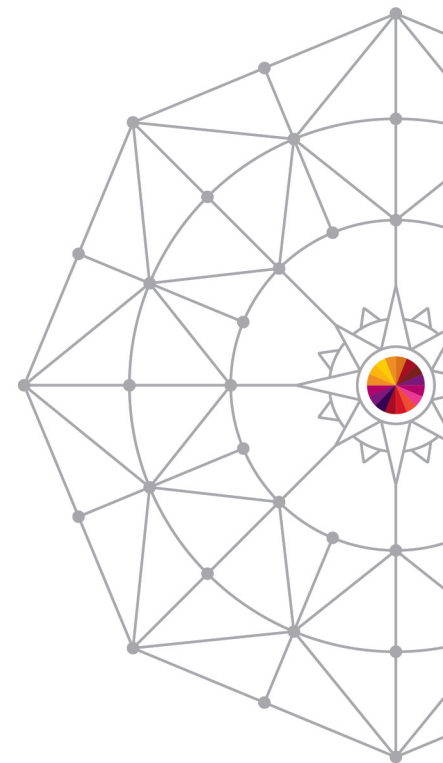


zHISR: Improving Application Performance using Hardware Instrumentation

Ed Jaffe
Phoenix Software International

5 August 2014
Session Number 16262



#SHAREorg



Copyright (c) 2014 by SHARE Inc.  Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Copyright (c) 2014 by SHARE Inc.  Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>

What is zHISR?

zHISR is an interactive application execution profiler that allows developers, performance analysts and others to easily interface with System z Hardware Instrumentation to perform near-zero overhead, high-resolution hot spot analysis of programs running under z/OS.

Software Timer-based Sampling Technologies

- 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.

