Measuring Throughput

Wednesday August 17, 2016
Seattle Information Technology Exchange
Seattle, WA USA

Stuart Kendrick

Systems Engineer

Allen Institute

This deck available at http://www.skendric.com/seminars

Outline

- Introduction

- cycle-file-copy
- cycle-iperf
- cycle-wget

- Wrap-Up

Intent

- I am a hands-on engineer who has worked in IT Infrastructure & Operations for ~30 years
- I have mostly worked for non-profit, biomedical research outfits
- By offering this ~twenty-minute talk, I hope to introduce peers to a toolkit I have found useful
- Please send me feature requests and bug reports!

The Challenge

- We are often faced with a ticket which says “X is slow”, where “X” can be *my computer*, *the network*, *the server*, *the application*, *the system* …

- Or, we are about to turn a knob and want to know if doing so will improve or degrade performance.

- In these situations, I want a quick & dirty way to measure performance. Not a subjective “it feels faster”, rather, actual data measuring performance.

- Plus, I want a way to easily store those results for posterity.

A Solution

So I've written three scripts

- *cycle-file-copy*
- *cycle-iperf*
- *cycle-wget*

Each script has two flavors: a .bat version for Windows, a BASH version for *nix

The basic idea is to run a file copy (or iPerf or wget) multiple times (10x ... 100x ... 1000x ... statistical power!), measure throughput, and chart the results

Start-Up

```
C:\Myth-Busting\Tools>cycle-file-copy.bat -n "Windows Desktop to Home Directory Server" -t s:\Destination-
Files


======================================================================

I will run the following command 10 times, sleeping 2 seconds between
each iteration:
xcopy C:\Myth-Busting\Temp\Source-Files S:\Destination-Files /s /y

and will store the output in:
     C:\Myth-Busting\Data\cycle-file-copy-output.csv
     C:\Myth-Busting\Data\cycle-file-copy-output.txt
     C:\Myth-Busting\Data\cycle-file-copy-output.gp
     C:\Myth-Busting\Data\cycle-file-copy-output.png

I will copy 1024MB of files

The chart produced from this run will be titled:
     Windows Desktop to Home Directory Server

Stalling for 5 seconds, hit Ctrl-C if you want to pause or cancel
```

Heavy-Lifting

```
Beginning file copy cycle:  10 iterations
    Starting copy iteration 1 at  4:45:20.65
      60MB/s:  1024MB of files copied in 17 seconds
    Starting copy iteration 2 at  4:45:39.38
      31MB/s:  1024MB of files copied in 33 seconds
    Starting copy iteration 3 at  4:46:14.30
      32MB/s:  1024MB of files copied in 32 seconds
    Starting copy iteration 4 at  4:46:48.16
      33MB/s:  1024MB of files copied in 31 seconds
    Starting copy iteration 5 at  4:47:21.26
      40MB/s:  1024MB of files copied in 25 seconds
    Starting copy iteration 6 at  4:47:48.54
      37MB/s:  1024MB of files copied in 27 seconds
    Starting copy iteration 7 at  4:48:17.63
      68MB/s:  1024MB of files copied in 15 seconds
    Starting copy iteration 8 at  4:48:34.22
      30MB/s:  1024MB of files copied in 34 seconds
    Starting copy iteration 9 at  4:49:10.90
      36MB/s:  1024MB of files copied in 28 seconds
    Starting copy iteration 10 at  4:49:40.60
      60MB/s:  1024MB of files copied in 17 seconds
Done with file copy cycle
```

