



银河麒麟限定域优化程序 用户手册 v1.1

版本:2025/03/07

表 1 手册更新说明

日期	说明
2024-10-25	初版；
2024-11-19	修改文档版本为 v1.0；
2025-03-07	编制 v1.1 版本文档；

前言

欢迎使用 [银河麒麟限定域优化程序]！本手册旨在为您提供使用指南和操作说明，帮助您充分了解和我们的软件。在开始使用之前，请先阅读本手册以获取必要的信息和指导。

目标读者：

本手册适用于 [银河麒麟限定域优化程序] 的所有用户，包括新用户、有一定经验的用户和管理员。无论您是第一次接触还是已经熟悉我们的软件，本手册都将为您提供有用的信息和操作建议。

目的：

本手册的目的是帮助用户更好地理解和使用 [银河麒麟限定域优化程序]。它包含了详细的操作指南、配置说明、故障排除和其他相关信息，旨在提供全面的支持，确保您能够充分利用我们软件的功能。

帮助与支持：

如果您在使用过程中遇到任何问题或需要进一步的帮助，请联系我们的技术支持团队。

版权信息：

版权所有 © [麒麟软件有限公司]。保留所有权利。

感谢您选择使用 [银河麒麟限定域优化程序]。请继续阅读本手册，以获得详细的操作指南和其他有用的信息。如果您有任何疑问或建议，请随时与我们联系。祝您使用愉快！

目录

前言3

术语和定义5

第 1 章 软件介绍5

 1.1. 功能概述5

 服务程序 5

 客户端5

第 2 章 部署环境6

 2.1. 安装环境 6

 3.1. 在线安装 7

 3.2. 离线安装 7

第 3 章 使用说明8

 6.1. 服务程序 8

 6.2. 客户端9

术语和定义

3.1

限定域优化 limit domain optimization techniques

用于操作系统应用程序优化的技术，保障对目标进程优化同时，在时间域、空间域上尽可能避免对其他进程的影响。

第 1 章 软件介绍

本软件产品全称为“银河麒麟限定域优化程序”，以下简称“限定域优化软件”，是部署在服务器上的一个性能优化程序；满足该规范规定的总体架构、功能、性能、兼容性、安全性方面的技术要求。

1.1. 功能概述

限定域优化软件在功能上包括两部分：客户端和服务程序；

服务程序

服务程序主要包括功能如下：

- 1) 主控程序模块，接收客户端操作请求，并调用对应的模块处理。
- 2) 方案管理模块，解析客户端的方案并存储数据库。
- 3) 方案实施模块，依照要求把优化方案的优化方法应用到操作系统上，或还原为原来设置。

客户端

客户端用于配置服务程序，包括添加优化方案、设置优化方案、查询优化方案状态。

第 2 章 部署环境

2.1. 安装环境

限定域优化软件目前已适配支持“银河麒麟高级服务器操作系统 V11 ”，并且分别支持 x86_64 平台、aarch64 平台、loongarch 平台，具体适配情况如下：

产品名称	对应架构	限定域优化软件适配
银河麒麟高级服务器操作系统 V11 2503	x86_64;aarch64;loongarch;	支持

安装软件

本章节介绍限定域优化软件的安装方式，限定域优化软件以 RPM 包的形式提供，其安装方式可以通过“在线安装”或“离线安装”的形式。“在线安装”指的是安装环境中 yum 源可用，可以为远程仓库，或者本地搭建的私有 yum 仓库源；当没有可用的 yum 源仓库时，可以通过离线的方式安装，需要提前下载好依赖的软件包。

3.1. 在线安装

下面银河麒麟高级服务器操作系统 V11 版本为例，在 x86 平台 演示其安装过程。

软件包: (下面 a.b.c-d 表示版本号，安装时请替换为实际版本。)

```
kyldop-a.b.c-d.ky11.x86_64.rpm
python3-kyldop-a.b.c-d.ky11.x86_64.rpm
kyldop-tools-a.b.c-d.ky11.x86_64.rpm
```

使用以下指令安装：

```
# yum install kyldop python3-kyldop kyldop-tools
```

3.2. 离线安装

如果没有可以访问的软件源，则需要离线安装 3 个软件包：

安装指令如下：（下面 a.b.c-d 表示版本号，安装时请替换为实际版本。）

```
# rpm -ivh kyldop-a.b.c-d.ky11.x86_64.rpm python3-kyldop-a.b.c-d.ky11.x86_64.rpm kyldop-tools-a.b.c-d.ky11.x86_64.rpm
```