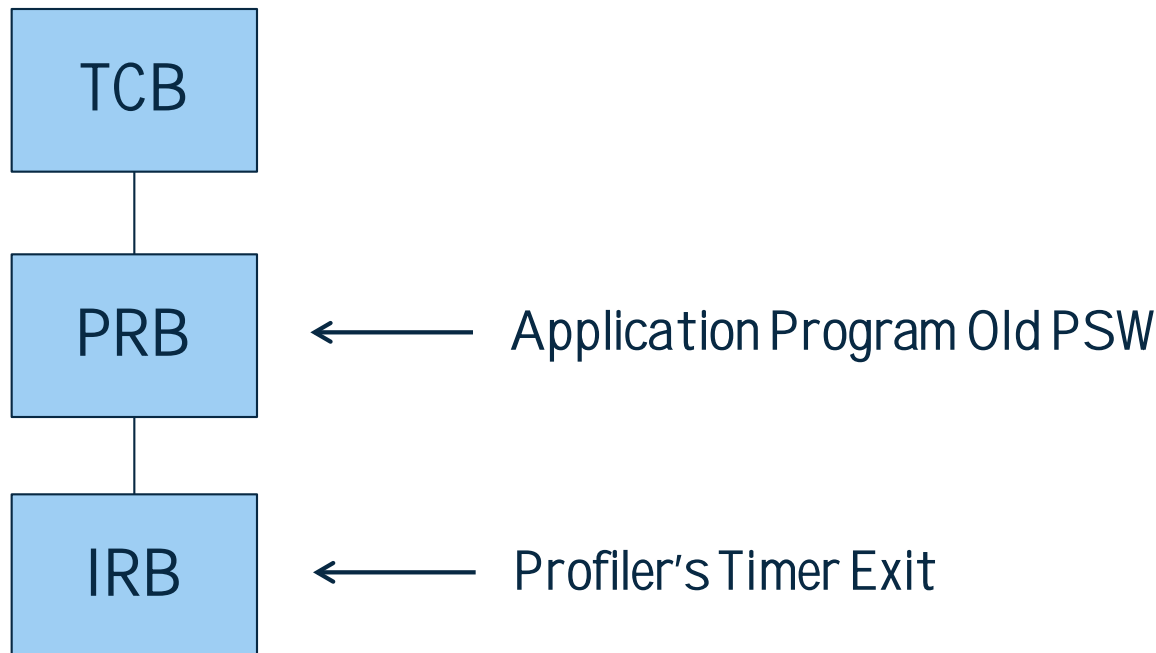
STIMER-Based Sampling

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



STIMER-Based Sampling

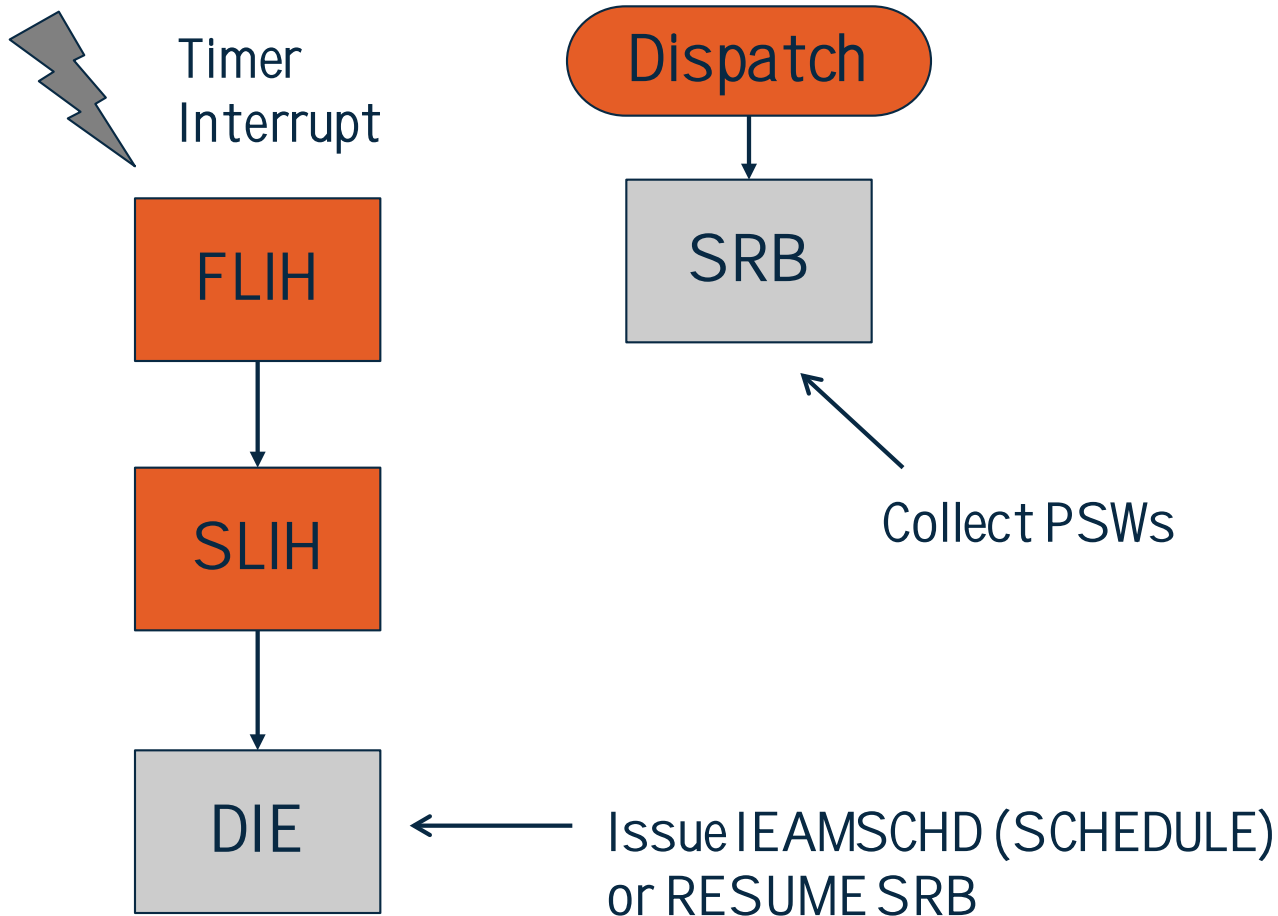
- When timer interrupt occurs, the operating system:
 - Schedules an SRB into the target address space
 - The SRB schedules an IRB to run the timer exit
 - The timer exit collects the PSW from the PRB



Timer DIE-Based Sampling

- 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



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.

System z Hardware Instrumentation

- Hardware Instrumentation is a mainframe hardware facility that was introduced long before System 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.

System 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.

Cycles Per Instruction

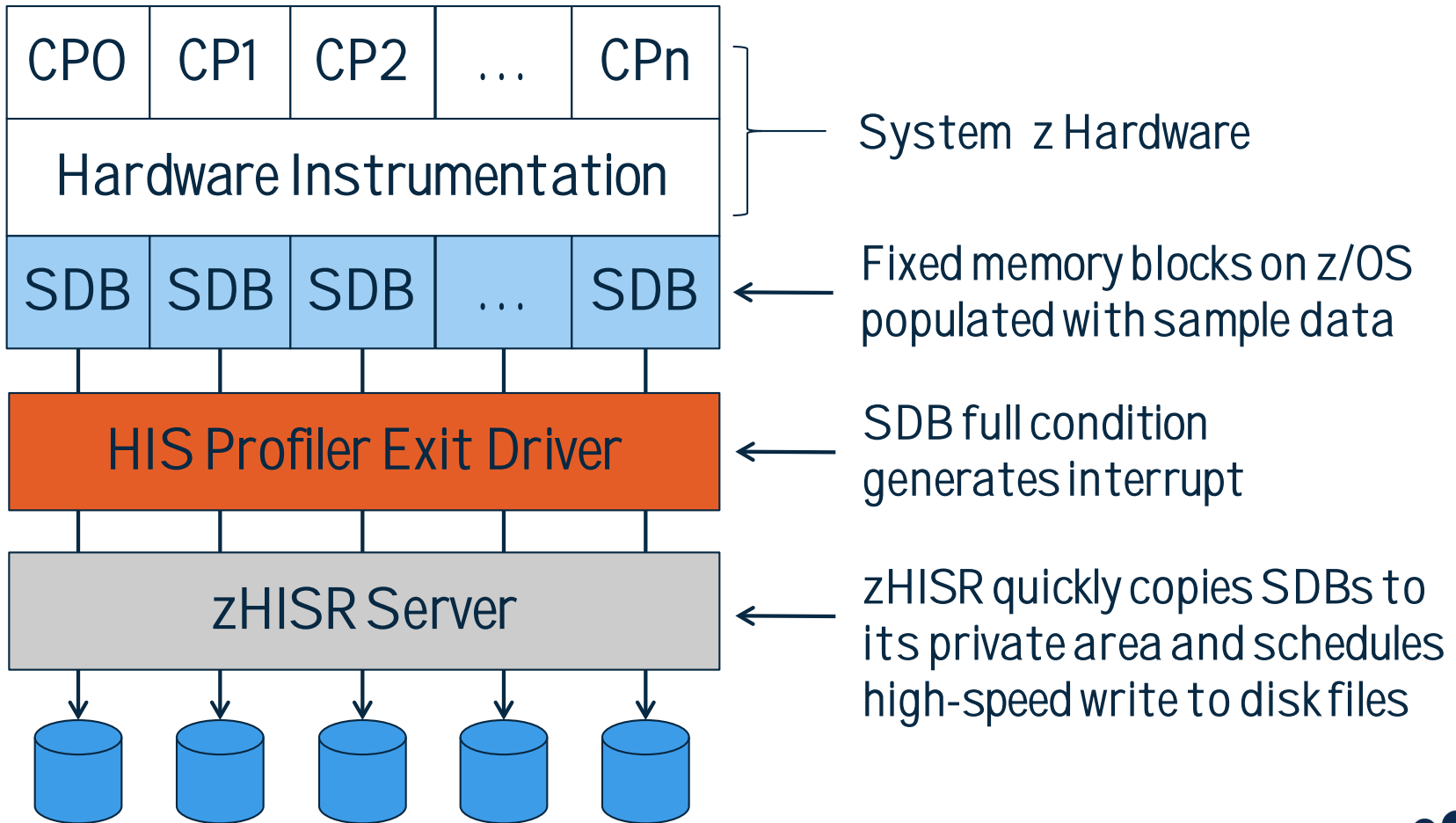
- If you have an increase in CPU cost in a module, it's often useful to know if the module or a loop in the module is executed more frequently (higher path length) or if the average instruction cost has gone up (higher CPI).
- Years ago, when instructions executed one at a time on a CPU, a signal called 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.