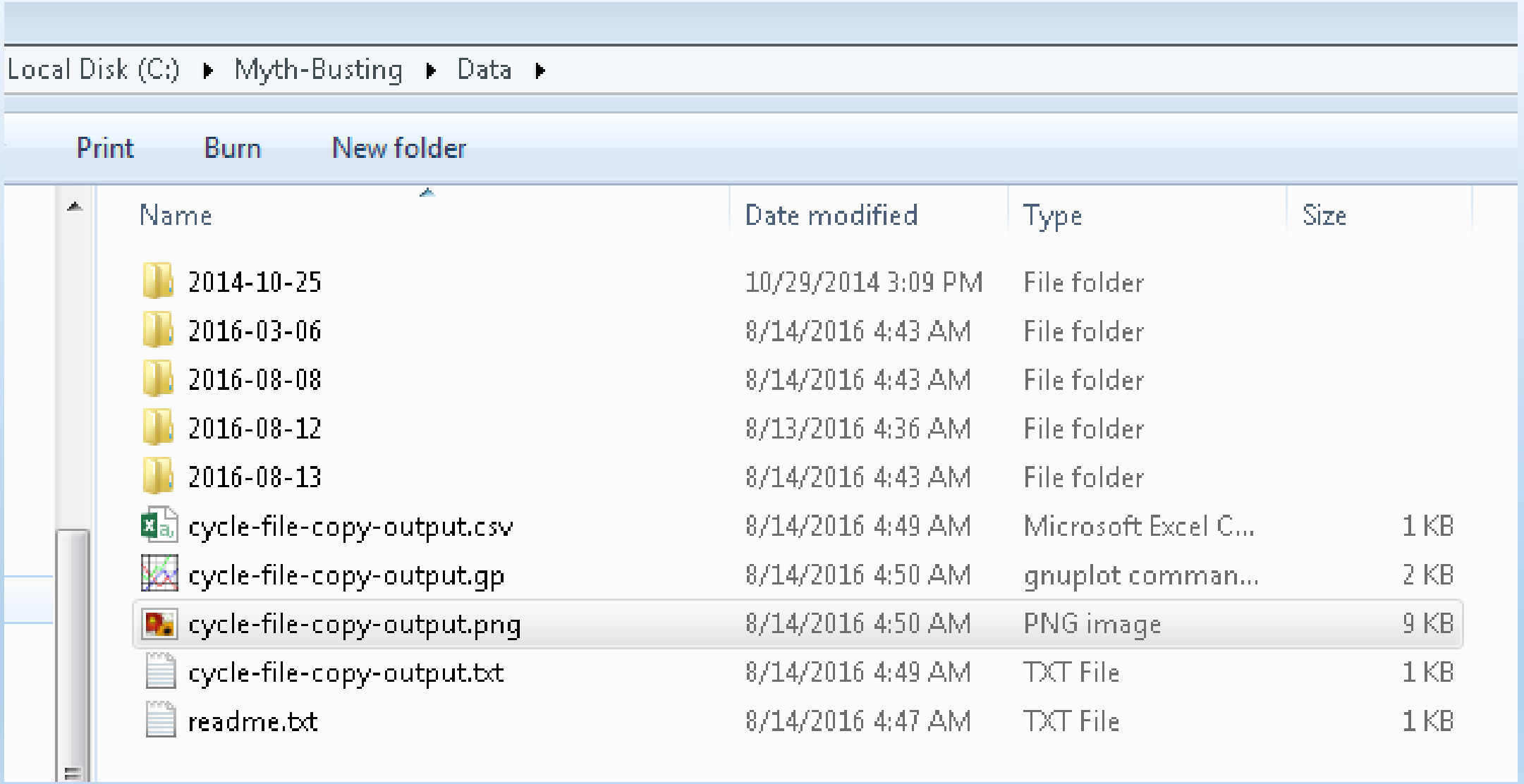
Process the results

```
Extracting throughput file

Creating gnuplot config file

Producing statistics and chart

* FILE:
  Records:      10
  Out of range:  0
  Invalid:       0
  Blank:         0
  Data Blocks:   1

* COLUMN:
  Mean:         42.7000
  Std Dev:      13.5281
  Sum:         427.0000
  Sum Sq.:   20063.0000

  Minimum:      30.0000 [ 7]
  Maximum:      68.0000 [ 6]
  Quartile:     32.0000
  Median:       36.5000
  Quartile:     60.0000
```

Save the Results

```
In C:\Myth-Busting\Data, see
    cycle-file-copy-output.csv for the raw output,
    cycle-file-copy-output.txt for throughput only,
    cycle-file-copy-output.gp  for the gnuplot config file,
    cycle-file-copy-output.png for the chart

Upload these four files plus readme.txt to the class Google Drive

This run of cycle-file-copy.bat started at Sun 08/14/2016  4:45:15.51
                             and ended at Sun 08/14/2016  4:49:59.51

Done with "Windows Desktop to Home Directory Server"
================================================================

C:\Myth-Busting\Tools>
```

You get four files



What do those files look like?

