



银河麒麟云底座操作系统 V10
一键式性能调优工具使用手册

麒麟软件有限公司

2024 年 7 月

版权所有 © 2014-2024 麒麟软件有限公司，保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他麒麟商标均为麒麟软件有限公司的商标。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受麒麟软件有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，麒麟软件有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容有可能变更，麒麟软件有限公司保留在没有任何通知或提示的情况下对内容进行修改的权利。除非另有约定，本文档仅作为使用指导，并不确保手册内容完全没有错误。本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1. extuner 介绍	1
1.1. 简介	1
1.1. 使用场景	1
1.2. 功能说明	2
1.2.1. 采集功能 (collection)	2
1.2.2. 分析功能 (analysis)	2
1.2.3. 静态调优功能 (profile)	2
1.3. 特性说明	2
2. 安装与使用	3
2.1. 使用说明	3
2.2. extuner collection	3
2.2.1. 采集基础性能数据	5
2.2.2. 采集 perf 数据	7
2.3. extuner analysis	7
2.4. extuner profile	9

1. extuner 介绍

1.1. 简介

为了辅助用户快速定位和处理操作系统性能问题，Kylin 开发一款轻量级系统调优指导工具 Extuner。Extuner 包含对操作系统的全量式数据采集功能、系统数据瓶颈分析功能、场景化自动调优功能。Extuner 简化了用户调优操作系统性能的工作，让系统发挥出色性能。

Extuner 是一款使用 python 语言开发，面向 Linux 平台的轻量级系统调优指导工具。工具能够对操作系统实时数据进行采集、处理等操作后得出分析报告。报告中包含系统数据、异常指标、调优指导等内容，使用者可以通过报告了解系统当前状态及瓶颈，从而根据指导内容，有针对性的对系统进行调优操作。

1.1. 使用场景

Extuner 工具使用场景比较丰富，推荐以下场景使用：

1. 客户现场应用调优
2. 应用性能 pk
3. 应用性能攻关
4. 查看性能瓶颈
5. 异常数据采集
6. Centos 替换前后的性能数据收集

1.2. 功能说明

Extuner 工具提供的功能，从以下三小节进行简要介绍。

1.2.1. 采集功能 (collection)

Extuner 采集功能可以一键式采集操作系统 CPU、内存、磁盘、网络、系统日志等资源的运行情况

1. 支持收集基础性能数据，包含 CPU、内存、网络、IO 等信息；
2. 支持 perf 采集热点数据；
3. 支持获取系统日志。

1.2.2. 分析功能 (analysis)

1. 支持 collection 功能的全部数据采集；
2. 采集数据中的异常数值进行高亮显示；
3. 智能识别系统瓶颈信息；
4. 智能输出当前系统调优建议。

1.2.3. 静态调优功能 (profile)

内置调优场景的调优参数下发。

1.3. 特性说明

工具提供 rpm 安装方式，一条命令集成所有功能；

1. 采集和下发功能均可选配；
2. 支持一键获取全量信息、方便易用；

3. 采集的数据不改变命令原有格式；
4. 支持调优方法一次配置下发；
5. 工具本身不影响原有系统的稳定性、兼容性；
6. 工具仅支持在麒麟操作系统上运行；
7. 工具本身脱敏处理，不泄露源码。

2. 安装与使用

2.1. 使用说明

(1) 安装依赖包

```
# yum install blktrace python3-netifaces python3-paramiko pcp-system-tools numactl
```

(2) 安装 rpm 包

```
# yum install extuner
```

(3) 运行 extuner

数据采集功能

```
# extuner collection [options]
```

数据分析功能

```
# extuner analysis [options]
```

2.2. extuner collection

全量采集系统性能数据，包括 CPU、内存、磁盘、网络、系统日志、综合数据等

extuner collection 命令参数说明

```
[root@localhost ~]# extuner collection -h
-----
collect system data, such as CPU/MEM/NET/IO

Usage:
  extuner collection [options]

Example:
  extuner collection --perf --pid -1

Options:
  -h, --help                show this help message and exit
  -o <path>, --out_path=<path> Generate extunerData/, include log and reports
                              default in /usr/share/extuner/extunerData/
  --perf [PERF_ENABLE_TYPE] collect perf data,
                              default is disabled if not specified
  --offcpu                  collect offcpu data,
                              default is disabled if not specified

Perf required parameters:
  --pid PERF_PID            process id to profile,
                              -1 to profile the whole system
  --duration VALUE         duration of perf trace, in seconds

OffCPU required parameters:
  --offcpu_pid OFFCPU_PID  process id to profile
  --offcpu_duration VALUE  duration of perf trace, in seconds
[root@localhost ~]#
```