$$CPI_{inModule} = \frac{Samples_{inModule}}{IFC_{samples}_{inModule}}$$

Cycles Per Instruction

- 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.

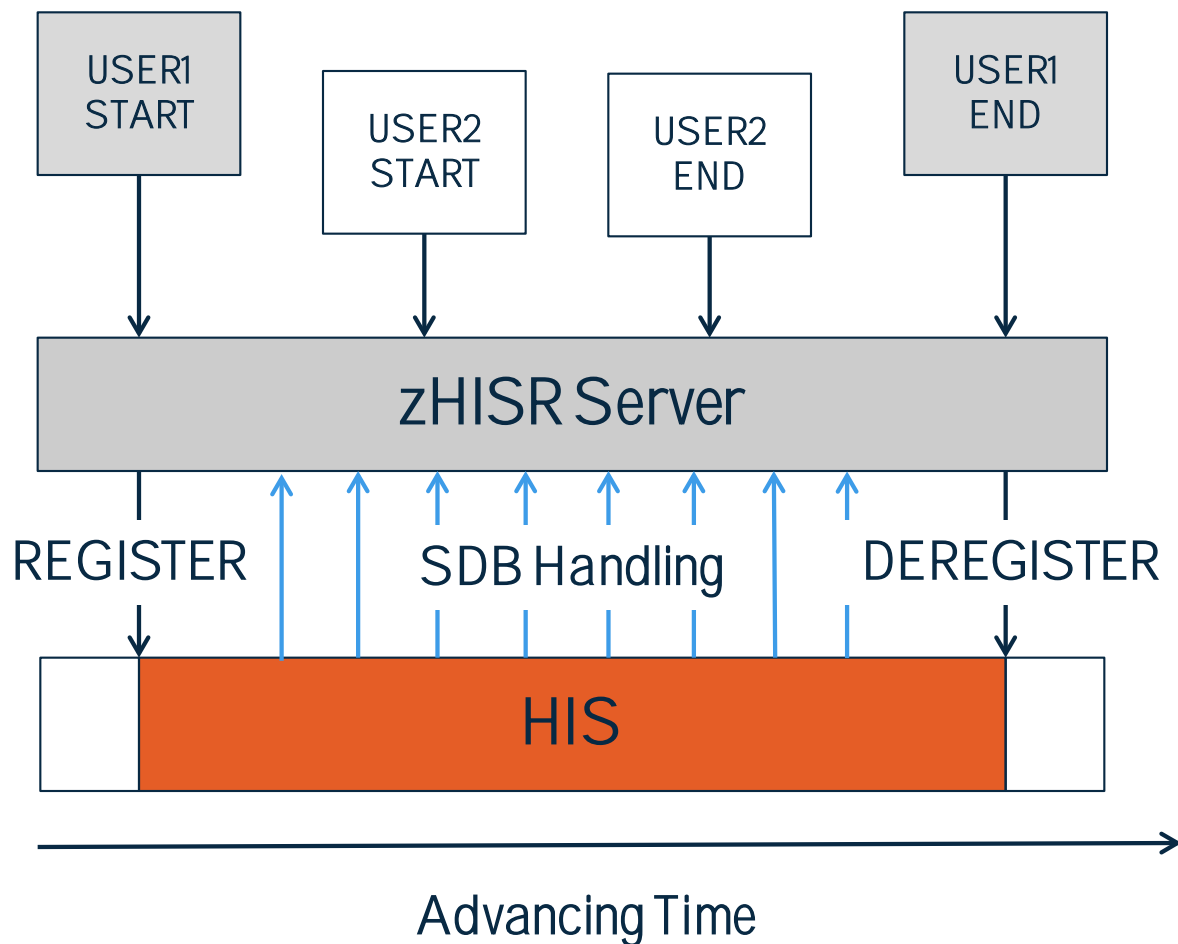
zHISR Data Collection Flow



zHISR HIS Profiler Registration

- 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.