安装效果如图所示：

```
[root@v11b07 x86_64]# rpm -ivh kyldop-0.1.0-1.ky11.x86_64.rpm python3-kyldop-0.1.0-1.ky11.x86_64.rpm kyldop-tools-0.1.0-1.ky11.x86_64.rpm
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
 1:python3-kyldop-1:0.1.0-1.ky11 ##### [ 33%]
 2:kyldop-1:0.1.0-1.ky11 ##### [ 67%]
Created symlink /etc/systemd/system/multi-user.target.wants/kyldop.service → /usr/lib/systemd/system/kyldop.service.
 3:kyldop-tools-1:0.1.0-1.ky11 ##### [100%]
/sbin/ldconfig: /lib/libactivation.so.1 is not a symbolic link
```

图 3-2 限定域优化软件安装过程

第 3 章 使用说明

6.1. 服务程序

1. 服务默认状态

(1) 安装限定域优化软件后，程序服务默认状态是关闭的。

```
[root@v11b07 ~]# systemctl status kyldop.service
● kyldop.service - kyldopd daemon
   Loaded: loaded (/usr/lib/systemd/system/kyldop.service; disabled; preset: disabled)
   Active: inactive (dead)
```

2. 启动服务程序

(1) 执行 `systemctl start kyldop.service` 命令，启动服务端。

```
[root@v11b07 ~]# systemctl start kyldop.service
[root@v11b07 ~]#
```

(2) 执行 `systemctl status kyldop.service` 命令查看服务状态。

```
[root@v11b07 ~]# systemctl status kyldop.service
● kyldop.service - kyldopd daemon
   Loaded: loaded (/usr/lib/systemd/system/kyldop.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-10-25 18:58:06 CST; 2s ago
     Main PID: 3033 (kyldopd)
       Tasks: 4 (limit: 21092)
      Memory: 18.7M ()
      CGroup: /system.slice/kyldop.service
              └─3033 /usr/bin/python3 /usr/sbin/kyldopd

10月 25 18:58:07 v11b07 kyldopd[3033]: manager _update_from_db:236 [DEBUG] update prof
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: source check_comms_use_netlink_rcv:370 [DEBUG]
10月 25 18:58:07 v11b07 kyldopd[3033]: manager start:719 [INFO] apply profile unixbenc
10月 25 18:58:07 v11b07 kyldopd[3033]: manager recovery:915 [INFO] recovery get enable
10月 25 18:58:07 v11b07 kyldopd[3033]: service run:124 [INFO] main loop run
```

3. 停止服务程序

(1) 执行 `systemctl stop kyldop.service` 命令，关闭服务程序。

```
[root@v11b07 ~]# systemctl stop kyldop.service
[root@v11b07 ~]#
```

(2) 执行 `systemctl status kyldop.service` 命令查看服务状态。

```
[root@v11b07 ~]# systemctl status kyldop.service
○ kyldop.service - kyldopd daemon
   Loaded: loaded (/usr/lib/systemd/system/kyldop.service; enabled; preset: disabled)
   Active: inactive (dead) since Fri 2024-10-25 19:24:16 CST; 1s ago
     Duration: 3min 28.963s
    Process: 4185 ExecStart=/usr/sbin/kyldopd (code=exited, status=0/SUCCESS)
   Main PID: 4185 (code=exited, status=0/SUCCESS)

10月 25 19:24:15 v11b07 systemd[4390]: [systemctl start irqbalance] called by PID 4185 (/
10月 25 19:24:15 v11b07 kyldopd[4185]: source check_comms_use_netlink_recv:370 [DEBUG] [Exc
10月 25 19:24:15 v11b07 kyldopd[4185]: method _service_set_status:190 [DEBUG] Set Service t
10月 25 19:24:16 v11b07 kyldopd[4185]: manager _update_from_db:236 [DEBUG] update profile s
10月 25 19:24:16 v11b07 kyldopd[4185]: manager _update_apply_state:465 [DEBUG] update unapp
10月 25 19:24:16 v11b07 kyldopd[4185]: manager stop:747 [INFO] unapply profile unixbench优化
10月 25 19:24:16 v11b07 kyldopd[4185]: service exit:136 [INFO] exit
10月 25 19:24:16 v11b07 kyldopd[4185]: kyldopd main:50 [INFO] exit
10月 25 19:24:16 v11b07 systemd[1]: kyldop.service: Deactivated successfully.
10月 25 19:24:16 v11b07 systemd[1]: Stopped kyldopd daemon.
```

6.2. 客户端

1. 查看已添加的优化方案

执行 `kyldop query`，查看已添加方案如下图所示。

```
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "e07e2a94fa5f11ef922e080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": true,
    "data": "",
    "state": false,
    "priority": 30,
    "exclusive": null,
    "create_time": "2025-03-06 15:51:56.697688",
    "update_time": "2025-03-06 15:51:56.697692"
  }
]
```

图 6-21 方案列表

2. 查看正在使用的优化方案

执行 `kyldop state`，查看当前正在使用方案如下图所示。

```
[root@b18 docs]# kyldop state
{
  "apply profile name": "",
  "active profile name": ""
}
```

