Automated Charting on CMS

Brian K. Wade, Ph.D.
IBM z/VM Performance, Endicott
bkw@us.ibm.com

Version 2021-07-21.1

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

Db2*
DirMaint
DS8000*
ECKD
FICON*
FlashCopy*
FlashSystem
GDPS*
ibm.com
IBM eServer
IBM (logo)*
IBM Z*
LinuxONE*
LinuxONE Emperor
LinuxONE Rockhopper
OMEGAMON*
PR/SM
RACF*
System z10*
XIV*
z13*
z13s
z14
z15
z10 BC
z10EC
z/Architecture*
zEnterprise*
z/OS*
zSecure
zSeries*
z/VM*
z Systems*

* Registered trademarks of IBM Corporation

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.
ITIL is a Registered Trade Mark of AXELOS Limited.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.
Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Abstract

- I will show you how I make useful charts and graphs,
- ... of data born on CMS,
- ... without the data ever leaving CMS,
- ... and sometimes totally unattended,
- ... and I give away the software on www.vm.ibm.com.

Agenda

- I live on CMS, and why I live there
- What was missing
- What I wanted, and what I had to work with
- What I found
- What I did
- Some examples

My Situation

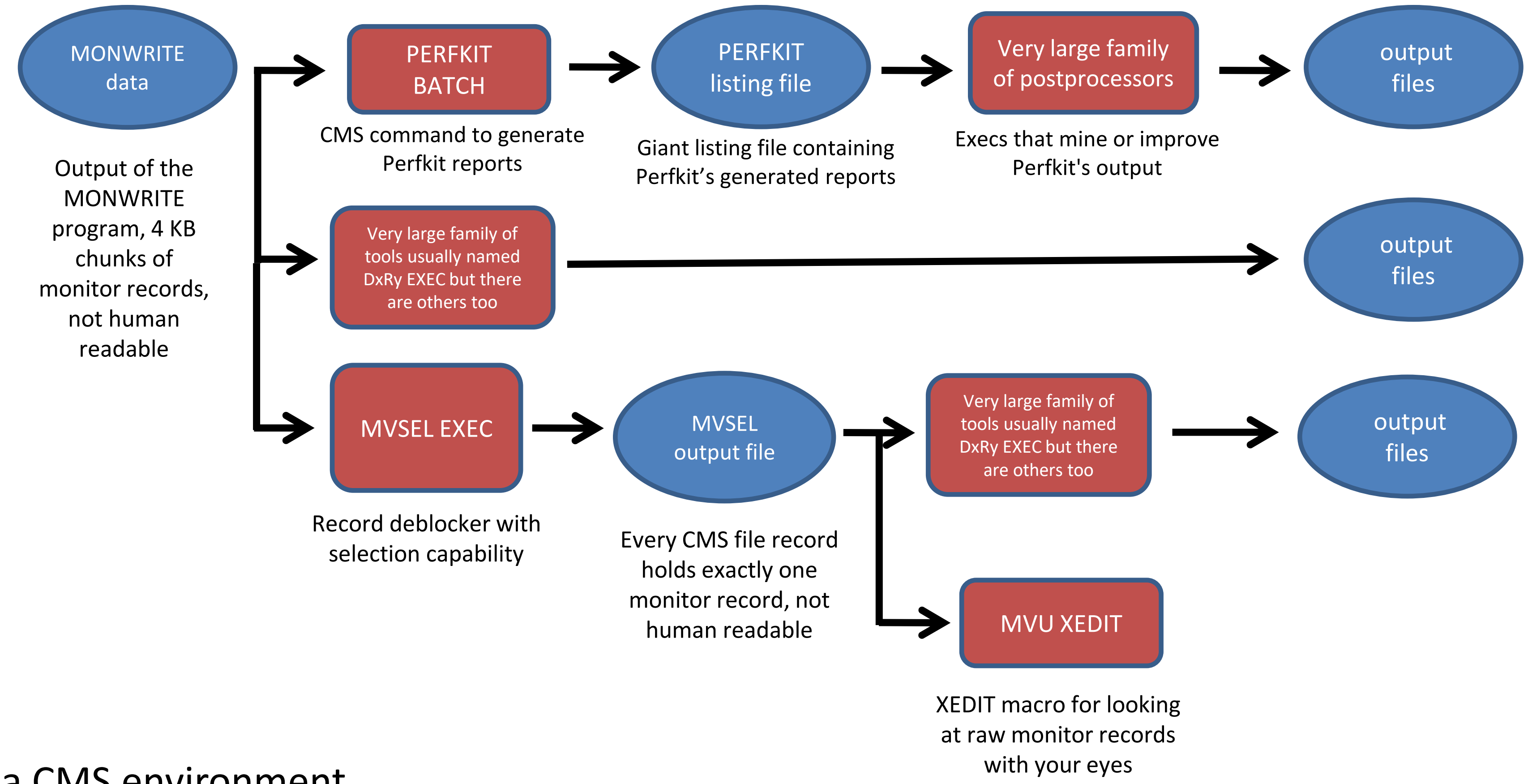
I Live on CMS

- We measure z/VM workloads
- MONWRITE is a CMS program that produces a CMS file
- PERFKIT BATCH is a CMS program that produces a CMS file
- Our measurement data gets archived in the CMS Shared File System

- So you can imagine that pretty much all day long, I live on CMS
  - I look at Perfkit listings
  - I slice-and-dice the monitor data in new ways, per each measurement
  - I talk to my friends about what I saw in the data

What We Do with MONWRITE Data

this input is a CMS file



MONWRITE data

Output of the MONWRITE program, 4 KB chunks of monitor records, not human readable