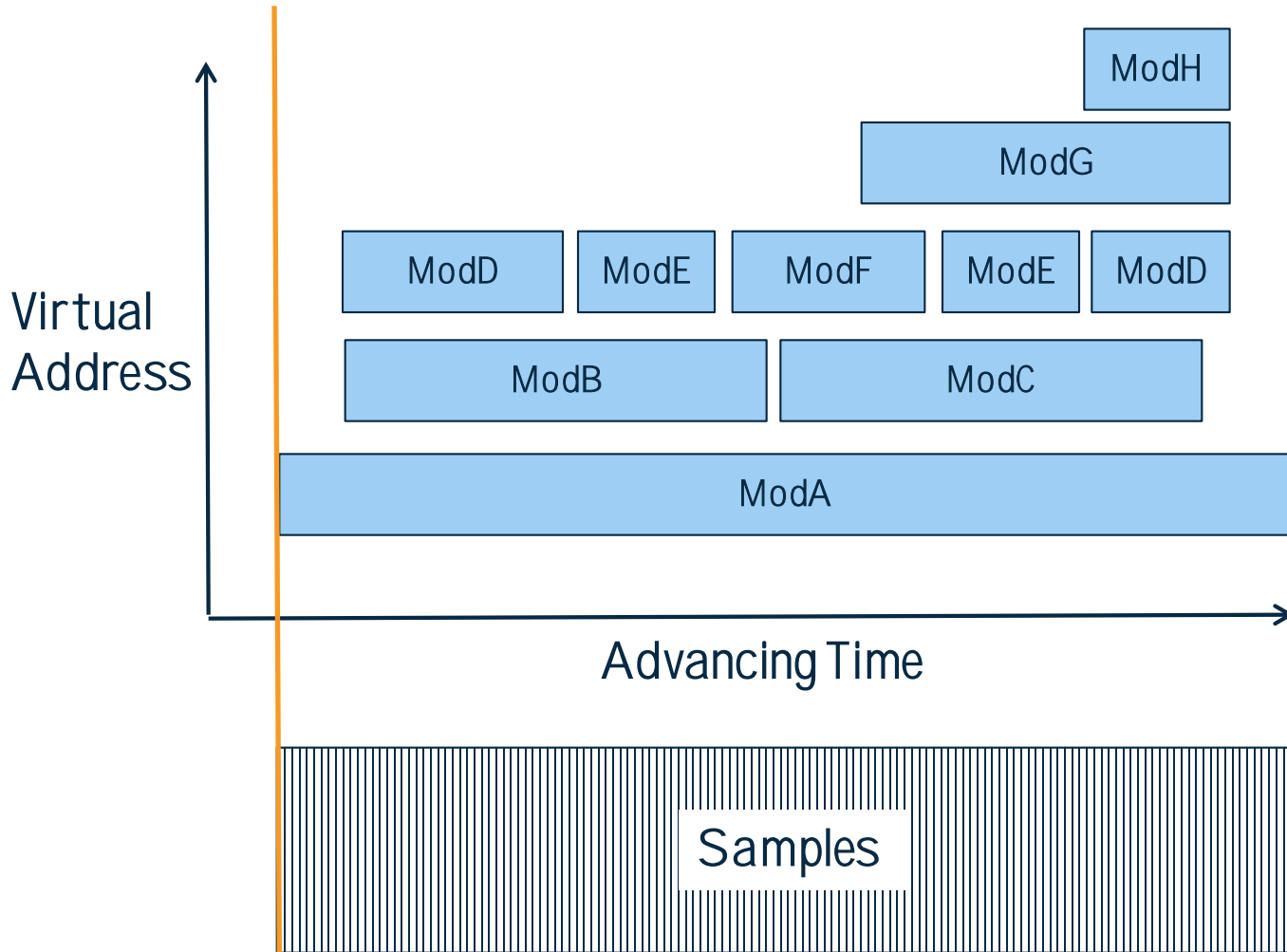
zHISR HIS Profiler Registration



zHISR Fetch/Unfetch Monitoring

- 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



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.

zHISR Server Characteristics

- 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! 😊

Files Created by zHISR

- zHISR creates sample and map files, no counters.
- Sample data format is identical to z/OS 2.1 HIS.
- Map data format is upward compatible to z/OS 2.1 HIS.
 - A format which is *totally incompatible* with earlier releases of z/OS HIS. :-/
- This means existing customer code that processes z/OS 2.1 HIS sample and map files can process zHISR sample and map files, unless the code is sensitive to record length or other things to which it should not be sensitive.

Starting a zHISR Data Collection

```

11/06/2013          zHISR: Manage H.I.S. Event Data Collection          16:23:09

System MVSA0      Proc ZHISR      Title Sample Collection
Duration 10.0    Space 0          MapUser GILBERT    MapOnly N
MapAsid _____
MapJob C3CONDOR _____ _____ _____ _____ _____ _____
_____ _____ _____ _____ _____ _____ _____
_____ _____ _____ _____ _____ _____ _____
Path disk
AutoStart_Id _____ Match_Limit 000001

1=Help 2=View 3=End 4=Start 5=HisInfo 7=Status 8=Jobs 9=Schedule 11=Clear
  
```

Starting a zHISR Data Collection Run

11/06/2013 zHISR: S y s t e m L o g / S t a r t 16:23:19

```
F ZHISR,START CLIENT=YES,  
F ZHISR,CONT DURATION=10.0,  
F ZHISR,CONT PATH=DISK,  
F ZHISR,CONT MAPJOB=(C3CONDOR),  
F ZHISR,CONT TITLE='Sample Collection'  
ZHS059I: SPACE= is required with PATH=DISK, set to 15620
```

0001 of 0001

3=End 7=Bwd 8=Fwd

XXXX FB

Displaying zHISR Data Collection Status

11/06/2013 zHISR: S y s t e m L o g / S t a t u s 16:24:03

F ZHISR,STATUS CLIENT=YES

```
ZHS004I: zHISR Server (V02R0.0037.253) is ONLINE
Connections in progress: 0  Client requests: 428
Collections in progress: 1  Collections taken: 3
Samples written: 0.31K      Samples lost: 0
Server CPU time: 00:00:09.910  Server zIIP time: 00:00:00.391
Client CPU time: 00:00:00.060  Client Delay time: 00:00:00.583
Client delay HWM: 0           Repository HWM: 0
CPU count: 4      Buffers per CPU: 14
```

Item	User_Id	Job/ASID	Time	Space	Title
1	GILBERT	C3CONDOR	000918	60.59M	Sample Collection