3. 添加优化方案

执行 `kyldop add -n demo xxx.yaml`，以加密方式添加优化方案如下图所示。

```
[root@b18 docs]# kyldop add -n demo v11-scene-demo.yaml
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "d337c4d4fa6511ef9436080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": true,
    "data": "",
    "state": false,
    "priority": 30,
    "exclusive": null,
    "create_time": "2025-03-06 16:34:31.406327",
    "update_time": "2025-03-06 16:34:31.406333"
  }
]
```

执行 `kyldop add xxx.yaml --cipher=0`，以明文方式添加优化方案如下图所示。

```
[root@b18 docs]# kyldop add -n demo v11-scene-demo.yaml --cipher=0
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "10e96e04fa6611ef9436080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": false,
    "data": "{\"identifier\": \"59cbb503-09fd-4f37-8ed9-c7bd3945dd33\",
rch64\", \"os\": \"V11\", \"priority\": 50, \"exclusive\": \"FALSE\", \
n\"}, \"rules\": [{\"rule\": {\"preset\": \"\", \"methods\": [{\"method
{\"method\": {\"type\": \"bind_cpus\", \"key\": \"cpus\", \"value\": \"
\"type\": \"sysctl\", \"key\": \"vm.swappiness\", \"value\": \"11\"}},
o\", \"value\": \"31\"}}]}]}]\",
    \"state\": false,
    \"priority\": 30,
    \"exclusive\": null,
    \"create_time\": \"2025-03-06 16:36:14.911661\",
    \"update_time\": \"2025-03-06 16:36:14.911667\"
  }
]
```

4. 删除优化方案

执行“ kyldop delete test 优化方案” ，以不加密方式添加优化方案如下图所示。

```
[root@b18 docs]# kyldop delete -n demo
1 profile removed!
```

5. 启动优化方案

执行“ kyldop query” ，先查看已添加的方案，如下图所示。

```
[root@b18 docs]# kyldop add -n demo v11-scene-demo.yaml
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "d337c4d4fa6511ef9436080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": true,
    "data": "",
    "state": false,
    "priority": 30,
    "exclusive": null,
    "create_time": "2025-03-06 16:34:31.406327",
    "update_time": "2025-03-06 16:34:31.406333"
  }
]
```

执行“kyldop apply -n demo”，启动“demo”优化方案如下图所示。

```
[root@b18 docs]# kyldop state
{
  "apply profile name": "",
  "active profile name": ""
}
[root@b18 docs]# kyldop apply -n demo
[root@b18 docs]# kyldop state
{
  "apply profile name": "demo",
  "active profile name": ""
}
```

6. 关闭优化方案

执行“kyldop unapply -n demo”，关闭“demo”优化方案如下图所示。

```
[root@b18 docs]# kyldop unapply -n demo
[root@b18 docs]#
[root@b18 docs]#
[root@b18 docs]# kyldop state
{
  "apply profile name": "",
  "active profile name": ""
}
```

7. 设置优化方案默认启动

执行“kyldop enable -n demo”，设置“demo”方案随 kyldop 服务启动，如下图所示。

```
[root@b18 docs]# kyldop enable -n demo
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "8d8a950afa6611ef9436080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": true,
    "data": "",
    "state": true,
    "priority": 30,
    "exclusive": null,
    "create_time": "2025-03-06 16:39:44.005272",
    "update_time": "2025-03-06 16:39:44.005277"
  }
]
```

8. 关闭优化方案默认启动

执行“kyldop disable -n demo”，关闭“demo”方案不随 kyldop 启动，执行 kyldop reload 或者重启 kyldop 服务生效，如下图所示。

```
[root@b18 docs]# kyldop disable -n demo
[root@b18 docs]# kyldop query
Profile found:
[
  {
    "uuid": "8d8a950afa6611ef9436080027d1b6b0",
    "name": "demo",
    "arch": "null",
    "os": "null",
    "scene": "null",
    "flags": true,
    "data": "",
    "state": false,
    "priority": 30,
    "exclusive": null,
    "create_time": "2025-03-06 16:39:44.005272",
    "update_time": "2025-03-06 16:39:44.005277"
  }
]
```

9. 查看正在使用的优化方案

执行” kyldop state” ，查看已启动的优化方案如下图所示。

```
[root@b18 docs]# kyldop state
{
  "apply profile name": "demo",
  "active profile name": ""
}
```

10. 重载优化方案

执行” kyldop reload” ，可以重新生效当前的优化方案，如果当前无应用方案，则启动 enable 状态方案。

```
[root@b18 docs]# kyldop reload
[root@b18 docs]# kyldop state
{
  "apply profile name": "demo",
  "active profile name": ""
}
```