openEuler
20.03 LTS

A-Tune User Guide

Date 2020-04-01

Contents

Legal Statement

Trademarks and Permissions

A-Tune and openEuler are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

This document is used only as a guide. Unless otherwise specified by applicable laws or agreed by both parties in written form, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, including but not limited to non-infringement, timeliness, and specific purposes.

Preface

Overview

This document describes how to install and use A-Tune, which is a performance self-optimization software for openEuler.

Intended Audience

This document is intended for developers, open-source enthusiasts, and partners who use the openEuler system and want to know and use A-Tune. You need to have basic knowledge of the Linux OS.

Symbol Conventions

The symbols that may be found in this document are defined as follows:

Symbol	Description
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

1 Getting to Know A-Tune

1.1 Introduction

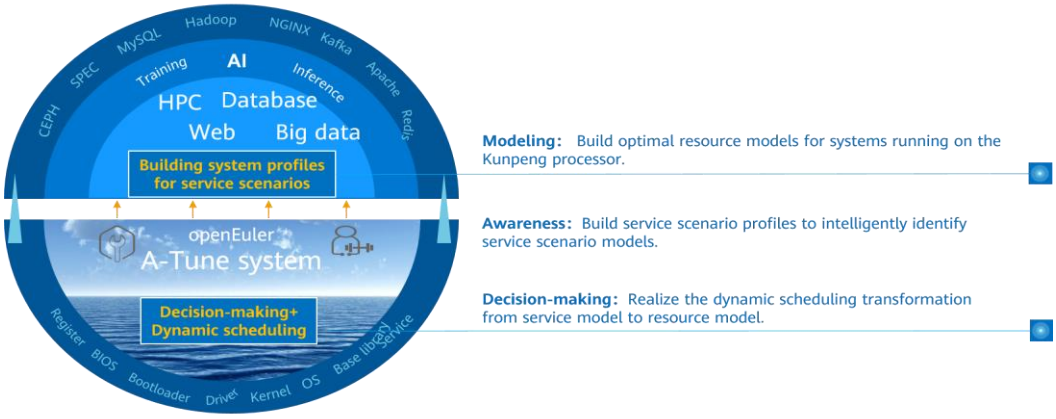
An operating system (OS) is basic software that connects applications and hardware. It is critical for users to adjust OS and application configurations and make full use of software and hardware capabilities to achieve optimal service performance. However, numerous workload types and varied applications run on the OS, and the requirements on resources are different. Currently, the application environment composed of hardware and software involves more than 7000 configuration objects. As the service complexity and optimization objects increase, the time cost for optimization increases exponentially. As a result, optimization efficiency decreases sharply. Optimization becomes complex and brings great challenges to users.

Second, as infrastructure software, the OS provides a large number of software and hardware management capabilities. The capability required varies in different scenarios. Therefore, capabilities need to be enabled or disabled depending on scenarios, and a combination of capabilities will maximize the optimal performance of applications.

In addition, the actual business embraces hundreds and thousands of scenarios, and each scenario involves a wide variety of hardware configurations for computing, network, and storage. The lab cannot list all applications, business scenarios, and hardware combinations.

To address the preceding challenges, openEuler launches A-Tune.

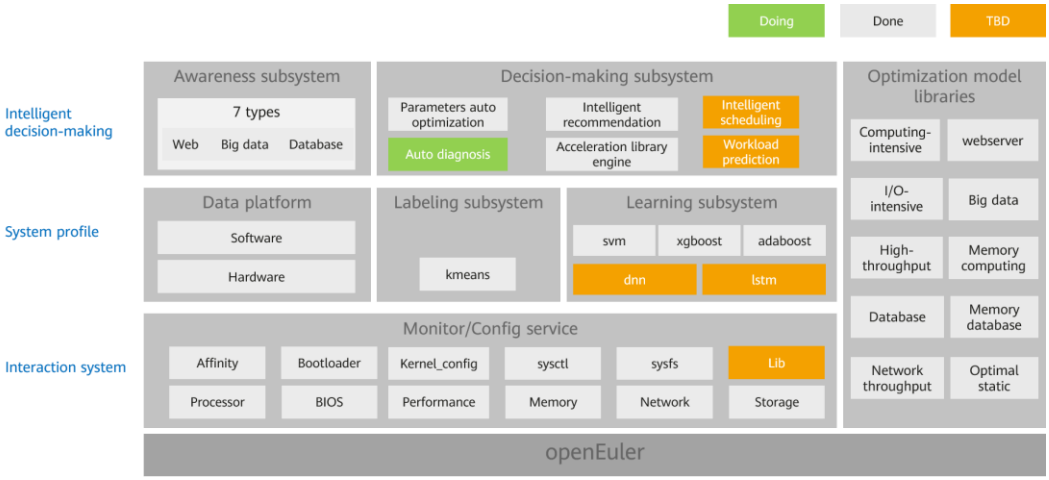
A-Tune is an AI-based engine that optimizes system performance. It uses AI technologies to precisely profile business scenarios, discover and infer business characteristics, so as to make intelligent decisions, match with the optimal system parameter configuration combination, and give recommendations, ensuring the optimal business running status.



