zHISR: Improving Application Performance using Hardware Instrumentation

Ray Mullins
Phoenix Software International

8 August 2017
What is zHISR?

zHISR is an interactive application execution profiler that allows developers, performance analysts and others to easily interface with IBM Z Hardware Instrumentation to perform near-zero overhead, high-resolution hot spot analysis of programs running under z/OS.
Most commercial application profilers use software timer-based sampling to obtain the data upon which to perform the analysis.

- STIMER(M), TIMER DIE (Disabled Interrupt Exit), etc.

The timer routines themselves are dispatched by z/OS. Therefore, they become part of the application execution path as seen by the system and its accounting routines.

Sampling this way can be expensive in terms of CPU consumption and is one reason that the use of application profilers is often strictly controlled.
STIMER-Based Sampling

- Normal condition of task-based execution:
  - Program executes under a Program Request Block (PRB)
STIMER-Based Sampling

- When timer interrupt occurs, the operating system:
  - Schedules an SRB into the target address space
  - The SRB schedules an IRB to run the timer exit
  - The timer exit collects the PSW from the PRB

![Diagram showing TCB, PRB, and IRB connections](image-url)
The operating system provides authorized programs with the Timer DIE (Disabled Interrupt Exit) function.

The Timer DIE gets control directly from the SLIH when the timer interrupt is handled. This can occur in any address space and within any unit of work (task or SRB) in the system.

The DIE executes disabled (must not create a page fault) and cannot obtain locks or reference private area storage.

The DIE can schedule (or resume) an SRB to do whatever collection is necessary.
Timer DIE-Based Sampling

Timer Interrupt

FLIH

SLIH

DIE

Dispatch

SRB

Collect PSWs

Issue IEAMSCHD (SCHEDULE) or RESUME SRB
Most Obvious Disadvantages of Software Timer-based Sampling

- z/OS timer services are efficient, but they are not designed for sampling. Significant CPU is consumed.
- Dispatch latency is unpredictable.
- Timer resolution higher than 100 samples per second adds significant complexity and even higher CPU consumption.
- Sampling code must make an educated “guess” at what the dispatcher would have run, if the sampling code was not there, and record those assumed PSWs.
- SRB routines (especially non-preemptible SRBs) are difficult to sample.
- Cycles Per Instruction (CPI) information is not available.
IBM Z Hardware Instrumentation

- Hardware Instrumentation is a mainframe hardware facility that was introduced long before IBM Z, but was accessible only to IBM internal tooling through activation of a special diagnostic mode on the machine.

- The facility was first externalized to customers with the z10 family of processors (z10EC and z10BC).

- Sampling using Hardware Instrumentation is almost “free.” There is no appreciable overhead.

- The default sampling frequency is 800,000 samples per minute. That’s 13,333 samples per second – PER CPU!

- Cycles Per Instruction (CPI) information is available if you know how to calculate it.
IBM Z Hardware Instrumentation

- The first operating system release to support Hardware Instrumentation was z/OS 1.9. For five releases, the IBM Hardware Instrumentation Services (HIS) address space performed all data collection and mapping activities.
  - Functionality extremely limited: only one data collection per system at a time, jobs to be mapped had to be running and execute for the entire duration, no recording of fetch/unfetch activities – mapping was a “snap shot” at the end.
- In z/OS 2.1, the capabilities of HIS were greatly expanded to allow authorized applications to become profilers.
- zHISR leverages these new HIS capabilities as well as other operating system functions to create an easy to use, near-zero overhead application profiler.
If you have an increase in CPU cost in a module, it's often useful to know if the module or a loop in the module is executed more frequently (higher path length) or if the average instruction cost has gone up (higher CPI).

Years ago, when instructions executed one at a time on a CPU, a signal called Instruction First Cycle (IFC) was turned on for the first cycle of an instruction.

IFC allowed us to estimate the average Cycles Per Instruction (CPI) in a module.

\[
CPI_{\text{in Module}} = \frac{Samples_{\text{in Module}}}{IFC_{\text{samples in Module}}}
\]
Samples provide an indication of CPU cost in a module or section of code. IFC samples provided an indication of frequency of various paths in the code. Regardless of how long the instruction took to execute, the IFC signal was only on for one cycle, providing instruction frequency, not instruction execution time.

Today, things are not so simple. Groups of instructions execute at the same time (superscalar) and OOO, but we still want the useful information from the old IFC signal.