F ZHISR,STATUS ID=ALL,CLIENT=YES

ZHS074I: No AutoStart Id's match selection criteria

0001 of 0001 1=Help 2=Refresh 3=End 7=Bwd 8=Fwd P=Stop K=Halt

XXXX FB ARE
gh 2014

Stopping a zHISR Data Collection Run

```

11/06/2013          zHISR: S y s t e m   L o g   /   S t a t u s          16:24:03

F ZHISR,STATUS CLIENT=YES

ZHS004I: zHISR Server (V02R0.0037.253) is ONLINE
Connections in progress: 0  Client requests: 428
Collections in progress: 1  Collections taken: 3
Samples written: 0.31K      Samples lost: 0
Server CPU time: 00:00:09.910  Server zIIP time: 00:00:00.391
Client CPU time: 00:00:00.060  Client Delay time: 00:00:00.583
Client delay HWM: 0          Repository HWM: 0
CPU count: 4      Buffers per CPU: 14

Item User_Id  Job/ASID Time    Space    Title
-----
p _  1 GILBERT  C3CONDOR 000918 60.59M  Sample Collection

F ZHISR,STATUS ID=ALL,CLIENT=YES
ZHS074I: No AutoStart Id's match selection criteria

0001 of 0001      1=Help 2=Refresh 3=End 7=Bwd 8=Fwd P=Stop K=Halt      XXXX FB

```

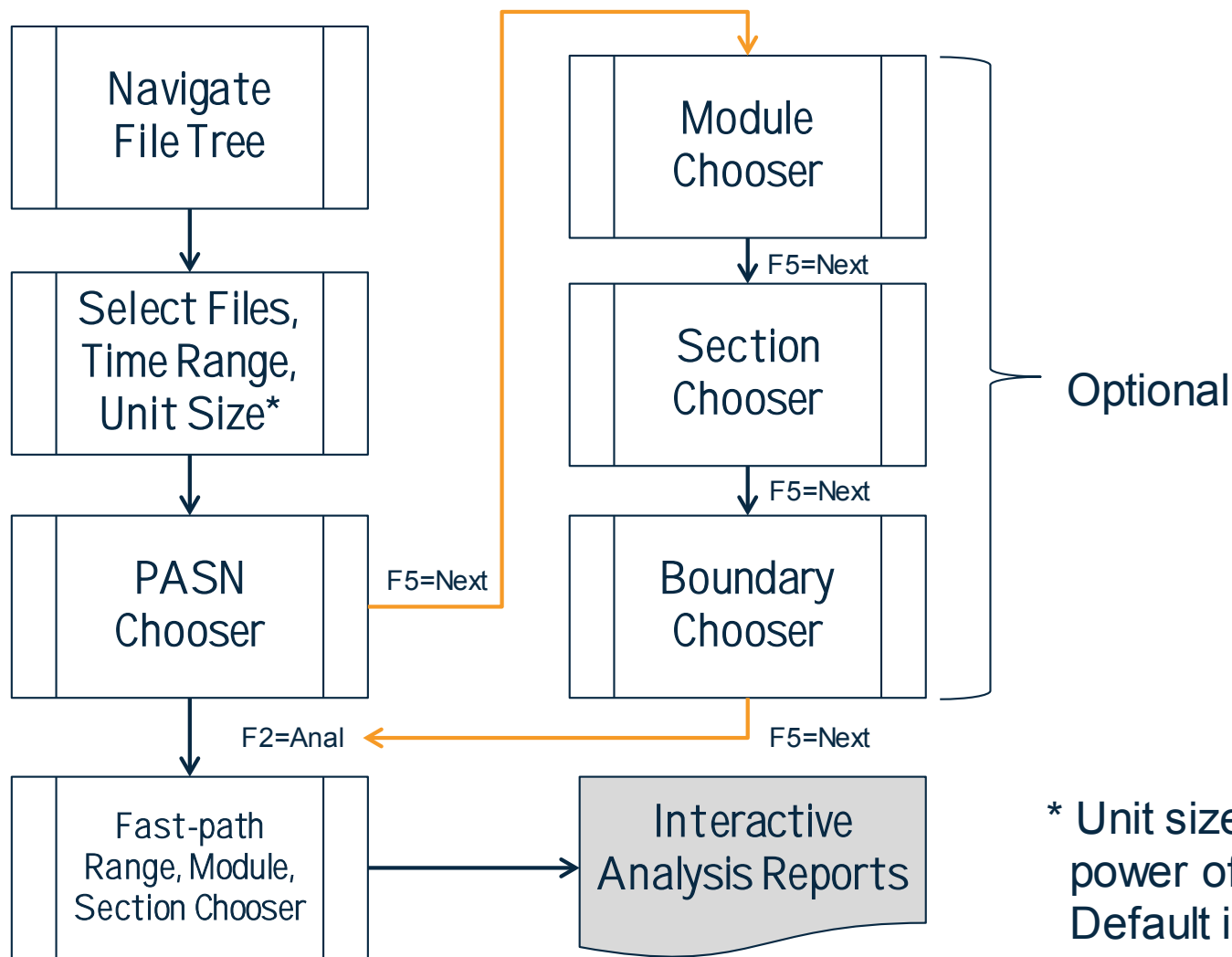

Stopping a zHISR Data Collection Run

```
11/06/2013          zHISR: S y s t e m   L o g   /   S t o p          16:24:38

F ZHISR,STOP 0001,CLIENT=YES
ZHS015I: STOP command has been serviced

0001 of 0001          3=End  7=Bwd  8=Fwd          XXXX FB 2014
```

zHISR Data Collection Analysis Wizard



* Unit size can be any power of two, 8 thru 4K. Default is 64 bytes.