PERFKIT BATCH

CMS command to generate Perfkit reports

PERFKIT listing file

Giant listing file containing Perfkit’s generated reports

Very large family of postprocessors

Execs that mine or improve Perfkit's output

output files

Very large family of tools usually named DxRy EXEC but there are others too

output files

MVSEL EXEC

Record deblocker with selection capability

MVSEL output file

Every CMS file record holds exactly one monitor record, not human readable

Very large family of tools usually named DxRy EXEC but there are others too

output files

MVU XEDIT

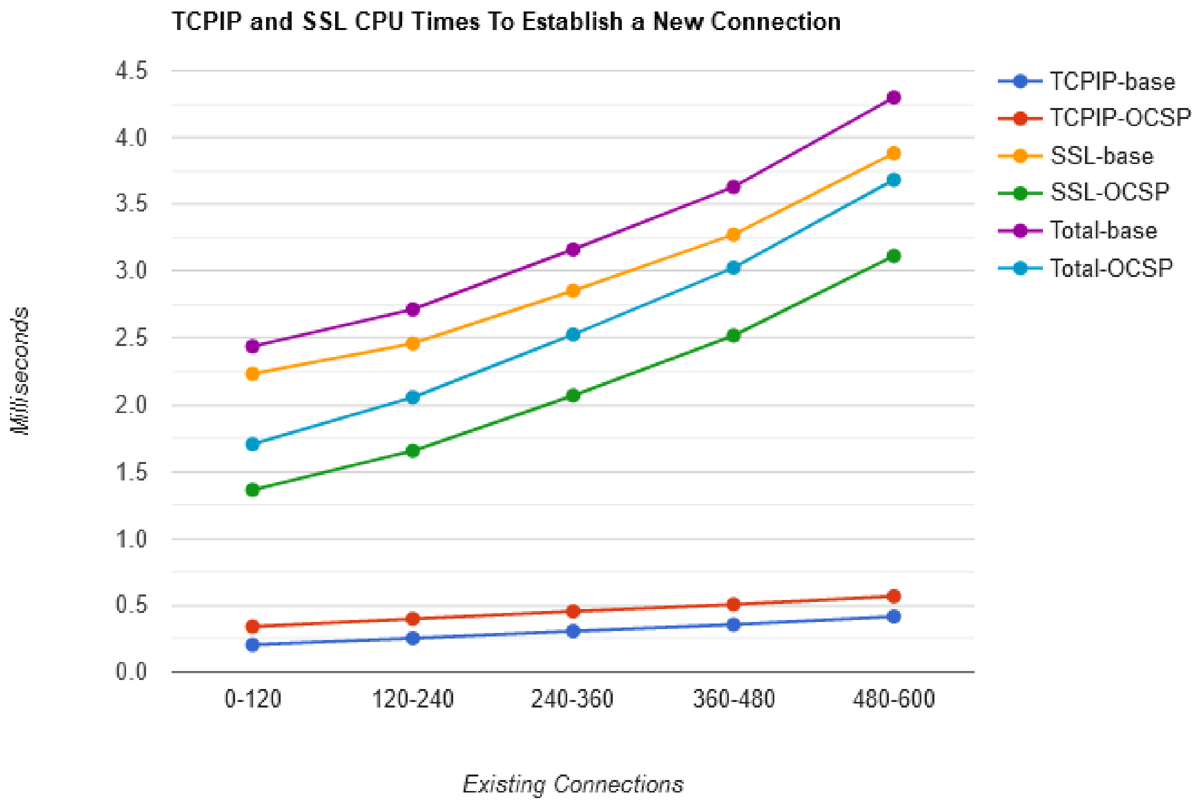
XEDIT macro for looking at raw monitor records with your eyes

all these outputs are CMS files

This is entirely a CMS environment.
I spend a good amount of my time in XEDIT, looking at reports.

What's Missing?

- Charts!
- Useful for:
  - Showing our friends what happened
  - Putting into the z/VM Performance Report
- In 2013 the state of the art was:
  - Cut-and-paste your data into Excel
  - Have Excel draw the chart for you
- Reasons that approach frustrated me
  - Tedious, error-prone, mind-numbing
  - Why do I have to drag my data to another platform just to make a chart?
  - The insult of "Oh, CMS can't do that."
  - The z/VM Performance Report did not look uniform
  - Not easily automated



So Here's Where I Sat

What I Had

- A bunch of data born on CMS

What I Wanted

- A way to draw a chart
  - Without cut-and-paste
  - Without moving the data elsewhere
  - Maybe even while I was asleep

Assets Already In Hand

- Rexx and Pipelines for data extraction and massaging
- A CMS-based web server
  - Serves HTML files out of SFS
- Browsers can draw things

Elements of the Solution

What I Found

- The Google Visualization API
  - https://developers.google.com/chart/

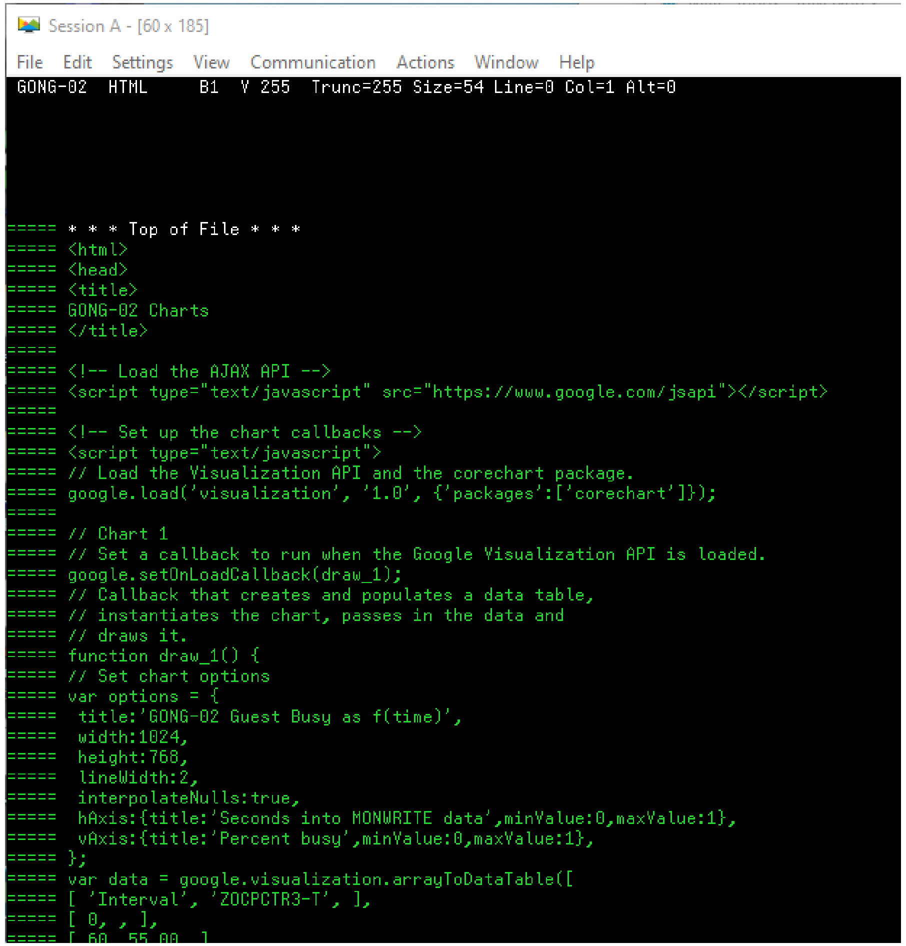
- This thing lets one use JavaScript to express a chart

