IOSTAT Analyzer for HPUX User Guide

文档历史：

版本	修改日期	修改者	注释
v1.0	**2014.01.07**	罗雪原	初始化文档。

目录

IOSTAT Analyzer for HPUX User Guide.....1
一、安装 strawberry perl for windows.....4
二、配置 PATH 环境变量.....4
三、安装 perl 相关模块.....5
1. 进入 cmd，运行 cpan.....5
2. 安装脚本需要模块.....6
四、iostat 的信息收集方法.....7
1. Oracle 提供的 OS Watcher 工具，简称 OSW.....7
2. 自定义脚本.....7
五、分析程序介绍和使用.....9
1. 配置文件.....9
2. 分析脚本 2014.01.07 版本.....10
3. 使用脚本生成 excel 图.....18
4. 生成的图像效果.....20

一、安装 strawberry perl for windows

Strawberry Perl 下载地址： http://strawberryperl.com/

安装过程比较简单，不在赘述。

二、配置 PATH 环境变量

当前使用的是 5.18.1 版本 perl，安装完毕后，配置环境变量，确认这个路径在其他可能包含 perl 软件的路径之前。

在 CMD 窗口里，使用 path 检查环境变量，使用 perl -version 查看 perl 版本。

```
C:\Users\Milo> path
PATH=D:\strawberry\perl\bin;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;D:\oracle\product\10.2.0\db_1\bin;
```

```
C:\Users\Milo> perl -version

This is perl 5, version 18, subversion 1 (v5.18.1) built for MSWin32-x64-multi-thread

Copyright 1987-2013, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on
this system using "man perl" or "perldoc perl".  If you have access to the
Internet, point your browser at http://www.perl.org/, the Perl Home Page.
```

三、安装 perl 相关模块

使用 cpan 命令进入 cpan 命令行，在线安装模块，此时需要**确保电脑连接互联网**。

1. 进入 cmd，运行 cpan

```
C:\Users\Milo> cpan
```

如果进入的时候，遇到下面的问题，这是因为 oracle 10g 自带的 perl 与当前的 perl 不兼容造成的。

```
C:\Users\Milo>cpan
Use of :unique is deprecated at D:\oracle\product\10.2.0\db_1\perl\5.8.3\lib/MSWin32-x64-multi-thread/Config.pm line 39.

Use of :unique is deprecated at D:\oracle\product\10.2.0\db_1\perl\5.8.3\lib/MSWin32-x64-multi-thread/Config.pm line 80.

Perl lib version (v5.8.3) doesn't match executable version (v5.18.1) at D:\oracle\product\10.2.0\db_1\perl\5.8.3\lib/MSW
in32-x64-multi-thread/Config.pm line 32.
Compilation failed in require at D:\oracle\product\10.2.0\db_1\perl\5.8.3\lib/CPAN.pm line 12.
BEGIN failed--compilation aborted at D:\oracle\product\10.2.0\db_1\perl\5.8.3\lib/CPAN.pm line 12.
Compilation failed in require at C:/strawberry/perl/lib/App/Cpan.pm line 218.
BEGIN failed--compilation aborted at C:/strawberry/perl/lib/App/Cpan.pm line 218.
Compilation failed in require at C:\strawberry\perl\bin/cpan.bat line 22.
BEGIN failed--compilation aborted at C:\strawberry\perl\bin/cpan.bat line 22.
```

直接把系统里的 PERL5LIB 环境变量改名，例如：

确定后，重新打开 CMD 窗口执行，就不会出现之前的问题了。

或者使用临时的解决办法（每次重新打开 cmd 进入 cpan 都需要这么做）：

```
C:\Users\Milo>set PERL5LIB=""

C:\Users\Milo>echo %PERL5LIB%
""

C:\Users\Milo>cpan

cpan shell -- CPAN exploration and modules installation (v2.00)
Enter 'h' for help.


cpan>
```

2. 安装脚本需要模块

安装前，确定计算机联网，进行在线安装模块：

```
cpan> install Excel::Writer::XLSX
```

最后如果看到类似字样，说明这个模块已经正确安装了：

```
Appending installation info to C:\strawberry\perl\lib/perllocal.pod
  JMCNAMARA/Excel-Writer-XLSX-0.76.tar.gz
  C:\strawberry\c\bin\dmake.exe install UNINST=1 -- OK
```