1.2 Architecture

The following figure shows the A-Tune core technical architecture, which consists of intelligent decision-making, system profile, and interaction system.

- Intelligent decision-making layer: consists of the awareness and decision-making subsystems, which implements intelligent awareness of applications and system optimization decision-making, respectively.
- System profile layer: consists of the labeling and learning subsystems. The labeling subsystem is used to cluster service models, and the learning subsystem is used to learn and classify service models.
- Interaction system layer: monitors and configures various system resources and executes optimization policies.



1.3 Supported Features and Service Models

Supported Features

Table 1-1 describes the main features supported by A-Tune, feature maturity, and usage suggestions.

Table 1-1 Feature maturity

Feature	Maturity	Usage Suggestion
Auto optimization of 11 applications in seven workload types	Tested	Pilot
User-defined workload types and service models	Tested	Pilot
Automatic parameter optimization	Tested	Pilot

Supported Service Models

Based on the workload characteristics of applications, A-Tune classifies services into seven types. For details about the workload characteristics of each type and the applications supported by A-Tune, see Table 1-2.

Table 1-2 Supported workload types and applications

Workload	Type	Workload Characteristic	Supported Application
default	Default type	The usage of CPU, memory bandwidth, network, and I/O resources is low.	N/A
webserver	HTTPS application	The CPU usage is high.	Nginx
big_database	Database	• Relational database Read: The usage of CPU, memory bandwidth, and network is high. Write: The usage of I/O is high. • Non-relational database The usage of CPU and I/O is high.	MongoDB, MySQL, PostgreSQL, and MariaDB
big_data	Big data	The usage of CPU and I/O is high.	Hadoop and Spark
in-memory_computing	Memory-intensive application	The usage of CPU and memory bandwidth is high.	SPECjbb2015
in-memory_database	Computing- and network-	The usage of a single-core CPU is high, and the network usage is high in multi-instance	Redis

Workload	Type	Workload Characteristic	Supported Application
	intensive application	scenarios.	
single_computer_intensive_jobs	Computing-intensive application	The usage of a single-core CPU is high, and the usage of memory bandwidth of some subitems is high.	SPECCPU2006
communication	Network-intensive application	The usage of CPU and network is high.	Dubbo
idle	System in idle state	The system is in idle state and no applications are running.	N/A

2 Installation and Deployment

This chapter describes how to install and deploy A-Tune.

2.1 Software and Hardware Requirements

Hardware Requirement

- Huawei Kunpeng 920 processor

Software Requirement

- OS: openEuler 20.03 LTS

2.2 Environment Preparation

For details about installing an openEuler OS, see *openEuler 20.03 LTS Installation Guide*.

2.3 A-Tune Installation

This chapter describes the installation modes and methods of the A-Tune.

2.3.1 Installation Modes

A-Tune can be installed in single-node or distributed mode.