- Put the JavaScript into an HTML file

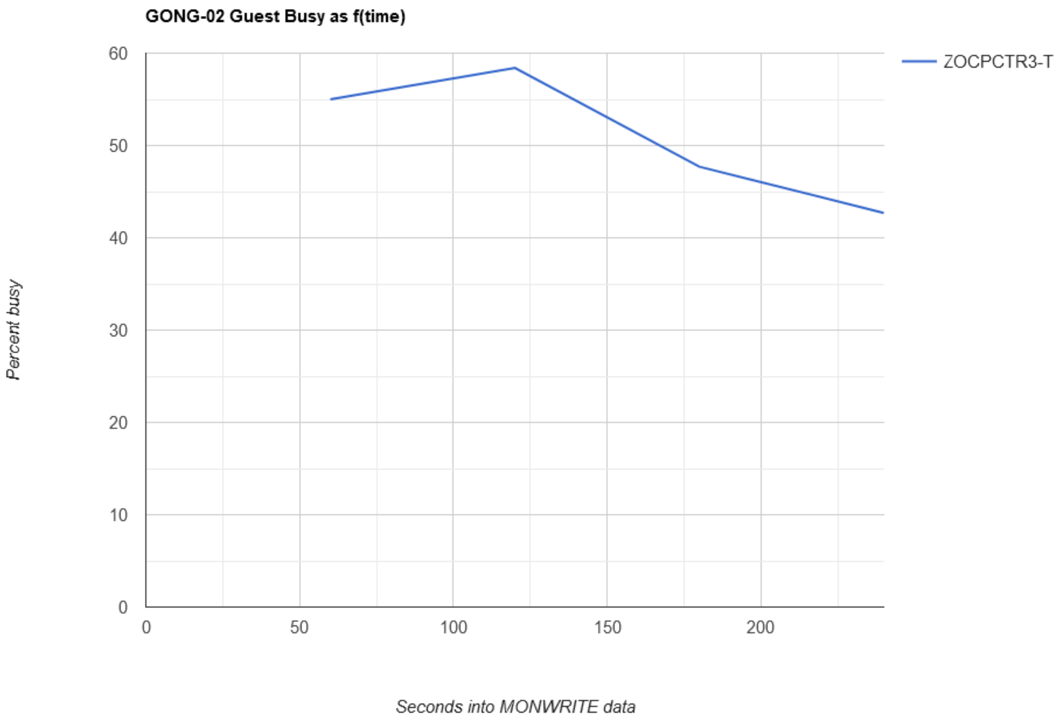
- Get a browser to fetch the HTML file

- Ta-da! The browser draws the chart for you

HTML File Containing JavaScript?



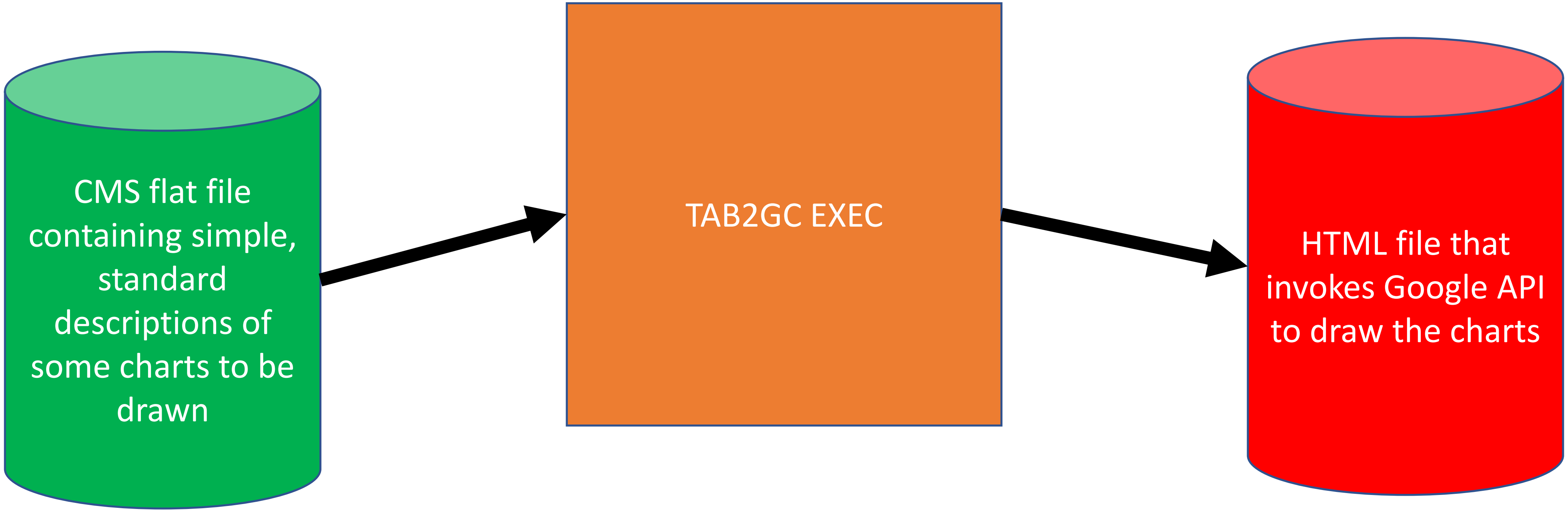
```
GONG-02  HTML      B1   V 255   Trunc=255 Size=54 Line=0 Col=1 Alt=0




===== * * * Top of File * * *
===== <html>
===== <head>
===== <title>
===== GONG-02 Charts
===== </title>
=====
===== <!-- Load the AJAX API -->
===== <script type="text/javascript" src="https://www.google.com/jsapi"></script>
=====
===== <!-- Set up the chart callbacks -->
===== <script type="text/javascript">
===== // Load the Visualization API and the corechart package.
===== google.load('visualization', '1.0', {'packages':['corechart']});
=====
===== // Chart 1
===== // Set a callback to run when the Google Visualization API is loaded.
===== google.setOnLoadCallback(draw_1);
===== // Callback that creates and populates a data table,
===== // instantiates the chart, passes in the data and
===== // draws it.
===== function draw_1() {
===== // Set chart options
===== var options = {
=====  title:'GONG-02 Guest Busy as f(time)',
=====  width:1024,
=====  height:768,
=====  lineWidth:2,
=====  interpolateNulls:true,
=====  hAxis:{title:'Seconds into MONWRITE data',minValue:0,maxValue:1},
=====  vAxis:{title:'Percent busy',minValue:0,maxValue:1},
===== };
===== var data = google.visualization.arrayToDataTable([
===== [ 'Interval', 'ZOCPCTR3-T', ],
===== [ 0, , ],
===== [ 60, 55.00, ]
```



