Product Introduction

YTC (Yashan Trace Collector) | Yashan One-Click Collection Tool

YTC (Yashan Trace Collector) is a tool designed for collecting information and logs from Yashan databases.

Target Users

- Yashan DBAs

Product Positioning

- Lightweight standalone tool
- Ready to use out of the box

Recommended Scenarios

- When the database encounters failures
- When the database experiences performance issues
- Any other scenario requiring quick information collection

Core Features

Basic Information Collection

- Yashan database basic information (version number/commit number, etc.)
- Basic information of the host server (operating system/hardware configuration/firewall, etc.)
- Host server load conditions (network traffic/CPU usage/IO load/memory/disk capacity, etc.)
- ...

Fault Information Collection

- Database status and process checks
- ADR logs
- Coredump files
- Database logs (run.log/alert.log, etc.)
- Operating system logs (Dmesg/message_log, etc.)
- ...

Performance Data Collection

- AWR reports
- Slow query logs
- ...

Flexible and Configurable Collection Strategies

- Support for custom time periods with absolute flexibility in time range selection
- Support for custom collection modules, all three major modules can be flexibly combined
- Support for custom path data collection, both directories and files can be batch selected
- ...

Rich and Comprehensive Data Management

- Support for custom storage paths for collected data
- Multiple collection data display formats (txt/md/html)
- ...

Specifications

Product Format

A standalone command-line tool

Platform Support

- Linux (ubuntu/centos/kylin)

Hardware Architecture

- x86
- arm

Deployment Method

- No deployment required, ready to use out of the box

Product Specifications

1. Only supports local information collection
2. Only supports information collection for YashanDB SE

Parameter Specifications

The following constraints apply to the parameters of the collect subcommand:

- –t must be one or more of base, diag, perf, separated by commas
- The -r parameter has higher priority than -s/-e
- –r maximum is 30d, minimum is 1m, can be modified through the configuration file ({ytc_home}/config/strategy.toml)
- --s/-e format is yyyy-MM-dd-ss-mm, maximum interval is 7d, minimum is 1m, can be modified through the configuration file ({ytc_home}/config/strategy.toml)

Feature Specifications

- The tool will attempt to collect historical load data for the specified time period (default 24h) based on the server's corresponding time period. This depends on sar (System Activity Reporter), a commonly used system performance monitoring tool. It is recommended to pre-install this tool, otherwise historical load data cannot be collected. Valid data can only be tracked after the sar tool is running, and by default, only data from the last 30 days is retained.

Quick Start

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Start Collection

It is strongly recommended to run this tool with the user that owns the database process to be collected and grant sudo privileges to that user, or run the tool as the root user.

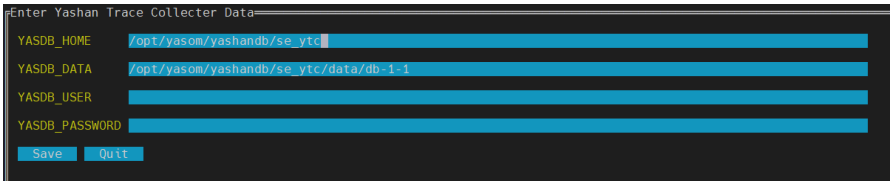
Executing `./ytcctl collect` will directly start a collection action. When all parameters are not specified, the tool will collect basic information, fault information, and performance data within the last 24 hours by default (specific metrics included in the modules can be found in [Reference Manual](./Reference Manual/)).

Language Selection

The tool supports both Chinese and English languages, which can be specified through the `--lang` parameter:

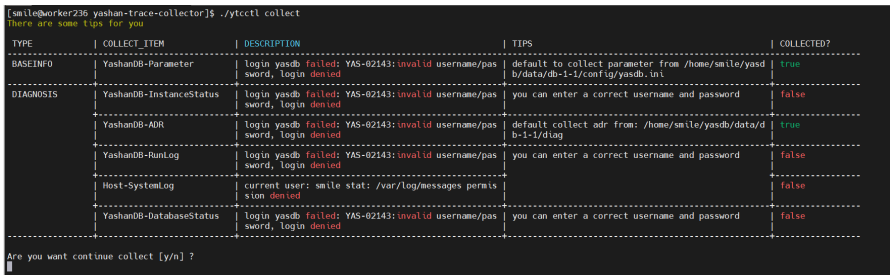
```
# Use Chinese (default)
./ytcctl collect --lang zh

# Use English
./ytcctl collect --lang en
```

1. Enter the pre-interaction page, fill in the relevant information, then Save to proceed to the next step:



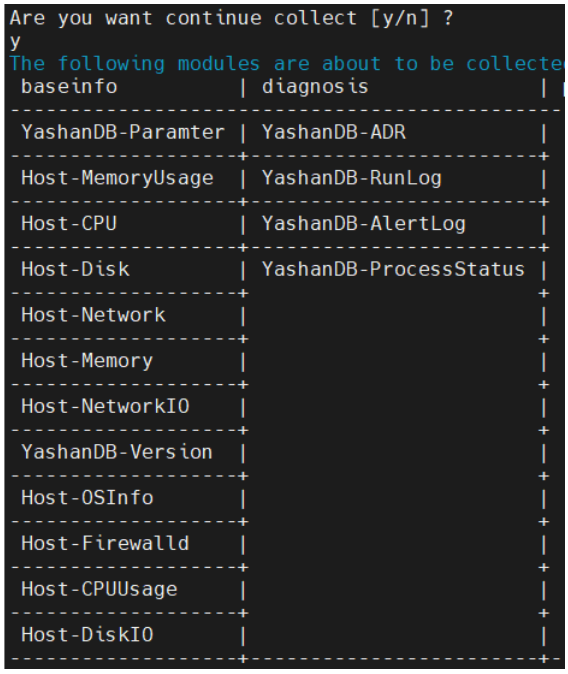

```
- YASDB_HOME: By default, it fills in the value of the current environment variable YASDB_HOME. If not available, it will
attempt to match the YASDB_HOME path corresponding to the existing yashandb process in the current environment. Users can
modify and confirm it themselves.
- YASDB_DATA: By default, it fills in the value of the current environment variable YASDB_DATA. If not available, it will
attempt to match the YASDB_DATA path corresponding to the existing yashandb process in the current environment. Users can
modify and confirm it themselves.
- YASDB_USER: Requires the user to provide a username that can access the current database with at least CREATE_SESSION
permission. Users can modify and confirm it themselves.
- YASDB_PASSWORD: Requires the user to provide a valid password corresponding to the database YASDB_USER. Users can modify
and confirm it themselves.
```

2. The tool will check whether the expected collected data can be correctly collected based on the information provided by the user, and display risk points and related suggestions, as shown below:



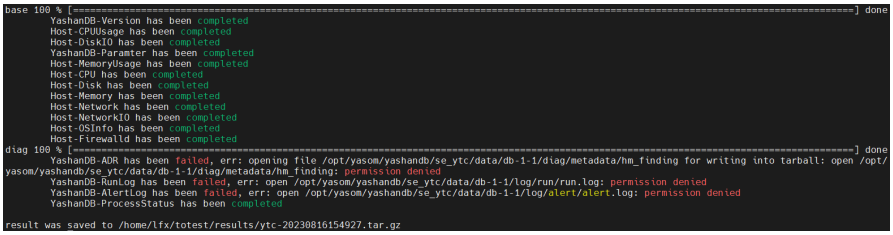
> Common risk warnings include:
>
> - 1. "you can run with sudo" - This is because collecting some log information requires the user to have root privileges. If the current executing user has sudo privileges, we recommend running the tool with sudo to collect more information.
> - 2. "login yasdb failed" - This is because the provided database username and password failed to log in. If the database cannot be connected, much database-related information cannot be accurately collected. We recommend providing a username and password that can normally access the database to collect more information.
> - 3. "current user xxx stat /yyy/zzz permission denied" - This is because the current user does not have permission to execute files in the path corresponding to certain metrics that need to be collected. Without -r-x permissions, this information cannot be collected. We recommend running the tool with a user that meets the expected information collection requirements (recommended: the database startup user with sudo privileges) to collect more information.