Select File, Choose Analysis Unit Size

```

11/19/2013          zHISR: Collection Dataset Catalog Display          14:41:02
                    A(n)=Analyze B(L|A)=Browse RM=Remove S=Slct Cursor=Slct
Directory disk john1

  Type  Permission MgmtClas Volume  FileName
a File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T1627318.C00000.HISMAP
_ File  zr--r--r--  STANDARD MVSNV0  JOHN1.D131019.T1627318.C00000.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T1627318.C00002.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T1627318.C00003.HISSMP
_ File  zr--r--r--  STANDARD MVSNV0  JOHN1.D131019.T1856090.C00000.HISMAP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T1856090.C00000.HISSMP
_ File  zr--r--r--  STANDARD MVSNV0  JOHN1.D131019.T1856090.C00002.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T1856090.C00003.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2017175.C00000.HISMAP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2017175.C00000.HISSMP
_ File  zr--r--r--  STANDARD MVSNV0  JOHN1.D131019.T2017175.C00002.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2017175.C00003.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2025020.C00000.HISMAP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2025020.C00000.HISSMP
_ File  zr--r--r--  STANDARD MVSNV0  JOHN1.D131019.T2025020.C00002.HISSMP
_ File  zr--r--r--  STANDARD MVSNV1  JOHN1.D131019.T2025020.C00003.HISSMP

0001 of 0003          1=Help 2=Refresh 3=End 4=Return 7=Bwd 8=Fwd          XXXX FB

```

Specify Time Period and Included CPUs

```
11/19/2013          zHISR: H.I.S. Sample Data CPU Selection          14:42:38
CPU Report          JOHN1.D131019.T1627318.C00000          Sort: Cpu#
```

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 YYDD.HH:MM:SS format: 13292.23:27:34 13292.23:28:35

```
JOHN1.D131019.T1627318.C00000.HISSMP      140.0K 13292.23:27:36 13292.23:28:35
JOHN1.D131019.T1627318.C00002.HISSMP      24.0K 13292.23:27:34 13292.23:28:27
JOHN1.D131019.T1627318.C00003.HISSMP      108.0K 13292.23:27:34 13292.23:28:35
```

0001 of 0001

1=Help 3=End 5=Next 7=Bwd 8=Fwd

XXXX FB 2014

Primary ASN Chooser

11/19/2013 zHISR: H.I.S. Sample Data PASN Selection 14:42:53
 ASID Report JOHN1.D131019.T1627318.C00000 Sort: Percent

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 PF5.

35.352%	003A-C3CONDOR	15.336%	0055	10.516%	0001
6.606%	002B	5.906%	0006	5.836%	000B
2.801%	0002	2.218%	0037-ZHISR	2.206%	0007
2.066%	0003	1.891%	0021	1.669%	0039
1.611%	002C	1.190%	0025	0.782%	003C
0.782%	005D	0.689%	0047	0.490%	0009
0.292%	000A	0.198%	0010	0.198%	0016
0.175%	0023-HIS	0.163%	0018	0.163%	001B
0.163%	003D	0.140%	0013	0.105%	002F
0.093%	0008	0.093%	0049	0.070%	0029
0.035%	002A	0.035%	0019	0.035%	000C
0.023%	003B	0.023%	0059	0.012%	0020
0.012%	0026	0.012%	0032	0.012%	0044

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

0001 of 0001 1=Help 2=Anal 3=End 5=Next 7=Bwd 8=Fwd 9=Sort_ASID

XXXX FB 2014

Module Chooser

11/19/2013

zHISR: H.I.S. Sample Data Module Selection

14:45:20

Module Report

JOHN1.D131019.T1627318.C00000

Sort: Owner

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.

003A-FMBCREXX	00000000_0CD67000	000209B0	C3CONDOR	13292.2327323	-
003A-FMBLIOCS	00000000_0CA8E000	00005000	C3CONDOR	13292.2327323	-
003A-FMBLOGIC	00000000_0CADD000	00033000	C3CONDOR	13292.2327323	-
003A-FMBLOGIC	00000000_0C9FB000	0000D000	C3CONDOR	13292.2327323	-
003A-FMBCCI2	00000000_0CF23000	00004000	C3CONDOR	13292.2327323	-
003A-FMBCDA2	00000000_0CF2D000	00010000	C3CONDOR	13292.2327323	-
003A-FMBCENVC	00000000_0CF80000	0000C8C8	C3CONDOR	13292.2327323	-
003A-FMBCLENU	00000000_0CFD8000	00058000	C3CONDOR	13292.2327323	-
003A-FMBCLIC	00000000_0CF14000	000000A0	C3CONDOR	13292.2327323	-
003A-FMBCML	00000000_0CF1D000	00006000	C3CONDOR	13292.2327323	-
003A-FMBCMS	00000000_0CF27000	000			
003A-FMBCOL	00000000_0CF54000	000			
003A-FMBCOPTS	00000000_0CF13000	000			
003A-FMBCRMGR	00000000_0CF12000	000			
003A-FMBCSE	00000000_0CF17000	000			

Provide a selection mask: *
To select all privates, Type *PVT.
3=End 4=Include 5=Exclude

0001 of 0101 1=Help 2=Mask 3=End 5=Next 7=Bwd 8=Fwd 9=Sort_Address

XXXX FB 2014

Control Section Chooser

11/19/2013

zHISR: H.I.S. Sample Data Csect Selection

14:46:07

Csect Report

JOHN1.D131019.T1627318.C00000

Sort: Owner

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.

```

_ 003A-@@KJUMP 00000000_0CD69968 000000A0 13292.2327323 -
  003A-@@TRT 00000000_0CD6BB30 00000100 13292.2327323 -
  003A-@@XTOVFN 00000000_0CD6BF00 00000220 13292.2327323 -
  003A-ACCESS 00000000_0CD5C000 00000473 13292.2327323 -
  003A-ALLOC 00000000_0C755000 0000128A 13292.2327323 -
  003A-ALTER 00000000_0C757000 00000A4A 13292.2327323 -
  003A-ATTRIB 00000000_0C74F000 00001500 13292.2327323 -
  003A-BPXWESTA 00000000_0CD6C1A0 0000008C 13292.2327323 -
  003A-BPXWREXX 00000000_0CD6C230 0001B27E 13292.2327323 -
  003A-BPXWRFM 00000000_0CD87658 000000AE 13292.2327323 -
  003A-BPXWRFMS 00000000_0CD87708 000000BC 13292.2327323 -
  003A-BPXWRGM 00000000_0CD875A0 000000B4 13292.2327323 -
  003A-BPXWRTIM 00000000_0CD877C8 000000FC 13292.2327323 -
  003A-BPXWRT2E 00000000_0CD878C8 000000E4 13292.2327323 -
  003A-BPXWRXST 00000000_0CD874B0 00000044 13292.2327323 -