The .csv and .txt files record the throughput of each trial

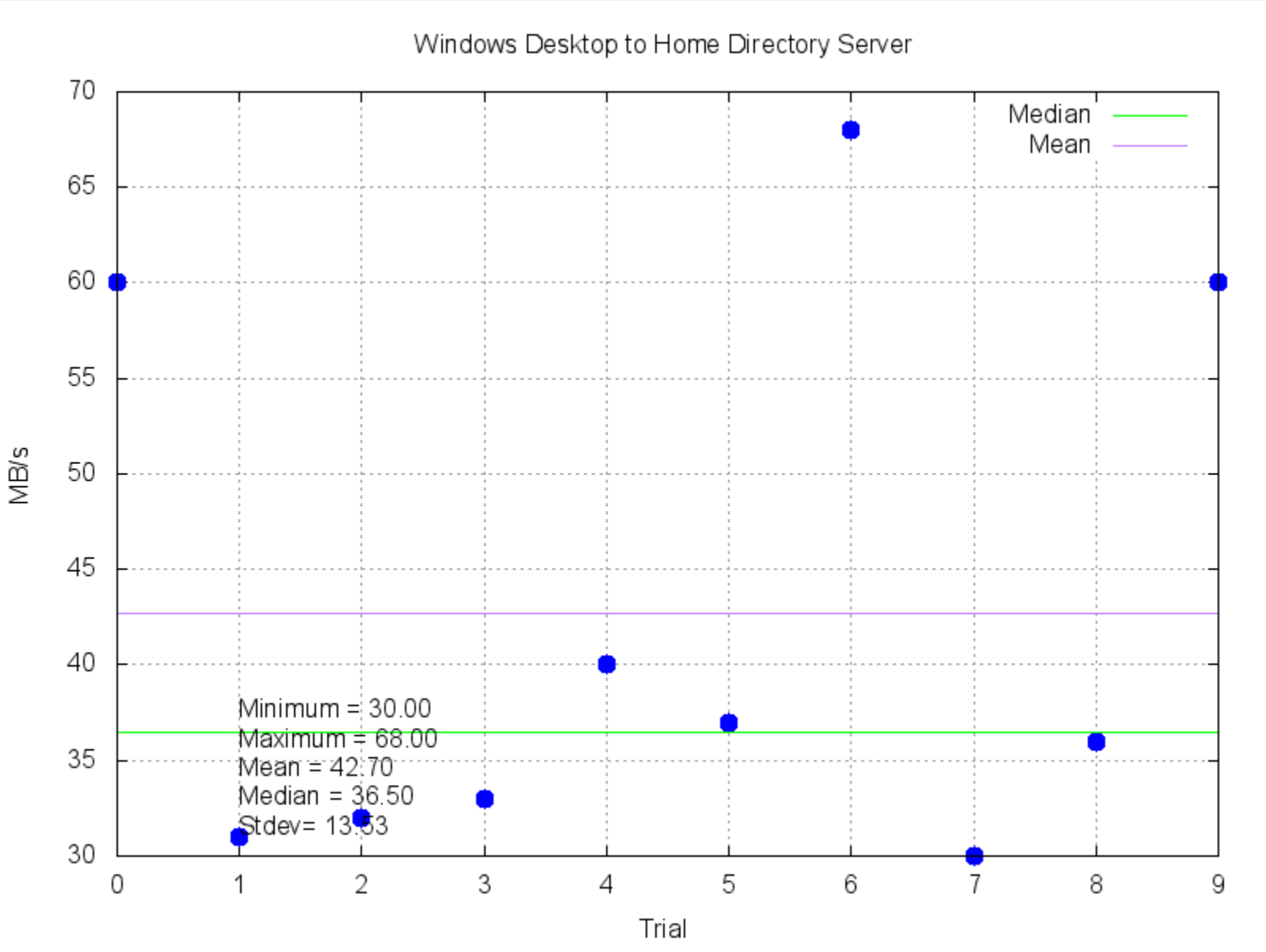
```
60
31
32
33
40
37
68
30
36
60
```

What do those files look like?