3. If you want to continue collection, enter y. If you want to follow the suggestions, select n to exit and re-execute, as shown below (example continues execution):



> Before starting collection, all metrics that can be collected this time will be listed (specific metrics can be found in [Reference Manual](./Reference Manual/))

4. Observe the specific progress of the tool collection and wait for completion, as shown below:



> The collection process will dynamically update the progress. After completion, the user will be prompted with the storage path of the results. The results of this collection are stored in `/home/lfx/totest/results/ytc-20230926090638.tar.gz`. By default, the collection results will be packaged and stored in the results path of the tool execution directory. Users can also use the -o parameter to customize the path.

Information Viewing

1. Extract the collected information package, which will generate a directory with the same name, as follows:

```
// Extract file
-bash-4.2$ tar -zxvf ytc-20230926090638.tar.gz
ytc-20230926090638/
ytc-20230926090638/yasdb/
ytc-20230926090638/yasdb/coredump/
ytc-20230926090638/yasdb/diag/
ytc-20230926090638/yasdb/diag/hm/
ytc-20230926090638/yasdb/diag/metadata/
ytc-20230926090638/yasdb/diag/metadata/hm_finding
ytc-20230926090638/yasdb/diag/metadata/hm_run
ytc-20230926090638/yasdb/diag/metadata/incident
ytc-20230926090638/yasdb/diag/metadata/problem
ytc-20230926090638/yasdb/diag/trace/
ytc-20230926090638/yasdb/log/
ytc-20230926090638/yasdb/log/alert.log
ytc-20230926090638/yasdb/log/run.log
ytc-20230926090638/yasdb/awr/
ytc-20230926090638/yasdb/awr/awrrpt-20230925090638-20230926090738.html
ytc-20230926090638/yasdb/slowsql/
ytc-20230926090638/yasdb/slowsql/slow.log
ytc-20230926090638/host/
ytc-20230926090638/host/log/
ytc-20230926090638/host/log/messages.log
ytc-20230926090638/host/log/dmesg.log
ytc-20230926090638/host/bashhistory/
ytc-20230926090638/host/bashhistory/root-bashhistory.txt
ytc-20230926090638/host/bashhistory/smile-bashhistory.txt
ytc-20230926090638/ytc-20230926090638.json
ytc-20230926090638/ytc_report_static/
ytc-20230926090638/ytc_report_static/css/
ytc-20230926090638/ytc_report_static/css/morris.css
ytc-20230926090638/ytc_report_static/js/
ytc-20230926090638/ytc_report_static/js/morris.js
ytc-20230926090638/ytc_report_static/js/raphael.min.js
ytc-20230926090638/ytc-report-20230926090638.txt
ytc-20230926090638/ytc-report-20230926090638.md
ytc-20230926090638/ytc-report-20230926090638.html

// View current directory status
-bash-4.2$ ll
total 216
drwxr-xr-x 5 smile smile    198 Sep 26 09:06 ytc-20230926090638
-rw-r--r-- 1 smile smile 219528 Sep 26 09:06 ytc-20230926090638.tar.gz

// Enter ytc-20230926090638 directory
-bash-4.2$ cd ytc-20230926090638/
-bash-4.2$ ll
total 2136
drwxr-xr-x 4 smile smile      36 Sep 26 09:06 host
drwxr-xr-x 7 smile smile      71 Sep 26 09:06 yasdb
-rw-r--r-- 1 smile smile 1323768 Sep 26 09:06 ytc-20230926090638.json
-rw-r--r-- 1 smile smile  383806 Sep 26 09:06 ytc-report-20230926090638.html
-rw-r--r-- 1 smile smile  102450 Sep 26 09:06 ytc-report-20230926090638.md
-rw-r--r-- 1 smile smile  367014 Sep 26 09:06 ytc-report-20230926090638.txt
drwxr-xr-x 4 smile smile      27 Sep 26 09:06 ytc_report_static
```

- host: Storage path for host resource-related source files collected this time (empty if none)
- yasdb: Storage path for YashanDB resource-related source files collected this time (empty if none)

- ytc-{generation time}.json: Source data json file collected this time, containing all underlying data collected and serving as the data source for generating reports
- ytc-report-{generation time}.html: HTML format file of the overall report collected this time
- ytc-report-{generation time}.md: MD format file of the overall report collected this time
- ytc-report-{generation time}.txt: TXT format file of the overall report collected this time
- ytc_report_static: js, css and other files that the html report depends on

> For more information, please refer to Report Interpretation

Best Practices

This section provides guidance on breaking down tool usage scenarios into minimal units. Users can integrate scenario units according to their actual needs to form best practices.

Parameter Explanation

ytc currently supports one subcommand, which is `collect`. The following is the parameter explanation for the collect command.

```
-bash-4.2$ ./ytcctl collect -h
Usage: ytcctl collect

The collect command is used to gather trace data.

Flags:
  -h, --help                          Show context-sensitive help.
  -v, --version                       Show version.
  -c, --config="./config/ytc.toml"    Configuration file.

  -t, --type="base,diag,perf"         The type of collection, choose one or more of (base|diag|perf) and split with ','.
  -r, --range=STRING                  The time range of the collection, such as '1M', '1d', '1h', '1m'. If <range> is given,
<start> and <end> will be discard.
  -s, --start=STRING                  The start datetime of the collection, such as 'yyyy-MM-dd', 'yyyy-MM-dd-hh', 'yyyy-MM-
dd-hh-mm'
  -e, --end=STRING                    The end timestamp of the collection, such as 'yyyy-MM-dd', 'yyyy-MM-dd-hh', 'yyyy-MM-
dd-hh-mm',, default value is current datetime.
  -o, --output=STRING                 The output dir of the collection.
      --include=STRING                Files or directories that need to be additionally collected, it is absolute path and
split with ',', such as '/tmp' or '/tmp,/root,/example.txt'.
      --exclude=STRING                Files or directories that no need to be additionally collected, it is absolute path and
split with ',', such as '/tmp' or '/tmp,/root,/example.txt'.
```

- -h: View help information for the collect subcommand
- -v: View the current version of the ytc tool
- -c: Specify the configuration file used by the tool, default is `{ytc_home}/config/ytc.toml`
- -t: Specify the modules expected to be collected. There are 3 modules in total (base information/diag fault information/perf performance data). By default, all modules are collected. When specifying multiple modules, separate them with commas
- -r: Specify time period. By default, data within the 24h period before the current time is collected. This configuration item takes effect globally and has higher priority than start/end parameters. Specification constraint: maximum 30d, minimum 1m, can be modified through configuration file ({ytc_home}/config/strategy.toml)
- -s: Specify the start time point for data collection. Default is the time point 24h before the current time. This configuration item takes effect globall y and has lower priority than the -r parameter. Specification constraint: format `yyyy-MM-dd-ss-mm` and the time period formed with end must be maximum 7d, minimum 1m, can be modified through configuration file ({ytc_home}/config/strategy.toml)
- -e: Specify the end time point for data collection. Default is the current time point. This configuration item takes effect globally and has lower priori ty than the -r parameter. Specification constraint: format `yyyy-MM-dd-ss-mm`, and the time period formed with start must be maximum 30d, mini mum 1m, can be modified through configuration file ({ytc_home}/config/strategy.toml)
- -o: Specify the path where collected data is stored. This path needs at least write permission for the current user executing the tool
- --include: Specify certain directories or files to be collected together with this collection. Separate multiple files or directories with commas
- --exclude: Specify files to be excluded during extra file collection. This applies to the include parameter. Files specified by the exclude parameter w ill not be collected. Separate multiple files or directories with commas

