操作类型
 *
 * @param userParams 用户信息，以 key,value 方式给定用户信息，key 为请求的参数名，value
为请求的参数值
 */
private static List<BasicNameValuePair> buildPostParams(IEncryptEngine
encryptEngine, String actionType, BasicNameValuePair[] userParams) throws
Exception {

    List<BasicNameValuePair> formParams = new ArrayList<BasicNameValuePair>();
    // 指定资源类型为用户

    formParams.add(new BasicNameValuePair("api_type", "user"));

```

```

// DEM 用户名

formParams.add(new BasicNameValuePair("dem_user", "admin"));

// DEM 用户登录密码，用加解密引擎加密传入

formParams.add(new BasicNameValuePair("dem_pwd",
encryptEngine.encrypt("888888")));

// 操作类型，1:创建 2:删除 3:查询 4:修改

formParams.add(new BasicNameValuePair("action_type", actionType));

//用户参数

for (BasicNameValuePair formParam : userParams)

{ formParams.add(formParam); }

return formParams;

}

/**
 * 执行请求并打印响应消息*
 * @param formParams 请求参数
 */
private static void doPost(HttpPost post, List<BasicNameValuePair> formParams)
throws Exception {

    // 设置请求参数

    HttpEntity entity = new UrlEncodedFormEntity(formParams, CHAR_SET);

    post.setEntity(entity);

    // 发送请求，获取响应消息

    HttpResponse response = new DefaultHttpClient().execute(post);

    if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK)

    {throw new Exception();}

    // 打印响应消息

    StringBuilder sb = new StringBuilder();

    BufferedReader br = new BufferedReader(

        new InputStreamReader(response.getEntity().getContent(), CHAR_SET));

    String str = null;

```

```

        while ((str = br.readLine()) != null)
        {
            sb.append(str);
        }
        br.close();
        System.out.println(sb.toString().trim());
    }
}

```

9.1.2 管理数据库连接的接口

使用接口安全的连接到 DEM 服务器，可以查询并管理 DEM 管理的数据库连接（客户端工具->对象管理中注册的连接）。

由于相关的接口访问需要 DEM 登录用户密码信息和连接信息的密码，因此



DEM 需要部署在 SSL 环境，才能保证相关信息的安全性，关于 DEM 部署

注意： 在 SSL 环境详见第 1 章 DEM 部署相关介绍。

获取用户信息访问的 URL：

```
http://<dem_ip>:<dem_port>/dem/api
```

通过请求参数 `api_type=SERVER_CONN` 来指定资源类型为告警信息。

通过接口可以对 DEM 管理的数据库连接进行创建、删除、查询和修改的操作，操作类型通过传递 `action_type` 参数指定，`action_type` 的值具体定义如下：

表 9.9 `action_type` 参数说明

| 值 | 说明 |
|---|----|
| 1 | 创建 |
| 2 | 删除 |
| 3 | 查询 |
| 4 | 修改 |

不同的操作传入参数和返回消息不一样，下面一一详细介绍。

9.1.2.1 创建

请求消息参数说明如下：

表 9.10 创建连接参数

| 名称 | 说明 |
|-------------|------------------------|
| api_type | SERVER_CONN (指定获取连接信息) |
| action_type | 1 (创建) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| conn_name | 连接名 |
| ip | 数据库 ip 地址 |
| port | 数据库端口号 |
| user_name | 数据库用户名 |
| user_pwd | 数据库密码 |
| save_pwd | 是否保存密码 (true, false) |

通常创建连接响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code 表示创建成功与否, msg 包括具体的信息(成功提示或错误信息)。



所有 DEM 登录用户都有创建连接的权限。

注意:

9.1.2.2 删除

请求消息参数说明如下:

表 9.11 删除连接参数

| 名称 | 说明 |
|-------------|------------------------|
| api_type | SERVER_CONN (指定获取连接信息) |
| action_type | 2 (删除) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 连接信息的 id |

通常删除连接响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code 表示删除成功与否, msg 包括具体的信息(成功提示或错误信息)。



DEM 用户只能删除自己创建的连接

注意:

9.1.2.3 查询

请求消息参数说明如下:

表 9.12 查询连接参数

| 名称 | 说明 |
|-------------|---------------------------------------|
| api_type | SERVER_CONN (指定获取连接信息) |
| action_type | 3 (查询) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 可选, 不设置则查询用户的所有连接信息, 设置则查询对应 id 的连接信息 |

通常查询连接响应消息包括 code, msg 和 data 信息(详见表 9.1 响应消息格式说明), code 表示查询操作成功与否, msg 包括具体的信息(成功提示或错误信息), data 返回具体的连接信息, 返回连接信息 data 字段参数说明如下:

表 9.13 返回连接信息 data 字段数据格式

| 名称 | 说明 |
|---------|-------------------|
| count | 连接总数 |
| records | 连接对象数组 (Record[]) |

表 9.14 返回连接信息 Record 字段数据格式

| 名称 | 说明 |
|-----------|----------------------|
| id | 连接信息的 ID |
| conn_name | 连接名 |
| ip | 数据库 ip 地址 |
| port | 数据库端口号 |
| user_name | 数据库用户名 |
| save_pwd | 是否保存密码 (true, false) |

例如:

```
{
  id: 1574735037086
  conn_name: "dmoa-test"
  ip: "192.168.0.156"
  port: 5236
  user_name: "SYSDBA"
  save_pwd: true
}
```



DEM 用户只有查询自己创建的连接的权限。

注意：

9.1.2.4 修改

请求消息参数说明如下：

表 9.15 修改连接参数

| 名称 | 说明 |
|-------------|------------------------|
| api_type | SERVER_CONN (指定获取连接信息) |
| action_type | 4 (修改) |
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |
| id | 连接信息的 id |
| conn_name | 连接名 |
| ip | 数据库 ip 地址 |
| port | 数据库端口号 |
| user_name | 数据库用户名 |
| user_pwd | 数据库密码 |
| save_pwd | 是否保存密码 (true, false) |

通常修改连接响应消息包括 code 和 msg 信息(详见表 9.1 响应消息格式说明), code 表示修改成功与否, msg 包括具体的信息(成功提示或错误信息)。



DEM 用户只有修改自己创建的连接的权限

注意：

9.1.2.5 使用示例

通过接口查询连接列表, 示例如下所示:

```
public class ConnApiTest
{
    private final static String CHAR_SET = "utf-8";

    public static void main(String[] args)
    {
        // 请求连接信息访问的 URL
        HttpPost post = new HttpPost("http://192.168.2.10:8080/dem/api");
    }
}
```

```

post.setHeader(HTTP.CONTENT_TYPE,
"application/x-www-form-urlencoded; charset=" + CHAR_SET);

post.setHeader(HTTP.CONN_DIRECTIVE, "close");

try{

    // DEM 用户加解密引擎

    IEncryptEngine encryptEngine = new DefaultEncryptEngine();

    /***** 新建连接 *****/
    createConn(post, encryptEngine);

    /***** 查询连接 *****/
    queryConn(post, encryptEngine);

    /***** 修改连接 *****/
    modifyConn(post, encryptEngine);

    /***** 删除连接 *****/
    deleteConn(post, encryptEngine);

} catch (Exception e)

{ e.printStackTrace();}

finally{try { post.abort();}catch (Exception ee) {}}

}

private static void createConn(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{

    // 新创建的连接信息

    BasicNameValuePair[] newConnParams = new BasicNameValuePair[] {

        new BasicNameValuePair("conn_name", "test"), new BasicNameValuePair("ip",
"192.168.1.16"), new BasicNameValuePair("port", "5236"), new
BasicNameValuePair("user_name", "SYSDBA"), new BasicNameValuePair("user_pwd",
encryptEngine.encrypt("SYSDBA")), new BasicNameValuePair("save_pwd", "true")};

    // 请求创建连接，并打印创建连接响应消息

    doPost(post, buildPostParams(encryptEngine, "1", newConnParams));

}

```

```
private static void queryConn(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{
    // 要查询的连接信息
    BasicNameValuePair[] queryConnParam = new BasicNameValuePair[] {/*new
BasicNameValuePair("id", "1111")*/};

    // 请求查询连接，并打印查询连接响应消息
    doPost(post, buildPostParams(encryptEngine, "3", queryConnParam));
}

private static void modifyConn(HttpPost post, IEncryptEngine encryptEngine)
throws Exception {
    // 要修改的连接信息
    BasicNameValuePair[] modifyConnParams = new BasicNameValuePair[] {
new BasicNameValuePair("i", "1111"), new BasicNameValuePair("conn_name",
"test1"), new BasicNameValuePair("ip", "192.168.1.16"), new
BasicNameValuePair("port", "5236"), new BasicNameValuePair("user_name",
"SYSDBA"), new BasicNameValuePair("user_pwd", encryptEngine.encrypt("SYSDBA")),
new BasicNameValuePair("save_pwd", "true")};

    // 请求查询连接，并打印查询连接响应消息
    doPost(post, buildPostParams(encryptEngine, "4", modifyConnParams));
}

private static void deleteConn(HttpPost post, IEncryptEngine encryptEngine)
throws Exception{
    // 要删除的连接信息
    BasicNameValuePair[] delConnParam = new BasicNameValuePair[] {new
BasicNameValuePair("id", "1111")};

