

orzdba 工具使用说明

Author : zhuxu@taobao.com

Date : 2012-08-03

Team : Taobao DBA

代码 SVN 地址:

<http://code.taobao.org/p/orzdba/src/trunk/>

说明:

Perl 脚本, 用于对 Linux 主机和 MySQL 相关指标进行实时监控。

工具主要有如下参数:

```
-h,--help          Print Help Info.
-i,--interval      Time(second) Interval.
-C,--count         Times.
-t,--time          Print The Current Time.
-nocolor          Print NO Color.

-l,--load          Print Load Info.
-c,--cpu           Print Cpu Info.
-s,--swap          Print Swap Info.
-d,--disk          Print Disk Info.
-n,--net           Print Net Info.

-P,--port          Port number to use for mysql connection(default 3306).
-S,--socket        Socket file to use for mysql connection.

-com              Print MySQL Status(Com_select,Com_insert,Com_update,Com_delete).
-hit             Print Innodb Hit%.
-innodb_rows      Print Innodb Rows Status(Innodb_rows_inserted/updated/deleted/read).
-innodb_pages     Print Innodb Buffer Pool Pages Status(Innodb_buffer_pool_pages_data/free/dirty/flushed)
-innodb_data      Print Innodb Data Status(Innodb_data_reads/writes/read/written)
-innodb_log       Print Innodb Log Status(Innodb_os_log_fsyncs/written)
-innodb_status    Print Innodb Status from Command: 'Show Engine Innodb Status'
                  (history list/ log unflushed/unchecked bytes/ read views/ queries inside/queued)
-T,--threads      Print Threads Status(Threads_running,Threads_connected,Threads_created,Threads_cached).
-rt              Print MySQL DB RT(us).
-B,--bytes        Print Bytes received from/send to MySQL(Bytes_received,Bytes_sent).

-mysql            Print MySQLInfo (include -t,-com,-hit,-T,-B).
-innodb           Print InnodbInfo(include -t,-innodb_pages,-innodb_data,-innodb_log,-innodb_status)
-sys              Print SysInfo (include -t,-l,-c,-s).
-lazy             Print Info (include -t,-l,-c,-s,-com,-hit).

-L,--logfile      Print to logfile.
-logfile_by_day   One day a logfile,the suffix of logfile is 'yyyy-mm-dd';
                  and is valid with -L.
```

下面分别说明一下:

1、查看 Linux 主机指标

-l,--load	Print Load Info.
-c,--cpu	Print Cpu Info.
-s,--swap	Print Swap Info.
-d,--disk	Print Disk Info.
-n,--net	Print Net Info.
-sys	Print SysInfo (include -t,-l,-c,-s)

这些指标的数据都来自 /proc 目录下的相关系统元数据:

LOAD : /proc/loadavg

CPU : /proc/stat

SWAP : /proc/vmstat

DISK : /proc/diskstats

NET : /proc/net/dev

其中, 查看 DISK (-d) 和 NET (-n) 需要带具体的设备名 (具体可以查看 /proc/diskstats 和 /proc/net/dev 中的设备或者可以取自 iostat 和 sar -n DEV)。

```
orzdaba -sys -n bond0 -d sda

=====
Welcome to use the orzdaba tool !
Yep...Chinese English~
===== Date : 2012-07-29 =====

HOST:      IP:

-----load-avg-----cpu-usage-----swap-----net(B)-----io-usage-----
time | 1m 5m 15m | usr sys idl iow | si so | recv send | r/s w/s | kB/s kB/s | queue await svctm %util |
14:42:39 | 1.21 1.27 1.22 | 3 1 94 3 | 0 0 | 0 0 | 164.9 28.3 | 3679.4 767.7 | 0.0 0.1 0.4 7.1 |
14:42:40 | 1.21 1.27 1.22 | 2 1 94 3 | 0 0 | 325k 1.5m | 140.1 6.9 | 2225.4 228.9 | 0.7 4.5 3.1 45.6 |
14:42:41 | 1.21 1.27 1.22 | 2 1 95 2 | 0 0 | 299k 1.1m | 114.6 7.9 | 1818.4 193.7 | 0.5 3.9 2.9 35.3 |
14:42:42 | 1.21 1.27 1.22 | 2 0 95 2 | 0 0 | 398k 1.7m | 126.4 6.9 | 2006.9 177.8 | 0.6 4.2 2.8 37.8 |
14:42:43 | 1.12 1.25 1.21 | 2 1 93 4 | 0 0 | 318k 1.4m | 149.1 178.8 | 2370.4 4404.9 | 0.9 2.8 1.5 49.4 |
14:42:44 | 1.12 1.25 1.21 | 2 1 95 3 | 0 0 | 391k 1.8m | 129.5 10.9 | 2072.7 217.6 | 0.7 4.9 3.1 44.2 |
14:42:45 | 1.12 1.25 1.21 | 2 0 95 3 | 0 0 | 352k 1.4m | 145.3 45.5 | 2356.0 948.7 | 0.7 3.7 1.8 35.3 |
14:42:46 | 1.12 1.25 1.21 | 2 0 94 3 | 0 0 | 304k 1.3m | 177.9 10.9 | 2814.6 173.9 | 0.8 4.4 2.3 44.0 |
14:42:47 | 1.12 1.25 1.21 | 2 0 94 3 | 0 0 | 273k 1.2m | 156.2 6.9 | 2499.9 185.9 | 0.7 4.4 2.4 39.6 |
14:42:48 | 1.11 1.25 1.21 | 1 0 94 4 | 0 0 | 355k 1.4m | 206.9 8.9 | 3295.0 202.0 | 0.9 4.2 2.7 59.1 |
14:42:49 | 1.11 1.25 1.21 | 1 1 95 3 | 0 0 | 291k 1.2m | 162.1 8.9 | 2593.2 158.1 | 0.8 4.7 2.0 34.8 |
14:42:50 | 1.11 1.25 1.21 | 2 0 95 3 | 0 0 | 328k 1.3m | 144.5 6.9 | 2263.9 146.4 | 0.7 4.4 2.1 32.5 |
14:42:51 | 1.11 1.25 1.21 | 3 1 94 3 | 0 0 | 380k 1.4m | 129.4 7.9 | 2070.1 177.8 | 0.6 4.4 3.0 40.9 |
14:42:52 | 1.11 1.25 1.21 | 1 0 96 2 | 0 0 | 319k 1.3m | 94.9 13.8 | 1503.1 158.2 | 0.4 4.0 2.8 30.7 |
14:42:53 | 1.18 1.26 1.21 | 3 1 93 3 | 0 0 | 283k 1.2m | 142.2 9.9 | 2275.6 221.2 | 0.7 4.5 2.5 38.2 |
```

2、查看 MySQL 指标

通过 show variables 列出经常关注的核心变量。

通过 show global status 列出相关 MYSQL status 信息。

```
-com
Print MySQL Status(Com_select,Com_insert,Com_update,Com_delete).

-hit
Print Innodb Hit%.

Hit% : (Innodb_buffer_pool_read_requests - Innodb_buffer_pool_reads) / Innodb_buffer_pool_read_requests * 100%

-innodb_rows
Print Innodb Rows Status(Innodb_rows_inserted/updated/deleted/read).

-innodb_pages
Print Innodb Buffer Pool Pages Status(Innodb_buffer_pool_pages_data/free/dirty/flushed)

-innodb_data
Print Innodb Data Status(Innodb_data_reads/writes/read/written)

-innodb_log
Print Innodb Log Status(Innodb_os_log_fsyncs/written)

-innodb_status
Print Innodb Status from Command: 'Show Engine Innodb Status' (history list/ log
unflushed/unchecked bytes/ read views/ queries inside/queued)
```

另外, -innodb_status 的信息来自如下:

```

$mysql -uroot -e "show engine innodb status" | grep -E -A4 -B1 --color '^TRANSACTIONS|LOG|ROW OPERATIONS'
-----
TRANSACTIONS
-----
Trx id counter 5FBD1896F
Purge done for trx's n:o < 5FBD165B2 undo n:o < 0
History list length 205
--
LOG
--
Log sequence number 1341571448062
Log flushed up to   1341571433421
Last checkpoint at  1341334896946
--
ROW OPERATIONS
-----
0 queries inside InnoDB, 0 queries in queue
1 read views open inside InnoDB
Main thread process no. 26735, id 1195481408, state: sleeping