Default Collection

Just execute `ytcctl collect` directly

Default collection means not specifying any parameters. Each parameter is executed according to its default value. For specific default values, please refer to Parameter Explanation

For detailed execution guidance on default collection, please refer to Quick Start

Specify Modules

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Start Collection

Specify via -t parameter

- -t: Supports specifying the module for data collection. Currently, there are 3 built-in modules: base/diag/perf, which are basic information/fault information/performance data (specific metric data contained in the modules can be viewed in [Reference Manual](../Reference Manual/)). The tool will collect all modules by default.

> For example, executing `./ytcctl collect -t base` means only collecting the basic information module data. Executing `./ytcctl collect -t base,diag,perf` means collecting basic information, fault information, and performance data module data.

Information Viewing

Find the report storage path prompted after the tool succeeds to view the detailed information of this collection (for specific report interpretation, please refer to [Report Interpretation](../Reference Manual/Report%20Interpretation.md)).

Specify Time Range

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Start Collection

> The following are two methods for specifying time range. The -r method has higher priority than -s/-e. If the -r parameter is specified, -s/-e will not take effect this time.

Specify via -r parameter

- Usage example: `-r 1d` means within one day, `-r 1h` means within one hour, `-r 1m` means within 1 minute
- -r maximum is 30d, minimum is 1m, can be modified through configuration file ({ytc_home}/config/strategy.toml)

> For example, executing `./ytcctl collect -r 7h` means specifying to collect data information from the current moment going back 7 hours

Specify via -s/-e parameters

- Usage example: `-s 2023-06-01 -e 2023-07-01` means collecting data information from June 1st to July 1st
- -s/-e format is yyyy-MM-dd-ss-mm, maximum interval is 7d, minimum is 1m, can be modified through configuration file ({ytc_home}/config/strategy.toml)
- -s and -e can be specified individually. Unspecified parameters use default values: -s defaults to the time point 24h before the current time, -e defaults to the current time

> For example, executing `./ytcctl collect -s 2023-06-01` specifies the start time as June 1st, 2023, while the end time uses the default current time. However, the time period length must not exceed the maximum specification of 30d. If adjustment is needed, you can modify the corresponding configuration file ({ytc_home}/config/strategy.toml)

Information Viewing

Find the report storage path prompted after the tool succeeds to view the detailed information of this collection (for specific report interpretation, please refer to [Report Interpretation](../Reference Manual/Report%20Interpretation.md)).

Specify Report Storage Path

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Start Collection

Specify via -o parameter

- -o: Supports specifying the file storage path after collection is complete. If the path does not exist, the tool will create it by default. If the current executing user does not have sufficient permissions, the tool cannot continue to run

> For example, executing `./ytcctl collect -o /home/lfx` means the data package collected this time will be stored in the `/home/lfx` path.

Use default path

- When not using the -o parameter to specify the file storage path after collection is complete, the output configuration in the configuration file will be used as the default storage path, which is saved in the `results` folder under the `yashan-trace-collector` directory by default

Information Viewing

Find the report storage path prompted after the tool succeeds to view the detailed information of this collection (for specific report interpretation, please refer to [Report Interpretation](../Reference Manual/Report%20Interpretation.md)).

Specify Collection Files

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Start Collection

Flexibly specify via --include and --exclude parameters

- --include: Supports customizing additional collection of directory or file data under specified paths
- --exclude: This parameter is used in conjunction with `--include`, supports excluding corresponding directory or file data within the range specified by `--include`

> 1. For example, executing `./ytcctl collect --include /home/lfx` means that after this collection, in addition to collecting the data files of the default modules, the files under the `/home/lfx` path will also be packaged and collected together.
> 2. If you need to exclude certain files or directories in the path specified by `--include`, you can use the `--exclude` parameter to exclude them. For example, executing `./ytcctl collect --include /home/lfx --exclude /home/lfx/test` means collecting all file data under the `/home/lfx` path except the test directory.

Information Viewing

Find the report storage path prompted after the tool succeeds to view the detailed information of this collection. If files were specified, they will be saved in the extra directory, which is the extra data module (for specific report interpretation, please refer to [Report Interpretation](../Reference Manual/Report%20Interpretation.md)).

Reference Manual

This section is mainly used to elaborate on the specific implementation of various data collection built into the tool, as well as the interpretation of data packages after collection is completed and tool configuration guidance, so that users can better understand and analyze the collected data information and control tool capabilities.

> If the `--include` parameter is specified, it will be tracked and collected based on the Extra-FileCollect metric item and stored in the corresponding extra directory.

Tool Structure

Tool Acquisition

1. Log in to the server where the database is located and access the URL to obtain the tool package

```
http://192.168.19.121:9988/product/ycm/release/ytc/
```

2. Extract to the corresponding directory (example: extract to the current directory by default):

```
tar -zxvf yashan-trace-collector-v0.1.1-linux-x86_64.tar.gz
```

3. Enter the `yashan-trace-collector` folder and execute `./ytcctl -v` to verify success

```
-bash-4.2$ ./ytcctl -v
v0.1.1
```

Tool Structure

```
-bash-4.2$ cd yashan-trace-collector
-bash-4.2$ ll
drwxrwxr-x 2 yashan yashan 32 Aug 30 10:24 bin
drwxrwxr-x 2 yashan yashan 43 Aug 30 10:24 config
drwxrwxr-x 3 yashan yashan 22 Aug 30 10:24 docs
drwxrwxr-x 2 yashan yashan 24 Aug 30 10:24 log
drwxrwxr-x 3 yashan yashan 98 Aug 30 10:44 results
drwxrwxr-x 2 yashan yashan 22 Aug 30 10:24 scripts
drwxrwxr-x 4 yashan yashan 27 Aug 30 10:24 static
lrwxrwxrwx 1 yashan yashan 12 Aug 30 10:24 ytcctl -> ./bin/ytcctl
```

The contents corresponding to each folder and file are as follows:

File/Folder	Description
bin	Executable binary file directory
config	Configuration file storage directory
docs	Tool documentation storage directory
log	Tool log storage directory
results	Default storage directory for collection results
scripts	Tool script storage directory
static	Storage directory for css, js files needed for html reports
ytcctl	Symbolic link pointing to ./bin/ytcctl