The Unique Instruction counts captured by Hardware Instrumentation are used by zHISR to calculate CPI for each execution analysis unit. The result is presented in terms of a ratio relative to the owning section or module.
# zHISR Data Collection Flow

<table>
<thead>
<tr>
<th>CPO</th>
<th>CP1</th>
<th>CP2</th>
<th>...</th>
<th>CPn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Instrumentation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDB</td>
<td>SDB</td>
<td>SDB</td>
<td>...</td>
<td>SDB</td>
</tr>
</tbody>
</table>

**System z Hardware**
- Fixed memory blocks on z/OS populated with sample data
- SDB full condition generates interrupt
- zHISR quickly copies SDBs to its private area and schedules high-speed write to disk files

![Diagram showing data collection flow]

**Notes:**
- CP0, CP1, CP2, ..., CPn represent the different hardware components.
- SDB stands for System Diagnostics Blocks.
The first data collection registers zHISR’s HIS Profiler.

Additional data collections do not register additional HIS Profilers. Only one is ever registered.

When no more data collections are running, zHISR’s HIS Profiler is deregistered.

This approach ensures the “performance” path, i.e., when copying the populated SDBs to zHISR’s private area in response to the full-SDB interrupt, is as short as possible.

The private area SDB copies are simultaneously written to disk, for each running data collection that needs them, and then made available for future copy/write operations.
zHISR HIS Profiler Registration

Advancing Time

USER1
START

USER2
START

USER2
END

USER1
END

zHISR Server

REGISTER

SDB Handling

DEREGISTER

HIS
Native HIS maps modules only at data collection end time.

In many applications, modules are fetched and unfetched throughout execution. A newly-loaded module can occupy the address range previously occupied by another module.

In some applications (e.g., CICS) “directed load” techniques are used. No CDE is created.

zHISR monitors module fetch/unfetch activity, including “directed” loads. The HIS module mapping format has been compatibly extended to record necessary timings.

At analysis time, a time-oriented module matrix is created and used to ensure samples are attributed to the proper module instance.
zHISR Fetch/Unfetch Monitoring

Virtual Address

Advancing Time

Samples
Which Jobs are Monitored and Mapped?

- All jobs are always monitored when a collection is running.
  - That’s just how Hardware Instrumentation works!

- Already-running jobs for which module mapping is desired can be identified by an ASID list and/or job name mask list.
  - A list of job names owned by a given userid can be generated for you on request.

- The **Auto Start Id** and **Match Limit** parameters allow collections to be deferred until a named job actually starts.
  - Parameters similar to SLIP ID= and MATCHLIM= keywords.
  - Makes it possible to monitor/map short-running batch jobs.

- A program can invoke the zHISR API to start/stop/pause its own data collection to target only a subset of its code.
Service access via space-switching PC routine interface.

Server fully supports ASN/LX reuse (REUSASID=YES).

Command interface allows full start/stop/modify control of data collections from MCS console.
  - End-user data collection management is via EMCS console.

Data collections are fully multi-tasked to minimize latency.

Files can be written to zFS using z/OS UNIX file system interfaces or to classic, multivolume MVS data sets using Phoenix Software International’s proprietary STARTIO driver, which performs like NO OTHER.
  - STARTIO driver fully supports advanced channel program technologies including ZHPF. Same driver used for (E)JES!
Starting a zHISR Data Collection
Starting a zHISR Data Collection Run

06/30/2017  zHISR: System Log / Start  07:48:19

F ZHISR,START CLIENT=YES,
F ZHISR,CONT DURATION=10.0,
F ZHISR,CONT PATH="/u/rmullin/zHISR",
F ZHISR,CONT MAPJOB=(RMULLI2),
F ZHISR,CONT TITLE='Demonstration'
ZHS015I: START command has been serviced
Displaying zHISR Data Collection Status

06/30/2017  zHISR: System Log / Status  07:49:08

F ZHISR,STATUS CLIENT=YES

ZH5004I: zHISR Server (V02R9.003C.253.00000) is ONLINE
Connections in progress: 0  Client requests: 9703
Collections in progress: 1  Collections taken: 6
Samples written: 0.27K  Samples lost: 0
Server CPU time: 00:02:27.572  Server zIIP time: 00:00:03.800
Client CPU time: 00:00:00.369  Client Delay time: 00:00:08.539
Client delay HWM: 1  Repository HWM: 24
CPU count: 10  Buffers per CPU: 11