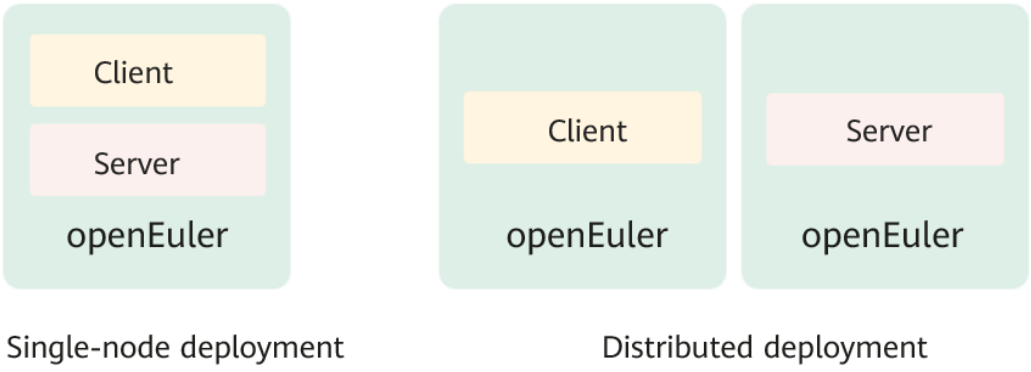
- Single-node mode

The client and server are installed on the same system.

- Distributed mode

  The client and server are installed on different systems.

The installation modes are as follows:



2.3.2 Installation Procedure

To install the A-Tune, perform the following steps:

Step 1 Mount an openEuler ISO file.

```
# mount openEuler-20.03-LTS-aarch64-dvd.iso /mnt
```

Step 2 Configure the local yum source.

```
# vim /etc/yum.repos.d/local.repo
```

The configured contents are as follows:

```
[local]
name=local
baseurl=file:///mnt
gpgcheck=0
enabled=1
```

Step 3 Install an A-Tune server.

NOTE

In this step, both the server and client software packages are installed. For the single-node deployment, skip **Step 4**.

```
# yum install atune -y
```

Step 4 For a distributed mode, install an A-Tune client.

```
# yum install atune-client -y
```

Step 5 Check whether the installation is successful.

```
# rpm -qa | grep atune
atune-client-xxx
atune-db-xxx
atune-xxx
```

If the preceding information is displayed, the installation is successful.

----End

2.4 A-Tune Deployment

This chapter describes how to deploy A-Tune.

2.4.1 Overview

The configuration items in the A-Tune configuration file **/etc/atuned/atuned.cnf** are described as follows:

- A-Tune service startup configuration

  You can modify the parameter value as required.

  - **protocol**: Protocol used by the gRPC service. The value can be **unix** or **tcp**. **unix** indicates the local socket communication mode, and **tcp** indicates the socket listening port mode. The default value is **unix**.
  - **address**: Listening IP address of the gRPC service. The default value is **unix socket**. If the gRPC service is deployed in distributed mode, change the value to the listening IP address.
  - **port**: Listening port of the gRPC server. The value ranges from 0 to 65535. If **protocol** is set to **unix**, you do not need to set this parameter.
  - **rest_port**: Listening port of the system REST service. The value ranges from 0 to 65535.
  - **sample_num**: Number of samples collected when the system executes the analysis process.

- System information

  System is the parameter information required for system optimization. You must modify the parameter information according to the actual situation.

  - **disk**: Disk information to be collected during the analysis process or specified disk during disk optimization.
  - **network**: NIC information to be collected during the analysis process or specified NIC during NIC optimization.
  - **user**: User name used for ulimit optimization. Currently, only the user **root** is supported.
  - **tls**: SSL/TLS certificate verification for the gRPC and HTTP services of A-Tune. This is disabled by default. After TLS is enabled, you need to set the following environment variables before running the **atune-adm** command to communicate with the server:
    - export ATUNE_TLS=yes
    - export ATUNE_CLICERT=<Client certificate path>
  - **tlsservercertfile**: path of the gPRC server certificate.
  - **tlsserverkeyfile**: gPRC server key path.
  - **tlshttpcertfile**: HTTP server certificate path.
  - **tlshttpkeyfile**: HTTP server key path.
  - **tlshttpcacertfile**: CA certificate path of the HTTP server.

- Log information

  Change the log path and level based on the site requirements. By default, the log information is stored in **/var/log/messages**.

- Monitor information

  Hardware information that is collected by default when the system is started.