`-o <path>, --out_path=<path>`: 指定采集数据存储路径, 默认存放在 `/usr/share/extuner/extunerData` 下。

`--perf [PERF_ENABLE_TYPE]`: 收集 perf data 类型, `PERF_ENABLE_TYPE` 为 1 采集 perf 函数报告信息, `PERF_ENABLE_TYPE` 为 2 采集 perf 指定事件统计信息

`--offcpu`: 是否采集 offcpu 数据

`--pid PERF_PID`: 需要进行 perf 采集的进程号, 与 `--perf` 搭配使用

`--duration VALUE`: 采集持续时间, 默认为 15s, 与 `--perf` 搭配使用

`--offcpu_pid OFFCPU_PID`: 需要进行 offcpu 采集的进程号, 与 `--offcpu` 搭配使用

`--offcpu_duration VALUE`: offcpu 采集持续时间, 默认为 15s, 与 `--offcpu` 搭配使用

2.2.1. 采集基础性能数据

(1) 功能介绍

基础性能数据采集功能是对 CPU、内存、网络、磁盘、系统参数、综合数据、系统日志的实时数据采集。

(2) 使用说明

Step1: 修改/usr/share/extuner/extuner.conf 配置文件 Common 部分的参数。

参数说明:

Common: 采集基础性能数据

- enable: 0 禁用, 1 开启, 默认开启
- CPU: 控制 CPU 相关数据采集
 - interval: 采集间隔, 以秒为单位, 默认间隔 3s
 - times: 采集次数, 默认采集 5 次
- Memory: 控制内存相关数据采集
 - interval: 采集间隔, 以秒为单位, 默认间隔 3s
 - times: 采集次数, 默认采集 5 次
- Net: 控制网络相关数据采集
 - interval: 采集间隔, 以秒为单位, 默认间隔 3s
 - times: 采集次数, 默认采集 5 次
- Disk: 控制磁盘 I/O 相关数据采集
 - interval: 采集间隔, 以秒为单位, 默认间隔 3s

- times: 采集次数, 默认采集 5 次
- enable_blktrace: 控制是否开启 blktrace 采集, 0 为禁用, 1 为开启, 默认关闭 blktrace 采集
- seconds: blktrace 采集时间, 默认采集 10s
- dev: 指定 blktrace 采集的块。多个块可填写为” /dev/sda,/dev/sdb”

Step2: 运行 extuner

extuner collection

(3) 输出结果

当配置文件 (extuner.conf) 中设置 Common 下的 enable 参数为 1 时, 运行工具会得到基础性能数据, 并将所得内容以文本的形式保存在 output 目录下。

数据采集完成后, 工具会自动抓取文本 (txt 文件) 中的内容, 进行二次处理后生成可视化报告 (html 文件), 报告样式如下图所示:



Kylin 性能数据报告

基本信息

主机名称:	localhost.localdomain
内核版本:	5.10.0-182.8.v2406.ky10h.x86_64
CPU型号:	Intel Xeon Processor (Cooperlake)
CPU架构:	x86_64
CPU核心数:	4
内存总大小:	2.58 GB
内存空闲:	1.07 GB
GCC版本:	gcc (GCC) 10.3.1
Glibc版本:	glibc 2.34
JDK版本:	/bin/sh: line 1: java: command not found
网络1:	ens3 192.168.122.47
磁盘1:	SDA (QEMU HARDDISK) 20G