四、iostat 的信息收集方法

要想使用这个工具做 iostat 的分析，那么，可以使用两个办法收集：

1. Oracle 提供的 OS Watcher 工具，简称 OSW

具体使用方法详见 OSW 的使用手册。

2. 自定义脚本

需要有特定格式的输出，下面提供一个简易的脚本仅供参考：

iostat_hpux.sh:

```
# Collect IOSTAT info for HPUX
# Time interval getting from input
interval=$1;

# while loop to execute iostat with define interval
while true
do
echo "zzz ***"`date`;
iostat $interval 1;
done
```

运行方法：脚本名 收集间隔

```
$ nohup sh iostat_hpux.sh 30 >> /tmp/iostat.log &
```

例如，上述命令会每 30 秒收集一次 iostat，只要把脚本放后台就可以持续以某个间隔收集，日志被记录在/tmp/iostat.log 里，不需要时可以 kill 掉这个后台运行的脚本，但是请确认 kill 的准确。

以上脚本运行成功的前提：

a. HPUX 系统；
b. 运行脚本用户有 iostat 命令权限；
c. date 命令格式为英文格式；

收集信息格式类似：

```
zzz ***Thu Nov 14 11:22:43 EAT 2013

    disk3      335    36.0     1.0
    disk5      281    26.0     1.0
   disk19        0     0.0     1.0
   disk20      449    29.0     1.0
   disk21      355    24.0     1.0
   disk24        0     0.0     1.0
   disk25      212    16.0     1.0
 c12t0d5         0     0.0     1.0
 c12t0d6         0     0.0     1.0
   disk34        0     0.0     1.0
   disk35        0     0.0     1.0
 c14t0d5         0     0.0     1.0
 c14t0d6         0     0.0     1.0
 c11t0d5         0     0.0     1.0
 c11t0d6         0     0.0     1.0
 c13t0d5         0     0.0     1.0
 c13t0d6         0     0.0     1.0

zzz ***Thu Nov 14 11:23:43 EAT 2013

    disk3       38     5.9     1.0
    disk5       36     5.0     1.0
   disk19        0     0.0     1.0
   disk20       33     3.0     1.0
   disk21        0     0.0     1.0
   disk24        0     0.0     1.0
   disk25       49     4.0     1.0
 c12t0d5         0     0.0     1.0
 c12t0d6         0     0.0     1.0
   disk34        0     0.0     1.0
   disk35        0     0.0     1.0
 c14t0d5         0     0.0     1.0
 c14t0d6         0     0.0     1.0
 c11t0d5         0     0.0     1.0
 c11t0d6         0     0.0     1.0
 c13t0d5         0     0.0     1.0
 c13t0d6         0     0.0     1.0
```

五、分析程序介绍和使用

分析程序包括 **配置文件** 和 **分析脚本** 两个部分。

1. 配置文件

里面写上在 iostat 里需要过滤或关注的 disk 名，如果不想要的 disk 可以删除或加#注掉。

配置 config.cfg 文件格式如下：

```
################################################################################
# This is a iostat analyzer program configuration file
# NOTE:
# 1. You can use "#" to comment the disk(s) you don't want.
# 2. It doesn't matter when you keep spaces before disk name,
#       but we dont recommend this,as it might let config file looks messy.
#
################################################################################

disk24
disk21
disk20
disk19
disk25
disk5
disk3
```