Basic Information

Metric Name	Description	Underlying Implementation	Notes
YashanDB-Version	YashanDB version and commit number	`{YASDB_HOME}/bin/yasdb -V`	
YashanDB-Parameter	YashanDB server configuration parameters	1. `cat{YASDB_DATA}/config/yasdb.ini` 2. `select * from v$parameter where value is not null`	This data item is the content of the ini file and all non-null parameter current values in the v$parameter view
Host-OSInfo	Server operating system basic information	github.com/shirou/gopsutil/host Info()	Calls a third-party dependency in golang, which in turn calls the basic commands of each OS
Host-Firewalld	Server firewall status	`systemctl is-active firewalld`	
Host-CPU	Server CPU basic information	github.com/shirou/gopsutil/cpu Info()	Calls a third-party dependency in golang, which in turn calls the basic commands of each OS
Host-Disk	Server disk basic information	github.com/shirou/gopsutil/disk Partitions()	Calls a third-party dependency in golang, which in turn calls the basic commands of each OS
Host-Network	Server network card basic information	github.com/shirou/gopsutil/net Interfaces()	Calls a third-party dependency in golang, which in turn calls the basic commands of each OS
Host-Memory	Server memory basic information	github.com/shirou/gopsutil/mem VirtualMemory()	Calls a third-party dependency in golang, which in turn calls the basic commands of each OS
Host-NetworkIO	Server network load conditions	1.Historical load (depends on sar tool) Check if sar has custom path configuration `sar_dir`: `/etc/sysconfig/sysstat` If not, use the default path for each OS (ubuntu: `/var/log/sysstat`; others: `/var/log/sa`) Execute specific query command: `sar -n DEV -f {sar_dir}/{sa file for corresponding date} -s 03:41:57 -e 15:41:57` 2.Current load If sar exists, query current network load through sar, e.g., execute `sar -n DEV 1 10` to query once every 1s for a total of 10 times If sar does not exist, query current network load through gopsutil's net.INCounters(), query once every 1s for a total of 10 times and calculate approximate load	1.When sar is not installed, historical load data cannot be obtained, only current load data can be calculated 2.gopsutil is calculated by a custom algorithm in the program and may have slight deviations, for reference only 3.Interval times and duration can be adjusted through the configuration file
Host-CPUUsage	Server CPU load conditions	1.Historical load (depends on sar tool) Check if sar has custom path configuration `sar_dir`: `/etc/sysconfig/sysstat` If not, use the default path for each OS (ubuntu: `/var/log/sysstat`; others: `/var/log/sa`) Execute specific query command: `sar -u -f {sar_dir}/{sa file for corresponding date} -s 03:41:57 -e 15:41:57` 2.Current load If sar exists, query current network load through sar, e.g., execute `sar -u 1 10` to query once every 1s for a total of 10 times If sar does not exist, query current network load through	1.When sar is not installed, historical load data cannot be obtained, only current load data can be calculated 2.gopsutil is calculated by a custom algorithm in the program and may have slight deviations, for reference only 3.Interval times and duration can be adjusted through the configuration file

Metric Name	Description	Underlying Implementation	Notes
		gopsutil's net.INCounters(), query once every 1s for a total of 10 times and calculate approximate load	
Host-DiskIO	Server disk I/O load conditions	1.Historical load (depends on sar tool) Check if sar has custom path configuration `sar_dir` : `/etc/sysconfig/sysstat` If not, use the default path for each OS (ubuntu: `/var/log/sysstat` ; others: `/var/log/sa`) Execute specific query command: `sar -d -f {sar_dir}/{sa file for corresponding date} -s 03:41:57 -e 15:41:57` 2.Current load If sar exists, query current network load through sar, e.g., execute `sar -d 1 10` to query once every 1s for a total of 10 times If sar does not exist, query current network load through gopsutil's net.INCounters(), query once every 1s for a total of 10 times and calculate approximate load	1.When sar is not installed, historical load data cannot be obtained, only current load data can be calculated 2.gopsutil is calculated by a custom algorithm in the program and may have slight deviations, for reference only 3.Interval times and duration can be adjusted through the configuration file
Host-MemoryUsage	Server memory load conditions	1.Historical load (depends on sar tool) Check if sar has custom path configuration `sar_dir` : `/etc/sysconfig/sysstat` If not, use the default path for each OS (ubuntu: `/var/log/sysstat` ; others: `/var/log/sa`) Execute specific query command: `sar -r -f {sar_dir}/{sa file for corresponding date} -s 03:41:57 -e 15:41:57` 2.Current load If sar exists, query current network load through sar, e.g., execute `sar -r 1 10` to query once every 1s for a total of 10 times If sar does not exist, query current network load through gopsutil's net.INCounters(), query once every 1s for a total of 10 times and calculate approximate load	1.When sar is not installed, historical load data cannot be obtained, only current load data can be calculated 2.gopsutil is calculated by a custom algorithm in the program and may have slight deviations, for reference only 3.Interval times and duration can be adjusted through the configuration file

Fault Information

Metric Name	Description	Underlying Implementation	Notes
YashanDB-ProcessStatus	Database process information	Match process by data path	
YashanDB-InstanceStatus	Database instance status	`select status from v$instance;`	Depends on the provided database user and password being able to correctly connect to the database and execute the corresponding sql query statement
YashanDB-DatabaseStatus	Database status	`select status,open_mode as openMode from v$database`	1.This data item depends on the database instance status being OPEN and the user having relevant permissions to query normally 2.Mainly queries the database running status and open mode
YashanDB-ADR	Database automatic diagnostic data	`select name,value from v$parameter where name=DIAGNOSTIC_DEST` to find the main path of diagnostic data	1.If any abnormal scenario prevents obtaining the ADR main path corresponding to this parameter from the database, the tool will default to looking for the diag directory under the YASDB_DATA path 2.The user executing the tool needs to have execution permission for the corresponding file
YashanDB-RunLog	Database run log	`select name,value from v$parameter where name=RUN_LOG_FILE_PATH` to find the main path of the run log	1.If any abnormal scenario prevents obtaining the main path corresponding to this parameter from the database, the tool will default to looking for run.log in the log directory under the YASDB_DATA path 2.The user executing the tool needs to have execution permission for the corresponding file
YashanDB-AlertLog	Database alert log	Find alert.log in the log directory based on the database DATA path	The user executing the tool needs to have execution permission for the corresponding file
YashanDB-Coredump	Coredump files on the server	1.Check the coredump file generation path configured in file `/proc/sys/kernel/core_pattern` 2.Redirect program (abrt-hook-ccpp, systemd-coredump) to `/var/spool/abrt`, absolute path, relative path to home/bin to find core	
Host-KernelLog	Kernel log	dmesg command filtering	Mainly Dmesg data
Host-SystemLog	Operating system log	/var/log/messages or /var/log/syslog	
Host-BashHistory	Operating system Bash history	Collect the contents of the `$HISTFILE` file and parse it	1.Since `history` is a Bash built-in command, it cannot be collected directly by executing the `history` command, only the contents of the `$HISTFILE` can be collected 2.If you need to collect the history of the current terminal, you can use the `history -a` command to write the current terminal's history to $HISTFILE before collection

Performance Data

Metric Name	Description	Underlying Implementation	Notes
YashanDB-AWR	Database AWR Report	Generate an AWR report based on snapshots within the time range specified by the tool: `exec sys.dbms_awr.awr_report({db_id},{instance_NUMBER}, snap_start_id), {snap_end_id};` , then execute yasql command	1.Depends on the database being able to connect normally, and the database user must be sys 2.If AWR report generation takes too long due to large snapshots or large differences between snapshots, this collection may fail. The maximum duration can be adjusted through the configuration file (default 10min) 3.If the number of snapshots within the collection period specified by the tool is less than 2, AWR cannot be collected by default 4.If the database was restarted during the collection period specified by the tool, this AWR report will attempt to use the database startup time as the start time for collection
YashanDB-SlowSQL	Database slow SQL information	Query the slow sql records in the corresponding file through the database slow sql related configuration parameters `ENABLE_SLOW_LOG, SLOW_LOG_TIME_THRESHOLD, SLOW_LOG_SQL_MAX_LEN, SLOW_LOG_OUTPUT` , and query the data within the corresponding time period in the view `SLOW_LOG$`	Querying the view depends on the database user being able to access the database normally and having corresponding query permissions. If configured to have a slow.log file, the user running the tool needs to have execution permission for that file

Report Interpretation

Report Acquisition

After executing the ytc tool, follow the tool's instructions to reach the corresponding path. Enter the corresponding path to view the current file list. You can see that each compressed package has a corresponding generation time. Extract the package you want to view.