HTML file describes chart

Browser draws chart

How I Exploited the Google Visualization API



CMS flat file containing simple, standard descriptions of some charts to be drawn

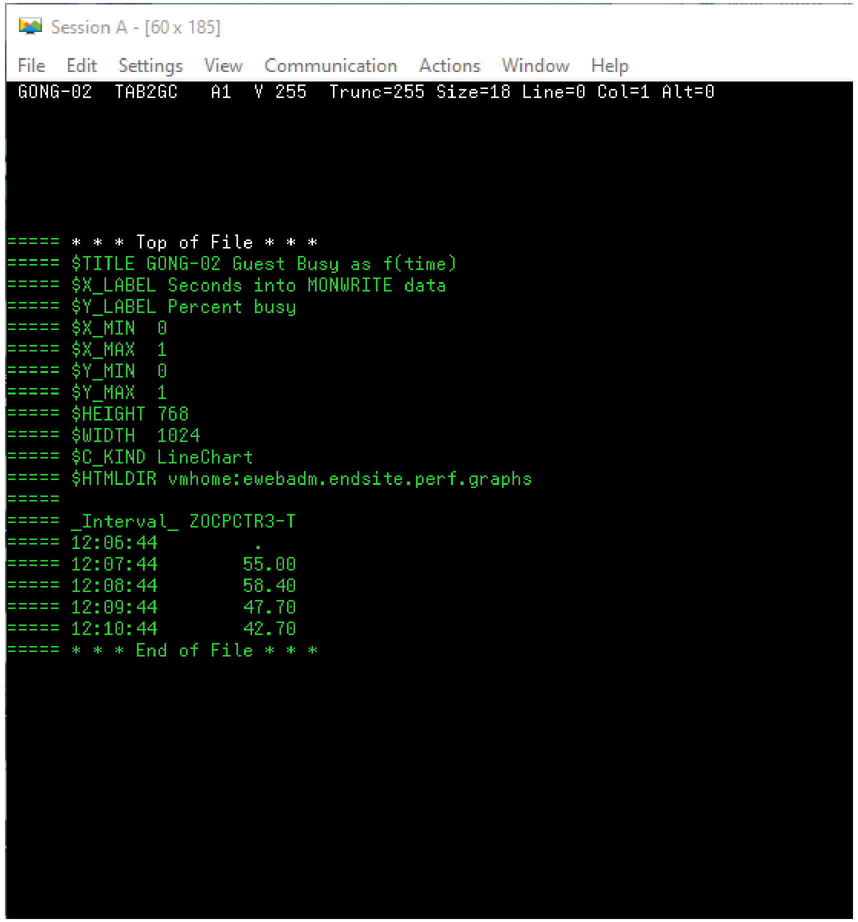
TAB2GC EXEC

HTML file that invokes Google API to draw the charts

"Table to Google Chart"

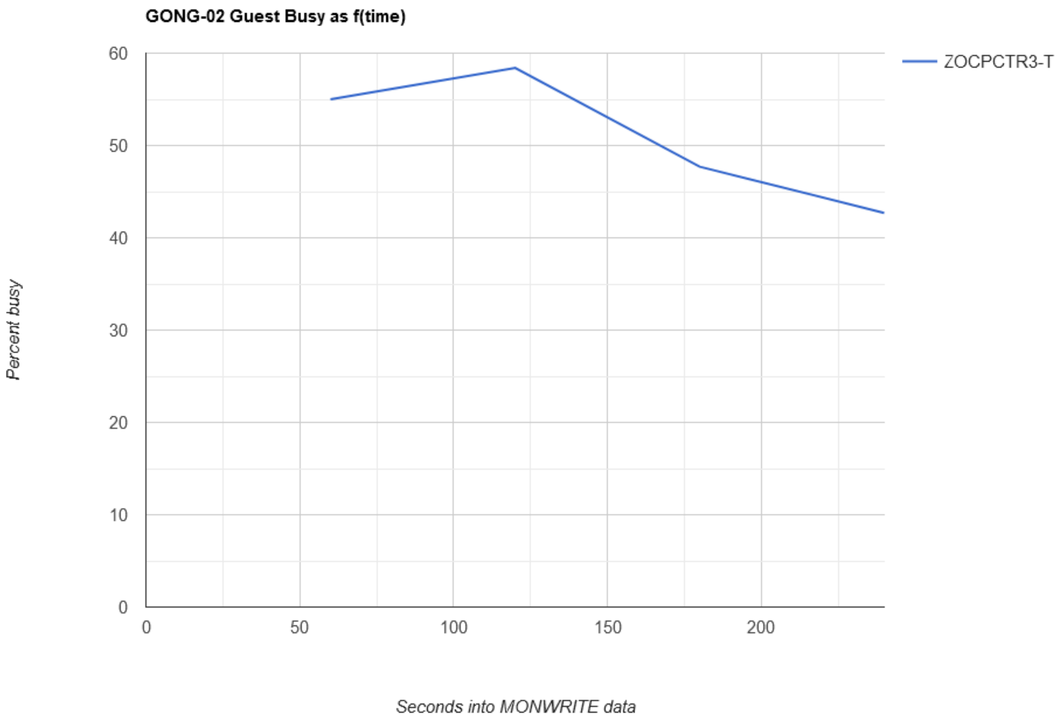
- CMS exec
- Reads flat file
- Generates JavaScript
- Emits HTML file