2. 分析脚本 2014.01.07 版本

io_analyzer_hpux.pl

```
#!/usr/bin/perl -w
# Program name: IOSTAT ANALYZER for hpux
# Purpose: From the Oracle(OSW) Tools, IOstat does not support generate gif of disk io, so this script is used to generate the disk gif, simple by this script.
# Author: Milo Luo
#
# Date                  Modifier          Comments
# ------------- ------------- ------------------------------------------------------
# Nov.08 2013      Milo Luo          Initialize the script.
# Nov.15 2013      Milo Luo          Replace variables with hard code on disks.
# Nov.16 2013      Milo Luo          Add the configure file.
# Nov.17 2013      Milo Luo          Add auto-plotted trend lines.
# Nov.18 2013      Milo Luo          Add handling to config file start with '#' and spaces before contents.
# Nov.25 2013      Milo Luo          Add range selected and optimize the structure of script.
# Jan.07 2014      Milo Luo          Optimize the GUI and test with strawberry perl 5.18.1.1 on Win 7 64bit.
#
#
#

use strict;
use diagnostics;
use Excel::Writer::XLSX;
use Excel::Writer::XLSX::Utility;
use feature "switch";
no warnings 'experimental::smartmatch';

############# Variables using handling iostat file and excel file ###################

# line contents
my $line = "";

# Define the disks you care about on iostat file.
my %mydisk;

# time lines
my $minus ="";
```

```
# if it's the init step or not
my $init_flag = 0;

# begin column
my $col=0;

# Define 2nd row as real iostat data write to , because row #1 will always be time lines
my $rows = 2;

# Iostat file row count
my $cnt = 0;

# Define the config file
my $conf="config.cfg";

# Define the excel file name
my $excel_name='iostat_result.xlsx';

################################ End Variables #######################################

########################################
# Define range flag
## 0 -- no start point
## 1 -- in
## 2 -- out
########################################

my $range_flag = 0;
my $begin_range = 'ALL DATA';
my $end_range = 'ALL DATA';

################################                                                    END
################################################


############################## Read config file #######################################

# Read configure file to get which disks you care about
open(FH1,"< $conf") or die "Can NOT open configure file: $conf !";

# Read the configure file
while ($line = <FH1>) {
      next if ($line =~ m/^\s*$/);
```

```
        next if ($line =~ m/^#$/);
        #if ((split(/\s+/,$line))[1] =~ m/disk/) {
        if ($line =~ m/^\s+disk/) {
                $mydisk{(split(/\s+/,$line))[1]} = 0;
        }elsif ($line =~ m/^disk/) {
                $mydisk{(split(/\s+/,$line))[0]} = 0;
        }

}

# Close File Header
close(FH1);
###################################                                    END
##############################################



########################## Get diskname & size
#######################################
# Aquired the hash size and diskname
my $size += scalar keys %mydisk;
my @diskname = keys %mydisk;

################################                                       End
##############################################



################### Read iostat file from command line ############################
# Define the iostat data filename in HPUX
my $fname=$ARGV[0];
################################                                       END
##############################################


$line = "";
$cnt = 0;

print "\n############################\n";
print "# IOSTAT for HPUX Analyzer #";
print "\n############################\n";
# Identified disks
print "\n*********************\n";
print "* Recongize disk(s): \n";
print "*********************\n";
```

```
for ($cnt=0;$cnt < $size; $cnt += 1) {
        print "$diskname[$cnt] \n";
}
print "Note: All the disks read from configuration file.\n";
print "**********************\n";

print "\n------------------ Choice Menu ----------------------------\n";
print "| 1. Generate all range graph of current iostat file.          |\n";
print "| 2. Generate specify range graph of current iostat file. |\n";
print "-------------------------------------------------------------\n";
print "Choice => ";

# Get customer input
chomp(my $choice=<STDIN>);

print "\nYour Choice: [$choice] \n";

given($choice){
        when(1) { }
        when(2) {
                print "\nPlease input the begin date format: (Nov 11 20:00)\n";
                # Remove the last new line chraracter
                chomp ($begin_range=<STDIN>);

                print "Please input the end date format: (Nov 11 20:00)\n";
                # Remove the last new line chraracter
                chomp ($end_range=<STDIN>);

                # Formatted range value
                $begin_range = ucfirst $begin_range;
                $end_range = ucfirst $end_range;

                # Open the iostat file for check the begin date range
                open(IOFILE, "< $fname") or die("Can't Open iostat file $fname !");

                my @match_begin = grep /^zzz.*$begin_range.*/, <IOFILE>;
                my $begin_cnt = @match_begin;
                close(IOFILE);

                # Open the iostat file for check the ending date range
                open(IOFILE, "< $fname") or die("Can't Open iostat file $fname !");
                my @match_end = grep /^zzz.*$end_range.*/, <IOFILE>;
                my $end_cnt = @match_end;
```

```
close(IOFILE);

print "\n\n----------------------------------------\n";
print "IOstat File Action: \n";
print "----------------------------------------\n";