<table>
<thead>
<tr>
<th>Item</th>
<th>User_Id</th>
<th>Job/ASID</th>
<th>Time</th>
<th>Space</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--------</td>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>--------------------</td>
</tr>
<tr>
<td>1</td>
<td>RMULLIN</td>
<td>RMULLI2</td>
<td>000911</td>
<td>38.15G</td>
<td>Demonstration</td>
</tr>
</tbody>
</table>

F ZHISR,STATUS ID=ALL,CLIENT=YES

ZH5074I: No AutoStart Id's match selection criteria
Stopping a zHISR Data Collection Run

06/30/2017 zHISR: System Log / Status 07:49:08

F ZHISR,STATUS CLIENT=YES

ZHS004I: zHISR Server (V02R9.003C.253.00000) is ONLINE
Connections in progress: 0  Client requests: 9703
Collections in progress: 1  Collections taken: 6
Samples written: 0.27K  Samples lost: 0
Server CPU time: 00:02:27.572  Server zIIP time: 00:00:03.800
Client CPU time: 00:00:08.369  Client Delay time: 00:00:08.539
Client delay HWM: 1  Repository HWM: 24
CPU count: 10  Buffers per CPU: 11

<table>
<thead>
<tr>
<th>Item</th>
<th>User_Id</th>
<th>Job/ASID</th>
<th>Time</th>
<th>Space</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>1</td>
<td>RMULLIN</td>
<td></td>
<td></td>
<td>Demonstration</td>
</tr>
</tbody>
</table>

F ZHISR,STATUS ID=ALL,CLIENT=YES

ZHS074I: No AutoStart Id's match selection criteria
Stopping a zHISR Data Collection Run

06/30/2017  zHISR: System Log  / Stop  07:52:56

F ZHISR,STOP 0001,CLIENT=YES
ZH3015I: STOP command has been serviced
**zHISR Data Collection Analysis Wizard**

- **Navigate File Tree**
  - **Select Files, Time Range, Unit Size***
    - **PASN Chooser**
      - **Fast-path Range, Module, Section Chooser**
          - **F2=Anal**
          - **F5=Next**
      - **F5=Next**
      - **Module Chooser**
          - **F5=Next**
      - **Section Chooser**
          - **F5=Next**
      - **Boundary Chooser**
          - **F5=Next**
      - **Interactive Analysis Reports**

* Optional

* Unit size can be any power of two, 8 thru 4K. Default is 64 bytes.
Select File: Choose Analysis Unit Size

<table>
<thead>
<tr>
<th>Type</th>
<th>Permission</th>
<th>Owner</th>
<th>Group</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dir</td>
<td>drwxr-xr-x</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>.</td>
</tr>
<tr>
<td>Dir</td>
<td>drwx--------</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>.</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170627.101835.000.MAP</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170627.101835.000.SMP.00</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170627.101835.000.SMP.02</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170627.101835.000.SMP.06</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170627.101835.000.SMP.07</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170628.160736.000.MAP</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170628.160736.000.SMP.00</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170628.160736.000.SMP.02</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170628.160736.000.SMP.06</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170629.095402.000.MAP</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170629.095402.000.SMP.00</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170629.095402.000.SMP.02</td>
</tr>
<tr>
<td>File</td>
<td>-rw-r--r--</td>
<td>RMULLIN</td>
<td>DEV</td>
<td>PSIHIS20170629.095402.000.SMP.06</td>
</tr>
</tbody>
</table>
Specify Time Period and Included CPUs

<table>
<thead>
<tr>
<th>Date</th>
<th>zHISR: H.I.S. Sample Data CPU Selection</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>06/30/2017</td>
<td>PSIHIS20170629.095402.000</td>
<td>07:54:34</td>
</tr>
</tbody>
</table>

CPU Report

Move the cursor to a line to be excluded or included in the analysis and press Enter. Alter the range data to change the time period that is to be analyzed. Once you have completed your selections press PF5.