```

其中:

$\text{log unflushed} = \text{Log sequence number} - \text{Log flushed up to}$

$\text{uncheckpointed bytes} = \text{Log sequence number} - \text{Last checkpoint at}$

```

-T,--threads
Print Threads Status(Threads_running,Threads_connected,Threads_created,Threads_cached).

-B,--bytes
Print Bytes received from/send to MySQL(Bytes_received,Bytes_sent).

-mysql
Print MySQLInfo (include -t,-com,-hit,-T,-B).

-innodb
Print InnoDBInfo(include -t,-innodb_pages,-innodb_data,-innodb_log,-innodb_status)

```

```
orzdaba -mysql -innodb_rows

=====
Welcome to use the orzdaba tool !
Yep...Chinese English~
===== Date : 2012-07-29 =====

HOST: IP:
DB :
Var : binlog_format[MIXED] max_binlog_cache_size[2G] max_binlog_size[500M]
max_connect_errors[50000] max_connections[1000] max_user_connections[950]
open_files_limit[65535] sync_binlog[0] table_definition_cache[2048]
table_open_cache[2048] thread_cache_size[256]

innodb_adaptive_flushing[ON] innodb_adaptive_hash_index[ON] innodb_buffer_pool_size[16G]
innodb_file_per_table[ON] innodb_flush_log_at_trx_commit[2] innodb_flush_method[O_DIRECT]
innodb_io_capacity[200] innodb_lock_wait_timeout[100] innodb_log_buffer_size[200M]
innodb_log_file_size[100M] innodb_log_files_in_group[4] innodb_max_dirty_pages_pct[50]
innodb_open_files[60000] innodb_read_io_threads[4] innodb_thread_concurrency[16]
innodb_write_io_threads[4]

----- -QPS- -TPS- -Hit%- ---innodb rows status--- -----threads----- ----bytes----
time ins upd del sel iud lor hit ins upd del read run con cre cac recv send
15:11:07 0 0 0 0 0 0 100.00 0 0 0 0 0 0 0 0 0
15:11:08 5 14 0 1147 19 79049 99.50 5 11 0 75886 6 223 0 190 242k 1.4m
15:11:09 8 50 0 1036 58 93991 99.37 8 45 0 66044 5 223 0 190 283k 1.6m
15:11:10 2 5 0 840 7 166306 99.96 2 5 0 224564 4 223 0 190 219k 1001k
15:11:11 3 18 0 901 21 32072 98.24 3 17 0 37971 4 223 0 190 245k 1.4m
15:11:12 6 25 0 635 31 150317 99.85 6 21 0 66086 4 223 0 190 233k 1.2m
15:11:13 4 27 0 820 31 85211 99.84 4 27 0 70329 3 223 0 190 285k 1.4m
15:11:14 6 30 0 642 36 21604 98.91 6 26 0 48047 5 223 0 190 231k 1.1m
15:11:15 2 32 0 587 34 35922 99.49 2 29 0 175949 3 223 0 190 221k 1.1m
15:11:16 5 34 0 625 39 56276 99.80 5 30 0 33564 4 223 0 190 229k 1.0m
15:11:17 2 24 0 583 26 68676 99.80 2 23 0 83928 4 223 0 190 206k 994k
15:11:18 11 46 0 636 57 60332 99.66 11 42 0 57969 5 223 0 190 260k 1.2m