# Check about the occurance of the begin and ending range
# if both ranges are not identical
if ($begin_cnt != 1 && $end_cnt != 1) {

    # print begin flag matching lines
    print "\n++++++++++++++++++++++++++++++++++++++++++++\n";
    print "Error ocurr!\n";
    print "++++++++++++++++++++++++++++++++++++++++++++\n";
    print "==> Begin count: $begin_cnt\n";
    print "Begin flag matching those lines:\n";
    print @match_begin;

    print "\n++++++++++++++++++++++++++++++++++++++++++++\n";
    # print end flag matching lines
    print "==> End count: $end_cnt\n";
    print "End flag matching those lines:\n";
    print @match_end;

    exit -1;

# if begin range are not identical
}elsif ($begin_cnt != 1) {

    # print begin flag matching lines
    print "++++++++++++++++++++++++++++++++++++++++++++\n";
    print "Error ocurr!\n";
    print "++++++++++++++++++++++++++++++++++++++++++++\n";
    print "==> Begin count: $begin_cnt\n";
    print "Begin flag matching those lines:\n";
    print @match_begin;
    exit -2;

# if ending range are not identical
}elsif ($end_cnt != 1) {

    # print end flag matching lines
    print "\n++++++++++++++++++++++++++++++++++++++++++++\n";
    print "Error ocurr!\n";
```

```
                print "++++++++++++++++++++++++++++++++++++++++++++\n";
                print "==> End count: $end_cnt\n";
                print "End flag matching those lines:\n";
                print @match_end;
                exit -3;

            }else{
                print "\nRanges seems to be ok!!!\n";
                print "Match begin line: ", @match_begin;
                print "Match end line:      ", @match_end;
                print "\nStarting Analyzing...\n";
                }
          }
          default { print "\nInput error!\n"; }
}


######################## Open Excel to load data
#######################################
# Open a Excel (xlsx format) for resultset, make sure this file is not opening.
my $Excel =    Excel::Writer::XLSX->new($excel_name);
my $Sheet =    $Excel-> add_worksheet();
################################### End
################################################

# Open the iostat file to fill data
open(IOFILE, "< $fname") or die("Can't Open iostat file $fname !");



print "\n\n----------------------------------------\n";
print "Excel Action: \n";
print "----------------------------------------\n";
# Begin fill data
print "Starting fill data into excel!\n";
while ($line = <IOFILE>) {
            $cnt += 1;
            #print "Processing $cnt lines.\n";
            next if ($line =~ m/^\s*$/);

            # Set flag for begin and end
            if ($choice == 1 || ($line =~ m/^zzz.*$begin_range.*$/ && $range_flag == 0) ){
                $range_flag = 1;
            }elsif($line =~ m/^zzz.*$end_range.*$/ && $range_flag == 1 && $choice == 2){
                # Determine if stop lookup iostat file immediate
```

```
                $range_flag = 2;
                last;
        }

        if ($line =~ m/\d{2}:\d{2}:\d{2}/ && $init_flag == 0 && $range_flag > 0){
                # Initialize the disk names
                for ($rows=1; $rows<= $size; $rows++) {
                        $Sheet->write($rows,$col,$diskname[$rows-1]);
                }
                $col += 1;
                $init_flag=1;
                $minus = (split(/\s+/,$line))[4];
        }
        elsif ($line =~ m/\d{2}:\d{2}:\d{2}/ && $init_flag == 1 && $range_flag > 0){
                # First flush the former result
                $Sheet->write(0,$col,$minus);
                #print "$minus\n";

                for ($rows=1; $rows<= $size; $rows++) {
                        $Sheet->write($rows,$col,$mydisk{$diskname[$rows-1]});
                }
                $col += 1;
                $minus = (split(/\s+/,$line))[4];
        }
        elsif ($line =~ m/disk/) {
                # store a group of values
                if ( exists($mydisk{(split(/\s+/,$line))[1]}) ) {
                        $mydisk{(split(/\s+/,$line))[1]} = (split(/\s+/,$line))[2];
                }

        }


}

# last flush
$Sheet->write(0,$col,$minus);
for ($rows=1; $rows<= $size; $rows++) {
        $Sheet->write($rows,$col,$mydisk{$diskname[$rows-1]});
}

# Close the iostat file
close(IOFILE);
```

```
######################### Plotted the graph ###################################

# Add a chart object
my $chart = $Excel -> add_chart( type    => 'line', embedded => 1);
my $colname = xl_col_to_name($col);
#print "\n","column name is ", $colname, "\n";
print "Starting plot the graph!\n";

# Add a chart title and some axis labels.
$chart -> set_title ( name => 'Results of iostat analysis on hpux' );
$chart -> set_x_axis( name => 'Time Lines' );
$chart -> set_y_axis( name => 'Kilobytes Per Second(bps)' );

# Set an Excel chart style. Colors with white outline and shadow.
#$chart -> add_series( values => '=Sheet1!$B$2:$E$2', trendline => {type => 'linear'} );
$chart -> set_style( 2 );
for ($rows=1; $rows<= $size; $rows++) {
        #$chart -> add_series( name => 'disk1', categories => 'Sheet1!$A$1:$ASC$1', values =>
'=Sheet1!$B$2:$ASC$2' );
        my $tmp_rl=$rows+1;
        $chart -> add_series( name => $diskname[$rows-1], categories =>
'Sheet1!$B$1:$'.$colname.'$'.$rows, values =>
'=Sheet1!$B'.'$'.$tmp_rl.':$'.$colname.'$'.$tmp_rl );
        #print $diskname[$rows-1],"\n";
}

# Insert the chart into the worksheet (with an offset).
#$worksheet->insert_chart( 'D2', $chart, 25, 10 );
$Sheet ->insert_chart( 'D'.($size+5), $chart,0,0,1.8,1.5 );
# clean up after ourselves
$Excel -> close();

print "Complet the Mission!\n";

######################### End Program ###################################
```