Here we use the scenario with the most complete data as an explanation example

Enter the folder where the report is stored. The folder contains the compressed packages of the collection results. The naming rule for compressed packages is `ytc-collection time.tar.gz`. Find the corresponding compressed package according to the collection time:

```
-bash-4.2$ cd results
-bash-4.2$ ll
total 216
-rw-r--r-- 1 smile smile 219528 Sep 26 09:06 ytc-20230926090638.tar.gz
```

Extract the corresponding collection result compressed package:

```
-bash-4.2$ tar -zxvf ytc-20230926090638.tar.gz
```

Enter the extracted folder. The folder contains the files collected this time, reports in different formats, the original json data of the report, and the css styles of the html report:

```
-bash-4.2$ cd ytc-20230926090638
-bash-4.2$ ll
total 2136
drwxr-xr-x 4 smile smile      36 Sep 26 09:06 host
drwxr-xr-x 7 smile smile      71 Sep 26 09:06 yasdb
drwxr-xr-x 7 smile smile      45 Sep 26 09:06 extra
-rw-r--r-- 1 smile smile 1323768 Sep 26 09:06 ytc-20230926090638.json
-rw-r--r-- 1 smile smile  383806 Sep 26 09:06 ytc-report-20230926090638.html
-rw-r--r-- 1 smile smile  102450 Sep 26 09:06 ytc-report-20230926090638.md
-rw-r--r-- 1 smile smile  367014 Sep 26 09:06 ytc-report-20230926090638.txt
drwxr-xr-x 4 smile smile      27 Sep 26 09:06 ytc_report_static
```

The contents corresponding to each folder and file are as follows:

```
host folder: Storage path for host resource-related source files collected this time (empty if none)
yasdb folder: Storage path for YashanDB resource-related source files collected this time (empty if none)
extra folder: Stores files collected by the extra collection module (empty if none)
ytc-{collection time}.json: Source data json file collected this time, containing all underlying data collected and serving
as the data source for generating reports
ytc-report-{collection time}.html: HTML format report file
ytc-report-{collection time}.md: Markdown format report file
ytc-report-{collection time}.txt: TXT format report file
ytc_report_static folder: Stores css and js files needed for html format reports
```

> Example collection command executed: sudo ./ytcctl collect -t base,diag,perf --include /tmp/test,/tmp/text.txt

Detailed Interpretation

report

One collection will produce three formats of reports: html, markdown, and txt. Taking the html format report as an example to interpret the report content, the html format report includes two main parts:

- Report overview information
- Collection item information

Report Overview Information

Report overview information includes basic overview and collection item overview

Basic Overview

The basic overview contains the database DATA path collected in this collection process, the database username used, the collection time range, the collection modules and the collection execution time, and other basic information. The main content is as follows (using this collection as an exampl e):

概览项	概览值
收集类型	基础信息，诊断信息，性能调优信息，额外收集项
收集范围--起始时间	2023-08-27 15:27
收集范围--截止时间	2023-08-28 15:28
YashanDB信息--YASDB_HOME	/opt/yasom/yashandb/test/yashandb/22.2.4.0
YashanDB信息--YASDB_DATA	/opt/yasom/yashandb/test/data/db-1-1
数据库用户(用于收集YashanDB信息)	sys
额外的收集文件	/tmp/test /test
收集结果存放目录	/home/golang/code/yashan-trace-collector/build/yashan-trace-collector/results
任务开始时间	2023-08-28 15:27:15
任务结束时间	2023-08-28 15:27:25

Collection Item Overview

The collection item overview information shows the metric names collected by each module during this collection process

基础信息	诊断信息	性能调优信息	额外收集项
数据库版本	数据库进程信息	AWR报告	额外文件收集
数据库配置	数据库实例状态	慢SQL	
操作系统信息	数据库状态		
防火墙配置	CoreDump		
CPU	数据库run.log日志		
内存	数据库alert.log日志		
磁盘	数据库ADR日志		
网络配置	操作系统日志		
CPU占用分析	操作系统内核日志		
内存容量检查			
网络流量			
磁盘I/O			

Collection Item Information

Collection item information contains the data collected by each collection module during this collection process. This collection process includes basi c information, diagnostic information, performance tuning information, and extra collection items (using this collection as an example)

Basic Information Collection Module

Version Information: Records the collected database version information and commit number

版本信息
YashanDB Server Release 22.2.4.0 x86_64 5314095

Database Configuration: Contains database instance configuration file (yasdb.ini) and database instance parameter view (v$parameter)

1.2.1 数据库实例配置文件：yasdb.ini

参数名称	参数值
CHARACTER_SET	UTF8
CONTROL_FILES	('?/dbfiles/ctrl1', '?/dbfiles/ctrl2')
DIN_ADDR	127.0.0.1:1690
LISTEN_ADDR	192.168.4.177:1688
NODE_ID	1-1:1
RUN_LOG_LEVEL	DEBUG
_CLUSTER_ID	c1aabc31e2e1720e06194b49913da4a8

1.2.2 数据库实例参数视图：v$parameter

参数名称	参数值
AC_MAX_SOURCE_SLICE_COUNT	20
AC_SLICE_THRESHOLD_SIZE	64M
ARCHIVE_LOCAL_DEST	?/archive
ARCH_CLEAN_IGNORE_MODE	NONE
ARCH_CLEAN_LOWER_THRESHOLD	12G
ARCH_CLEAN_UPPER_THRESHOLD	16G
AUDIT_FLUSH_INTERVAL	100
AUDIT_QUEUE_SIZE	16M
AUDIT_QUEUE_WRITE	TRUE

> The database instance parameter view only shows the parameter names and values of parameters whose values are not empty in the view

Operating System Information: Current host's operating system type, boot time, and kernel version information

检查项	检查结果
主机名称	mg_4
开机时间	2022-05-26 04:36:51
操作系统	linux
发行版本	centos 7.9.2009 (rhel系列)
内核版本	3.10.0-1160.el7.x86_64
内核架构	x86_64

Firewall Configuration: Current host's firewall status (enabled/disabled)

防火墙状态
已关闭

CPU: Current host's CPU model, number of cores, frequency, vendor ID and other basic information

检查项	检查结果
CPU型号	Intel Xeon Processor (Skylake, IBRS)
CPU核心数量	物理CPU核心：4，逻辑CPU核心：4
CPU主频	@2.10GHz
厂商标识ID	GenuineIntel

Memory: Current host's memory-related information, usually including system memory and swap partition information

	内存大小	已使用	空闲	共享内存	缓冲/缓存	可用
系统内存	15.51G	7.51G	287.32M	135.52M	7.72G	7.54G
交换分区	7.87G	658.46M	7.23G	/	4.04M	

- Memory size: Total system memory size
- Used: Amount of memory used, including memory in use and in cache
- Free: Amount of unused memory, including free memory and memory used for kernel cache
- Shared memory: Size of shared memory, this part of memory is shared by multiple processes
- Buffer/Cache: Amount of memory used for kernel cache, including file system cache and other buffers
- Available: Amount of available memory, indicating memory that can be immediately allocated to new processes or provided to existing processes. It takes into account the file system cache portion, so it can be quickly allocated by the system