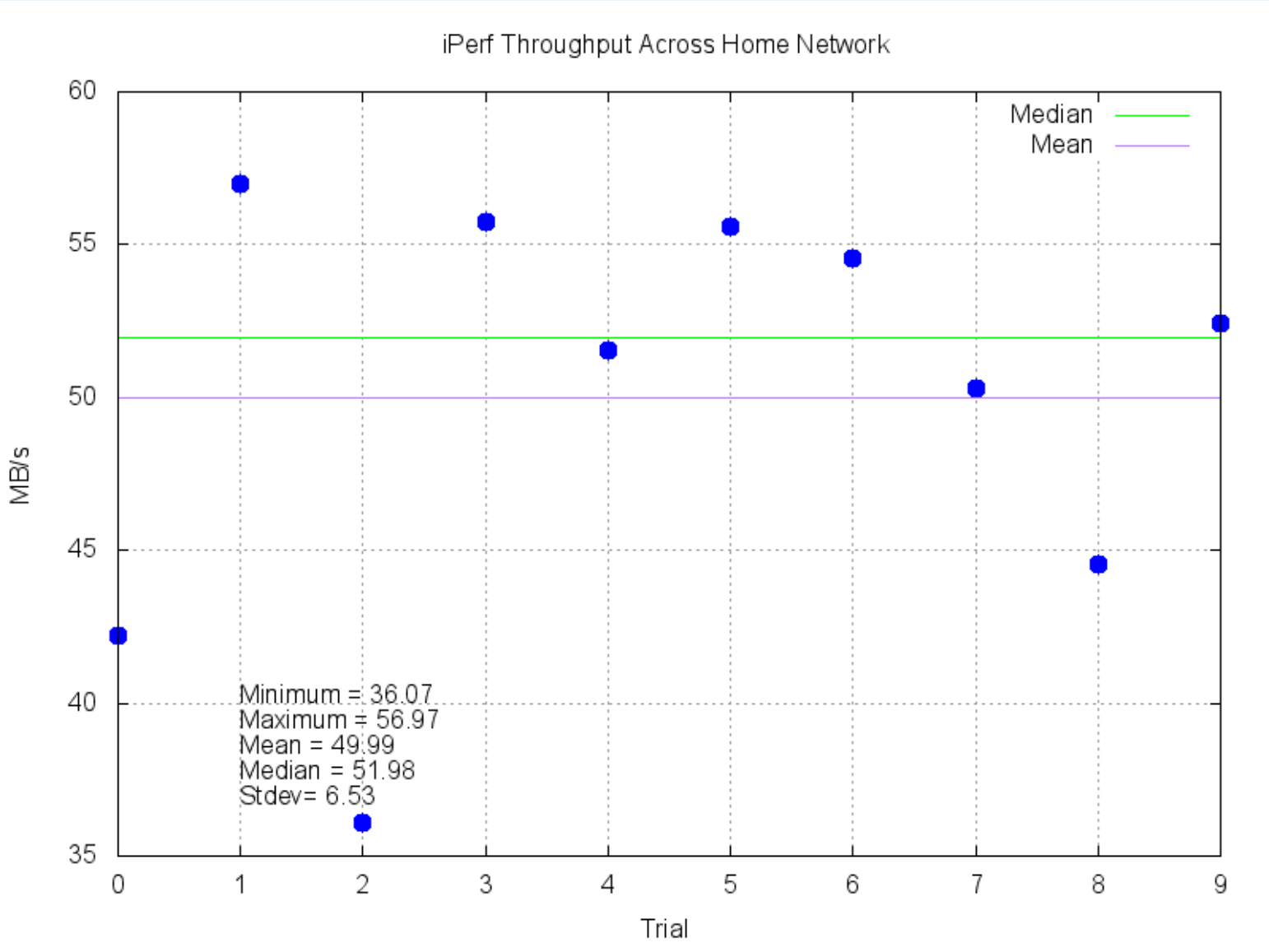
The .gp file is a gnuplot configuration file

```
# gnuplot configuration file for Myth-Busting:  The Network Layer -- CasitConf 2017
# Typically invoked as follows:  cat file-copy-throughput.gp | gnuplot
#
# V      Who       When        What
# ------------------------------------------------------------------------------
# 1.0.0  skendric  2014-08-08  First version
#
#

# Define variables
long_title  = "Windows Desktop to Home Directory Server"
input_file  = "cycle-file-copy-output.txt"
output_file = "cycle-file-copy-output.png"

# Exit cleanly if this install of gnuplot does not support stats
if (!strstrt(GPVAL_COMPILE_OPTIONS,"+STATS")) {
    print "No support for stats command"
    exit
}


# Chart details
set title long_title
set terminal png size 800,600
set grid
set xlabel "Trial"
set ylabel "MB/s"


# File specifics
set datafile sep ','
set output output_file

# Calculate statistics
stats input_file index 0 using 1

# http://www.phyast.pitt.edu/~zov1/gnuplot/html/statistics.html
# Add shading to show stddev --sk

# Figure out where to put the labels
line_width = (STATS_max - STATS_min) / 25
y_offset   = STATS_records * .1

# Stats labels
set label 1 sprintf("Minimum = %.2f", STATS_min)    at y_offset, STATS_min   + line_width*5 front
set label 2 sprintf("Maximum = %.2f", STATS_max)    at y_offset, STATS_min   + line_width*4 front
set label 3 sprintf("Mean = %.2f"   , STATS_mean)   at y_offset, STATS_min   + line_width*3 front
set label 4 sprintf("Median = %.2f",  STATS_median) at y_offset, STATS_min   + line_width*2 front
set label 5 sprintf("Stdev= %.2f",    STATS_stddev) at y_offset, STATS_min   + line_width*1 front

# Do the work
plot input_file index 0 using 0:1 title '' with points pointtype 7 pointsize 2 linecolor rgb "blue", \
     STATS_median title " Median" linecolor rgb "green", STATS_mean title " Mean" linecolor rgb "purple"
```