A TAB2GC Input File



```
GONG-02  TAB2GC   A1  V 255  Trunc=255 Size=18 Line=0 Col=1 Alt=0




===== * * * Top of File * * *
===== $TITLE GONG-02 Guest Busy as f(time)
===== $X_LABEL Seconds into MONWRITE data
===== $Y_LABEL Percent busy
===== $X_MIN  0
===== $X_MAX  1
===== $Y_MIN  0
===== $Y_MAX  1
===== $HEIGHT 768
===== $WIDTH  1024
===== $C_KIND LineChart
===== $HTMLDIR vmhome:ewebadm.endsite.perf.graphs
=====
===== _Interval_ ZOCPCTR3-T
===== 12:06:44          .
===== 12:07:44        55.00
===== 12:08:44        58.40
===== 12:09:44        47.70
===== 12:10:44        42.70
===== * * * End of File * * *
```

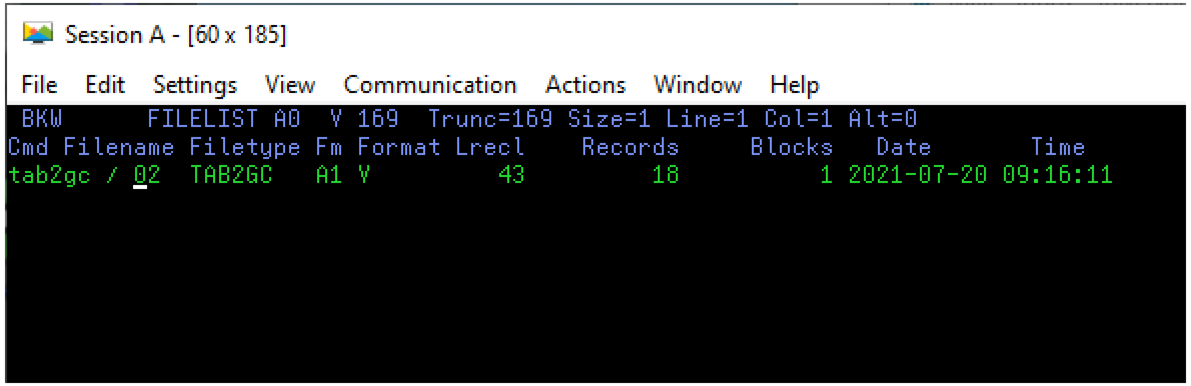
XEDIT session of file GONG-02 TAB2GC



Browser's rendering of file GONG-02 HTML

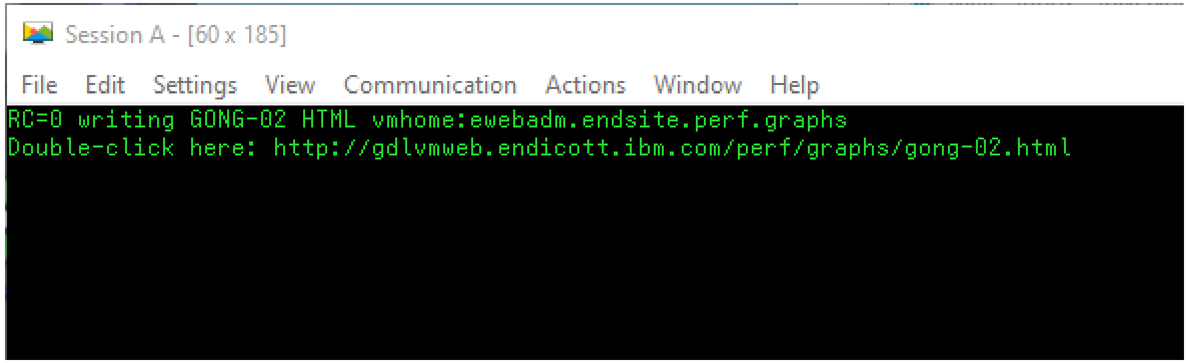
Running and Clicking

Run from FILELIST



```
 BKW      FILELIST A0  V 169  Trunc=169 Size=1 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl    Records     Blocks   Date       Time
tab2gc / 02   TAB2GC   A1  V         43         18           1 2021-07-20 09:16:11
```

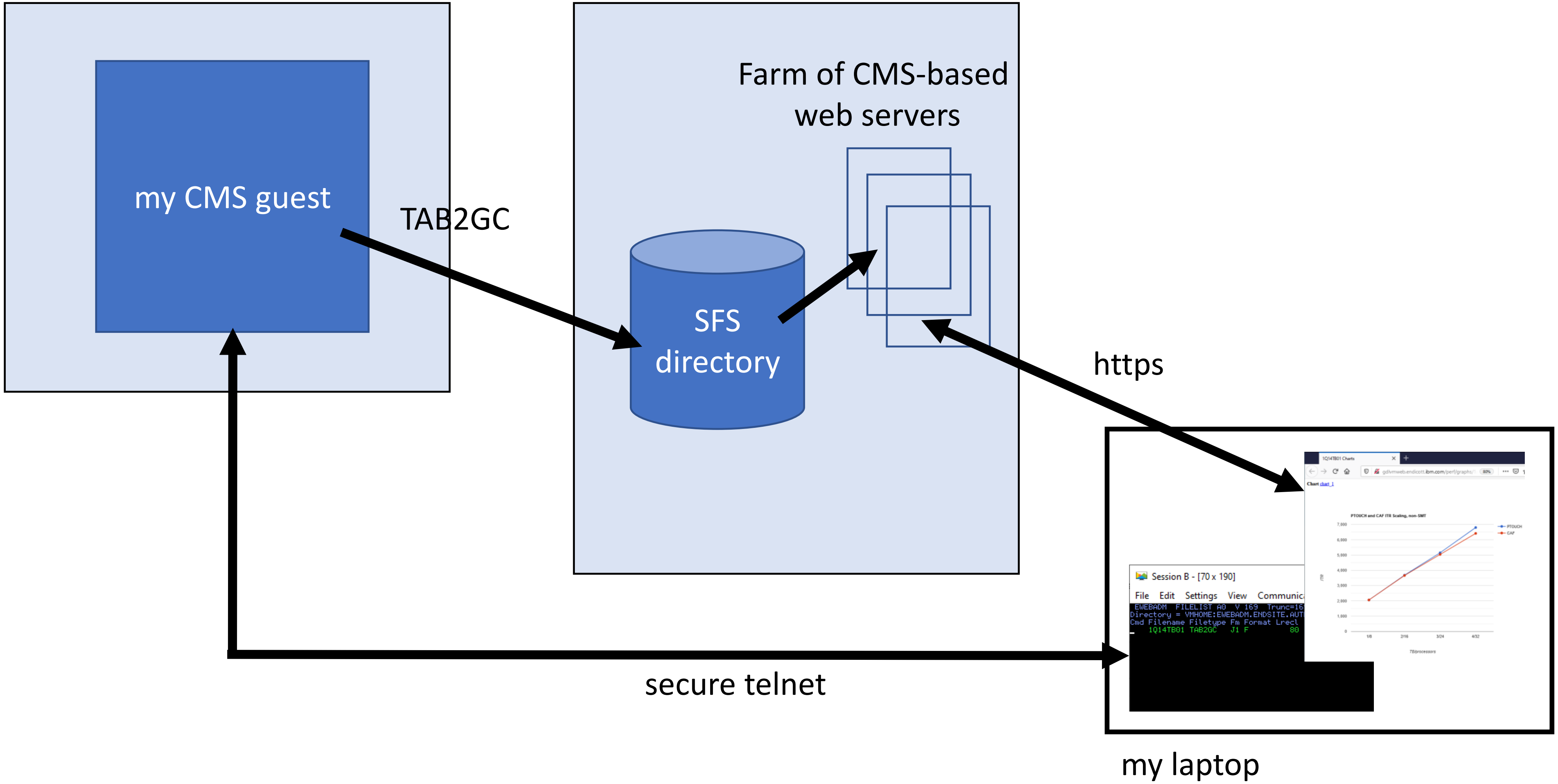
TAB2GC EXEC prints URL on 3270 screen



```
RC=0 writing GONG-02 HTML vmhome:ewebadm.endsite.perf.graphs
Double-click here: http://gdlvmweb.endicott.ibm.com/perf/graphs/gong-02.html
```