Collection time period in YYDDD.HH:MM:SS format: 17180.16:52:33 17180.16:54:12

<table>
<thead>
<tr>
<th>Location</th>
<th>Time</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSIHIS20170629.095402.000.SMP.00</td>
<td>200.0K</td>
<td>17180.16:52:33</td>
<td>17180.16:54:10</td>
</tr>
<tr>
<td>PSIHIS20170629.095402.000.SMP.02</td>
<td>44.0K</td>
<td>17180.16:52:40</td>
<td>17180.16:54:05</td>
</tr>
<tr>
<td>PSIHIS20170629.095402.000.SMP.06</td>
<td>80.0K</td>
<td>17180.16:52:40</td>
<td>17180.16:53:56</td>
</tr>
<tr>
<td>PSIHIS20170629.095402.000.SMP.07</td>
<td>84.0K</td>
<td>17180.16:52:40</td>
<td>17180.16:54:12</td>
</tr>
</tbody>
</table>
### Primary ASN Chooser

**Move the cursor to the ASID of the address space to be included in the analysis and press Enter. Only one ASID may be selected.**

Once you have completed your selection press PF2 or PF5.

<table>
<thead>
<tr>
<th>ASID</th>
<th>Percentage</th>
<th>ASID</th>
<th>Percentage</th>
<th>ASID</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.251% 0001</td>
<td>23.735% 0065</td>
<td>9.621% 0002-RMULL1Z</td>
<td>7.458% 0045</td>
<td>4.168% 0010</td>
<td>3.646% 0077</td>
</tr>
<tr>
<td>1.583% 0006</td>
<td>1.068% 0039</td>
<td>0.677% 007E</td>
<td>0.590% 0008</td>
<td>0.443% 0007</td>
<td>0.440% 004C</td>
</tr>
<tr>
<td>0.392% 0066</td>
<td>0.312% 003A</td>
<td>0.297% 002E</td>
<td>0.416% 0066</td>
<td>0.250% 0009</td>
<td>0.211% 002D</td>
</tr>
<tr>
<td>0.296% 0047</td>
<td>0.175% 0027</td>
<td>0.166% 0026</td>
<td>0.192% 0050</td>
<td>0.133% 006C</td>
<td>0.101% 0016</td>
</tr>
<tr>
<td>0.149% 003F</td>
<td>0.058% 004E</td>
<td>0.052% 004B</td>
<td>0.089% 0051</td>
<td>0.031% 0002</td>
<td>0.028% 0023</td>
</tr>
<tr>
<td>0.041% 0010</td>
<td>0.026% 0008</td>
<td>0.025% 0042</td>
<td>0.028% 0048</td>
<td>0.022% 001E</td>
<td>0.017% 0025</td>
</tr>
<tr>
<td>0.025% 0042</td>
<td>0.016% 0080</td>
<td>0.015% 001E</td>
<td>0.016% 0080</td>
<td>0.009% 0017</td>
<td>0.007% 0019</td>
</tr>
<tr>
<td>0.009% 0017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proceed directly to Fast-path Range, Module, Section Chooser.

1=Help  2=Anal  3=End  5=Next  7=Bwd  8=Fwd  9=Sort ASID
Module Chooser

Move the cursor to a module to be excluded or included in the analysis and press Enter. Continue this process as often as needed. Once you have completed your selections press PF5.

0082-DPXWREXX 00000000_0000C000 00021F30 RMULL12 17180.1652326 -
0082-EJESBENV 00000000_00271000 00000800 RMULL12 17180.1652326 -
0082-EJESC13 00000000_00250000 00005000 RMULL12 17180.1652326 -
0082-EJESA03 00000000_002C8000 00010800 RMULL12 17180.1653418 -
0082-EJESD13 00000000_00280000 00013000 RMULL12 17180.1652326 -
0082-EJESF13 00000000_0029E000 0000C000 RMULL12 17180.1652494 -
0082-EJESL13 00000000_001EC000 0005E000 RMULL12 17180.1652326 -
0082-EJESL13 00000000_00168000 0000B000 RMULL12 17180.1652326 -
0082-EJESML 00000000_00172000 00007000 RMULL12 17180.1652326 -
0082-EJESMS 00000000_00272000 00006000 RMULL12 17180.1652326 -
0082-EJESOPTS 00000000_00D8B000 000
0082-EJESPDK 00000000_002C6000 000
0082-EJESPDK 00000000_0013A000 000
0082-EJESPDK 00000000_0066000 000
0082-EJESPDK 00000000_0067000 000

Include/Exclude mask: **EJE* **
To select all privates, type *PVT.
3=End 4=Include 5=Exclude

0082 00100 1=Help 2=Mask 3=End 5=Next 7=Bwd 8=Fwd 9=Sort Address FB
Control Section Chooser

Move the cursor to a Csect to be excluded or included in the analysis and press Enter. Continue this process as often as needed. Once you have completed your selections press PF5.