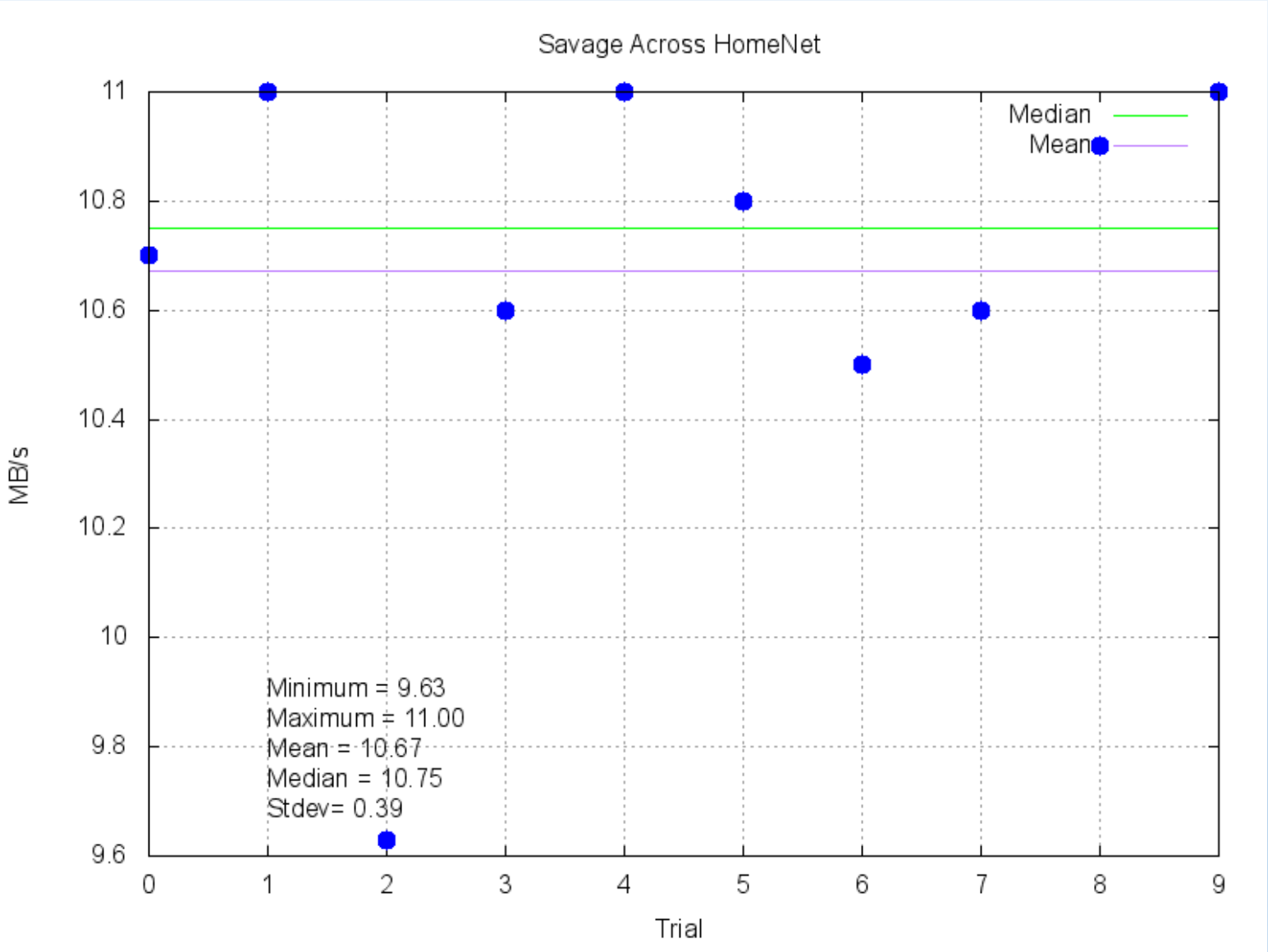
View the Results



cycle-iperf


iPerf Throughput Across Home Network
Median
Mean
Minimum = 36.07
Maximum = 56.97
Mean = 49.99
Median = 51.98
Stdev= 6.53
MB/s
Trial
60
55
50
45
40
35
0
1
2
3
4
5
6
7
8
9

cycle-wget



At first glance, looks like a low-performing outlier on Trial 2

But pay attention to stdev ... in fact, all these results are pretty darn close to each other

If you figure out how to tell *gnuplot* to scale the chart more appropriately, drop me a note and tell me how you did it

Send me email with questions --sk

Help

C:\Myth-Busting\Tools>cycle-file-copy --help

Program = cycle-file-copy

Version = 1.0.0

* Must specify the Title of the chart

Remaining arguments are optional (although you'll probably want to change destination directory)

Usage:

cycle-file-copy

-n {title for this run, inscribed on chart}

[-i iterations]

[-o source directory]

[-s sleep time in seconds between iterations]

[-t destination directory]

[-d debug]

Defaults as follows:

-i iterations = 10

-o source = C:\Myth-Busting\Temp\Source-Files

-s sleepTime = 2

-t destination = C:\Myth-Busting\Temp\Destination-Files

-d debug = 0

Standardized Language

I imagine the following benefits:

The cycle-* scripts give us techs a common language

> *Let's look at the cycle-{x} chart before we made the change compared with after*

I would like to think that power users can employ it also, to perform their own measurements.

With Power Comes Responsibility

The scripts allow you to specify number of iterations and sleep time between iterations.

You can combine these two to create a DoS attack on your systems (e.g. lots of iterations, little pause time between them).

Shameless Plug

These scripts form the core of a class which my buddy Chris Shaiman and I teach at IT conferences

Want to spend a fun day of team-oriented exercises measuring throughput across a fancy laboratory? (Chris and I together bring ~100K of gear to this class)

Join us at Cascadia IT Conference in March 2017

We call the class *Myth-Busting*, inspired by a popular television show. See http://www.skendric.com/seminar/ for details

Questions, Comments, Complaints?

Download the *Myth-Busting Toolkit* to install the scripts
http://www.skendric.com/app/

Send me feature requests & bug reports!

stuartk {at} alleninstitute {dot} org