Example

```
################################### server ##############################
# atuned config
[server]
# the protocol grpc server running on
# ranges: unix or tcp
protocol = unix

# the address that the grpc server to bind to
# default is unix socket /var/run/atuned/atuned.sock
# ranges: /var/run/atuned/atuned.sock or ip
address = /var/run/atuned/atuned.sock

# the atuned grpc listening port, default is 60001
# the port can be set between 0 to 65535 which not be used
port = 60001

# the rest service listening port, default is 8383
# the port can be set between 0 to 65535 which not be used
rest port = 8383

# when run analysis command, the numbers of collected data.
# default is 20
sample num = 20

# Enable gRPC and http server authentication SSL/TLS
# default is false
# tls = true
# tlsservercertfile = /etc/atuned/server.pem
# tlsserverkeyfile = /etc/atuned/server.key
# tlshttpcertfile = /etc/atuned/http/server.pem
# tlshttpkeyfile = /etc/atuned/http/server.key
# tlshttpcacertfile = /etc/atuned/http/cacert.pem

################################### log ##############################
# Either "debug", "info", "warn", "error", "critical", default is "info"
level = info

################################### monitor ##############################
[monitor]
# With the module and format of the MPI, the format is {module} {purpose}
# The module is Either "mem", "net", "cpu", "storage"
# The purpose is "topo"
module = mem topo, cpu topo

################################### system ##############################
# you can add arbitrary key-value here, just like key = value
# you can use the key in the profile
```

```
[system]
# the disk to be analysis
disk = sda

# the network to be analysis
network = enp189s0f0

user = root
```

2.5 Starting A-Tune

After the A-Tune is installed, you need to start the A-Tune service.

- Start the atuned service.

  ```
  # systemctl start atuned
  ```

- To query the status of the atuned service, run the following command:

  ```
  # systemctl status atuned
  ```

  If the following information is displayed, the service is started successfully:

  ```
  ● atuned.service - A-Tune Daemon
     Loaded: loaded (/usr/lib/systemd/system/atuned.service; disabled; vendor preset: disabled)
     Active: active (running) since Sat 2019-12-21 19:28:47 CST; 7s ago
  Main PID: 94790 (atuned)
     Tasks: 19 (limit: 104856)
    Memory: 132.2M
    CGroup: /system.slice/atuned.service
            ├─94790 /usr/bin/atuned
            ├─94797 python3 /usr/libexec/atuned/analysis/app.py /etc/atuned/atuned.cnf
            └─94801 /usr/bin/python3 -c from multiprocessing.semaphore_tracker import main;main(3)
  ```

3 Application Scenarios

You can use functions provided by A-Tune through the CLI client atune-adm. This chapter describes the functions and usage of the A-Tune client.

3.1 Overview

- You can run the **atune-adm help/--help/-h** command to query commands supported by atune-adm.
- All example commands are used in single-node mode. For distributed mode, specify an IP address and port number. For example:

```
# atune-adm -a 192.168.3.196 -p 60001 list
```

- The **define**, **update**, **undefine**, **collection**, **train**, and **upgrade** commands do not support remote execution.
- In the command format, brackets ([]) indicate that the parameter is optional, and angle brackets (<>) indicate that the parameter is mandatory. The actual parameters prevail.
- In the command format, meanings of each command are as follows:
  - **WORKLOAD_TYPE**: name of a user-defined workload type. For details about the supported workload types, see the query result of the **list** command.

- **PROFILE_NAME**: user-defined profile name.
- **PROFILE_PATH**: path of the user-defined profile.

3.2 Querying Workload Types

3.2.1 list

Function

Query the supported workload types, profiles, and the values of Active.

Format

atune-adm list

Example

```
# atune-adm list

Support WorkloadTypes:
+-----------------------------------+------------------------+-----------+
| WorkloadType                      | ProfileName            | Active    |
+===================================+========================+===========+
| default                           | default                | true      |
+-----------------------------------+------------------------+-----------+
| webserver                         | ssl webserver          | false     |
+-----------------------------------+------------------------+-----------+
| big database                      | database               | false     |
+-----------------------------------+------------------------+-----------+
| big data                          | big data               | false     |
+-----------------------------------+------------------------+-----------+
| in-memory computing               | in-memory computing    | false     |
+-----------------------------------+------------------------+-----------+
| in-memory database                | in-memory database     | false     |
+-----------------------------------+------------------------+-----------+
| single computer intensive jobs    | compute-intensive      | false     |
+-----------------------------------+------------------------+-----------+
| communication                     | rpc communication      | false     |
+-----------------------------------+------------------------+-----------+
| idle                              | default                | false     |
+-----------------------------------+------------------------+-----------+
```

NOTE

If the value of Active is **true**, the profile is activated. In the example, the profile of the default type is activated.