Include/Exclude mask: EJE*
To select all privates, type *PVT.
3=End 4=Include 5=Exclude

0082-B0EKJUMP 00000000_0000E568 00000000 17180.1652326 -
0082-B0ETRT 00000000_00010630 00000100 17180.1652326 -
0082-B0EXDVFN 00000000_00010F00 00000220 17180.1652326 -
0082-BPXWESTA 00000000_008111A0 000008BC 17180.1652326 -
0082-BPXWREXX 00000000_00811230 00003C7C 17180.1652326 -
0082-BPXWRFM 00000000_00820B08 000000AE 17180.1652326 -
0082-BPXWRFMS 00000000_00820C88 000000BC 17180.1652326 -
0082-BPXWRFMG 00000000_00820F20 000000B4 17180.1652326 -
0082-BPXWRIM 00000000_00820D48 000000FC 17180.1652326 -
0082-BPXWRTZG 00000000_00820E48 000000E4 17180.1652326 -
0082-BPXWRXST 00000000_0082 -
0082-BPXKIL 00000000_0012 -
0082-BPXK1IL 00000000_001E -
0082-BPX1MPC 00000000_0012 -
0082-BPX1MPC 00000000_001E -
Virtual Storage Boundary Chooser

Move the cursor to a boundary to be excluded or included in the analysis and press Enter. Continue this process as often as needed. Once you have completed your selections press PF5.
Fast-path Range, Module, Section Chooser

All available modules, cssects and boundaries have been selected

Up to ten modules, Csects or address ranges may be entered into this screen. To specify a module or Csect, enter its name in the left field while setting the right field to blanks. Press PF2 to display a usage report that includes all the data that you have selected for address space 0082.

Wait state: N  Include Wait dispatched, Y or N

Address range:  

Press <F2> to Create Interactive Analysis Reports

Press <F9> to include all virtual storage ranges. Useful when nothing else has been selected previously.

1=Help 2=Run 3=End 4=Return 7=Counter 8=Info 9=Select All 10=Memory 11=Clear
Interactive Analysis Report Navigation

- The Full Analysis shows all execution analysis units with the most frequently-executed at the top of the display.
  - Control section, module and boundary are displayed for every execution analysis unit.
  - Change sort order as desired using cursor-based selection.
- Use cursor-based selection to drill down to the Spot Analysis, where all execution analysis units for a given control section, module or virtual storage boundary are shown.
- From there, you can display control section source code with execution analysis unit highlighted – if ADATA or COBOL SYSDEBUG information is available.
### Full Analysis

**Run Summary**

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Address</th>
<th>TotPct</th>
<th>SelPct</th>
<th>Csect</th>
<th>Module</th>
<th>Bndry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0082</td>
<td>00000000_002575C0</td>
<td>1.876</td>
<td>19.500</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00257680</td>
<td>1.126</td>
<td>11.703</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00257640</td>
<td>0.699</td>
<td>7.257</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_001E4A00</td>
<td>0.683</td>
<td>7.100</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_018AD5B0</td>
<td>0.342</td>
<td>3.557</td>
<td>IAXFH</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00257660</td>
<td>0.273</td>
<td>2.846</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00257570</td>
<td>0.256</td>
<td>2.664</td>
<td>EJEOPATF</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00257600</td>
<td>0.205</td>
<td>2.134</td>
<td>IAXFH</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_002555C0</td>
<td>0.119</td>
<td>1.241</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_01899C00</td>
<td>0.103</td>
<td>1.074</td>
<td>IAXFH</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00179200</td>
<td>0.103</td>
<td>1.074</td>
<td>EJESSUBS</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00137A00</td>
<td>0.005</td>
<td>0.85</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00179580</td>
<td>0.005</td>
<td>0.85</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_00255340</td>
<td>0.005</td>
<td>0.85</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_7F336F00</td>
<td>0.068</td>
<td>0.711</td>
<td>-</td>
<td>-</td>
<td>EPRV</td>
</tr>
</tbody>
</table>

- **Point and <F6> for Location Pop-Up**
- **Click for Spot Analysis of Section, Module or Boundary.**
- **Press <F9> to rotate through available sorts.**
Full Analysis with Location Pop-up