    // 请求删除连接，并打印删除连接响应消息
    doPost(post, buildPostParams(encryptEngine, "2", delConnParam));
}

/*
 * 构造请求参数

```

```

    * @param encryptEngine 密码加密引擎
    * @param actionType 操作类型
    * @param connParams 连接信息, 以 key,value 方式给定连接信息, key 为请求的参数名, value
为请求的参数值
    */

    private static List<BasicNameValuePair> buildPostParams(IEncryptEngine
encryptEngine, String actionType, BasicNameValuePair[] connParams) throws
Exception{
    List<BasicNameValuePair> formParams = new ArrayList<BasicNameValuePair>();
    // 指定资源类型为连接
    formParams.add(new BasicNameValuePair("api_type", "server_conn"));
    // DEM 用户名
    formParams.add(new BasicNameValuePair("dem_user", "admin"));
    // DEM 用户登录密码, 用加解密引擎加密传入
    formParams.add(new BasicNameValuePair("dem_pwd",
encryptEngine.encrypt("888888")));
    // 操作类型, 1:创建 2:删除 3:查询 4:修改
    formParams.add(new BasicNameValuePair("action_type", actionType));
    // 连接参数
    for (BasicNameValuePair formParam : connParams)
    {
        formParams.add(formParam);
    }
    return formParams;
}
/***
 * 执行请求并打印响应消息
 * @param formParams 请求参数
 */
private static void doPost(HttpPost post, List<BasicNameValuePair> formParams)
throws Exception{
    // 设置请求参数

```

```

HttpEntity entity = new UrlEncodedFormEntity(formParams, CHAR_SET);

post.setEntity(entity);

// 发送请求，获取响应消息

HttpResponse response = new DefaultHttpClient().execute(post);

if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK)

{ throw new Exception();}

// 打印响应消息

StringBuilder sb = new StringBuilder();

BufferedReader br = new BufferedReader(new

InputStreamReader(response.getEntity().getContent(), CHAR_SET));

String str = null;

while ((str = br.readLine()) != null)

{sb.append(str); }

br.close();

System.out.println(sb.toString().trim());

}

}

```

9.1.3 管理 DEM 告警信息的接口

使用接口安全的连接到 DEM 服务器，可以查询 DEM 的告警信息。



注意：由于相关的接口访问需要 DEM 登录用户密码信息，因此 DEM 需要部署在 SSL 环境，才能保证相关信息的安全性，关于 DEM 部署在 SSL 环境详见

第 1 章 DEM 部署相关介绍。

获取用户信息访问的 URL：

```
http://<dem_ip>:<dem_port>/dem/api
```

通过请求参数 api_type=ALERT 来指定资源类型为告警信息。

通过接口可以对告警信息进程查询操作。

使用接口安全的连接到 DEM 服务器，可以查询并管理 DEM 管理的数据库连接（客户端工具->对象管理中注册的连接）。



注意：由于相关的接口访问需要 DEM 登录用户密码信息和连接信息的密码，因此 DEM 需要部署在 SSL 环境，才能保证相关信息的安全性，关于 DEM 部署在 SSL 环境详见第 1 章 DEM 部署相关介绍。

获取用户信息访问的 URL：

```
http://<dem_ip>:<dem_port>/dem/api
```

通过请求参数 api_type=ALERT 来指定资源类型为告警信息。

通过接口可以对 DEM 管理的数据库连接进行创建、删除、查询和修改的操作，操作类型通过传递 action_type 参数指定，action_type 的值具体定义如下：

表 9.9 action_type 参数说明

| 值 | 说明 |
|---|----|
| 1 | 创建 |
| 2 | 删除 |
| 3 | 查询 |
| 4 | 修改 |

不同的操作传入参数和返回消息不一样，下面一一详细介绍。

9.1.3.1 查询

请求消息参数说明如下：

表 9.16 查询告警参数

| 名称 | 说明 |
|---------------|---|
| api_type | ALERT（指定获取连接信息） |
| dem_user | DEM 的用户名，必填 |
| dem_pwd | DEM 的用户密码，必填 |
| limit | 返回记录最大行数，缺省值为 10 |
| offset | 返回记录的偏移值，缺省值为 0 |
| level | 告警级别，0 全部，1 致命，2 严重，3 一般，4 轻微。缺省值为 0；多个值以, 分隔 |
| order | 0 降序，1 升序。缺省降序 |
| order_by | 排序字段，0 告警产生时间，1 告警级别，2 告警名称。缺省值为 0；多个值以, 分隔 |
| name | 告警名称，缺省则表示全部告警，多个值逗号, 分隔 |
| resource_name | 主机或者数据库名称，缺省则表示全部；多个值以, 分隔 |

| | |
|------------|-----------------------------|
| event_id | 告警事件 id, 缺省则表示全部, ;多个值以, 分隔 |
| start_time | 起始时间, 缺省则表示按持续时间过滤 |
| end_time | 终止时间, 缺省则表示按持续时间过滤 |
| last_time | 最近多长时间, 单位秒, 缺省值为 20 |

其中告警事件 id 定义如下:

表 9.17 告警事件 id

| 告警资源类型 | 告警事件 ID | 告警事件类型 |
|--------|---------|-----------------------------|
| 主机 | 1001 | 连接异常 |
| 主机 | 1002 | CPU 使用率 |
| 主机 | 1003 | 内存使用率 |
| 主机 | 1004 | 磁盘使用率 |
| 主机 | 1006 | 交换区使用率 |
| 主机 | 1007 | 进程监控 |
| 主机 | 1005 | 自定义监控脚本或程序的执行情况 |
| 数据库 | 2001 | 连接异常 |
| 数据库 | 2002 | CPU 使用率 |
| 数据库 | 2003 | 内存使用率 |
| 数据库 | 2004 | 文件 IO 读 |
| 数据库 | 2005 | 文件 IO 写 |
| 数据库 | 2006 | 等待任务数 |
| 数据库 | 2007 | 归档等待任务数 |
| 数据库 | 2008 | 备份失败 |
| 数据库 | 2009 | 数据库状态或模式变更 |
| 数据库 | 2101 | 表空间使用率 |
| 数据库 | 2102 | 表数据行数 |
| 数据库 | 2100 | 数据文件大小 |
| 数据库 | 2201 | 总会话数 |
| 数据库 | 2202 | 活动会话数 |
| 数据库 | 2301 | 执行 SQL 出错 |
| 数据库 | 2302 | 执行 SQL 超时 |
| 数据库 | 2303 | 自定义 SQL 监控 |
| 数据库 | 2403 | 运行日志 |
| 数据库 | 2404 | 事务日志文件空闲大小 |
| 数据库 | 2405 | 事务日志 CUR_LSN 与 FILE_LSN 差值 |
| 数据库 | 2406 | 事务日志 FILE_LSN 与 CKPT_LSN 差值 |
| 数据库 | 2401 | 死锁 |
| 数据库 | 2402 | 用户登录异常 |

通常查询告警响应消息包括 code, msg 和 data 信息(详见表 9.1 响应消息格式说明), code 表示查询操作成功与否, msg 包括具体的信息(成功提示或错误信息), data 返回具体的告警信息, 返回告警信息 data 字段参数说明如下:

表 9.18 返回告警信息 data 字段数据格式

| 名称 | 说明 |
|---------|-------------------|
| count | 告警总数 |
| records | 告警对象数组 (Record[]) |

表 9.19 返回告警信息 Record 数据格式

| 名称 | 说明 |
|--------------|--------|
| name | 告警名称 |
| resourceType | 告警资源类型 |
| resourceName | 告警资源名称 |
| eventName | 告警事件名称 |
| occurTime | 告警时间 |
| level | 告警级别 |
| description | 告警描述信息 |

例如:

```
{
  id: 1576124021889
  login_name: "t2"
  real_name: "1"
  email: ""
  phone: ""
  sex: "男"
  enable: true
  role: 3
}
```



DEM 管理员用户和超级管理员用户才拥有获取用户信息的权限。

注意:

9.1.3.2 使用示例

通过接口查询用户列表，示例如下所示：

```
public class AlertApiTest

{
    private final static String CHAR_SET = "utf-8";

    public static void main(String[] args)
    {
        // 请求查询用户信息访问的 URL

        HttpPost post = new HttpPost("http://192.168.2.10:8080/dem/api");

        post.setHeader(HTTP.CONTENT_TYPE,
                    "application/x-www-form-urlencoded; charset=" + CHAR_SET);

        post.setHeader(HTTP.CONN_DIRECTIVE, "close");

        try {
            // DEM 用户加解密引擎

            IEncryptEngine encryptEngine = new DefaultEncryptEngine();

            /***** ***** ***** ***** ***** ***** ***** ***** *****
            // 要查询的告警信息

            DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

            BasicNameValuePair[] queryUserParam = new BasicNameValuePair[] {new
BasicNameValuePair("start_time", String.valueOf(dateFormat.parse("2020-1-1
00:00:00").getTime())), new BasicNameValuePair("end_time",
String.valueOf(dateFormat.parse("2021-3-20 00:00:00").getTime()))};

            // 请求查询告警，并打印查询告警响应消息

            doPost(post, buildPostParams(encryptEngine, queryUserParam));