3.3 Workload Type Analysis and Auto Optimization

3.3.1 analysis

Function

Collect real-time statistics from the system to identify and automatically optimize workload types.

Format

atune-adm analysis [OPTIONS]

Parameter Description

- OPTIONS

Parameter	Description
--model, -m	Model generated by user-defined training

Example

- Use the default model for classification and identification.

```
# atune-adm analysis
```

- Use the user-defined training model for recognition.

```
# atune-adm analysis --model /usr/libexec/atuned/analysis/models/new-model.m
```

3.4 User-defined Model

A-Tune allows users to define and learn new models. To define a new model, perform the following steps:

Step 1 Run the **define** command to define workload_type and profile.

Step 2 Run the **collection** command to collect the profile data corresponding to workload_type.

Step 3 Run the **train** command to train the model.

----End

3.4.1 define

Function

Add a user-defined workload type and the corresponding profile optimization item.

Format

atune-adm define <WORKLOAD_TYPE> <PROFILE_NAME> <PROFILE_PATH>

Example

Add a workload type. Set workload type to **test_type**, profile name to **test_name**, and configuration file of an optimization item to **example.conf**.

```
# atune-adm define test_type test_name ./example.conf
```

The **example.conf** file can be written as follows (the following optimization items are optional and are for reference only). You can also run the **atune-adm info** command to view how the existing profile is written.

```
[main]
# list its parent profile
[tip]
# the recommended optimization, which should be performed manunaly
[check]
# check the environment
[affinity.irq]
# to change the affinity of irqs
[affinity.task]
# to change the affinity of tasks
[bios]
# to change the bios config
[bootloader.grub2]
# to change the grub2 config
[kernel config]
# to change the kernel config
[script]
# the script extention of cpi
[sysctl]
# to change the /proc/sys/* config
[sysfs]
# to change the /sys/* config
[systemctl]
# to change the system service config
[ulimit]
# to change the resources limit of user
```

3.4.2 collection

Function

Collect the global resource usage and OS status information during service running, and save the collected information to a CSV output file as the input dataset for model training.

NOTE

- This command depends on the sampling tools such as perf, mpstat, vmstat, iostat, and sar.
- Currently, only the Kunpeng 920 CPU is supported. You can run the **dmidecode -t processor** command to check the CPU model.

Format

atune-adm collection <OPTIONS>

Parameter Description

- OPTIONS

Parameter	Description
--filename, -f	Name of the generated CSV file used for training: *name-timestamp*.csv
--output_path, -o	Path for storing the generated CSV file. The absolute path is required.
--disk, -b	Disk used during service running, for example, /dev/sda.
--network, -n	Network port used during service running, for example, eth0.
--workload_type, -t	Workload type, which is used as a label for training.
--duration, -d	Data collection time during service running, in seconds. The default collection time is 1200 seconds.
--interval, -i	Interval for collecting data, in seconds. The default interval is 5 seconds.

Example

```
# atune-adm collection --filename name --interval 5 --duration 1200 --output_path
/home/data --disk sda --network eth0 --workload_type test_type
```

3.4.3 train

Function

Use the collected data to train the model. Collect data of at least two workload types during training. Otherwise, an error is reported.

Format

atune-adm train <OPTIONS>

Parameter Description

- OPTIONS

Parameter	Description
--data_path, -d	Path for storing CSV files required for model training
--output_file, -o	Model generated through training

Example

Use the CSV file in the **data** directory as the training input. The generated model **new-model.m** is stored in the **model** directory.

```
# atune-adm train --data_path /home/data --output_file
/usr/libexec/atuned/analysis/models/new-model.m
```

3.4.4 undefine

Function

Delete a user-defined workload type.

Format

atune-adm undefine <WORKLOAD_TYPE>

Example

Delete the **test_type** workload type.

```
# atune-adm undefine test_type
```

3.5 Querying Profiles

3.5.1 info

Function

View the profile content of a workload type.