3. 使用脚本生成 excel 图

脚本可以生成 iostat 文件里所有时间的 disk 的 io 图像，也可以过滤生成脚本里出现的一个时间段图像。

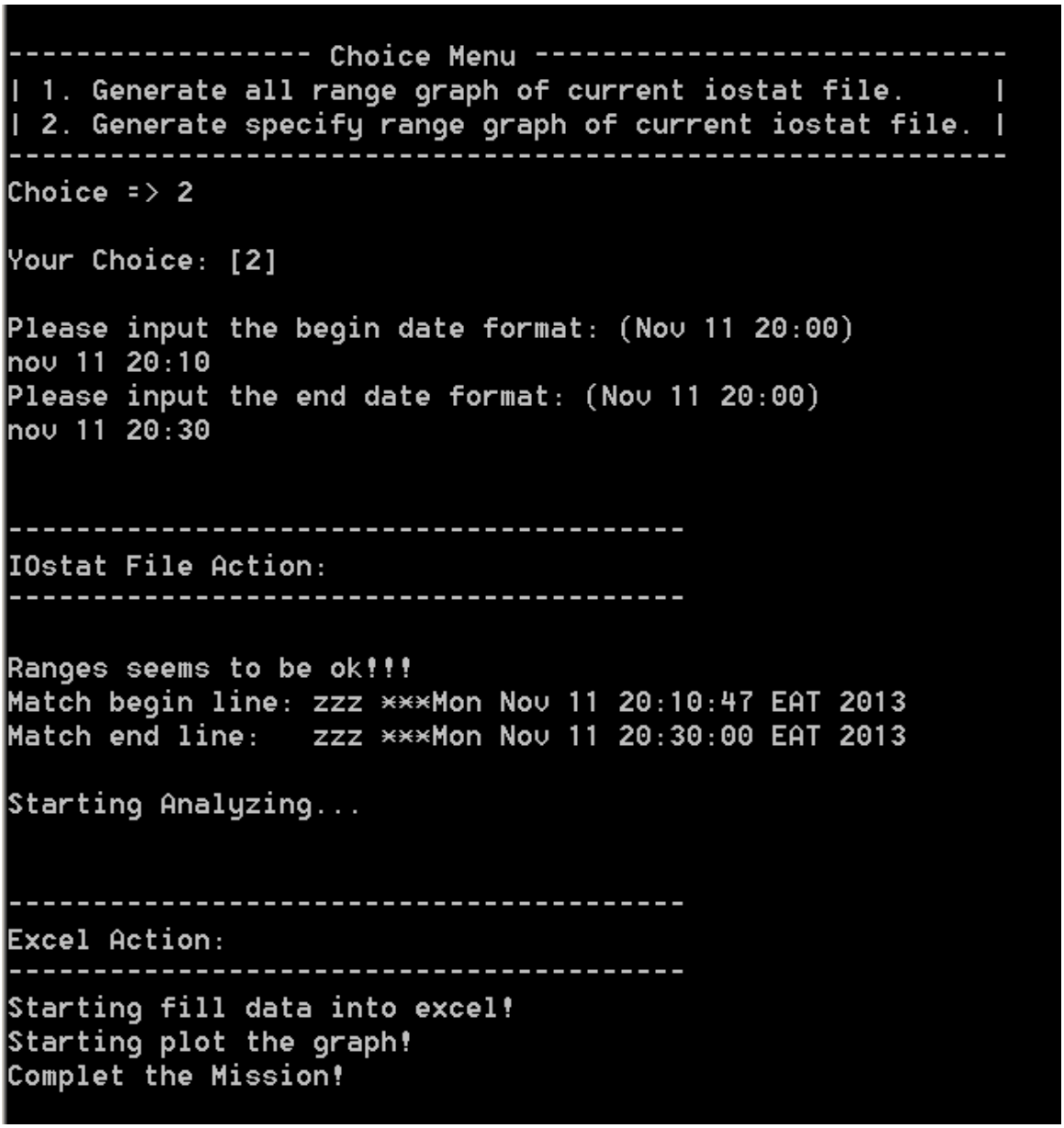
使用方法:
perl io_analyzer_hpux.pl <iostat 的文件名>