            }catch (Exception e) { e.printStackTrace();}

            finally

            {try{ post.abort();}catch (Exception ee) {}}

        }
    }

    /*
     * 构造请求参数
     *
     * @param encryptEngine 密码加密引擎
     */
}
```

```

* @param actionType 操作类型

* @param alertParams 告警信息,以 key,value 方式给定告警信息, key 为请求的参数名,
value 为请求的参数值

*/
private static List<BasicNameValuePair> buildPostParams(IEncryptEngine
encryptEngine, BasicNameValuePair[] alertParams) throws Exception{
List<BasicNameValuePair> formParams = new ArrayList<BasicNameValuePair>();

// 指定资源类型为告警
formParams.add(new BasicNameValuePair("api_type", "alert"));

// DEM 用户名
formParams.add(new BasicNameValuePair("dem_user", "admin"));

// DEM 用户登录密码, 用加解密引擎加密传入
formParams.add(new BasicNameValuePair("dem_pwd",
encryptEngine.encrypt("888888")));

// 告警参数
for (BasicNameValuePair formParam : alertParams)
{
    formParams.add(formParam);
}

return formParams;
}

/**
 * 执行请求并打印响应消息
 * @param formParams 请求参数
*/
private static void doPost(HttpPost post, List<BasicNameValuePair> formParams)
throws Exception{

// 设置请求参数
HttpEntity entity = new UrlEncodedFormEntity(formParams, CHAR_SET);
post.setEntity(entity);

// 发送请求, 获取响应消息
HttpResponse response = new DefaultHttpClient().execute(post);
}

```

```

        if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK)

        { throw new Exception();}

        // 打印响应消息

        StringBuilder sb = new StringBuilder();

        BufferedReader br = new BufferedReader(new

InputStreamReader(response.getEntity().getContent(), CHAR_SET));

        String str = null;

        while ((str = br.readLine()) != null)

        {sb.append(str); }

        br.close();

        System.out.println(sb.toString().trim());

    }

}

```

9.1.4 管理 prometheus 监控数据的接口

DEM 为 prometheus 提供了数据收集功能。

获取 prometheus 访问的 URL:

```
http://<dem_ip>:<dem_port>/<dem_web_app_name>/metrics
```

具体收集哪些主机和数据库的数据需要在 DEM 系统配置中设置。设置步骤为:

1. “系统配置” → “其他功能” → “prometheus_metric_mf_nodes” 中设置主机列表。
2. “系统配置” → “其他功能” → “prometheus_metric_db_nodes” 中设置数据库列表。

DEM 将根据此列表将相关主机和数据库的收集信息提供给 prometheus。



指定的数据库需要先在数据库监控中添加对该数据库的监控，才可以在

注意： prometheus_metric_db_nodes 中配置

9.1.4.1 使用示例

DEM 提供 prometheus 监控数据的获取接口，要使用 DEM 获取 prometheus 监控数据

需要以下配置：

1. 在 DEM 系统配置中设置需要收集信息的主机和数据库；
2. 配置 prometheus.yml，指定要监控的资源，示例如下：

```
- job_name: 'dem'

metrics_path: /dem/metrics

static_configs:

- targets: ['192.168.0.1:8080']
```

prometheus.yml 配置的 Endpoint 为 `http://192.168.0.1:8080/dem/metrics`。

prometheus 收集数据的时间频率在 `prometheus.yml` 内配置，详情请参见 `prometheus` 相关文档，DEM 只负责被动发送数据。

9.1.4.2 指标信息

DEM 发给 `prometheus` 监控数据的相关指标信息，如下所示。

表 9.13 `prometheus` 指标信息

| 名称 | 数据类别 | 说明 |
|-------------------------------|------|----------------------|
| dm_global_status_tps | 数据库 | 每秒事务数 |
| dm_global_status_qps | 数据库 | 每秒执行 select SQL 语句数 |
| dm_global_status_ips | 数据库 | 每秒执行 insert SQL 语句数。 |
| dm_global_status_dps | 数据库 | 每秒执行 delete SQL 语句数 |
| dm_global_status_ups | 数据库 | 每秒执行 update SQL 语句数 |
| dm_global_status_ddlps | 数据库 | 每秒执行 DDL SQL 语句数 |
| dm_global_status_nio_ips | 数据库 | 每秒从客户端接收字节数 |
| dm_global_status_nio_ops | 数据库 | 每秒往客户端发送字节数 |
| dm_global_status_fio_ips | 数据库 | 每秒读取字节数 |
| dm_global_status_fio_ops | 数据库 | 每秒写入字节数 |
| dm_global_status_mem_used | 数据库 | 内存占用字节数 |
| dm_global_status_cpu_use_rate | 数据库 | cpu 使用率，单位% |

| | | |
|---|-----|---|
| dm_global_status_swap | 数据库 | 系统交换分区相关, 单位 B。属性 type: 1) total: 总交换分区。 2) used: 已使用交换分区。 |
| dm_global_status_sessions | 数据库 | 会话数相关。属性 type: 1) total: 会话总数。 2) active: 活动会话数。 |
| dm_global_status_task_waiting | 数据库 | 等待处理任务数 |
| dm_global_status_task_ready | 数据库 | 已处理任务数 |
| dm_global_status_task_total_wait_time | 数据库 | 已处理任务的总等待时间 |
| dm_global_status_avg_wait_time | 数据库 | 已处理任务的平均等待时间 |
| dm_global_status_threads | 数据库 | 活动线程数 |
| dm_mf_status_cpu_use_rate | 主机 | cpu 使用率, 单位%。属性 type: 1) sys: 系统进程 cpu 使用率。 2) used: 总 cpu 使用率。 3) user: 用户进程 cpu 使用率。 4) wait: cpu 等待。 |
| dm_mf_status_memory_mem_total_bytes | 主机 | 总内存大小, 单位 B |
| dm_mf_status_memory_mem_used_bytes | 主机 | 已使用的内存大小, 单位 B |
| dm_mf_status_memory_swap_total_bytes | 主机 | 总交换区大小, 单位 B |
| dm_mf_status_memory_swap_used_bytes | 主机 | 已使用的交换区大小, 单位 B |
| dm_mf_status_memory_swap_page_in | 主机 | 交换区读取页数 |
| dm_mf_status_memory_swap_page_out | 主机 | 交换区写入页数 |
| dm_mf_status_disk_total_bytes | 主机 | 总磁盘大小, 单位 B |
| dm_mf_status_disk_used_bytes | 主机 | 已使用的磁盘大小, 单位 B |
| dm_mf_status_disk_write_speed_bytes | 主机 | 磁盘写入速率, 单位 Bps |
| dm_mf_status_disk_read_speed_bytes | 主机 | 磁盘读取速率, 单位 Bps |
| dm_mf_status_network_receive_speed_bytes | 主机 | 网络接收速率, 单位 Bps |
| dm_mf_status_network_transmit_speed_bytes | 主机 | 网络发送速率, 单位 Bps |

| | | |
|-------------------------------------|----|-------------------------------------|
| dm_mf_status_filesystem_total_bytes | 主机 | 分区总大小，单位 B。属性 directory 为分区路径。 |
| dm_mf_status_filesystem_used_bytes | 主机 | 分区已使用大小，单位 B。属性 directory 为分区路径。 |

9.2 基于 UDP 协议开发的接口

DEM 还为用户提供了访问 DEM 资源的 UDP 接口。SNMP AGENT 就是基于 UDP 协议开发的接口。

9.2.1 SNMP AGENT

简单网络管理协议（SNMP）是专门设计用于在 IP 网络管理网络节点的一种标准的应用层协议。通过实现 SNMP AGENT，DEM 支持使用 SNMP 监控和管理数据库节点信息。SNMP AGENT 的配置、实现和使用举例如下介绍。

9.2.1.1 DEM 配置

DEM 可以在“系统配置”->“其他功能”->“snmp_enable”中设置 SNMP AGENT 监听的开启和关闭；在“系统配置”->“其他功能”->“snmp_udp_port”中设置 SNMP 监听的端口号。DEM 将根据这两个配置信息配置 SNMP AGENT，监听 SNMP 请求并响应。

请注意，如果在监听开启时，新的数据库实例被 DEM 监控，SNMP AGENT 无法获取到这些新数据库实例的相关参量值；如需获取最新的所有数据库实例的相关参量值，请重启 SNMP 监控。

9.2.1.2 DM SNMP AGENT 的实现

本章节提供了 PUBLIC RDBMS MIB 在达梦实现中的各个变量的信息。请注意，本章节只强调了达梦的实现。有关这些变量的其他实现替代方案的信息，请参阅 RDBMS-MIB 相关文档。

本章节包括一些小节，介绍达梦在下表中实现的 PUBLIC RDBMS MIB 变量

- rdbmsDbTable

- rdbmsDbInfoTable
- rdbmsSrvTable
- rdbmsSrvInfoTable
- rdbmsSrvParamTable
- rdbmsRelTable

请注意，达梦没有实现 PUBLIC RDBMS MIB 的下表：

- rdbmsDbParamTable
- rdbmsDbLimitedResourceTable
- rdbmsSrvLimitedResourceTable
- rdbmsTraps

尝试访问达梦未实现的任何变量将导致返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。

9.2.1.2.1 rdbmsDbTable

rdbmsDbTable 包含 DEM 监控的每个数据库的基本信息。任意被监控数据库（无论是否处于存活状态）都由 rdbmsDbTable 中的一行表示，并且每行都由 rdbmsDbIndex 索引唯一确定。rdbmsDbTable 变量的值从 DEM 后台数据库检索得到。

下面列出了达梦实现的 rdbmsDbTable 表变量、对应的 OID 及其实现方式。

表 9.14 rdbmsDbTable

| 变量 | OID | DM 实现方式 |
|----------------------|----------------------|--|
| rdbmsDbIndex | 1.3.6.1.2.1.39.1.1.1 | 语义更改为 OctetString，其值是此数据库的唯一标识值，可以在 DEM 的数据库管理界面查看 |
| rdbmsDbPrivateMIBOID | 1.3.6.1.2.1.39.1.1.2 | 不支持 |
| rdbmsDbVendorName | 1.3.6.1.2.1.39.1.1.3 | 硬编码为“Dameng” |
| rdbmsDbName | 1.3.6.1.2.1.39.1.1.4 | 此变量的值是从 DEM 后台数据库中获取。 此名称是全局唯一的，格式为“ip:port” |

| | | |
|----------------|----------------------------|----------------------|
| rdbmsDbContact | 1.3.6.1.2.1.39.1. 1.1.5 | 硬编码为“www.dameng.com” |
|----------------|----------------------------|----------------------|

9.2.1.2.2 rdbmsDbInfoTable

`rdbmsDbInfoTable` 包含 DEM 监控的处于存活状态的数据库实例的其他基本信息。如果数据库处于非存活状态，则尝试访问此表中的相应变量实例可能会导致返回 `noSuchName` (SNMPv1) 或 `noSuchInstance` (SNMPv2)。“存活状态”由该数据库对应的 `rdbmsRelTable` 表中 `rdbmsRelState` 值确定。

`rdbmsDbInfoTable` 中的每行表示 DEM 监控的处于存活状态的数据库实例，每行都由 `rdbmsDbIndex` 索引唯一确定。但是，由于 `rdbmsDbTable` 包含 DEM 监控的每个数据库的一行数据，无论其是否处于存活状态，因此在任意时间点，`rdbmsDbTable` 和 `rdbmsDbInfoTable` 两个表中的数据之间都可能存在一对一的对应关系。`rdbmsDbInfoTable` 表变量的值从 DEM 后台数据库检索得到。

下表列出了达梦实现的 `rdbmsDbInfoTable` 表变量、对应的 OID 及其实现方式。

表 9.15 `rdbmsDbInfoTable`

| 变量 | OID | DM 实现方式 |
|---------------------------------------|----------------------------|-------------------|
| <code>rdbmsDbInfoProductName</code> | 1.3.6.1.2.1.39.1.2. 1.1 | 数据库实例名 |
| <code>rdbmsDbInfoVersion</code> | 1.3.6.1.2.1.39.1.2. 1.2 | 数据库版本号 |
| <code>rdbmsDbInfoSizeUnits</code> | 1.3.6.1.2.1.39.1.2. 1.3 | 1~5 |
| <code>rdbmsDbInfoSizeAllocated</code> | 1.3.6.1.2.1.39.1.2. 1.4 | 数据库实例所有表空间的大小总和 |
| <code>rdbmsDbInfoSizeUsed</code> | 1.3.6.1.2.1.39.1.2. 1.5 | 数据库实例所有表空间已使用大小总和 |
| <code>rdbmsDbInfoLastBackup*</code> | 1.3.6.1.2.1.39.1.2. 1.6 | 不支持 |

达梦不支持用星号标记的变量。试图访问这些变量可能会导致 noSuchName(SNMPv1) 或 noSuchInstance(SNMPv2)。

PUBLIC RDBMS MIB 规定的 rdbmsDbInfoSizeUnits 共有 5 种可选值，代表 rdbmsDbInfoSizeAllocated 和 rdbmsDbInfoSizeUsed 的单位，用于度量此数据库实例大小，分别为 bytes(1)、kbytes(2)、mbytes(3)、gbytes(4)、tbytes(5)，每个单位的换算值都是二进制倍数 (1K=1024)。达梦设置 rdbmsDbInfoSizeUnits 的值为允许 rdbmsdbinfoallocates 表示为 32 位整数的最小值。rdbmsDbInfoSizeUsed 应始终小于或等于 rdbmsDbInfoSizeAllocated。