Format

atune-adm info <WORKLOAD_TYPE>

Example

View the profile content of webserver.

```
# atune-adm info webserver

*** ssl webserver:

#
# webserver tuned configuration
#
[main]
#TODO CONFIG
```

```
[kernel config]
#TODO CONFIG

[bios]
#TODO CONFIG

[sysfs]
#TODO CONFIG

[sysctl]
fs.file-max=6553600
fs.suid dumpable = 1
fs.aio-max-nr = 1048576
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.tcp tw reuse = 1
net.ipv4.tcp syncookies = 1
net.ipv4.ip local port range = 1024    65500
net.ipv4.tcp max tw buckets = 5000
net.core.somaxconn = 65535
net.core.netdev max backlog = 262144
net.ipv4.tcp max orphans = 262144
net.ipv4.tcp max syn backlog = 262144
net.ipv4.tcp timestamps = 0
net.ipv4.tcp synack retries = 1
net.ipv4.tcp syn retries = 1
net.ipv4.tcp fin timeout = 1
net.ipv4.tcp keepalive time = 60
net.ipv4.tcp mem =  362619     483495   725238
net.ipv4.tcp rmem = 4096        87380   6291456
net.ipv4.tcp wmem = 4096        16384   4194304
net.core.wmem default = 8388608
net.core.rmem default = 8388608
net.core.rmem max = 16777216
net.core.wmem max = 16777216

[systemctl]
sysmonitor=stop
irqbalance=stop

[bootloader.grub2]
selinux=0
iommu.passthrough=1

[tip]
bind your master process to the CPU near the network = affinity
bind your network interrupt to the CPU that has this network = affinity
relogin into the system to enable limits setting = OS

[script]
openssl hpre = 0
prefetch = off
```

```
[ulimit]
{user}.hard.nofile = 102400
{user}.soft.nofile = 102400

[affinity.task]
#TODO CONFIG

[affinity.irq]
#TODO CONFIG

[check]
#TODO CONFIG
```

3.6 Updating a Profile

You can update the existing profile as required.

3.6.1 update

Function

Update an optimization item of a workload type to the content in the **new.conf** file.

Format

atune-adm update <WORKLOAD_TYPE> <PROFILE_NAME> <PROFILE_FILE>

Example

Update the workload type to **test_type** and the optimization item of test_name to **new.conf**.

```
# atune-adm update test_type test_name ./new.conf
```

3.7 Activating a Profile

3.7.1 profile

Function

Manually activate a profile of a workload type.

Format

atune-adm profile <WORKLOAD_TYPE>

Parameter Description

You can run the **list** command to query the supported workload types.

Example

Activate the profile configuration of webserver.

```
# atune-adm profile webserver
```

3.8 Rolling Back Profiles

3.8.1 rollback

Function

Roll back the current configuration to the initial configuration of the system.

Format

atune-adm rollback

Example

```
# atune-adm rollback
```

3.9 Updating Database

3.9.1 upgrade

Function

Update the system database.

Format

atune-adm upgrade <DB_FILE>

Parameter Description

- DB_FILE
  New database file path.

Example

The database is updated to **new_sqlite.db**.

```
# atune-adm upgrade ./new_sqlite.db
```

3.10 Querying System Information

3.10.1 check

Function

Check the CPU, BIOS, OS, and NIC information.

Format

atune-adm check

Example

```
# atune-adm check
 cpu information:
    cpu:0   version: Kunpeng 920-6426  speed: 2600000000 HZ   cores: 64
    cpu:1   version: Kunpeng 920-6426  speed: 2600000000 HZ   cores: 64
 system information:
    DMIBIOSVersion: 0.59
    OSRelease: 4.19.36-vhulk1906.3.0.h356.eulerosv2r8.aarch64
 network information:
    name: eth0            product: HNS GE/10GE/25GE RDMA Network Controller
    name: eth1            product: HNS GE/10GE/25GE Network Controller
    name: eth2            product: HNS GE/10GE/25GE RDMA Network Controller
    name: eth3            product: HNS GE/10GE/25GE Network Controller
    name: eth4            product: HNS GE/10GE/25GE RDMA Network Controller
    name: eth5            product: HNS GE/10GE/25GE Network Controller
    name: eth6            product: HNS GE/10GE/25GE RDMA Network Controller
    name: eth7            product: HNS GE/10GE/25GE Network Controller
    name: docker0          product:
```

3.11 Automatic Parameter Optimization

A-Tune provides the automatic search capability for optimal configurations, eliminating the need for repeated manual parameter adjustment and performance evaluation. This greatly improves the search efficiency of optimal configurations.