Cursor positioned to second line; press <F6> repeatedly to step through Location Pop-Ups.
# Spot Analysis for Control Section

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Address</th>
<th>Offset</th>
<th>TotPct</th>
<th>SelPct</th>
<th>Csect</th>
<th>Module</th>
<th>Endry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0082</td>
<td>00000000_0D1EA300</td>
<td>00000210</td>
<td>0.034</td>
<td>0.363</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_0D1EA3C0</td>
<td>00000200</td>
<td>0.052</td>
<td>0.545</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_0D1EA400</td>
<td>00000310</td>
<td>0.052</td>
<td>0.545</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_0D1EA440</td>
<td>00000350</td>
<td>0.034</td>
<td>0.363</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_0D1EA4C0</td>
<td>00000300</td>
<td>0.663</td>
<td>7.100</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
<tr>
<td>0082</td>
<td>00000000_0D1EA500</td>
<td>00000410</td>
<td>0.119</td>
<td>1.241</td>
<td>EJEEXMOV</td>
<td>EJESSUBS</td>
<td>EPRV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Address</th>
<th>Offset</th>
<th>TotPct</th>
<th>SelPct</th>
</tr>
</thead>
<tbody>
<tr>
<td>0081</td>
<td>000001</td>
<td>000001</td>
<td>0.97</td>
<td>10.15</td>
</tr>
</tbody>
</table>

Click here to show source code via ADATA
Define the SYSADATA, SYSPRINT, SYSCPRT or SYSDEBUG dataset name using the Catalog Search mask and the PF6 key. Correct the Target member name if needed. If using SYSDEBUG, provide the Target program name. After these fields have been provided press the PF2 key to view the associated data.

Catalog Search mask **ADATA**
Target member name **EJEOXMOV**

Target program name
(Only if SYSDEBUG)
(Case sensitive)

Report page limit **1000**

Instruction address **00000000_001EA4C0**
Instruction offset **000003D0**

1=Help 2=Adata 3=End 5=Dataset(s) 6=Catalog Search 11=Clear
Scrollable ADATA with Highlighted Code from Execution Analysis Unit

06/30/2017  zHISR: Associated Data  (06-22-2017,15:19)  08:08:59
EJES.PRODGEN.ADATA

0003CC 5133 0000 0000 79730  L A E  R3,0(R3,0)  Ens
0003D0 A544 00FF 00FF 79731  N I H H  R4,'00FF'  Zer
0003D4 B219 0200 0020 79732  S A C  'X'200'  *  Swi
0003D8 A748 FF00 FF00 79733  A G H I  R4,-256  Sub
0003DC A7D4 000D 003F6 79734  I F  P  T f
0003E0 A7D4 000D 003F6 79744+  B R C  13,#0LB20
0003E0 79746  D D  U N T I L = N P  D
0003E0 79754+#0LB23  D C  O H
0003E0 E50E 3000 2000 0000 0000 79764  M V C S K  0(R3),0(R2)
0003E6 4120 2100 00100 79765  L A  R2,256(.R2)
0003EA 4130 3100 00100 79766  L A  R3,256(.R3)
0003EE A748 FF00 FF00 79767  A G H I  R4,-256
0003F2 79768  E N D D O ,
0003F2 79773+#0LB24  D C  O H
0003F2 A724 FF7 003E0 79774+  B R C  2,#0LB20
0003F6 79777  E N D I F ,
0003F6 79782+#0LB20  D C  O H
0003F6 0908 0004 0000 79784  A G R  R0,R4
0003F6 79785  I F  N M  T f
00012 00025 1=Help 3=End 4=Return 7=Owd 8=Fwd 10=Lft 11=Rht 12=Print  F8
Print, Save or Export Results

- The Full Analysis, Spot Analysis and ADATA source code reports can be printed or saved. These reports are text versions of the 3270-based reports – all rows shown.
- Exporting the Full Analysis or Spot Analysis report to a CSV (comma-separated values) file allows you to easily import the data into your favorite spreadsheet or charting utility.
To print the current report provide a valid SYSOUT class and press the PF4 key. To save the current report supply the name of a PDS or PDSE dataset that you are authorized to update and the member name that is to contain the report and press the PF5 key. To export the current report as a CSV file specify the output path name and press the PF7 key.