15:11:19 4 20 0 763 24 26243 99.49 4 20 0 103495 4 223 0 190 255k 1.4m
15:11:20 7 37 0 637 44 67901 99.79 7 32 0 62859 3 223 0 190 271k 1.3m
15:11:21 3 37 0 836 40 18453 99.04 3 36 0 39502 3 223 0 190 262k 1.3m
```

```
orzdaba -innodb

=====
Welcome to use the orzdaba tool !
Yep...Chinese English~
===== Date : 2012-07-29 =====

HOST: IP:
DB :
Var : binlog_format[MIXED] max_binlog_cache_size[2G] max_binlog_size[500M]
max_connect_errors[50000] max_connections[1000] max_user_connections[950]
open_files_limit[65535] sync_binlog[0] table_definition_cache[2048]
table_open_cache[2048] thread_cache_size[256]

innodb_adaptive_flushing[ON] innodb_adaptive_hash_index[ON] innodb_buffer_pool_size[16G]
innodb_file_per_table[ON] innodb_flush_log_at_trx_commit[2] innodb_flush_method[O_DIRECT]
innodb_io_capacity[200] innodb_lock_wait_timeout[100] innodb_log_buffer_size[200M]
innodb_log_file_size[100M] innodb_log_files_in_group[4] innodb_max_dirty_pages_pct[50]
innodb_open_files[60000] innodb_read_io_threads[4] innodb_thread_concurrency[16]
innodb_write_io_threads[4]

----- ---innodb bp pages status--- -----innodb data status--- --innodb log-- his --log(byte)-- read ---query---
time data free dirty flush reads writes read written fsyncs written list uflush uckpt view inside que
15:15:55 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15:15:56 1017426 0 49127 0 99 31 1.5m 27k 1 27k 19 0 222.1m 2 1 0
15:15:57 1017425 0 49138 0 215 41 3.4m 31k 1 31k 38 0 222.1m 2 1 0
15:15:58 1017425 0 49148 0 130 27 2.0m 21k 1 21k 46 0 222.1m 1 0 0
15:15:59 1017425 0 49155 0 174 31 2.7m 25k 1 25k 59 2k 222.1m 2 1 0
15:16:00 1017425 0 49167 0 157 61 2.5m 46k 1 46k 83 2k 222.2m 1 0 0
15:16:01 1017424 0 49183 0 244 67 3.8m 63k 1 63k 112 8k 222.2m 1 0 0
15:16:02 1017422 0 49077 120 264 192 4.1m 3.8m 1 58k 145 2k 222.2m 4 4 0
15:16:03 1017424 0 49092 0 267 75 4.2m 55k 1 55k 182 4k 222.2m 1 0 0
15:16:04 1017425 0 49101 0 149 41 2.3m 36k 1 36k 200 2k 222.2m 1 0 0
15:16:05 1017424 0 49117 23 205 100 3.2m 812k 2 76k 7 5k 221.8m 1 0 0
15:16:06 1017428 0 49126 0 347 41 5.4m 37k 1 37k 23 7k 221.8m 1 0 0
15:16:07 1017435 0 49134 0 424 37 6.6m 32k 1 32k 38 10 221.8m 3 2 0
15:16:08 1017436 0 49028 119 488 176 7.6m 3.8m 1 48k 67 4k 221.8m 2 2 0
15:16:09 1017439 0 49036 0 228 33 3.6m 25k 1 25k 82 2k 221.8m 2 1 0
```