2.2.2. 采集 perf 数据

(1) 功能介绍

可选择是否开启，可采集 perf record/perf report 和 perf stat 数据。

(2) 使用说明

方法 1: 命令行执行 `extuner collection --perf --pid *`

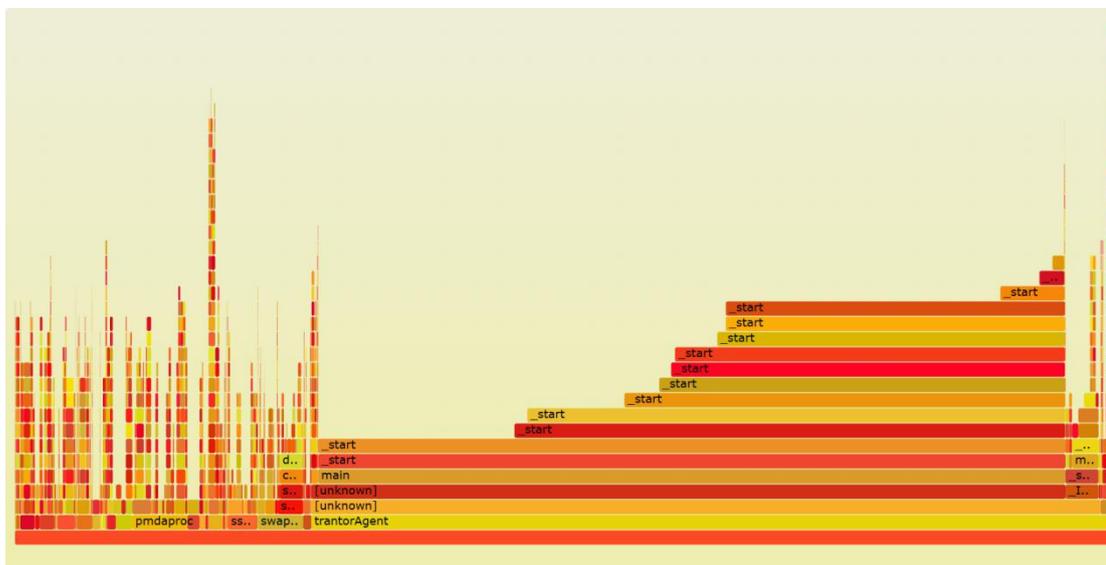
*为 pid 号

方法 2: 修改配置文件 `extuner.conf` 中 perf 的配置，然后执行 `extuner collection`

(3) 输出结果

perf 采集完成后在报告 (html) 的热点函数栏中。

perf.svg 图如下



2.3. extuner analysis

在采集全量数据的基础上，进行瓶颈数据分析，提供调优建议。

extuner analysis 参数说明:

```
[root@localhost ~]# extuner analysis -h
-----
collect system data, analysis performance and provide tuning advice

Usage:
  extuner analysis [options]

Example:
  extuner analysis --perf --pid -1

Options:
  -h, --help                show this help message and exit
  -o <path>, --out_path=<path>  Generate extunerData/, include log and reports
                                default in /usr/share/extuner/extunerData/
  --test                    test mode. Default is False. Only developer use
  --perf [PERF_ENABLE_TYPE] collect perf data,
                                default is disabled if not specified
  --offcpu                  collect offcpu data,
                                default is disabled if not specified

Perf required parameters:
  --pid PERF_PID            process id to profile,
                                -1 to profile the whole system
  --duration VALUE         duration of perf trace, in seconds

OffCPU required parameters:
  --offcpu_pid OFFCPU_PID  process id to profile
  --offcpu_duration VALUE  duration of perf trace, in seconds
[root@localhost ~]#
```

`-o <path>, --out_path=<path>`: 指定采集数据存储路径，默认存放在 `/usr/share/extuner/extunerData` 下。

`--perf [PERF_ENABLE_TYPE]`: 收集 perf data 类型, `PERF_ENABLE_TYPE` 为 1 采集 perf 函数报告信息, `PERF_ENABLE_TYPE` 为 2 采集 perf 指定事件统计信息

`--offcpu`: 是否采集 offcpu 数据

`--pid PERF_PID`: 需要进行 perf 采集的进程号, 与 `--perf` 搭配使用

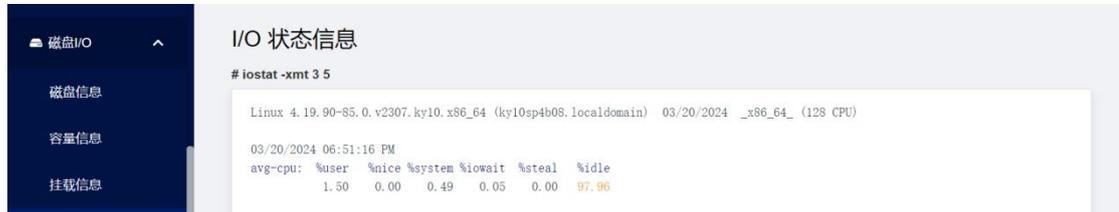
`--duration VALUE`: 采集持续时间, 默认为 15s, 与 `--perf` 搭配使用