注意：运行前，分析脚本里写的 excel 文件不能被打开，否则程序会中断。

下面显示的是 生成所有文件里**所有出现时间**的图像的形式：
可以看到，首先会程序读取配置文件里设置的 disk 信息（显示在 excel 结果里的 disk），
其次，提供是否生成 所有时间段 iostat 图像，还是 某个时间段的 iostat 图像。
下面显示的是生成所有时间段图像的程序输出。

```
C:\Users\Milo>perl io_analyzer_hpux.pl case7.dat

##############################
# IOSTAT for HPUX Analyzer #
##############################

**********************
* Recongize disk(s):
**********************
disk24
disk20
disk21
disk19
disk25
disk3
disk5
Note: All the disks read from configuration file.
**********************

------------------ Choice Menu ----------------------------
| 1. Generate all range graph of current iostat file.     |
| 2. Generate specify range graph of current iostat file. |
-----------------------------------------------------------
Choice => 1

Your Choice: [1]


---------------------------------------
Excel Action:
---------------------------------------
Starting fill data into excel!
Starting plot the graph!
Complet the Mission!
```

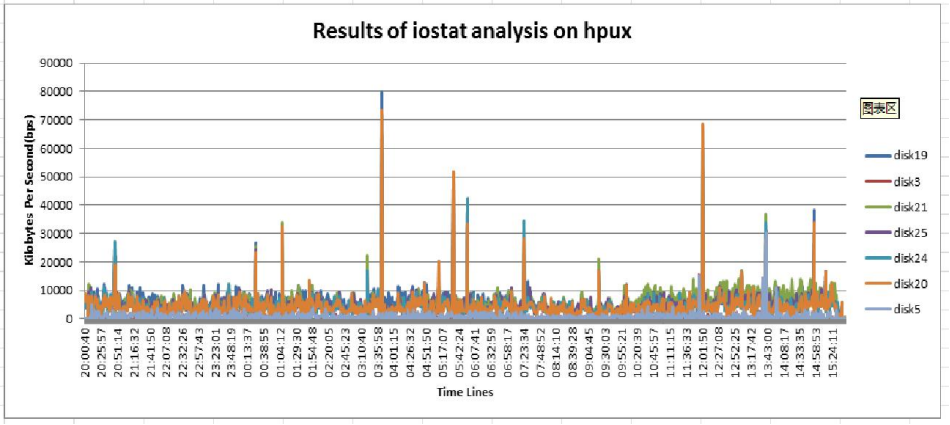
这里显示的是按某个时间段生成图像的输出：
可以看到，如果程序**正常会把匹配的开始行**和**结束行**打印出来。