Disk: Information about each disk on the current host, including device name, file system type, disk size, used space, available space, usage rate, mount point, and mount options

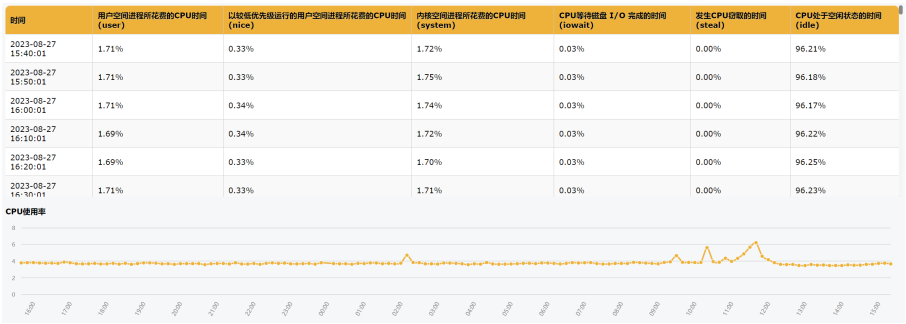
磁盘设备	文件系统类型	磁盘大小	已使用	可用	使用率	挂载路径	挂载选项
/dev/dm-0	xfs	49.98G	21.37G	28.6G	42.77%	/	rw,relatime
/dev/vda1	xfs	1014M	182.98M	831.02M	18.05%	/boot	rw,relatime
/dev/dm-2	xfs	141.05G	25.24G	115.81G	17.90%	/home	rw,relatime

- Disk device: Disk device name
- File system type: Current disk's file system type
- Disk size: Total file system size
- Used: Used disk space size
- Available: Available disk space size
- Mount path: File system mount point
- Mount options: Parameters and behaviors specified when mounting the file system

Network Configuration: Current host's network interface name, corresponding IP address, and MAC address information

网络接口	IP地址	MAC地址
lo	IPv4: 127.0.0.1/8 IPv6: ::1/128	
eth0	IPv4: 192.168.4.177/24 172.25.1.1/16 192.168.4.231/24 IPv6: fe80::e0bd:9b88:7de4:d691/64 fe80::b6a7:dd83:5f48:9fdc/64 fe80::976c:a0eb:2e2b:9e7d/64	52:54:00:c7:5b:f5
docker0	IPv4: 172.17.0.1/16 IPv6: fe80::42:35ff:fe80:3ae4/64	02:42:35:80:3a:e4

CPU Usage Analysis: Current host's CPU usage, including historical CPU usage and current CPU usage



时间	用户空间进程所花费的CPU时间 (user)	以较低优先级运行的用户空间进程所花费的CPU时间 (nice)	内核空间进程所花费的CPU时间 (system)	CPU等待磁盘 I/O 完成的时间 (iowait)	发生CPU窃取的时间 (steal)	CPU处于空闲状态的时间 (idle)
2023-08-27 15:40:01	1.71%	0.33%	1.72%	0.03%	0.00%	96.21%
2023-08-27 15:50:01	1.71%	0.33%	1.75%	0.03%	0.00%	96.18%
2023-08-27 16:00:01	1.71%	0.34%	1.74%	0.03%	0.00%	96.17%
2023-08-27 16:10:01	1.69%	0.34%	1.72%	0.03%	0.00%	96.22%
2023-08-27 16:20:01	1.69%	0.33%	1.70%	0.03%	0.00%	96.25%
2023-08-27 16:30:01	1.71%	0.33%	1.71%	0.03%	0.00%	96.23%

CPU使用率

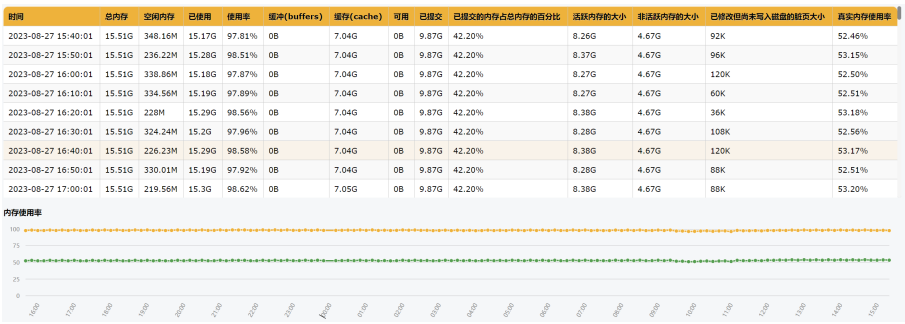
Detailed CPU usage information is presented in table format, specifically including:

- Time: Data collection moment
- user: CPU time spent by user space processes
- nice: CPU time spent by user space processes running at lower priority
- system: CPU time spent by kernel space processes
- iowait: Time CPU waits for disk I/O to complete
- steal: Time when CPU stealing occurs
- idle: Time when CPU is idle

CPU usage (user+system) is presented as a time series chart for intuitive representation of CPU usage fluctuations.

> Server load information (CPU usage, memory usage, network IO, and disk IO) is based on data collected by the System Activity Reporter (SAR) tool. If SAR tool is not installed on the server or SAR tool has not retained historical monitoring data for the specified time, historical load information w ill not be collected by default, and current load information will also be collected using a custom algorithm.
> Historical load information collects data within the specified time range. Its data collection time interval is affected by the operating system cron c onfiguration, with a default interval of 10 minutes for data collection.
> Current load information collects data from now to the next 10 seconds (controlled by configuration file).

Memory Capacity Check: Current host system memory usage, including historical memory usage and current memory usage



时间	总内存	空闲内存	已使用	使用率	缓冲(buffers)	缓存(cache)	可用	已提交	已提交的内存占总内存的百分比	活跃内存的大小	非活跃内存的大小	已修改但尚未写入磁盘的脏页大小	真实内存使用率
2023-08-27 15:40:01	15.51G	348.16M	15.17G	97.81%	0B	7.04G	0B	9.87G	42.20%	8.26G	4.67G	92K	52.46%
2023-08-27 15:50:01	15.51G	236.22M	15.28G	98.51%	0B	7.04G	0B	9.87G	42.20%	8.37G	4.67G	96K	53.15%
2023-08-27 16:00:01	15.51G	338.86M	15.18G	97.87%	0B	7.04G	0B	9.87G	42.20%	8.27G	4.67G	120K	52.50%
2023-08-27 16:10:01	15.51G	334.56M	15.19G	97.89%	0B	7.04G	0B	9.87G	42.20%	8.27G	4.67G	60K	52.51%
2023-08-27 16:20:01	15.51G	228M	15.29G	98.56%	0B	7.04G	0B	9.87G	42.20%	8.38G	4.67G	36K	53.18%
2023-08-27 16:30:01	15.51G	324.24M	15.2G	97.96%	0B	7.04G	0B	9.87G	42.20%	8.28G	4.67G	108K	52.56%
2023-08-27 16:40:01	15.51G	226.23M	15.29G	98.58%	0B	7.04G	0B	9.87G	42.20%	8.38G	4.67G	120K	53.17%
2023-08-27 16:50:01	15.51G	330.01M	15.19G	97.92%	0B	7.04G	0B	9.87G	42.20%	8.28G	4.67G	88K	52.51%
2023-08-27 17:00:01	15.51G	219.56M	15.3G	98.62%	0B	7.05G	0B	9.87G	42.20%	8.38G	4.67G	88K	53.20%