Sysout Class_ _
Output Dataset Name ________________________________
Output Member Name _____

Output Path Name ________________________________
Character encoding ASCII

1=Help  3=End  4=Print  5=Save  7=Export  11=Clear
### Import CSV File into Your Spreadsheet

The image shows a spreadsheet with a chart and a table. The chart is a bar chart with categories on the x-axis labeled 'PASN' and 'PSW_Add', and numerical values on the y-axis labeled 'TotPct'. The table includes columns labeled 'PASN', 'PSW_Add', 'TotPct', 'SelPct', 'Csect', 'Module', and 'Bndry'. The data appears to be related to some form of analysis or reporting. The spreadsheet is open in Microsoft Excel, with the file name 'demo.CSV'.

#### Table Example

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Add</th>
<th>TotPct</th>
<th>SelPct</th>
<th>Csect</th>
<th>Module</th>
<th>Bndry</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>000000000</td>
<td>1.87618</td>
<td>19.5004</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>1.126</td>
<td>11.7033</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.6992</td>
<td>7.26722</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.68318</td>
<td>7.10068</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.34232</td>
<td>3.55791</td>
<td>IAXFN</td>
<td>IAXFN</td>
<td>ERON</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.27385</td>
<td>2.84633</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.25637</td>
<td>2.66465</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.20539</td>
<td>2.13475</td>
<td>IAXFN</td>
<td>IAXFN</td>
<td>ERON</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.11945</td>
<td>1.24148</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.11945</td>
<td>1.24148</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.10342</td>
<td>1.07494</td>
<td>IAXFN</td>
<td>IAXFN</td>
<td>ERON</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.10342</td>
<td>1.07494</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.08594</td>
<td>0.89326</td>
<td>GABUFRG</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
<tr>
<td>82</td>
<td>000000000</td>
<td>0.08594</td>
<td>0.89326</td>
<td>EJESSUB3</td>
<td>EJESSUB3</td>
<td>EPRV</td>
</tr>
</tbody>
</table>
**C++ Spot Analysis for Control Section**

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Address</th>
<th>Offset</th>
<th>TotPct</th>
<th>SelPct</th>
<th>Csect</th>
<th>Module</th>
<th>Endry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0078</td>
<td>00000000 00000260 000001E0</td>
<td>0.381</td>
<td>0.668</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000268 000001E0</td>
<td>15.900</td>
<td>27.876</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000270 000001F0</td>
<td>3.275</td>
<td>5.742</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000278 000001F0</td>
<td>4.359</td>
<td>7.642</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000280 00000200</td>
<td>0.071</td>
<td>0.125</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000200</td>
<td>0.071</td>
<td>0.125</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000220</td>
<td>0.559</td>
<td>0.981</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000220</td>
<td>0.262</td>
<td>0.354</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000230</td>
<td>0.063</td>
<td>0.146</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000230</td>
<td>0.559</td>
<td>0.981</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000240</td>
<td>0.273</td>
<td>0.400</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000240</td>
<td>0.035</td>
<td>0.062</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000250</td>
<td>1.543</td>
<td>2.881</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000250</td>
<td>0.119</td>
<td>0.208</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
<tr>
<td>0078</td>
<td>00000000 00000220 00000250</td>
<td>0.976</td>
<td>1.712</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>CPPSAM</td>
<td>EPRV</td>
</tr>
</tbody>
</table>

Click here to show source listing.
# Scrollable C++ Source with Highlighted Code from Execution Analysis Unit

```cpp
#pragma csect (CODE,"CPPSAMP")
typedef class Main {

    private:

    int i1, i2, result;
    char reply, junk[3910];

    void zeroit() {
        result = 0;
    }

    void loop() {
        zeroit();
        result = i1 + i2;
        result = i1 + i2;
        result = i1 + i2;
        result = i1 + i2;
        result = (((i1 / i2) + (i1 / i2)) / 2) / 1 / 1 / 1;
    }
```