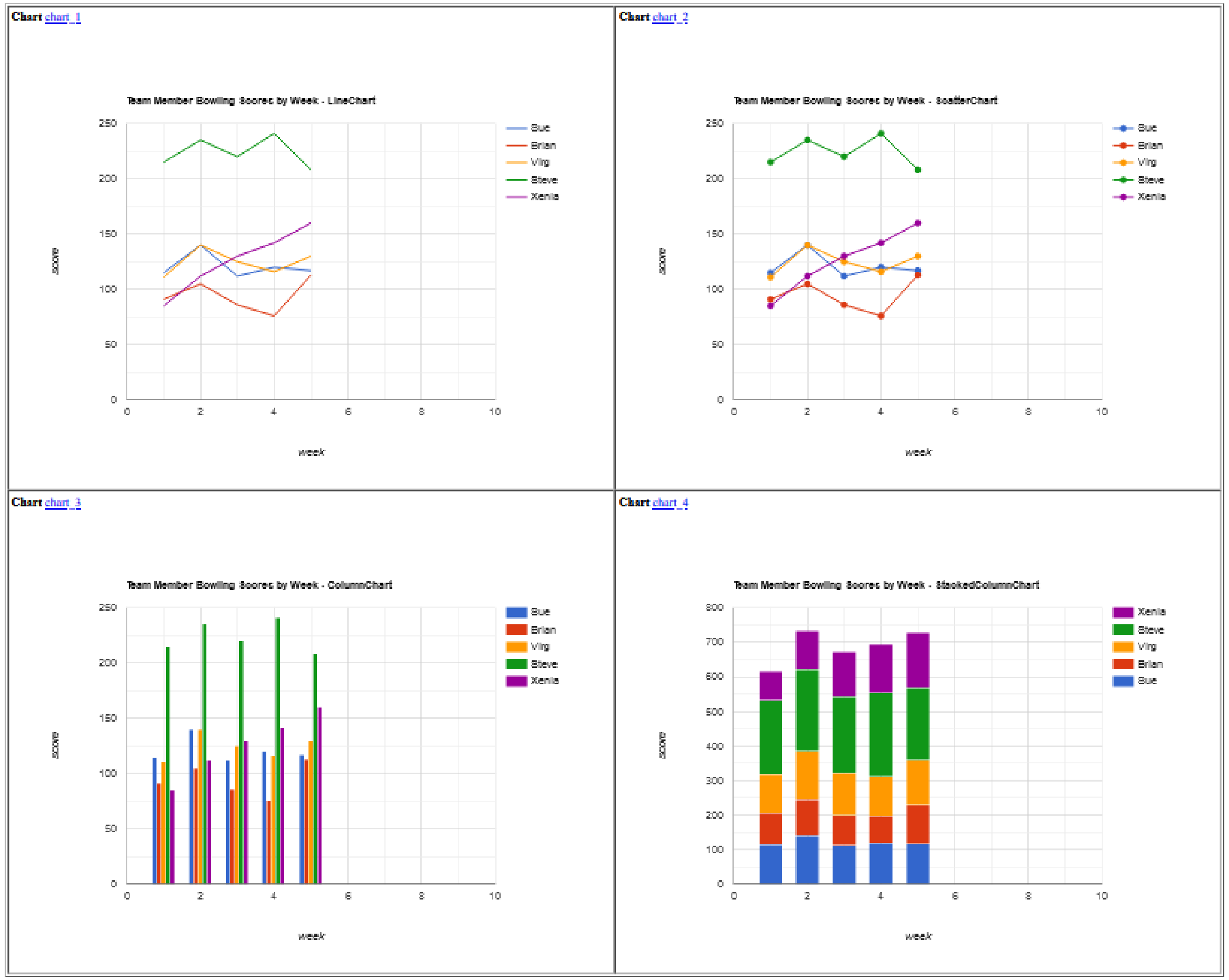
Just double-click on the URL. PComm launches the browser.

In Practice



```
 EWEBADM  FILELIST A0  V 169  Trunc=16
Directory = VMHOME:EWEBADM.ENDSITE.AUT
Cmd Filename Filetype Fm Format Lrecl
    1Q14TB01 TAB2GC   J1 F        80
```

Kinds of Charts TAB2GC Can Draw



- Line chart
- Scatter chart
- Column chart
- Stacked column chart

The Google API can do a lot more kinds. These were just the ones I needed every day as a performance analyst.

https://gdlvmweb.endicott.ibm.com/perf/graphs/bowling.html

What Data Can We Chart?

What Can We Chart?