注意，DM 没有实现 rdbmsDbInfoLastBackup，因为 DM 允许备份单个表空间；因此，最后一次完整数据库备份的时间没有意义。

9.2.1.2.3 rdbmsSrvTable

rdbmsSrvTable 包含 DEM 监控的处于存活状态的数据库服务的基本信息。在 DM 标准的数据库配置中，一个服务支持一个数据库。因此，虽然 rdbmsSrvTable 的内容本质上反映了 rdbmsDbTable 的内容，但 DM 选择实现这两个表，以便遵守 PUBLIC RDBMS MIB 标准的中心原则之一。这一原则是，为了考虑其他数据库供应商的体系结构必须区分数据库及其服务。

任意被监控的数据库（无论是否处于存活状态）都由 rdbmsDbTable 中的一行表示。rdbmsSrvTable 中的每行表示 DEM 上给定的数据库服务器，每行都由 applIndex 索引唯一确定。rdbmsSrvTable 表变量的值从 DEM 后台数据库检索得到。

下表列出了达梦实现的 rdbmsDbSrvTable 表变量、对应的 OID 及其实现方式。

表 9.16 rdbmsDbSrvTable

| 变量 | OID | DM 实现方式 |
|------------------------|----------------------------|---------------------|
| rdbmsSrvPrivateMIBOID* | 1.3.6.1.2.1.39.1.5.1 .1 | 不支持 |
| rdbmsSrvVendorName | 1.3.6.1.2.1.39.1.5.1 .2 | 硬编码为“Dameng” |
| rdbmsSrvProductName | 1.3.6.1.2.1.39.1.5.1 .3 | 硬编码为“Dameng Server” |

| | | |
|-----------------|----------------------------|----------------------|
| rdbmsSrvContact | 1.3.6.1.2.1.39.1.5.1 .4 | 硬编码为“www.dameng.com” |
|-----------------|----------------------------|----------------------|

如果私有 MIB 不存在对象标识符，尝试访问此对象将返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。

9.2.1.2.4 rdbmsSrvInfoTable

rdbmsSrvInfoTable 包含 DEM 监控的每个数据库的服务的附加信息。如果数据库处于非存活状态，则尝试访问此表中的相应变量实例可能会返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。“存活状态”由该数据库对应的 rdbmsRelTable 表中 rdbmsRelState 值确定。

rdbmsDbSrvInfoTable 中的每行表示 DEM 监控处于存活状态的数据库服务器实例，每行都由 applIndex 索引唯一确定。rdbmsSrvInfoTable 表变量的值从 DEM 后台数据库检索得到。

下表列出了达梦实现的 rdbmsDbInfoTable 表变量、对应的对象 ID 及其实现方式。

表 9.17 rdbmsDbSrvInfoTable

| 变量 | OID | DM 实现方式 |
|----------------------------------|----------------------------|-----------------------------------|
| rdbmsSrvInfoStartTime | 1.3.6.1.2.1.39. 1.6.1.1 | 数据库实例最近一次启动时间 |
| rdbmsSrvInfoFinishedTransactions | 1.3.6.1.2.1.39. 1.6.1.2 | 语义更改为 Counter64, 数据库实例完成的总事务数 |
| rdbmsSrvInfoDiskReads | 1.3.6.1.2.1.39. 1.6.1.3 | 语义更改为 Counter64, 数据库实例物理磁的盘读取字节总数 |
| rdbmsSrvInfoLogicalReads* | 1.3.6.1.2.1.39. 1.6.1.4 | 不支持 |
| rdbmsSrvInfoDiskWrites | 1.3.6.1.2.1.39. 1.6.1.5 | 语义更改为 Counter64, 数据库实 |

| | | 物理的磁盘写入字节总数 |
|--|-----------------------------|-------------|
| rdbmsSrvInfoLogicalWrites* | 1.3.6.1.2.1.39. 1.6.1.6 | 不支持 |
| rdbmsSrvInfoPageReads* | 1.3.6.1.2.1.39. 1.6.1.7 | 不支持 |
| rdbmsSrvInfoPageWrites* | 1.3.6.1.2.1.39. 1.6.1.8 | 不支持 |
| rdbmsSrvInfoDiskOutOfSpaces* | 1.3.6.1.2.1.39. 1.6.1.9 | 不支持 |
| rdbmsSrvInfoRequestsHandled* | 1.3.6.1.2.1.39. 1.6.1.10 | 不支持 |
| rdbmsSrvInfoRequestsRecvs* | 1.3.6.1.2.1.39. 1.6.1.11 | 不支持 |
| rdbmsSrvInfoRequestSends* | 1.3.6.1.2.1.39. 1.6.1.12 | 不支持 |
| rdbmsSrvInfoHighwaterIn-boundAssociations* | 1.3.6.1.2.1.39. 1.6.1.13 | 不支持 |
| rdbmsSrvInfoMaxInboundAssociations* | 1.3.6.1.2.1.39. 1.6.1.14 | 不支持 |

达梦不支持用星号标记的变量。试图访问这些变量可能会返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。

9.2.1.2.5 rdbmsSrvParamTable

rdbmsSrvParamTable 包含 DEM 监控的每个数据库服务的配置参数信息。如果数据库处于非存活状态，则尝试访问此表中的相应变量实例可能会返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。“存活状态”由该数据库对应的 rdbmsRelTable 表中 rdbmsRelState 值确定。

rdbmsSrvParamTable 中的每行表示 DEM 监控的数据库服务的配置参数，每行首先

由 applIndex 索引，然后由 rdbmsSrvParamName 索引，最后由 rdbmsSrvParamSubIndex 索引；因此，数据库服务与其配置参数之间通常存在一对多关系，配置参数与其多个值之间通常存在一对多关系。但是，在 DM 实现中，配置参数的多个值被连接到该参数的单行中。每个 rdbmsSrvParamTable 变量的值直接从 DEM 后台数据库检索得到。

下表列出了达梦实现的 rdbmsSrvParamTable 表变量、对应的 OID 及其实现方式。

表 9.18 rdbmsDbSrvParamTable

| 变量 | OID | DM 实现方式 |
|----------------------------|----------------------------|--------------------------------|
| rdbmsSrvParamName | 1.3.6.1.2.1.39.1.7.1 .1 | 动态视图 v\$dm_ini 中参数名 |
| rdbmsSrvParamSubIndex x | 1.3.6.1.2.1.39.1.7.1 .2 | 硬编码为 1 |
| rdbmsSrvParamID* | 1.3.6.1.2.1.39.1.7.1 .3 | 不支持 |
| rdbmsSrvParamCurrVal ue | 1.3.6.1.2.1.39.1.7.1 .4 | 动态视图 v\$dm_ini 中参数对 应的实际参数值 |
| rdbmsSrvComment* | 1.3.6.1.2.1.39.1.7.1 .5 | 不支持 |

达梦不支持用星号标记的变量。试图访问这些变量可能会返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。

9.2.1.2.6 rdbmsRelTable

rdbmsRelTable 表示 DEM 监控的数据库和数据库服务的关联。在 DM 标准的数据库配置中，一个服务支持一个数据库；因此，实现 rdbmsRelTable 表是为了遵循 PUBLIC RD BMS MIB 的标准。

rdbmsDbRelTable 中的每行表示数据库服务和 DEM 监控的数据库之间的映射，每行首先由 rdbmsDbIndex 索引，然后由 applIndex 索引；因此，对于每个数据库一个服务的 DM 标准配置，这两个索引是相同的。

下表列出了 DM 实现的 rdbmsRelTable 表变量、对应的 OID 及其实现方式。

表 9.19 rdbmsRelTable

| 变量 | OID | DM 实现方式 |
|---------------------|------------------------|---------|
| rdbmsRelState | 1.3.6.1.2.1.39.1.9.1.1 | 2 或 5 |
| rdbmsRelActiveTime* | 1.3.6.1.2.1.39.1.9.1.2 | 不支持 |

达梦不支持用星号标记的变量。试图访问这些变量可能会返回 noSuchName (SNMPv1) 或 noSuchInstance (SNMPv2)。

PUBLIC RDBMS MIB 规定了 rdbmsRelState 共有五种状态值，但是达梦对于 rdbmsRelState 设置的可选项只有 Active (2) 和 Unavailable (5)。Active (2) 表示数据库被 DEM 监控并且处于存活状态。Unavailable (5) 表示数据库被 DEM 监控但是不处于存活状态。

9.2.1.3 应用举例

按照如下步骤使用 SNMP 协议查询达梦数据库实例的相关信息。请注意，SNMP 请求中的 OID 参数需要附加非末尾索引的长度信息。同时，第三方 SNMP 工具 iReasoning MIB Brower 和 SNMPUTIL 等不支持字符串类型的 OID 自动转换为 SNMP AGENT 可识别 OID，需要手动将例子中引号内容转换为对应的 ASCLL 码；NET-SNMP 支持字符串类型的 OID 自动转换为 SNMP AGENT 可识别 OID；具体工具的使用请查阅相关官网。

DEM 系统配置中打开 DEM SNMP 服务，并且配置 SNMP 监听端口号，默认是 161。

| | | |
|---------------|------|---------------------|
| snmp_enable | true | (1/0),开启时必须实现SNMP协议 |
| snmp_udp_port | 161 | SNMP协议监听网络端口号 |

图 9.1 SNMP 系统配置

1. DEM 数据库监控中查看要查询的数据库实例的唯一标识 ID，例如数据库实例 192.168.1.45:1077 的编号 1612660573027；此编号可以在“操作”->“查看基本信息”对话框中查看和复制。

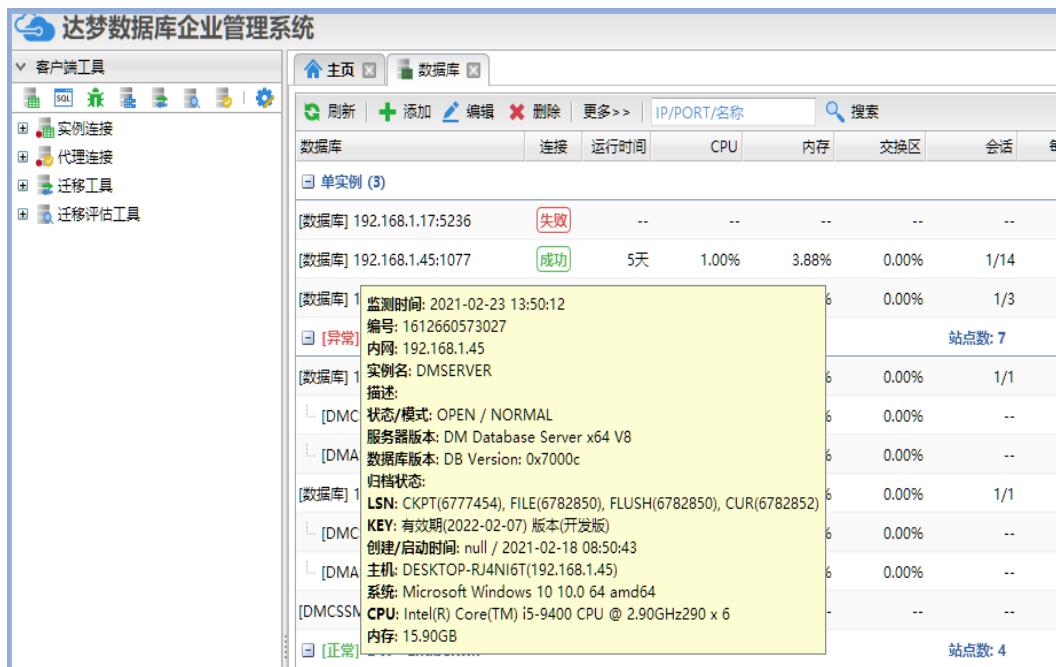


图 9.2 数据库唯一标识

2. 使用第三方工具（以下示例使用 NET-SNMP 作为第三方工具）按照 SNMP 请求规则发起请求等待 DEM SNMP 的响应。下面举例介绍。

1) 查询某个数据库实例 OID 为 1.3.6.1.2.1.39.1.1.1.4 的值，表示该数据库实例 rdbmsDbName 参量值；利用步骤 2 查看的该数据库的唯一标识 ID，拼接“1612660573027”作为查询索引（因为此索引为末尾索引，所以不需要附加其长度信息），其中，“1612660573027”为该数据库实例的唯一标识 ID；最后，使用 net-snmp 执行 snmpget 命令即可得到 SNMP 响应结果。