内存使用率

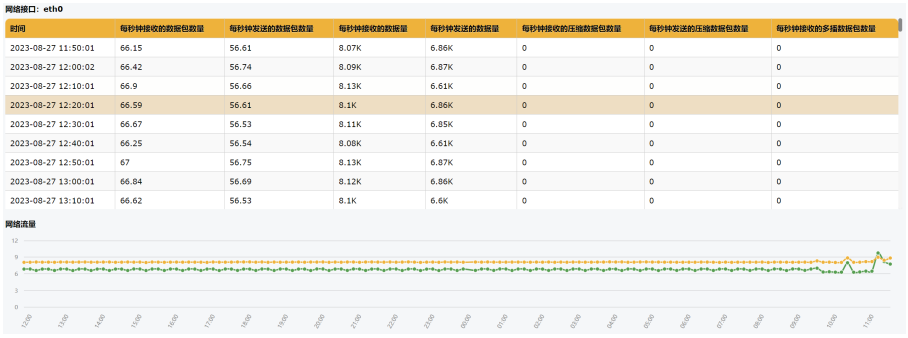
Memory capacity detailed information is presented in table format, specifically including:

- Time: Data collection moment
- Total memory: Total system memory size
- Free space: Free memory size
- Used: Used memory size, including buffer/cache size
- Usage rate: Used/Total memory*100%
- Buffer: Memory size used as buffer
- Cache: Memory size used as cache
- Committed: Size of committed memory (allocated but not yet written to disk)
- Committed memory percentage of total memory: Committed/Total memory*100%
- Active memory size: Size of active memory (memory in use)
- Inactive memory size: Size of inactive memory (unused but still retained in memory)
- Dirty page size modified but not yet written to disk: Size of dirty pages (modified but not yet written to disk)
- Real memory usage rate: (Total memory - Free memory - Buffer - Cache)/Total memory*100%

Memory usage and real memory usage are presented as time series charts for intuitive display of memory usage.

> Since buffers and caches can be reused, when calculating memory usage, buffers and caches need to be considered. On some operating systems (such as CentOS), the usage field only considers the actual memory used without considering buffers and caches, so the data is larger than the act ual memory usage.
> Real memory usage takes into account buffer and cache conditions when calculated, representing the operating system's true memory usage.

Network Traffic: Network IO conditions of different network interfaces on the current host



网络接口：eth0

时间	每秒钟接收的数据包数量	每秒钟发送的数据包数量	每秒钟接收的数据量	每秒钟发送的数据量	每秒钟接收的压缩数据包数量	每秒钟发送的压缩数据包数量	每秒钟接收的多播数据包数量
2023-08-27 11:50:01	66.15	56.61	8.07K	6.86K	0	0	0
2023-08-27 12:00:02	66.42	56.74	8.09K	6.87K	0	0	0
2023-08-27 12:10:01	66.9	56.66	8.13K	6.61K	0	0	0
2023-08-27 12:20:01	66.59	56.61	8.1K	6.86K	0	0	0
2023-08-27 12:30:01	66.67	56.53	8.11K	6.85K	0	0	0
2023-08-27 12:40:01	66.25	56.54	8.08K	6.61K	0	0	0
2023-08-27 12:50:01	67	56.75	8.13K	6.87K	0	0	0
2023-08-27 13:00:01	66.84	56.69	8.12K	6.86K	0	0	0
2023-08-27 13:10:01	66.62	56.53	8.1K	6.6K	0	0	0

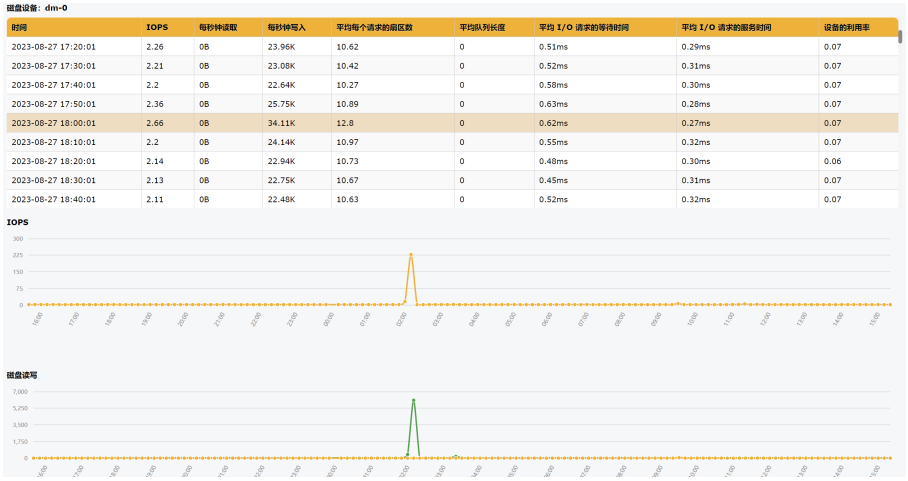
网络流量

Network traffic detailed data is presented in table format, specifically including:

- Time: Data collection time
- Number of packets received per second: Represents the number of packets received per second from the network interface
- Number of packets sent per second: Represents the number of packets sent per second from the network interface
- Data received per second: Represents the amount of data received per second from the network interface
- Data sent per second: Represents the amount of data sent per second from the network interface
- Number of compressed packets received per second: Represents the number of compressed packets received per second from the network interf ace
- Number of compressed packets sent per second: Represents the number of compressed packets sent per second from the network interface
- Number of multicast packets received per second: Represents the number of multicast packets received per second from the network interface

Send traffic and receive traffic are presented as time series charts for intuitive representation of network IO conditions.

Disk IO: IO conditions of each disk on the current host



磁盘设备：dm-0

时间	IOPS	每秒钟读取	每秒钟写入	平均每个请求的扇区数	平均队列长度	平均 I/O 请求的等待时间	平均 I/O 请求的服务时间	设备的利用率
2023-08-27 17:20:01	2.26	0B	23.96K	10.62	0	0.51ms	0.29ms	0.07
2023-08-27 17:30:01	2.21	0B	23.08K	10.42	0	0.52ms	0.31ms	0.07
2023-08-27 17:40:01	2.2	0B	22.64K	10.27	0	0.58ms	0.30ms	0.07
2023-08-27 17:50:01	2.36	0B	25.75K	10.89	0	0.63ms	0.28ms	0.07
2023-08-27 18:00:01	2.66	0B	34.11K	12.8	0	0.62ms	0.27ms	0.07
2023-08-27 18:10:01	2.2	0B	24.14K	10.97	0	0.55ms	0.32ms	0.07
2023-08-27 18:20:01	2.14	0B	22.94K	10.73	0	0.48ms	0.30ms	0.06
2023-08-27 18:30:01	2.13	0B	22.75K	10.67	0	0.45ms	0.31ms	0.07
2023-08-27 18:40:01	2.11	0B	22.48K	10.63	0	0.52ms	0.32ms	0.07

IOPS

磁盘读写

Disk IO detailed information is presented in table format, specifically including:

- Time: Data collection moment
- IOPS: Number of I/O operations completed per second, representing the total number of read and write requests processed by the disk device per second
- Read per second: Amount of data read per second from the disk device
- Write per second: Amount of data written per second to the disk device
- Average sectors per request: Average number of sectors requested per I/O request
- Average queue length: Average length of the queue waiting to be processed
- Average I/O request wait time: Average time from request submission to completion
- Average I/O request service time: Average service time for each I/O request

- Device utilization: Device usage rate during the specified time

Disk read/write and IOPS are presented as time series charts for intuitive representation of disk IO conditions