- **Any data we can reach from CMS**
  - Just array the data into a file of filetype TAB2GC
  - Then invoke TAB2GC EXEC

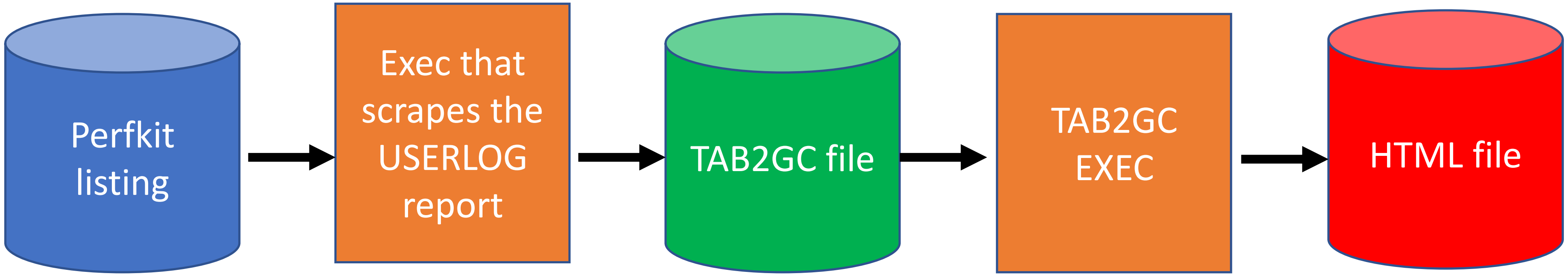
- This creates an HTML file containing your charts.
- Then just use a web browser to visit the HTML file.

Perfkit Listings Are Attractive Targets

- Especially LOG reports
- Or the same line from every INTERIM report

```
1FCX162  Run 2021/03/19 15:36:58         USERLOG  ZOCPCTR3
                                          User Resource Consumption Log
 From 2020/04/20 12:06:44
 To   2020/04/20 12:10:44
 For     240 Secs 00:04:00                Result of GONG-02 Run
 ____________________________________________________________________________

  Resource Usage Log for User ZOCPCTR3

            <----- CPU Load -----> <----- Virtual IO/s ----->
  Interval       <-Seconds->   T/V                  Diag
  End Time  %CPU  TCPU  VCPU Ratio Total DASD Avoid   98   UR Pg/s  User Status
  >>Mean>>  51.0 30.58 29.35  1.04    .3   .0    .0   .0   .0   .0  ---,---,----
  12:07:44  55.0 33.03 31.75  1.04    .3   .0    .0   .0   .0   .0  EME,CL3,DISP
  12:08:44  58.4 35.05 33.73  1.04    .3   .0    .0   .0   .0   .0  EME,CL3,DISP
  12:09:44  47.7 28.62 27.37  1.05    .3   .0    .0   .0   .0   .0  EME,CL3,DISP
  12:10:44  42.7 25.63 24.56  1.04    .3   .0    .0   .0   .0   .0  EME,CL3,DISP
```

Graphing a PERFKIT BATCH Listing



Lots of Perfkit reports are time-oriented. USERLOG, PRCLOG, DEVLOG, ... on and on.
So are other files derived from MONWRITE data. The $CPUMFLG CPU MF report, for example.
These are very amenable to being scraped and charted.

The scraper reads the report and writes a TAB2GC file.
I have lots of scrapers. They mine Perfkit reports and other similar reports.

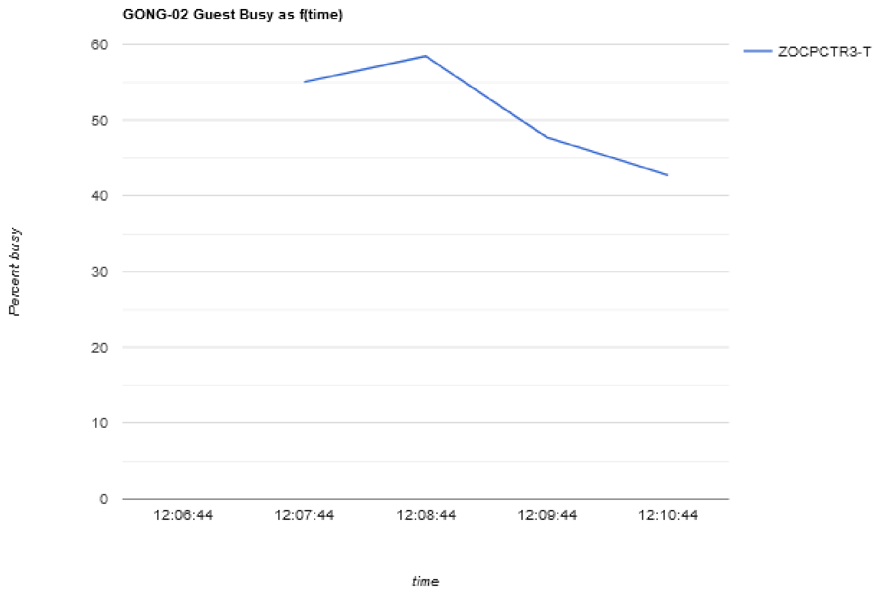
Perfkit to TAB2GC to HTML

```
1FCX162  Run 2021/03/1

 From 2020/04/20 12:06
 To   2020/04/20 12:10
 For    240 Secs 00:04
 _____________________

  Resource Usage Log f

            <----- CPU
  Interval       <-Sec
  End Time  %CPU  TCPU
  >>Mean>>  51.0 30.58
  12:07:44  55.0 33.03
  12:08:44  58.4 35.05
  12:09:44  47.7 28.62
  12:10:44  42.7 25.63
```

GONG-02 PERFKIT

```
$TITLE GONG-02 Guest Busy as f(time)
$X_LABEL time
$Y_LABEL Percent busy
$X_MIN   0
$X_MAX   1
$Y_MIN   0
$Y_MAX   1
$HEIGHT 768
$WIDTH  1024
$C_KIND LineChart
$HTMLDIR vmhome:ewebadm.endsite.perf.graphs

_Interval_ ZOCPCTR3-T
12:06:44          .
12:07:44         55.00
12:08:44         58.40
12:09:44         47.70
12:10:44         42.70
```