3、查看 MySQL 的响应时间

-rt	Print MySQL DB RT(us).
-----	------------------------

通过调用 [tcprstat](#) 来监控 MySQL 的响应时间。

-rt 中列的具体意义参考 <http://www.percona.com/docs/wiki/tcprstat:start> 中的解释。

```

$orzdab -lazy -rt

=====
| Welcome to use the orzdab tool ! |
| Yep...Chinese English~ |
|===== Date : 2012-07-29 =====|
=====

HOST:  IP:
DB :
Var : binlog_format[ROW] max_binlog_cache_size[2G] max_binlog_size[500M]
      max_connect_errors[50000] max_connections[3500] max_user_connections[3000]
      open_files_limit[65535] sync_binlog[0] table_definition_cache[2048]
      table_open_cache[2048] thread_cache_size[256]

      innodb_adaptive_flushing[ON] innodb_adaptive_hash_index[ON] innodb_buffer_pool_size[72G]
      innodb_file_per_table[ON] innodb_flush_log_at_trx_commit[2] innodb_flush_method[O_DIRECT]
      innodb_io_capacity[2000] innodb_lock_wait_timeout[100] innodb_log_buffer_size[200M]
      innodb_log_file_size[1.26953125G] innodb_log_files_in_group[3] innodb_max_dirty_pages_pct[80]
      innodb_open_files[60000] innodb_read_io_threads[8] innodb_thread_concurrency[64]
      innodb_write_io_threads[8]

-----load-avg-----cpu-usage-----swap-----
time | 1m | 5m | 15m | usr | sys | idl | iow | si | so | ins | upd | del | sel | iud | lor | hit | count | avg | 95-avg | 99-avg |
15:40:55 | 2.88 | 2.84 | 2.74 | 3 | 1 | 96 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.00 | 0 | 0 | 0 | 0 |
15:40:56 | 2.88 | 2.84 | 2.74 | 11 | 3 | 86 | 1 | 0 | 0 | 92 | 228 | 0 | 8423 | 320 | 130217 | 99.45 | 8647 | 213 | 187 | 200 |
15:40:57 | 2.88 | 2.84 | 2.74 | 8 | 3 | 89 | 0 | 0 | 0 | 53 | 245 | 0 | 8696 | 298 | 102098 | 99.33 | 9107 | 212 | 188 | 199 |
15:40:58 | 2.88 | 2.84 | 2.74 | 7 | 3 | 89 | 1 | 0 | 0 | 20 | 269 | 0 | 8823 | 289 | 105779 | 99.26 | 9220 | 231 | 193 | 206 |
15:40:59 | 2.88 | 2.84 | 2.74 | 7 | 3 | 89 | 2 | 0 | 0 | 10 | 257 | 0 | 8318 | 267 | 102112 | 99.30 | 8628 | 306 | 194 | 227 |
15:41:00 | 2.73 | 2.81 | 2.73 | 7 | 3 | 90 | 0 | 0 | 0 | 15 | 278 | 0 | 8653 | 293 | 94269 | 99.24 | 8944 | 218 | 191 | 203 |
15:41:01 | 2.73 | 2.81 | 2.73 | 7 | 3 | 89 | 1 | 0 | 0 | 37 | 266 | 0 | 8896 | 303 | 100442 | 99.30 | 9268 | 250 | 192 | 205 |
15:41:02 | 2.73 | 2.81 | 2.73 | 7 | 3 | 89 | 0 | 0 | 0 | 48 | 335 | 0 | 8834 | 383 | 96747 | 99.26 | 9279 | 216 | 192 | 203 |
15:41:03 | 2.73 | 2.81 | 2.73 | 7 | 4 | 89 | 0 | 0 | 0 | 59 | 421 | 0 | 9266 | 480 | 102378 | 99.24 | 9867 | 236 | 191 | 203 |
15:41:04 | 2.73 | 2.81 | 2.73 | 7 | 3 | 88 | 2 | 0 | 0 | 57 | 225 | 0 | 8469 | 282 | 120472 | 99.42 | 8877 | 277 | 193 | 227 |
15:41:05 | 2.67 | 2.79 | 2.73 | 7 | 3 | 89 | 0 | 0 | 0 | 33 | 282 | 0 | 8633 | 315 | 168297 | 99.53 | 9023 | 239 | 197 | 212 |
15:41:06 | 2.67 | 2.79 | 2.73 | 8 | 3 | 89 | 0 | 0 | 0 | 13 | 309 | 0 | 9089 | 322 | 138385 | 99.41 | 9303 | 222 | 196 | 209 |
15:41:07 | 2.67 | 2.79 | 2.73 | 7 | 3 | 90 | 0 | 0 | 0 | 31 | 315 | 0 | 8609 | 346 | 96843 | 99.23 | 8999 | 216 | 190 | 202 |
15:41:08 | 2.67 | 2.79 | 2.73 | 7 | 3 | 90 | 0 | 0 | 0 | 30 | 245 | 0 | 8446 | 275 | 97439 | 99.17 | 8939 | 230 | 193 | 206 |
15:41:09 | 2.67 | 2.79 | 2.73 | 7 | 3 | 85 | 4 | 0 | 0 | 18 | 269 | 0 | 8519 | 287 | 95612 | 99.25 | 8870 | 408 | 104 | 278 |