3.11.1 Tuning

Function

Use the specified project file to search the dynamic space for parameters and find the optimal solution under the current environment configuration.

Format

NOTE

Before running the command, ensure that the following conditions are met:

1. The YAML configuration file of the server has been edited and placed in the **/etc/atuned/tuning/** directory on the server by the server administrator.
2. The YAML configuration file of the client has been edited and placed in an arbitrary directory on the client.

atune-adm tuning [OPTIONS] <PROJECT_YAML>

Parameter Description

- OPTIONS

Parameter	Description
--restore, -r	Restores the initial configuration before tuning.
--project, -p	Specifies the project name in the YAML file to be restored.

NOTE

The preceding two parameters must be used at the same time, and the -p parameter must be followed by the specific project name.

- **PROJECT_YAML**: YAML configuration file of the client.

Configuration Description

Table 3-1 YAML file on the server

Name	Description	Type	Value Range
project	Project name.	Character string	-
startworkload	Script for starting the service to be optimized.	Character string	-
stopworkload	Script for stopping the service to be optimized.	Character string	-
maxiterations	Maximum number of optimization iterations, which is used to limit the number of	Integer	>10

Name	Description	Type	Value Range
	iterations on the client. Generally, the more optimization iterations, the better the optimization effect, but the longer the time required. Set this parameter based on the site requirements.		
object	Parameters to be optimized and related information. For details about the object configuration items, see Table 3-2.	-	-

Table 3-2 Description of object configuration items

Name	Description	Type	Value Range
name	Parameter to be optimized.	Character string	-
desc	Description of parameters to be optimized.	Character string	-
get	Script for querying parameter values.	-	-
set	Script for setting parameter values.	-	-
needrestart	Specifies whether to restart the service for the parameter to	Enumeration	**true** or **false**

Name	Description	Type	Value Range
	take effect.		
type	Parameter type. Currently, the **discrete** and **continuous** types are supported.	Enumeration	**discrete** or **continuous**
dtype	This parameter is available only when type is set to **discrete**. Currently, only **int** and **string** are supported.	Enumeration	int, string
scope	Parameter setting range. This parameter is valid only when type is set to **discrete** and dtype is set to **int**, or type is set to **continuous**.	Integer	The value is user-defined and must be within the valid range of this parameter.
step	Parameter value step, which is used when **dtype** is set to **int**.	Integer	This value is user-defined.
items	Enumerated value of which the parameter value is not within the scope. This is used when **dtype** is set to **int**.	Integer	The value is user-defined and must be within the valid range of this parameter.
options	Enumerated value range of the parameter value, which is used when **dtype** is set to **string**.	Character string	The value is user-defined and must be within the valid range of this parameter.
ref	Recommended initial value of the parameter	Integer or character string	The value is user-defined and must be within the valid range of this parameter.

Table 3-3 Description of configuration items of a YAML file on the client

Name	Description	Type	Value Range
project	Project name, which must be the same as that in the configuration file on the server.	Character string	-
iterations	Number of optimization iterations.	Integer	≥ 10
benchmark	Performance test script.	-	-
evaluations	Performance test evaluation index. For details about the evaluations configuration items, see Table 3-4.	-	-

Table 3-4 Description of evaluations configuration item

Name	Description	Type	Value Range
name	Evaluation index name.	Character string	-
get	Script for obtaining performance evaluation results.	-	-
type	Specifies a positive or negative type of the evaluation result. The value **positive** indicates that the performance value is minimized, and the value **negative** indicates that the performance value is maximized.	Enumeration	**positive** or **negative**

Name	Description	Type	Value Range
weight	Weight of the index. The value ranges from 0 to 100.	Integer	0-100
threshold	Minimum performance requirement of the index.	Integer	User-defined

