Formatez vos résultats IOMETER

IOMETER est un logiciel permettant de benchmarker votre stockage en le sollicitant avec des conditions proches d'une production voire même un test de stress intensif ! Cependant les résultats obtenus avec IOMETER sont assez catastrophiques en terme de mise en page.

Bref, trève de blabla, vous observez que le résultat obtenu est immédiatement exploitable !

Voici donc le script PERL qui permet cette merveille !

[perl]

```
#!/usr/bin/perl
#
# IOMeter results parser, 2005, Paul Venezia
#
#       Simply run this script in the directory containing the CSV output of
IOMeter.
#       Results are organized by filename.
#
#

use File::Glob ':glob';
use Data::Dumper;

@files = <*.csv>;
my %results;
my $parmcount = 1;
my $lcount = 0;
foreach my $file (@files) {
        undef @lines;
        open (LOG, "<$file");
        my @lines = <LOG>;
        foreach my $line (@lines) {
                $lcount++;
                if ($line =~ /'Access\sspecification\sname.+/) {
                        $line = @lines[$lcount];
                        $line =~ /(.+),\d/;
                        #%results->{$file}->{test} = "$1";
                        $thistest = "$1";
                        #print $thistest;
```

```
                }
                if ($line =~ /ALL,All.+/) {
                        #$line = @lines[$lcount];
                        @thisres = (split(",", $line));
                        %results->{$file}->{$thistest}{IOps} = @thisres[6];
                            %results->{$file}->{$thistest}{ReadIOps} =
@thisres[7];
                            %results->{$file}->{$thistest}{WriteIOps} =
@thisres[8];
                        %results->{$file}->{$thistest}{MBps} = @thisres[9];
                            %results->{$file}->{$thistest}{ReadMBps} =
@thisres[10];
                        %results->{$file}->{$thistest}{CPU} = @thisres[45];
                }

        }
$parmcount = 0;
$lcount = 0;

}

sub swrite {
  my $fmt = shift(@_);
  $^A = '';
  formline($fmt,@_);
  return $^A;
}

my %output;
foreach my $logfile ( sort keys %results ) {
        #$TXT .= "$logfile: \n";
        $header = 1;
        foreach $test ( sort keys %{$results{$logfile}} ) {
                foreach $tsize ( sort keys %{$results{$logfile}{$test}} ) {
                                $output{$test}{$logfile}{$tsize} =
swrite("@<<<<<<<<<<<<<<<<<<", $results{$logfile}{$test}{$tsize});
                }

        }
}

#print Dumper %results;
foreach $testsize ( sort keys %output ) {
        $TXT .= "$testsize\n";
        $header = 1;
```

```
        foreach $unit ( sort keys %{$output{$testsize}} ) {
                if ($header) {
                        $TXT .= "\t\t";
                                        foreach $test ( sort keys
%{$output{$testsize}{$unit}} ) {
                                $TXT .= swrite("@>>>>>>>>>>>>>> @<<<<<<",
$test);
                        }
                        $TXT .= "\n";
                        undef $header;
                }
                $testunit = $unit;
                $testunit =~ s/\.csv//;
                    $TXT .= swrite("@<<<<<<<<<<<<<<<<<<<<<<<<< @<<<<<",
$testunit);

                foreach $test ( sort keys %{$output{$testsize}{$unit}} ) {
                            $TXT .= swrite("@<<<<<<<<<<<<<< @<<<<<<<",
$output{$testsize}{$unit}{$test});
                }
                $TXT .= "\n";
        }
        $TXT .= "\n";
}

print $TXT;
#print Dumper %output;
```

[/perl]

Merci donc à Paul Venezia de nous avoir pondu ce script téléchargeable ici http://pvenezia.com/iw/iometer-parse.pl