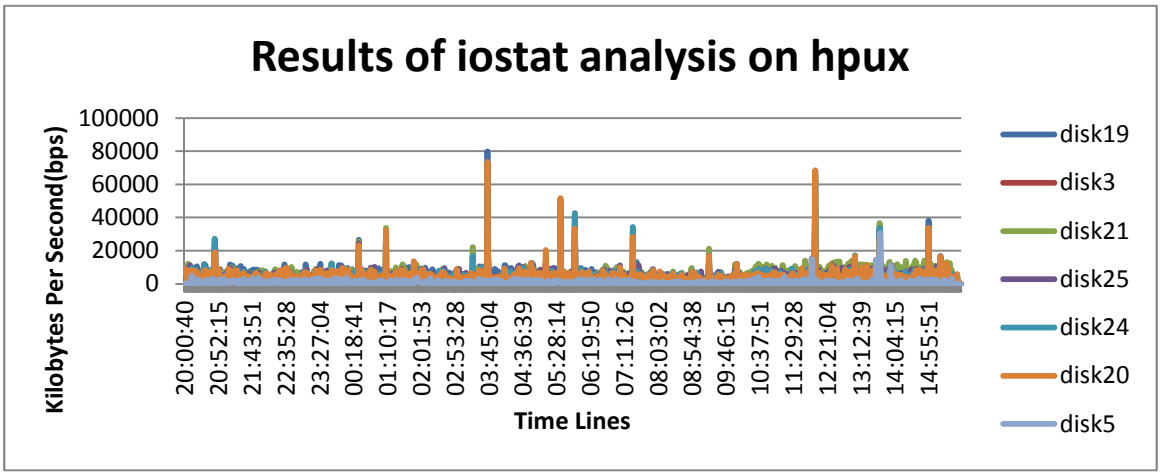
4. 生成的图像效果

1) 所有的 Excel：(Nov 11 20:00 ~ Nov 12 15:45)

	20:00:40	20:01:40	20:02:41	20:03:42	20:04:42	20:05:43	20:06:44	20:07:45	20:08:45	20:09:46	20:10:47	20:11:47	20:12:48	20:13:49	20:14:50	20:15
disk19	8111	4055	5961	6147	8111	0	9758	7319	9093	10080	7059	9600	2478	9537	8630	
disk3	166	166	190	192	23	18	241	129	229	318	853	4620	323	75	835	
disk21	8681	7731	5741	4214	8238	0	12040	8269	11565	10080	4298	8768	3059	6907	6210	
disk25	6610	7068	8081	8178	5228	48	9401	4230	7224	11046	4847	8192	3137	9066	5941	
disk24	4059	8115	8035	8115	8112	1	3580	3462	6337	6048	5296	9439	2843	6165	4363	
disk20	6068	8571	8519	7304	5086	33	8136	6341	5815	6641	4377	8177	2119	5672	4782	
disk5	166	166	188	190	23	16	229	123	189	316	839	4987	315	71	1170	

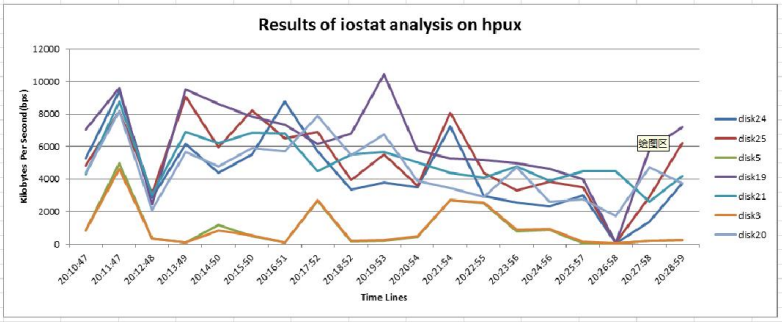


iostat 图像：



2) 某个时间的 Excel：（Nov 11 20:10 ~ Nov 20:30）

	20:10:47	20:11:47	20:12:48	20:13:49	20:14:50	20:15:50	20:16:51	20:17:52	20:18:52	20:19:53	20:20:54	20:21:54	20:22:55	20:23:56	20:24:56	20:25:57	20:26:58	20:27:58	20:28:59
disk24	5296	9439	2843	6165	4363	5513	8782	5701	3324	3802	3488	7247	2947	2573	2290	3008	51	1394	3733
disk25	4847	8192	3137	9066	5941	8247	6528	6882	3930	5513	3616	8087	4366	3310	3859	3488	102	2915	6202
disk5	839	4987	315	71	1170	504	94	2678	143	205	453	2682	2489	807	861	36	51	175	220
disk19	7059	9600	2478	9537	8630	7826	7360	6141	6804	10461	5760	5256	5164	4957	4612	4000	51	5799	7184
disk21	4298	8768	3059	6907	6210	6859	6816	4477	5499	5671	5026	4397	4089	4756	3899	4486	4486	2598	4188
disk3	853	4620	323	75	835	536	98	2702	211	223	503	2713	2541	877	943	134	59	185	245
disk20	4377	8177	2119	5672	4782	5891	5713	7891	5450	6732	3883	3423	2916	4729	2589	2737	1729	4712	3766



iostat 图像：