```
[test@test210 bin]$ ./snmpget -v 2c -c public 192.168.1.17:161 .1.3.6.1.2.1.39.1.1.1.4."1612660573027"
SNMPv2-SMI::mib-2.39.1.1.4.49.54.49.50.54.54.48.53.55.51.48.50.55 = STRING: "192.168.1.45:1077"
[test@test210 bin]$
```

图 9.3 使用 SNMP 查询指定数据库名字

2) 查询某个数据库实例 OID 为 1.3.6.1.2.1.39.1.7.1.4，表示该数据库实例 rdbmsSrvParamCurrValue 参量值；利用步骤 2 查看的该数据库的唯一标识 ID、要查询的某个参数名（例如“HAGR_DISTINCT_OPT_FLAG”），拼接 13.'1611191105106'.22.'HAGR_DISTINCT_OPT_FLAG'.1 作为查询索引（因为此索引包含三个信息，唯一标识 ID 和参数名不是末尾索引，按照 SNMP 请求规则需要附加其长度信息），最后，使用 net-snmp 执行 snmpget 命令即可得到 SNMP 响应结果。

```
[test@test210 bin]$ ./snmpget -v 2c -c public 192.168.1.17:161 .1.3.6.1.2.1.39.1.7.1.4."13.'1612660573027'.22.'HAGR_DISTINCT_OPT_FLAG'.1"
SNMPv2-SMI::mib-2.39.1.7.1.4.13.49.54.49.50.54.54.48.53.55.51.48.50.55.22.72.65.71.82.95.68.73.83.84.73.78.67.84.95.79.80.84.95.70.76.65.71.1 = STRING: "2"
[test@test210 bin]$
```

图 9.4 使用 SNMP 查询指定数据库的指定参数值

3) 查询某个数据库实例 OID 为 1.3.6.1.2.1.39.1.9.1.1, 表示该数据库实例 的状态信息参量值；利用步骤 2 查看的该数据库的唯一标识 ID，拼接 13.'1612660573027'.'161266057302' 作为查询索引（因为此索引包含两个信息，唯一标识 ID 不是末尾索引，按照 SNMP 请求规则需要附加其长度信息），最后，使用 net-snmp 执行 snmpget 命令即可得到 SNMP 响应结果。

```
[test@test210 bin]$ ./snmpget -v 2c -c public 192.168.1.17:161 .1.3.6.1.2.1.39.1.9.1.1."13.'1612660573027'.'1612660573027'"  
SNMPv2-SMI::mib-2.39.1.9.1.1.13.49.54.49.50.54.48.53.55.51.48.50.55.49.54.49.50.54.54.48.53.55.51.48.50.55 = INTEGER: 2  
[test@test210 bin]$
```

图 9.5 使用 SNMP 查询指定数据的指定参数值

10 自动登录

为给其他应用提供自动跳转登录 DEM 的功能，DEM 提供自动登录功能。自动登录功能方便用户将 DEM 与其他应用配合使用。用户通过在 DEM 的 URL 地址中添加自动登录相关参数信息，即可实现 DEM 的自动登录功能。

DEM 自动登录目前分为两种：

1. 自动登录 DEM
2. 自动登录 DEM 并且自动连接 DB

加密登录 DEM 的 URL 地址。如下所示：

```
http://<dem_ip>:8080/dem/auto_login_ex_action
```

10.1 参数说明

自动登录相关参数说明。区分大小写。

表 10.1 自动登录参数说明

| 名称 | 说明 |
|-----------------|--|
| auto_login_type | <p>自动登录功能类型。 1：登录 DEM，2：打开客户端工具节点，4：打开面板。可使用 1,2,4 的值进行按位或计算，配置多种自动登录功能。</p> <p>功能示例：</p> <ul style="list-style-type: none"> 1：登录 DEM。 2：打开客户端工具节点。（不登录 DEM，此功能无效） 3：登录 DEM，打开客户端工具节点。 4：打开面板。（不登录 DEM，此功能无效） 5：登录 DEM，打开面板。 6：打开客户端工具节点，打开面板。（不登录 DEM，此功能无效） 7：登录 DEM，打开客户端工具节点，打开面板 |
| valid_time | url 有效时间，单位：秒。 |
| version | 自动登录版本号，值：2。 |
| dem_params | 登录 DEM 参数，JSON 格式，当 auto_login_type 包含登录 DEM 功能时生效。 |
| node_params | 打开客户端工具节点参数，JSON 格式，当 auto_login_type 包含登录打开客户端工具节点功能时生效。 |
| panel_params | 打开面板参数，JSON 格式，当 auto_login_type 包含登录面板功能时生效。 |

| | |
|--|------------|
| | 打开面板功能时生效。 |
|--|------------|

10.2 参数配置

参数均为 KEY=<value>模式，其中 dem_params, node_params, panel_params 的值为 JSON 格式，所有属性值为字符串类型。具体配置下面详细介绍。

10.2.1 dem_params

属性说明。

| 名称 | 说明 |
|----------|-----------|
| dem_user | DEM 的用户名 |
| dem_pwd | DEM 的用户密码 |

10.2.2 node_params

属性说明。

| 名称 | 说明 |
|---------|---|
| type | 打开客户端工具节点类型。 类型值如下所示： manager：新建对象管理节点。 |
| db_host | 数据库地址。 (type 值为 manager 时生效) |
| db_port | 数据库端口号。 (type 值为 manager 时生效) |
| db_user | 数据库用户名。 (type 值为 manager 时生效) |
| db_pwd | 数据库密码。 (type 值为 manager 时生效) |

10.2.3 panel_params

属性说明。

| 名称 | 说明 |
|------|---------|
| type | 打开面板类型。 |

| | |
|---------------|---|
| | 类型值如下所示: mpp_dashboard: MPP 集群数据大盘。 mpp_awr: MPP 集群 AWR 面板。 |
| full_view | 全部页面用于显示当前面板。目前仅支持值为 true。(type 值为 mpp_dashboard 时生效) |
| db_host | 数据库地址。(type 值为 mpp_dashboard 时生效) |
| db_port | 数据库端口号。(type 值为 mpp_dashboard 时生效) |
| db_user | 数据库用户名。(type 值为 mpp_dashboard 时生效) |
| db_pwd | 数据库密码。(type 值为 mpp_dashboard 时生效) |
| serial_number | MPP 集群编号, 可在 DEM -> 资源监控 -> 集群属性中查看。(type 值为 mpp_awr 时生效) |

10.3 参数传递方式

传递参数有以下两种方式：

1. 明文传递

2. 登录 DEM

明文 URL 如下所示：

```
http://192.168.2.10:8080/dem/?auto_login_type=1&version=2&dem_params={"dem_u
ser":"admin","dem_pwd":"888888"}&valid_time= 10000000
```

3. 密文传递

在实际应用场景中, 相关字段明文暴露出来是不安全的, 所以建议用户使用密文方式进行自动登录。密文生成方式参见 [10.4 自动登录参数加密](#)。

4. 登录 DEM

密文 URL 如下所示：

```
http://192.168.2.10:8080/dem/?login_token=V01H1RDvVmfl1NcQU52rHvznT5xh40Yusd
v6IvH24rJq3lssU16UxjU8uhavF98i6IjazpIuII2aIAtnCVsD12XqDz6FnKibV3jvSIzbb901sH
qa1BfCdsoJodEkE4Z4dun4QodHF172IGbt7OKXi9OQLcn7cjBK0HY2pK9
```

10.4 自动登录参数加密

自动登录参数加密的 URL 地址：

```
http://<dem_ip>:8080/<dem_web_app_name>/API/AutoLoginEncryptParam
```

用户通过 POST 方法将自动登录参数发送到此 URL 地址，DEM 将返回加密后的自动登录参数。用户将加密后的自动登录参数添加到 DEM 的 URL 地址中，实现自动登录功能。

10.4.1 示例

1. 通过第三方软件或程序将自动登录参数发送至待加密 URL 地址。

输入的待加密 URL 地址如下：

```
http://192.168.2.10:8080/dem/API/AutoLoginEncryptParam?auto_login_type=1&version=2&dem_params={"dem_user":"admin","dem_pwd":"888888"}&valid_time=10000000
```

如下图所示：

| KEY | VALUE | KEY | VALUE | KEY | VALUE | KEY | VALUE |
|-----------------|-------|---------|-------|------------|--|------------|--------|
| auto_login_type | 1 | version | 2 | dem_params | {"dem_user": "admin", "dem_pwd": "888888"} | valid_time | 100000 |
| Key | Value | | | | | | |

2. DEM 返回加密后的相关信息。

加密成功后返回信息，如下所示：