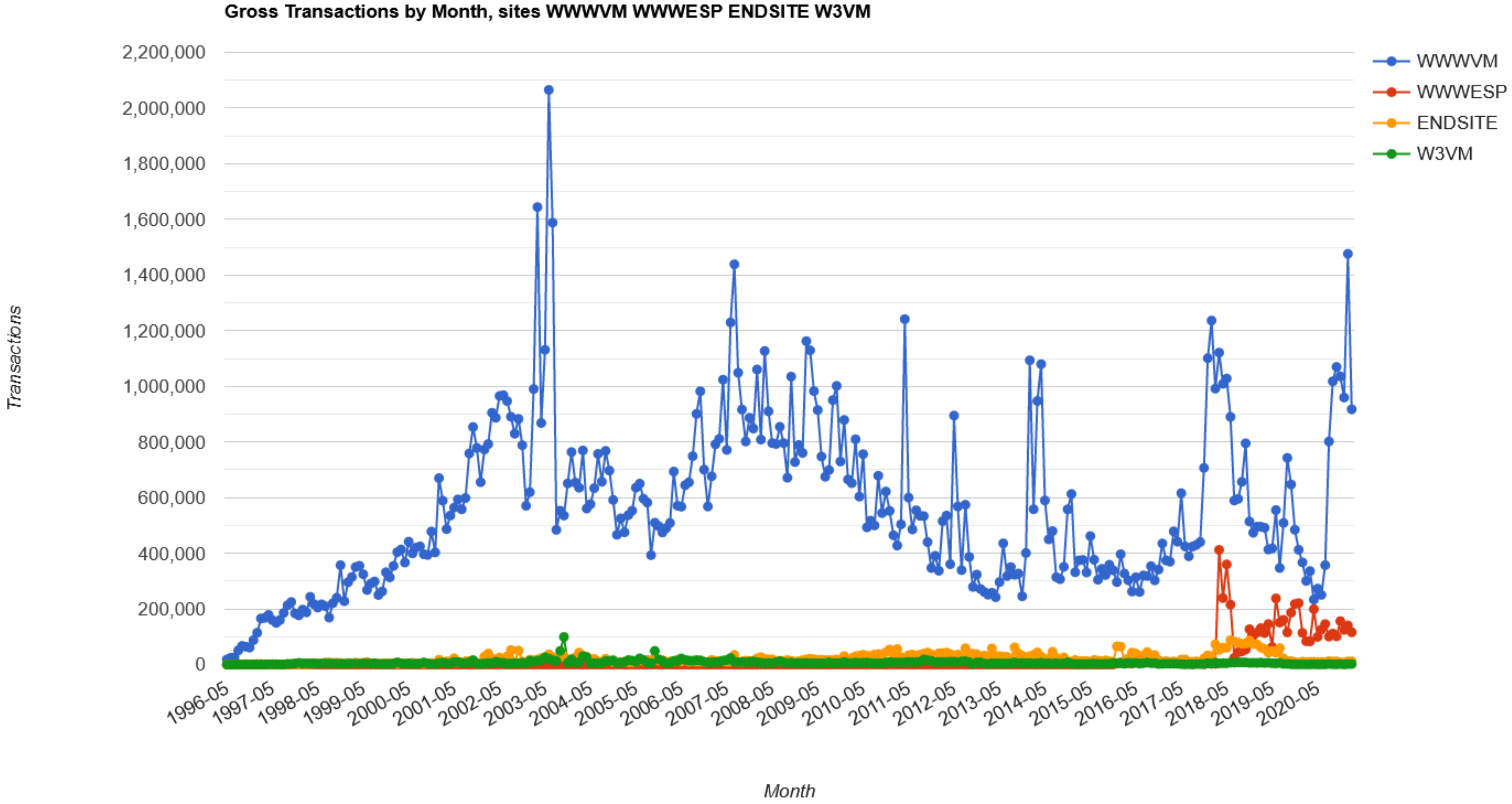
GONG-02 TAB2GC



GONG-02 HTML

The Real Power Here

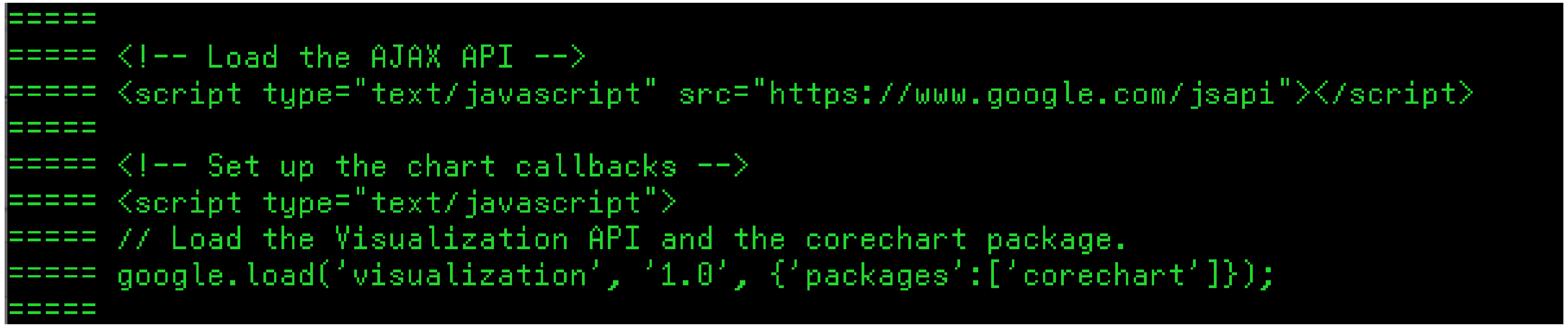
- **Your "scraper" is a CMS app. It can:**
  - **Gather up information from pretty much anywhere**
  - **Emit a TAB2GC file**
- And you've got a graph
- The graph at right gets maintained automatically by our web servers. Every month some automation updates it.
- Look Ma, no hands!



From the NCSA-standard http logs kept by our web servers

One Caution

One Caution



```
=====
===== <!-- Load the AJAX API -->
===== <script type="text/javascript" src="https://www.google.com/jsapi"></script>
=====
===== <!-- Set up the chart callbacks -->
===== <script type="text/javascript">
===== // Load the Visualization API and the corechart package.
===== google.load('visualization', '1.0', {'packages':['corechart']});
=====
```

The HTML file loads a JavaScript library from google.com.

Your company might like to consider this carefully.

Summary

Summary

- I live on CMS
- I wanted to make charts
  - without cutting and pasting
  - without moving the data off CMS
  - unattended via automation
  - in a form I could use in the z/VM Performance Report
- TAB2GC is what happened
- Thank you

Where to Get It

- https://www.vm.ibm.com/download/packages/descript.cgi?TAB2GC
- For instructions, read the prologue of the exec