`--offcpu_pid OFFCPU_PID`: 需要进行 offcpu 采集的进程号, 与 `--offcpu` 搭配使用

`--offcpu_duration VALUE`: offcpu 采集持续时间, 默认为 15s, 与 `--offcpu` 搭配使用

报告输出展示

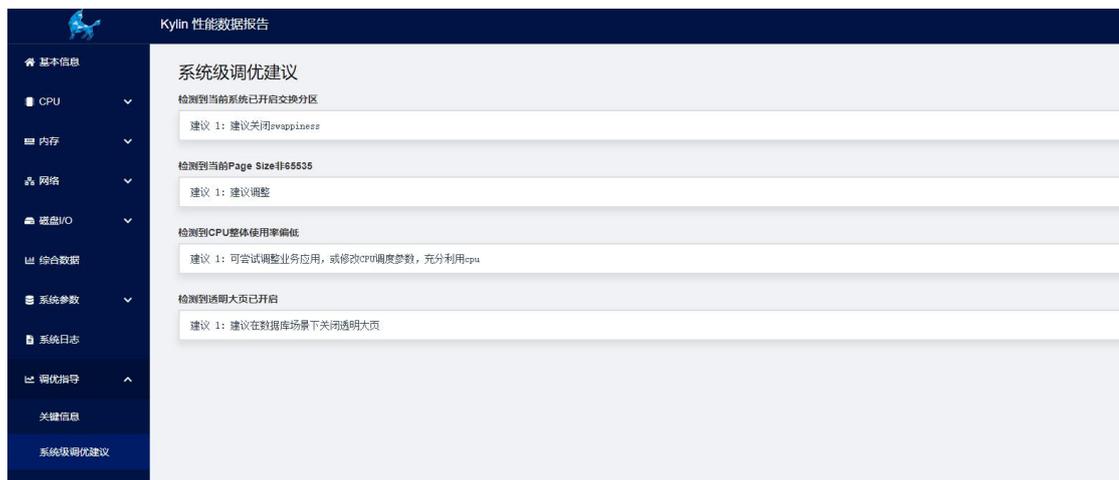
● 关键指标数据高亮展示



● 关键信息



● 系统级调优建议



2.4. extuner profile

可一键下发指定场景的调优配置, 当前支持 mysql、kafka、ats(apache traffic server)、specweb_config、dm 数据库场景。

extuner profile 参数说明:

```
[root@localhost ~]# extuner profile -h
-----
activate the preset scene profile

Usage:
  extuner profile COMMAND [options]

Examples:
  extuner profile mysql_config /etc/
  extuner profile recover
  extuner profile list

Commands:
  mysql_config CONF_DIR      MySQL scene set, CONF_DIR is MySQL conf path
  kafka_config CONF_DIR     Kafka scene set, CONF_DIR is Kafka conf path
  ats_config CONF_DIR       ATS scene set, CONF_DIR is ATS conf path
  specweb_config CONF_DIR   SPECweb scene set, CONF_DIR is apache conf path
  dm_config CONF_DIR        DM scene set, CONF_DIR is DM conf path
  recover                    recover
  list                       list all profile

Options:
  -h, --help                show this help message and exit
  -o <path>, --out_path=<path> Generate extunerData/, include log and reports
                              default in /usr/share/extuner/extunerData/

[root@localhost ~]#
```

mysql_config: mysql 场景

kafka_config: kafka 场景

ats_config: ats 场景

specweb_config: specweb 场景

dm_config: dm 数据库场景

CONF_DIR: 配置文件路径

recover: 备份还原

list: 列出支持的场景

注: 仅在第一次运行 extuner profile COMMAND 时进行环境数据备份, 之后运行 extuner profile 不再备份,

还原则只会还原第一次 extuner profile COMMAND 之前的环境

2.使用方法

以 mysql 场景为例

extuner profile mysql_config /etc

```
[root@localhost ~]# extuner profile mysql_config /etc
2024-04-18 15:44:22,039 - INFO - main.py:13 - Extuner 开始执行
2024-04-18 15:44:22,043 - INFO - main.py:13 - scene parameters for MySQL set success, please restart mysql.service to take effect.
2024-04-18 15:44:22,043 - INFO - main.py:13 - Sysctl parameters set success.
2024-04-18 15:44:22,047 - INFO - main.py:13 - Extuner 执行结束
[root@localhost ~]#
```

还原配置

```
[root@localhost ~]# extuner profile recover
2024-04-18 15:45:40,459 - INFO - main.py:13 - Extuner 开始执行
2024-04-18 15:45:40,459 - INFO - main.py:13 - scene parameters restore success, please restart mysql to take effect.
2024-04-18 15:45:40,460 - INFO - main.py:13 - scene parameters for sysctl clear successfully.
2024-04-18 15:45:40,460 - INFO - main.py:13 - Extuner 执行结束
[root@localhost ~]#
```