Diagnostic Information

Database Process Information: Database process PID, command line, owner, creation time, CPU and memory usage, and status information

进程ID	命令行	所属用户	创建时间	CPU使用率	内存使用率	状态
15591	/opt/yasom/yashandb/test/yashandb/22.2.4.0/bin/yasdb nomount -D /opt/yasom/yashandb/test/data/db-1-1	golang	2023-08-22 18:47:37	2.20%	2.48%	S

Database Instance Status: STATUS field in the database V$INSTANCE view

状态
OPEN

Database Status: STATUS and OPEN_MODE fields in the database V$DATABASE view

状态	开启模式
NORMAL	READ_WRITE

Coredump: Records the path of the collected coredump files

存放路径
ytc-20230926090638/yasdb/coredump

Database run.log: Records the path of the database run.log within the specified time range

存放路径
ytc-20230926090638/yasdb/log/run.log

Database alert.log: Records the path of the database alert.log within the specified time range

存放路径
ytc-20230926090638/yasdb/log/alert.log

Database ADR Log: Records the path of the collected database ADR log files

存放路径
ytc-20230926090638/yasdb/diag

Operating System Log: Records the path of the operating system logs within the specified time range

存放路径
ytc-20230926090638/host/log/messages.log

Operating System Kernel Log: Records the path of the operating system kernel logs within the specified time range

存放路径
ytc-20230926090638/host/log/dmesg.log

Operating System Bash History: Records the storage path of the collected operating system Bash history

存放路径
ytc-20230926090638/host/bashhistory

Performance Tuning Information

AWR Report: Records the path of the AWR report generated in this collection

存放路径
ytc-20230926090638/yasdb/awr/awrrpt-20230925090638-20230926090738.html

Slow SQL Parameters: Database slow SQL log related parameter configuration

慢SQL参数名称	参数值
ENABLE_SLOW_LOG	FALSE
SLOW_LOG_OUTPUT	FILE
SLOW_LOG_FILE_PATH	?/log/slow
SLOW_LOG_TIME_THRESHOLD	1000
SLOW_LOG_SQL_MAX_LEN	2000

SLOW_LOG$ System Table: Slow log information within the specified time range collected from the database SLOW_LOG$ system table

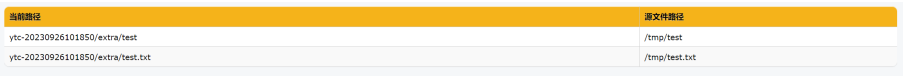
DATABASE_NAME	USER_NAME	START_TIME	USER_HOST	QUERY_TIME	ROWS_SENT	SQL_ID	SQL_TEXT

Slow SQL Log File: Path of the database slow log file within the specified time range

存放路径
ytc-20230926090638/yasdb/slowsql/slow.log

Extra Collection Items

Extra File Collection: Mapping relationship between the original path and current path of specified collection files



当前路径	源文件路径
ytc-20230926101850/extra/test	/tmp/test
ytc-20230926101850/extra/test.txt	/tmp/test.txt

> When files or folders with the same name exist, files or folders with smaller depth will be renamed. For example, when collecting both /test and /tmp/test folders, the content of /test folder will be saved in ./extra/_test folder, while the content of /tmp/test folder will be saved in ./extra/test folder

Some collection modules need to collect files. The report records the storage path of the collected files. The specific files are stored in the folder of the corresponding module. Below is the interpretation of the contents of each folder in the collection results.

base

Base information does not have source file data to save, so there is no folder for this module. Valid data can be viewed in the report file.

diag and perf modules

diag is the files collected by the fault information collection module, perf is the files collected by the performance data module. The file directory structure of the collection results is not divided by module, but by resource, specifically divided into host resources and database resources. The corresponding folder names are: host and yasdb, their file structure is as follows:

```
-bash-4.2$ cd ytc-20230926090638/host/
-bash-4.2$ tree
.
├── bashhistory
│   ├── root-bashhistory.txt
│   └── yashan-bashhistory.txt
└── log
    ├── dmesg.log
    └── messages.log

2 directories, 4 files
```

- bashhistory folder stores the host's Bash history records, including root user (if permission to collect) and YashanDB user
- log folder stores the collected host logs, including host operating system kernel log (dmesg.log), host messages log (messages.log), host syslog (syslog.log)

```
-bash-4.2$ cd ytc-20230926090638/yasdb/
-bash-4.2$ tree
.
├── awr
│   └── awrrpt-20230925090638-20230926090738.html
├── coredump
├── diag
│   ├── hm
│   ├── metadata
│   │   ├── hm_finding
│   │   ├── hm_run
│   │   ├── incident
│   │   └── problem
│   └── trace
├── log
│   ├── alert.log
│   └── run.log
└── slowsql
    └── slow.log

8 directories, 8 files
```

- awr folder stores the database AWR report generated in this collection

- coredump folder stores the collected core files
- diag folder stores the database adr information, corresponding to the content under the `${YASDB_DATA}/diag` folder of the collected database
- log folder stores the collected database logs, including database run.log and alert.log
- slowsql folder stores the collected database slow log files

extra

extra folder stores the files collected additionally this time

> - If the --include parameter specifies multiple files or folders with parent-child relationships, they will be merged for collection, keeping only the parent folder.
> - If folders or files have the same name, folders at shallower levels will be renamed according to certain rules.
> - The file path mapping before and after collection can be referred to in the extra file collection module in the report.

ytc-{time}.json

The source data json file collected this time, containing all underlying data collected and serving as the data source for generating reports

ytc_report_static

ytc_report_static folder stores js and css files needed for html format reports

```
-bash-4.2$ cd ytc-20230926090638/ytc_report_static/
-bash-4.2$ tree
.
├── css
│   └── morris.css
└── js
    ├── morris.js
    └── raphael.min.js

2 directories, 3 files
```

Configuration File

Current version tool effective configuration file path: `{ytc_home}/config/strategy.toml`

Default Factory Configuration

```
[collect]
max_duration = "30d"
min_duration = "1m"
network_io_discard = "^lo$,^veth.*,^virbr.*,^br.*,^tap.*,^tun.*,^docker.*,^flannel.*"
output = "./results"
range = "24h"
scrape_interval = 1
scrape_times = 10
awr_timeout = "10m"
```

- max_duration: Maximum time range for tool information collection, i.e., the effective function of specified -r or -s/-e, default 30 days
- min_duration: Minimum time range for tool information collection, i.e., the effective function of specified -r or -s/-e, default 1 minute
- network_io_discard: Data discarded by default when querying server network load conditions
- output: Main path where collected data is finally stored, default is in the results directory under the tool's current running directory
- range: Time range for tool collection, default is 24h before the current time
- scrape_interval: Time interval for network load related data collection, default is 1s
- scrape_times: Total number of repetitions based on time interval when collecting network load related data, default is 10 times
- awr_timeout: Maximum timeout for AWR generation, default 10 minutes

Other Non-default Configuration Items

- `sar_dir`: Storage path for sar's historical monitoring data, default is empty, the tool will obtain this path based on system configuration. Users can specify a specific path to take effect in extreme scenarios
- `core_dump_path`: Main path for core dump files, default is empty, the tool will obtain this path based on system configuration. Users can specify a specific path to take effect in extreme scenarios
- `core_file_key`: Keyword for core dump file names, default is empty, the tool will obtain the keyword based on system configuration. Users can specify a specific path to take effect in extreme scenarios, recommended to use in combination with `core_dump_path`

All the above configuration items can be adjusted according to the specification and will take effect upon next execution

Q&A

TODO Common Problem Solutions Guide

Terminology

References