```
{
  "code": 0,
  "message": "",
  "data": "login_token=V01zuCnDYnQQ4cUgs1SQvJWpyFwHff1YaB8jfgyZb8evctEPXygOH
p7OxLgEnKZWPMDOjxypYwnHI7xSdMrzczDHDgXKGCrADYeRkR1BR11G98SA8kXV9rK3BGtLpsix
2q5uE0z12Crr2vJBS9P1QGZ8TUbfduxyST"
```

}

3. 将步骤 2 返回信息的 data 字段内容，添加至 DEM 的 URL 中。

原明文 URL。格式如下所示：

`http://<dem_ip>:8080/<dem_web_app_name>/`

添加 data 字段内容后的密文 URL。格式如下所示：

`http://<dem_ip>:8080/<dem_web_app_name>/?login_token=XXX`

4. 将添加自动登录加密信息后的 URL 输入到浏览器中，实现 DEM 的自动登录功能。

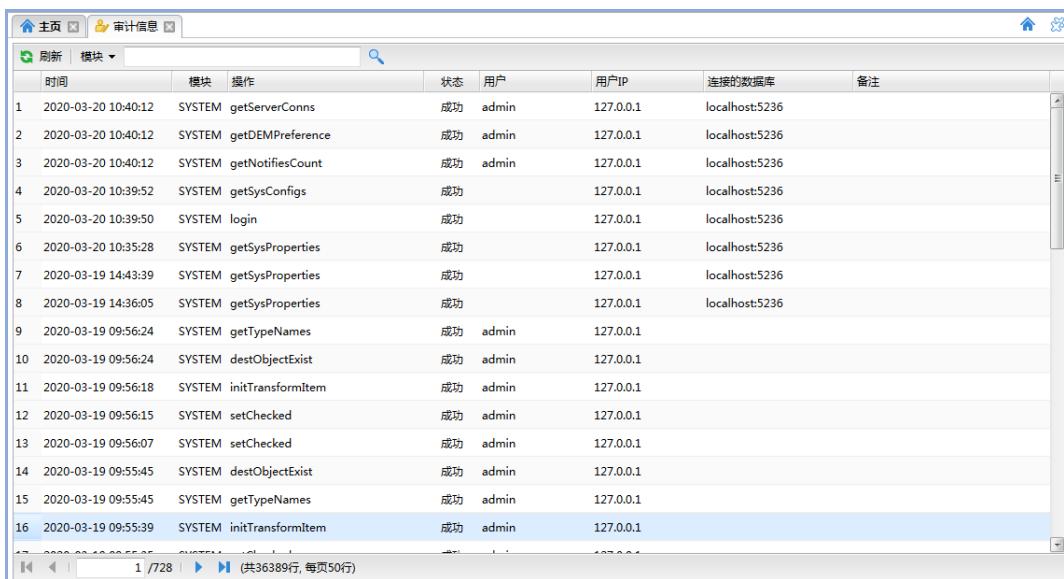
11 DEM 审计

DEM 系统自带审计功能，可以记录用户每一步操作信息。并且审计功能可以通过参数配置来决定是否开启。配置参数 `audit_enable`，默认为 0，表示不开启 DEM 系统审计。

如果开启 DEM 系统审计，则 DEM 每隔一个 `audit_flush_freq`（系统配置参数，默认值 30 秒）将审计信息存储到 DEM 后台库。

管理员和超级管理员用户可以通过 DEM 页面访问审计信息。路径：[系统管理->审计信息](#)。

打开审计信息面板，查看具体审计信息。



The screenshot shows a table with 16 rows of audit log entries. The columns are: Time, 模块 (Module), 操作 (Operation), 状态 (Status), 用户 (User), 用户IP (User IP), 连接的数据库 (Connected Database), and 备注 (Notes). The data includes various system operations like 'getServerConns', 'getDEMPreference', etc., performed by user 'admin' from IP '127.0.0.1' on 'localhost:5236'.

| 时间 | 模块 | 操作 | 状态 | 用户 | 用户IP | 连接的数据库 | 备注 |
|------------------------|--------|-------------------|----|-------|-----------|----------------|----|
| 1 2020-03-20 10:40:12 | SYSTEM | getServerConns | 成功 | admin | 127.0.0.1 | localhost:5236 | |
| 2 2020-03-20 10:40:12 | SYSTEM | getDEMPreference | 成功 | admin | 127.0.0.1 | localhost:5236 | |
| 3 2020-03-20 10:40:12 | SYSTEM | getNotifiesCount | 成功 | admin | 127.0.0.1 | localhost:5236 | |
| 4 2020-03-20 10:39:52 | SYSTEM | getSysConfigs | 成功 | | 127.0.0.1 | localhost:5236 | |
| 5 2020-03-20 10:39:50 | SYSTEM | login | 成功 | | 127.0.0.1 | localhost:5236 | |
| 6 2020-03-20 10:35:28 | SYSTEM | getSysProperties | 成功 | | 127.0.0.1 | localhost:5236 | |
| 7 2020-03-19 14:43:39 | SYSTEM | getSysProperties | 成功 | | 127.0.0.1 | localhost:5236 | |
| 8 2020-03-19 14:36:05 | SYSTEM | getSysProperties | 成功 | | 127.0.0.1 | localhost:5236 | |
| 9 2020-03-19 09:56:24 | SYSTEM | getTypeNames | 成功 | admin | 127.0.0.1 | | |
| 10 2020-03-19 09:56:24 | SYSTEM | destObjectExist | 成功 | admin | 127.0.0.1 | | |
| 11 2020-03-19 09:56:18 | SYSTEM | initTransformItem | 成功 | admin | 127.0.0.1 | | |
| 12 2020-03-19 09:56:15 | SYSTEM | setChecked | 成功 | admin | 127.0.0.1 | | |
| 13 2020-03-19 09:56:07 | SYSTEM | setChecked | 成功 | admin | 127.0.0.1 | | |
| 14 2020-03-19 09:55:45 | SYSTEM | destObjectExist | 成功 | admin | 127.0.0.1 | | |
| 15 2020-03-19 09:55:45 | SYSTEM | getTypeNames | 成功 | admin | 127.0.0.1 | | |
| 16 2020-03-19 09:55:39 | SYSTEM | initTransformItem | 成功 | admin | 127.0.0.1 | | |

图 11.1 DEM 审计信息面板

审计信息面板详细记录了时间、模块、操作、操作结果、用户、用户 IP 和连接的数据

库等信息。

其中模块分 SYSTEM 和 DBTOOL，客户端工具相关操作属于 DBTOOL 模块，其他属于 SYSTEM 模块。

12 版本信息

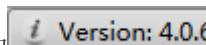
DEM 版本信息可以通过 DEM 右下角的  按钮查看，点击可以查看具体 DEM 版本信息（对应的 dmagent 的版本信息与 DEM 版本信息保持一致）。



图 12.1 DEM 版本信息

13 demsdk 文件

DEM 提供了 demsdk.jar 和 demsdk-sources.jar 文件(在达梦数据库安装目录下的 web/demsdk 目录下)，demsdk.jar 中包含了部分功能接口，用户可以通过自定义类来实现这些接口。demsdk-sources.jar 为 demsdk.jar 的源码包，用户在开发工具中导入 demsdk-sources.jar 后，可在开发过程中查看 demsdk.jar 接口和类的相关注释，方便开发。

13.1 接口/类文件

demsdk.jar 提供的接口和类，如下所示：

表 13.1 demsdk.jar 提供的接口和实现类

| 接口/类名称 | 作用 | 说明 |
|--|---|---|
| com.dameng.dem.server.util.I Notify | 自定义推送接口 | 发送消息方法：boolean send(NotifyUser user, Notify notify) |
| com.dameng.dem.server.util.IPhoneNotify | 自定义推送接口(旧版本兼容)，新功能开发建议使用 com.dameng.dem.server.util.INotify 接口。 | 发送消息方法： 1. boolean send(String phone, String message) 2. boolean send(String phone, Notify notify) 3. boolean send(NotifyUser user, Notify notify) |
| com.dameng.dem.server.util.Notify | 通知信息类 | 警告的相关信息 |
| com.dameng.dem.server.util.NotifyUser | 通知接收者 | 通知接收者的相关信息 |
| com.dameng.dem.server.util.IEncryptEngin | 加解密引擎接口 | 加密方法：String encrypt(String str) |

| | | |
|---|-------------|-------------------------------------|
| e | | 解密方法: String decrypt(String str) |
| com.dameng.dem.server.util.DefaultEncryptEngine | 默认的加解密引擎实现类 | IEncryptEngine 接口的实现类 |

13.1.1 INotify

com.dameng.dem.server.util.INotify, 自定义推送接口。用户使用自定义推送功能, 需实现此接口, 并在 DEM 中配置自定义推送相关信息。

接口相关信息, 如下所示:

```
/**
 * 第三方系统推送（短信或其他外部系统）
 *
 * @param user
 *          接收者
 *
 * @param notify
 *          推送的信息
 *
 * @return 信息发送成功, 返回true, 信息发送失败, 返回false
 */
boolean send(NotifyUser user, Notify notify);
```

13.1.2 IPhoneNotify

com.dameng.dem.server.util.IPhoneNotify, 自定义推送接口。旧版本兼容使用, 新开发功能建议使用 com.dameng.dem.server.util.INotify 接口。

接口相关信息, 如下所示:

```
/**
 * 短信发送接口, 短信内容格式由平台拼接, 不关注短信内容格式, 重写该方法即可
 *
```

```

* @param phone
*
*          手机号

* @param message
*
*          短信内容

*
* @return 信息发送成功,返回true,信息发送失败,返回false
*/
default boolean send(String phone, String message)
{
    System.out.println(phone + " -> " + message);

    return true;
}

/**
* 短信发送接口,短信内容格式可定制,需调整默认的短信内容格式,可重写改方法
*
* @param phone
*
*          手机号

* @param notify
*
*          待发送的通知对象

*
* @return 信息发送成功,返回true,信息发送失败,返回false
*/
default boolean send(String phone, Notify notify)
{
    return this.send(phone, notify.toString());
}

/**
* 短信发送接口,短信内容格式可定制,需调整默认的短信内容格式,可重写改方法
*
* @param user

```

```

    *      接收者
    * @param  notify
    *      通知信息
    *
    * @return 信息发送成功,返回true,信息发送失败,返回false
*/
@Override
default boolean send(NotifyUser user, Notify notify)
{
    return this.send(user.phone, notify);
}

```

13.1.3 Notify

com.dameng.dem.server.util.Notify, 通知信息类。包含推送通知的相关信息。

类相关信息，如下所示：

```

/**
 * 警告ID, 警告的唯一标识
 */
public long id;

/**
 * 异常发现时间
 */
public String gmtHappen;

/**
 * 警告生成时间
 */
public String gmtCreate;

/**
 * 警告所属告警配置的名称

```

```
 */     public String alertName;  
  
    /**  
     * 警告所属的规则的名称（具体到告警配置的规则）  
    */     public String ruleName;  
  
    /**  
     * 资源类型 (DB/MF/SP/MP/BP/ASM/CSS/CSSM/MONITOR/WATCHER)  
    */  
    public String resType;  
  
    /**  
     * 警告所属的资源的名称（所属数据库或主机等）  
    */  
    public String resName;  
  
    /**  
     * 警告级别 (致命/严重/一般/轻微/恢复)  
    */  
    public String level;  
  