Example

The following is an example of the YAML file configuration on a server:

```
project: "example"
maxiterations: 10
startworkload: ""
stopworkload: ""
object :
  -
    name : "vm.swappiness"
    info :
       desc : "the vm.swappiness"
       get : "sysctl -a | grep vm.swappiness"
       set : "sysctl -w vm.swappiness=$value"
       needrestart: "false"
       type : "continuous"
       scope :
         - 0
         - 10
       ref : 1
  -
    name : "irqbalance"
    info :
       desc : "system irqbalance"
       get : "systemctl status irqbalance"
       set : "systemctl $value sysmonitor;systemctl $value irqbalance"
       needrestart: "false"
       type : "discrete"
       options:
         - "start"
         - "stop"
       dtype : "string"
       ref : "start"
  -
    name : "net.tcp min tso segs"
    info :
       desc : "the minimum tso number"
       get : "cat /proc/sys/net/ipv4/tcp min tso segs"
       set : "echo $value > /proc/sys/net/ipv4/tcp min tso segs"
       needrestart: "false"
       type : "continuous"
```

```
      scope:
        - 1
        - 16
      ref : 2
 -
   name : "prefetcher"
   info :
      desc : ""
      get : "cat /sys/class/misc/prefetch/policy"
      set : "echo $value > /sys/class/misc/prefetch/policy"
      needrestart: "false"
      type : "discrete"
      options:
        - "0"
        - "15"
      dtype : "string"
      ref : "15"
 -
   name : "kernel.sched min granularity ns"
   info :
      desc : "Minimal preemption granularity for CPU-bound tasks"
      get : "sysctl kernel.sched min granularity ns"
      set : "sysctl -w kernel.sched min granularity ns=$value"
      needrestart: "false"
      type : "continuous"
      scope:
        - 5000000
        - 50000000
      ref : 10000000
 -
   name : "kernel.sched latency ns"
   info :
      desc : ""
      get : "sysctl kernel.sched latency ns"
      set : "sysctl -w kernel.sched latency ns=$value"
      needrestart: "false"
      type : "continuous"
      scope:
        - 10000000
        - 100000000
      ref : 16000000
```

The following is an example of the YAML file configuration on a client:

```
project: "example"
iterations : 10
benchmark : "sh /home/Benchmarks/mysql/tunning mysql.sh"
evaluations :
 -
   name: "tps"
   info:
      get: "echo -e '$out' |grep 'transactions:' |awk '{print $3}' | cut -c 2-"
      type: "negative"
```

```
    weight: 100
    threshold: 100
```

Example

- Perform tuning.

  ```
  # atune-adm tuning example-client.yaml
  ```

- Restore the initial configuration before tuning. The example value is the project name in the YAML file.

  ```
  # atune-adm tuning --restore --project example
  ```

4 FAQs

Q1: An error occurs when the **train** command is used to train a model, and the message "training data failed" is displayed.

Cause: Only one type of data is collected by using the **collection** command.

Solution: Collect data of at least two data types for training.

Q2: The atune-adm cannot connect to the atuned service.

Possible cause:

1. Check whether the atuned service is started and check the atuned listening address.

```
# systemctl status atuned
# netstat -nap | atuned
```

2. The firewall blocks the atuned listening port.
3. The HTTP proxy is configured in the system. As a result, the connection fails.

Solution:

1. If the atuned service is not started, run the following command to start the service:

```
# systemctl start atuned
```

2. Run the following command on the atuned and atune-adm servers to allow the listening port to receive network packets. In the command, **60001** is the listening port number of the atuned server.

```
# iptables -I INPUT -p tcp --dport 60001 -j ACCEPT
# iptables -I INPUT -p tcp --sport 60001 -j ACCEPT
```

3. Run the following command to delete the HTTP proxy or disable the HTTP proxy for the listening IP address without affecting services:

```
# no_proxy=$no_proxy, Listening IP address
```

Q3: The atuned service cannot be started, and the message "Job for atuned.service failed because a timeout was exceeded." is displayed.

Cause: The hosts file does not contain the localhost information.

Solution: Add localhost to the line starting with **127.0.0.1** in the **/etc/hosts** file.

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
```

5 Appendixes

5.1 Acronyms and Abbreviations

Table 5-1 Terminology

Term	Description
workload_type	Workload type, which is used to identify a type of service with the same characteristics.
profile	Set of optimization items and optimal parameter configuration.