```

0001 of 1494 1=Help 2=Mask 3=End 5=Next 7=Bwd 8=Fwd 9=Sort_Address

XXXX FB 2014

Virtual Storage Boundary Chooser

11/19/2013

zHISR: H.I.S. Sample Data Bndry Selection

14:46:35

Boundary List

JOHN1.D131019.T1627318.C00000

Sort: Owner

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 pres PF5.

```

_ CSA      00000000_00B00000  00000000_00D25FFF
DONUC     00000000_7FD7C000  00000000_7FD7FFFF
ECSA      00000000_076BA000  00000000_0C6FFFFFFF
EFLPA     00000000_076A9000  00000000_076ABFFF
EMLPA     00000000_076AC000  00000000_076B9FFF
EPLPA     00000000_03C0B000  00000000_076A8FFF
EPRV      00000000_0C700000  00000000_7FFFFFFF
ERON      00000000_01000000  00000000_01B354FF
ERWN      00000000_01B36000  00000000_01B98FFF
ESQA      00000000_01B99000  00000000_03C0AFFF
HCSA      000001EF_80000000  000001FF_FFFFFFFF
PLPA      00000000_00D26000  00000000_00F07FFF
PRIVATE   00000000_00000000  00000000_00AFFFFFFF
RON       00000000_00FE1000  00000000_00FFFFFFF
RWNUC     00000000_00FD5000  00000000_00FE0387
  
```

0001 of 0002 1=Help 2=Mask 3=End 5=Next 7=Bwd 8=Fwd 9=Sort_Address

XXXX FB 2014

Fast-path Range, Module, Section Chooser

11/19/2013
Analysis

zHISR: H.I.S. Data Analysis
JOHN1.D131019.T1627318.C00000

14:48:44
PASN=003A

All available modules, csects 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 003A.

Wait state: N Include Wait dispatched, Y or N

Address range: _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____
 _____ to _____

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

11/19/2013 zHISR: H.I.S. Full Analysis 14:52:11
 Run Summary JOHN1.D131019.T1627318.C00000 Sort: Tot_Pct

PSW_Address	Tot_Pct	Sel_Pct	Csect	Module	Boundary
00000000_01864C40	2.451%	6.933%	IAXFP	-	ERON
00000000_0CF321C0	1.190%	3.367%	FMBCDA2	FMBCDA2	EPRV
00000000_01864C80	1.155%	3.268%	IAXFP	-	ERON
00000000_0CF320C0	1.120%	3.169%	FMBCDA2	FMBCDA2	EPRV
00000000_0D050580	0.969%	2.740%	ERBSMFI	ERBSMFI	EPRV
00000000_0CF7700	0.700%	1.981%	-	-	EPRV
00000000_0D037040	0.654%	1.849%	GAAUX	-	RV
00000000_0D15C000	0.514%	1.453%	IGVCP	-	N
00000000_0D15C000	0.479%	1.354%	FMBCS	-	RV
00000000_0CF31700	0.444%	1.255%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF33000	0.444%	1.255%	FMBCDA2	FMBCDA2	EPRV
00000000_0D052D00	0.397%	1.122%	ERBGASD0	ERBSMFI	EPRV
00000000_0CB280C0	0.385%	1.080%	CAOHX	GAAUX	EPRV
00000000_0CF327C0	0.385%	1.080%	FMBCDA2	FMBCDA2	EPRV
00000000_0D15D200	0.303%	0.850%	FMBCSUBS	FMBCSUBS	EPRV
00000000_7FD7C3C0	0.292%	0.825%	-	-	EPRV
00000000_011BC580	0.268%	0.759%	IEAVETPC	-	ERON

Point and <F6> for Location Pop-Up

Click for Spot Analysis of Section, Module or Boundary.

Press <F9> to rotate through available sorts.

0001 of 0059 1=Help 3=End 6=Loc 7=Bwd 8=Fwd 9=Sel_Pct 10=Mem 12=Prnt XXXX FB

Full Analysis with Location Pop-up

11/19/2013 zHISR: H.I.S. Full Analysis 14:52:11
 Run Summary JOHN1.D131019.T1627318.C00000 Sort: Tot_Pct

PSW_Address	Tot_Pct	Sel_Pct	Csect	Module	Boundary
-------------	---------	---------	-------	--------	----------

00000000_01864C40	2.451%	6.933%	IAXFP	-	ERON
00000000_0CF321C0	1.190%	3.367%	FMBCDA2	FMBCDA2	EPRV

01 of 01 Module: FMBCDA2 3=End 6=Loc 7=Bwd 8=Fwd

Start 00000000_0CF2D000 End 00000000_0CF3CFFF Length 00010000

CPI ratio: MOD FMBCDA2 21.2927 Csect FMBCDA2 21.2927

VolSer MVSNO DSN PHOENIX.TEST.LOADLIB

Press <F6>
repeatedly to step
through Location
Pop-Ups

00000000_0D052D00	0.397%	1.122%	ERBGA	-	EPRV
00000000_0CB280C0	0.385%	1.089%	GAAUX	-	EPRV
00000000_0CF327C0	0.385%	1.089%	FMBCD	-	EPRV
00000000_0D15D200	0.303%	0.858%	FMBCS	-	EPRV
00000000_7FD7C3C0	0.292%	0.825%	-	-	EPRV
00000000_011BC580	0.268%	0.759%	IEAVETRC	-	ERON

0001 of 0059 1=Help 3=End 6=Loc 7=Bwd 8=Fwd 9=Sel_Pct 10=Mem 12=Prnt XXXX FB

Spot Analysis for Control Section

11/19/2013 zHISR: H.I.S. Spot Analysis 14:52:25
 By Csect JOHN1.D131019.T1627318.C00000 Sort: Tot_Pct