The offending line
C Spot Analysis for Control Section

<table>
<thead>
<tr>
<th>PASN</th>
<th>PSW_Address</th>
<th>Offset</th>
<th>TotPct</th>
<th>SelPct</th>
<th>Csect</th>
<th>Module</th>
<th>Ordry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0078</td>
<td>00000000_00000218_00000198</td>
<td>7.197</td>
<td>22.282</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000220_000001A0</td>
<td>3.445</td>
<td>10.666</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000200_00000180</td>
<td>3.034</td>
<td>9.394</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001C8_00000148</td>
<td>2.300</td>
<td>7.122</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000100_000001C0</td>
<td>1.988</td>
<td>6.155</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000100_000001C0</td>
<td>1.550</td>
<td>4.799</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000100_000001C0</td>
<td>1.013</td>
<td>3.137</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000100_000001C0</td>
<td>0.903</td>
<td>2.798</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000198_00000118</td>
<td>0.816</td>
<td>2.526</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001D0_00000150</td>
<td>0.547</td>
<td>1.695</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001D0_00000150</td>
<td>0.509</td>
<td>1.577</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001D0_00000150</td>
<td>0.482</td>
<td>1.492</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000230_000001A0</td>
<td>0.471</td>
<td>1.458</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001A8_0000012B</td>
<td>0.465</td>
<td>1.441</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000002F0_00000270</td>
<td>0.460</td>
<td>1.424</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_00000168_000000E8</td>
<td>0.454</td>
<td>1.407</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>00000000_000001C0_00000140</td>
<td>0.454</td>
<td>1.407</td>
<td>CSAMP</td>
<td>CSAMP</td>
<td>EPRV</td>
<td></td>
</tr>
</tbody>
</table>

Click here to show source listing
The offending line is:

```c
result = ((i1 / i2) + (i1 / i2)) / 2 / 1 / 1 / 1;
```
The biggest single enhancement is normalization across engines with different cycle times.

For example, suppose an equal number of samples is collected from a zIIP and a sub-capacity CP running only half as fast.

Prior zHISR releases would have reported a 1:1 ratio between consumption of the two cores/threads.

The new release will report the ratio between the two cores/threads as 2:1 when an equal number of samples is collected. That is, only \( \frac{1}{3} \) of normalized CPU consumption will be attributed to the zIIP.
What’s New?

- z/Architecture Instruction Speed Tester
  - Generalized comparisons
  - Many variables can affect real-time instruction performance
Relative performance: Cpu 90   Clock 90
Seconds per instruction: Cpu 0.0000000090   Clock 0.0000000090

Supply the mnemonic of the instruction to be tested and press the enter key. Some instructions may request one or more optional operand specifics to be used when testing the instruction. Once the mnemonic and operand specifics have been supplied press the PF2 key to run the speed test.

Instruction mnemonic **LGHI**
06/30/2017         z/Architecture Instruction Speed Tester         08:43:01
XGR

Relative performance:     Cpu 119     Clock 119
Seconds per instruction: Cpu 0.0000000120     Clock 0.0000000120

Supply the mnemonic of the instruction to be tested and press the enter key. Some instructions may request one or more optional operand specifics to be used when testing the instruction. Once the mnemonic and operand specifics have been supplied press the PF2 key to run the speed test.

Instruction mnemonic XGR
What’s New?

- Hardware Counter Data Collection and Analysis
  - CPU-Measurement and Sampling Facilities feature
Counter Data Collection and Analysis

06/30/2017  zHISR: Counters Analysis  08:56:24
PSIHE520170638.084521.000

Optionally change the three input fields, position the cursor to the counter number(s) to be graphed and press Enter. Press PF2 to view.

System: MVS70  Interval: 5  Intervals: 120  Jobs: 5  Counters: 135
Start time: 084521  End time: 085516  Number of jobs: 5  Interval: 5

<table>
<thead>
<tr>
<th>Basic:</th>
<th>000 001 002 003 004 005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem:</td>
<td>032 033 034 035 036 037</td>
</tr>
<tr>
<td>Crypto:</td>
<td>064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079</td>
</tr>
<tr>
<td>Extended:</td>
<td>120 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143</td>
</tr>
<tr>
<td></td>
<td>144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159</td>
</tr>
<tr>
<td></td>
<td>160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175</td>
</tr>
<tr>
<td></td>
<td>176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191</td>
</tr>
<tr>
<td></td>
<td>192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207</td>
</tr>
<tr>
<td></td>
<td>208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223</td>
</tr>
<tr>
<td></td>
<td>224 225</td>
</tr>
<tr>
<td>Diagnostic:</td>
<td>448 449 450 451 452 453 454 455 456</td>
</tr>
</tbody>
</table>

1=Help  2=View  3=End  11=Clear
Questions?

Contact Information:

Phoenix Software International
831 Parkview Drive North
El Segundo, CA 90245
http://www.phoenixsoftware.com
sales@phoenixsoftware.com