    /**  
     * 警告具体内容  
    */  
    public String message;
```

13.1.4 NotifyUser

com.dameng.dem.server.util.NotifyUser，通知接收者类。包含通知接受者的相关信息。

类相关信息，如下所示：

```
/**  
 * 用户ID  
 */  
public long id;  
  
/**  
 * 用户登录名  
 */  
public String loginName;  
  
/**  
 * 用户真实名  
 */  
public String realName;  
  
/**  
 * 用户手机号  
 */  
public String phone;  
  
/**  
 * 用户邮件地址  
 */  
public String email;
```

13.1.5 IEncryptEngine

com.dameng.dem.server.util.IencryptEngine, 加解密引擎的接口。

接口相关信息，如下所示：

```
/**  
 * 加密字符串  
 *  
 * @param str 待加密字符串  
 * @return 加密后字符串  
 */  
public String encrypt(String str) throws Exception;  
  
/**  
 * 解密字符串  
 *  
 * @param str 待解密字符串  
 * @return 解密后字符串  
 *  
 */  
public String decrypt(String str) throws Exception;
```

13.1.6 DefaultEncryptEngine

demsdk.jar 中提供了一个默认的加解密引擎 com.dameng.dem.server.util.DefaultEncryptEngine。DefaultEncryptEngine 是 IEncryptEngine 接口的实现类。

13.2 应用举例

13.2.1 db.xml 相关密码字段加密

在 db.xml 中需要配置连接 DEM 后台数据库的密码信息，详细参见 [3.1.3 db.xml 配置参数说明](#) 章节的 password 和 SSLPassword 参数配置说明。通常密码信息需要以密文形式呈现，该配置文件由用户配置，需要由用户将相关密码信息生成密文，该密码信息被 DEM 读取后用于连接 DEM 后台数据库，因此 DEM 需要对其进行解密，故用户需告知 DEM 解密方法，或者用户可以使用 DEM 提供的默认加密方法对密码信息加密。

具体包含 2 种方式：

- 使用 DEM 默认加解密引擎

DEM 提供默认加解密引擎 DefaultEncryptEngine，DEM 使用该加解密引擎中的解密方法去解密从 db.xml 读取到的密码信息，因此用户需要按该加解密引擎中的加密方法对密码信息进行加密，具体步骤如下：

1. 加密密码，生成密文

例如，需要加密的密码为 A:

```
java -cp demsdk.jar com.dameng.dem.server.util.DefaultEncryptEngine encrypt A
```

生成密文 B。

2. 在配置文件 db.xml 中使用密文

配置 db.xml 相关密码字段，以 Password 为例，具体如下：

```
<Password engine="com.dameng.dem.server.util.DefaultEncryptEngine">B
</Password>
```

- 使用自定义的加解密引擎

用户也可以使用自定义加解密引擎对 db.xml 中的密码信息进行加密，DEM 需要按对应的方式去解密读取到的密码信息，因此需要用户提供解密方法，具体步骤如下：

1. 自定义加解密引擎实现

自定义加解密引擎需要实现 demsdk.jar 包的

com.dameng.dem.server.util.IEncryptEngine 接口，具体举例如下：

1) 新建 java 工程，引用 demsdk.jar 包。用户可同时导入 demsdk-sources.jar 文件，方便开发工作。

2) 创建包 com.my.engine;
3) 在包 com.my.engine 下创建 MyEncryptEngine 类，并实现
com.dameng.dem.server.util.IEncryptEngine 接口，即实现 String
encrypt(String str), String decrypt(String str) 方法。

以下代码仅作为参考示例：

```
package com.my.engine;

import com.dameng.dem.server.util.IEncryptEngine;

public class MyEncryptEngine implements IEncryptEngine

{
    /**
     * @param str 待加密字符串
     * @return 加密后字符串
     *
     * 实现 encrypt 方法，对字符串进行加密
     */
    @Override

    public String encrypt(String str) throws Exception
    {
        // 简易示例：通过对字符串添加前缀"A_"进行加密
        return "A_" + str;
    }

    /**
     * @param str 待解密加密字符串
     * @return 解密后字符串
     *
     * 实现 decrypt 方法，对字符串进行解密
     */
    @Override

    public String decrypt(String str) throws Exception
```

```

{
    // 简易示例：对应本示例的encrypt方法，去除密文的前两个字符"A_",
    // 还原字符串
    return str.substring(2);
}
}

```

- 4) 将上述 java 工程打包成 jar 文件 myEncryptEngine.jar。jar 文件中只需包含用户的实现类，不必包含 demsdk.jar 文件。

2. 配置自定义加解密引擎

加解密引擎实现后，需要在 DEM 中配置相关信息，供 DEM 使用，具体步骤如下：

- 1) 将 myEncryptEngine.jar 放入 DEM 的 WEB-INF/lib 目录下。WEB-INF/lib 目录已包含 demsdk.jar，不用再放入 demsdk.jar。
- 2) 配置 db.xml 相关密码字段，以 Password 为列，具体如下：
<Password engine="com.my.engine.MyEncryptEngine">加密后的密码</Password>
- 3) 重启 DEM 应用。

13.2.2 自定义推送

DEM 中的告警通知等消息可以通过自定义推送方式推送给用户（具体配置参见 7.2.3 创建告警配置章节），具体自定义推送方式需要用户自己实现，如下：

- **自定义推送类实现**

- 自定义推送方式需要实现 demsdk.jar 包的 com.dameng.dem.server.util.INotify 接口，具体举例如下：
1. 新建 java 工程，引用 demsdk.jar 包。用户可同时导入 demsdk-sources.jar 文件，方便开发工作。
 2. 创建包 com.my.notify；
 3. 在包 com.my.notify 下创建 MyNotify 类，实现 com.dameng.dem.server.util.INotify 接口，即实现 boolean send(NotifyUser user, Notify notify) 方法。

以下代码仅作为参考示例：

```
package com.my.notify;
```

```

import com.dameng.dem.server.util.INotify;
import com.dameng.dem.server.util.Notify;
import com.dameng.dem.server.util.NotifyUser;

public class MyNotify implements INotify
{
    public static final String LINE_SEPARATOR =
        System.getProperty("line.separator");

    /**
     * @param user 通知接收者
     * @param notify 通知信息封装
     *
     * @return 信息发送成功,返回true,信息发送失败,返回false
     *
     * @see
     */
    com.dameng.dem.server.util.INotify#send(com.dameng.dem.server.util.NotifyUser,
        com.dameng.dem.server.util.Notify)

    @Override
    public boolean send(NotifyUser user, Notify notify)
    {
        // 获取user对象数据, 拼接用户信息
        StringBuilder userContent = new StringBuilder(256);
        userContent.append("用户ID: " + user.id);
        userContent.append(LINE_SEPARATOR);
        userContent.append("用户登录名: " + user.loginName);
        userContent.append(LINE_SEPARATOR);
        userContent.append("用户真实名: " + user.realName);
        userContent.append(LINE_SEPARATOR);
    }
}

```

```
userContent.append("用户手机号: " + user.phone);

userContent.append(LINE_SEPARATOR);

userContent.append("用户邮件地址: " + user.email);

// 打印user相关信息

System.out.println(userContent.toString());

System.out.println(LINE_SEPARATOR);

System.out.println(LINE_SEPARATOR);

// 获取notify对象数据，拼接警告消息

StringBuilder notifyContent = new StringBuilder(256);

notifyContent.append("警告ID: " + notify.id);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("异常发现时间: " + notify.gmtHappen);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告生成时间: " + notify.gmtCreate);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告所属告警配置的名称: " + notify.alertName);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告所属的规则的名称: " + notify.ruleName);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("资源类型: " + notify.resType);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告所属的资源的名称: " + notify.resName);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告级别: " + notify.level);

notifyContent.append(LINE_SEPARATOR);

notifyContent.append("警告具体内容: " + notify.message);

// 打印notify相关信息
```

```
System.out.println(notifyContent.toString());  
  
    // 上述代码为打印user和notify对象包含的信息  
    // 用户可根据自身需求，将用户和警告相关信息发送至第三方系统  
  
    // TODO  
  
    return false;  
}  
}
```

上述自定义推送类 MyNotify 实现了将 NotifyUser user 和 Notify notify 对象信息打印至控制台。

用户也可获取用户和警告相关信息，并根据第三方系统接口的要求，将信息发送至第三方系统。

4. 将上述 java 工程打包成 jar 文件 myNotify.jar。jar 文件中只需包含用户的实现类，不必包含 demsdk.jar。

● 配置自定义推送

自定义推送实现后，需要在 DEM 中配置相关信息，供 DEM 使用，具体步骤如下：

1. 将 myNotify.jar 放入 DEM 的 WEB-INF/lib 目录下。WEB-INF/lib 目录已包含 demsdk.jar，不用再放入 demsdk.jar。

2. 启动 DEM 后，访问 DEM 页面，在“系统配置”页面中配置“自定义推送配置”相关，具体包括：

- 1) 配置“custom_notify_enable”为 true，开启自定义推送功能；
- 2) 配置“custom_notify_class”为 com.my.notify.MyNotify。

配置好后，可以在“系统配置”页面的工具栏点击“自定义推送测试”按钮，测试自定义推送功能。

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