```

-rt 参数依赖:

(1) 需要安装 [tcprstat](#)

由于安装后运行 [tcprstat](#) 需要 root 用户，或者 sudo 进行，可以通过如下命令来避免:

```

$sudo chown root:root /usr/bin/tcprstat
$sudo chmod u+s /usr/bin/tcprstat
$ll /usr/bin/tcprstat
-rwxr-xr-x 1 root root 1183465 May 28 15:38 /usr/bin/tcprstat

```

默认工具读取 [tcprstat](#) 的代码如下:

```

$grep -n 'my $TCPRSTAT' orzdab
166:my $TCPRSTAT = "/usr/bin/tcprstat --no-header -t 1 -n 0 -p $port";

```

不同的路径可以自行修改。

(2) 需要安装 Perl 的 File::Lockfile 模块

File::Lockfile 模块同时要依赖 Module-Build/version/Class-Data-Inheritable 这 3 个模块，可以通过下面方式进行安装:

安装 version 模块:

```

$wget http://search.cpan.org/CPAN/authors/id/J/JPEACOCK/version-0.99.tar.gz
$tar -zxvf version-0.99.tar.gz
$cd version-0.99
$perl Makefile.PL
$make
$make test
$sudo make install

```

安装 Class-Data-Inheritable 模块:

```

$wget http://search.cpan.org/CPAN/authors/id/T/TM/TMTM/Class-Data-Inheritable-0.08.tar.gz

```

```
$tar -zxvf Class-Data-Inheritable-0.08.tar.gz
$cd Class-Data-Inheritable-0.08
$perl Makefile.PL
$make
$make test
$sudo make install
```

安装 **Module-Build** 模块:

```
$wget http://search.cpan.org/CPAN/authors/id/K/KW/KWILLIAMS/Module-Build-0.31.tar.gz
$tar -zxvf Module-Build-0.31.tar.gz
$cd Module-Build-0.31
$perl Build.PL
$./Build
$./Build test
$sudo ./Build install
```

安装 **File::Lockfile** 模块:

```
$wget http://search.cpan.org/CPAN/authors/id/G/GL/GLORYBOX/File-Lockfile-v1.0.5.tar.gz
$tar -zxvf File-Lockfile-v1.0.5.tar.gz
$cd File-Lockfile-v1.0.5
$perl Build.PL
$perl ./Build
$perl ./Build test
$sudo perl ./Build install
```

可以通过如下脚本，检查安装了哪些 **perl** 模块的脚本:

```
$cat check_module.pl
#!/usr/bin/perl
use ExtUtils::Installed;
my $inst = ExtUtils::Installed->new();
print join "\n", $inst->modules();
print "\n";
```

4、其他

(1) MySQL 相关参数配置

-P,--port	Port number to use for mysql connection(default 3306).
-S,--socket	Socket file to use for mysql connection.

用于指定端口或者 **socket**。

如果需要 **MYSQL** 的其他参数要添加，直接改代码:

```
$grep -n -A1 'my $MYSQL' orzdba
166:my $MYSQL = qq{mysql -s --skip-column-names -uroot -P$port };
167:$MYSQL .= qq{-S$socket } if defined $socket;
```

(2) 控制输出间隔和次数，以及是否输出颜色。

-h,--help	Print Help Info.
-----------	------------------

<code>-i,--interval</code>	Time(second) Interval.
<code>-C,--count</code>	Times.
<code>-t,--time</code>	Print The Current Time.
<code>-nocolor</code>	Print NO Color.

(3) 日志输出

<code>-L,--logfile</code>	Print to Logfile.
<code>-logfile_by_day</code>	One day a logfile,the suffix of logfile is 'yyy-mm-dd'; and is valid with <code>-L</code> .

用于将数据输出到日志文件-L 后面带输出的日志文件，另外如果加上-`logfile_by_day` 参数的话，将按天输出日志。