Psw_Address	Offset	Tot_Pct	Sel_Pct	Csect	Module	Boundary
00000000_0CF321C0	000051C0	1.190%	3.367%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF320C0	000050C0	1.120%	3.169%	FMBSDA2	FMBSDA2	EPRV
00000000_0CF31700	00004700	0.444%	1.255%	FMBSDA2	FMBSDA2	EPRV
00000000_0CF33000	00006000	0.444%	1.255%	FMBSDA2	FMBSDA2	EPRV
00000000_0CF327C0	000057C0	0.385%	1.089%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF333C0	000063C0	0.187%	0.528%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF32080	00005080	0.163%	0.462%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF31200	00004200	0.128%	0.363%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF32780	00005780	0.117%	0.330%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF30D80	00003D80	0.105%	0.297%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF32E80	00005E80	0.105%	0.297%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF32D80	00005D80	0.093%	0.264%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF31400	00004400	0.082%	0.231%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF36240	00009240	0.082%	0.231%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF36600	00009600	0.082%	0.231%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF31900	00004900	0.058%	0.165%	FMBCDA2	FMBCDA2	EPRV
00000000_0CF328C0	000058C0	0.058%	0.165%	FMBCDA2	FMBCDA2	EPRV

Click here to show source code via ADATA

0001 of 0005 1=Help 3=End 6=Loc 7=Bwd 8=Fwd 9=Sel_Pct 10=Mem 12=Prnt XXXX FB

ADATA Location Prompt

```

11/19/2013          zHISR: Associated Data Prompt          14:53:03
                    JOHN1.D131019.T1627318.C00000

Define the SYSADATA and SYSDEBUG dataset name(s) using the catalog mask and
the PF5 and PF6 key functions. Correct the target member name if necessary
and if using SYSDEBUG, provide the program name. After all required fields
have been provided press the PF2 key to view the associated data.

Catalog Search mask ** .ADATA
Target member name  FMBCDA2

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

Report page limit   1000

Instruction address  00000000_0CF30A80

Instruction offset   00003A80

1=Help  2=Adata  3=End  5=Dataset(s)  6=Catalog_Search  11=Clear

```

ADATA Library Concatenation Prompt

11/19/2013

zHISR: A s s o c i a t e d D a t a

14:53:46

FMBCRKE .ADATA
FMBCRKE .EJES51 .ADATA
FMBCRKE .EJES52 .ADATA
FMBCRK2 .ADATA
FMBCRK3 .ADATA
FMBCADM .ADATA
FMBC1 .ADATA
FMBC2 .ADATA
FMBC .PRODGEN .ADATA
FMBC .PROD360 .ADATA
FMBC .PROD410 .ADATA
FMBC .PROD420 .ADATA
FMBC .PROD430 .ADATA
FMBC .PROD440 .ADATA
FMBC .PROD450 .ADATA
FMBC .PROD460 .ADATA
FMBC .PROD470 .ADATA
FMBC .PROD480 .ADATA
FMBC .PROD510 .ADATA
FMBC .PROD520 .ADATA

FMBC .PROD510 .ADATA

— FMBC .PRODGEN .ADATA —
— FMBCADM .ADATA —
— FMBCRK2 .ADATA —
— FMBCRKE .ADATA —

3=End 4=Before 5=After 6=Replace

0001 of 0002 1=Help 3=End 4=Return 7=Bwd 8=Fwd

XXXX FB ARE
burgh 2014

Scrollable ADATA with Highlighted Code from Execution Analysis Unit

11/19/2013

zHISR: A s s o c i a t e d D a t a
FMBC.PRODGEN.ADATA

14:54:30

More➔

003A7C	41E0	E0C0		000C0	29599+	LA	R14,ESTKESIZ(,R14)	Adv	
003A80	E3E0	A010	0024	00000	00000	29600+	STG	R14,ESTKCUR	Sav
003A86	D207	E000	A028	00000	00018	29601+	MVC	0(8,R14),ESTKWORK	Sav
003A8C	EBFD	E008	0024	00000	00008	29607+	STMG	15,13,8(R14)	Mul
003A92	E3E0	A028	0004	00000	00018	29610+	LG	R14,ESTKWORK	Res
003A98	B904	0041		00000		29612	LGR	R4,R1	Loa
003A9C	D703	4137	4137	00137	00137	29613	XC	DASECPUP,DASECPUP	Set
003AA2	D703	413B	413B	0013B	0013B	29614	XC	DASEACPP,DASEACPP	(sa
003AA8	D703	413F	413F	0013F	0013F	29615	XC	DASEZAPP,DASEZAPP	(sa
003AAE	D703	4147	4147	00147	00147	29616	XC	DASEGCPP,DASEGCPP	(sa
003AB4	D703	414B	414B	0014B	0014B	29617	XC	DASEUCPP,DASEUCPP	(sa
003ABA	D703	414F	414F	0014F	0014F	29618	XC	DASEUACP,DASEUACP	(sa
003AC0	D703	4153	4153	00153	00153	29619	XC	DASEPAGR,DASEPAGR	(sa
003AC6	D703	4157	4157	00157	00157	29620	XC	DASESIOR,DASESIOR	(sa
003ACC	E370	A0C0	0017	00000	000C0	29621	LLGT	R7,EMRACTRD	Poi
003AD2	B902	0077		00000		29622	LTGR	R7,R7	Doe
003AD6	A784	0024		03B1E		29623	JZ	DARATE1A	Bra
003ADA	E310	4027	0090	00000	00027	29624	LLGC	R1,DASESYID	Loa
003AE0	A71B	FFFF		FFFFF		29625	AGHI	R1,-1	Mak

0186 of 1000 1=Help 3=End 4=Return 7=Bwd 8=Fwd 10=Lft 11=Rht 12=Print XXXX FB

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.

Print, Save or Export Results

11/19/2013

zHISR: Print/Save/Export - Spot Analysis

14:54:51

JOHN1.D131019.T1627318.C00000

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 _____

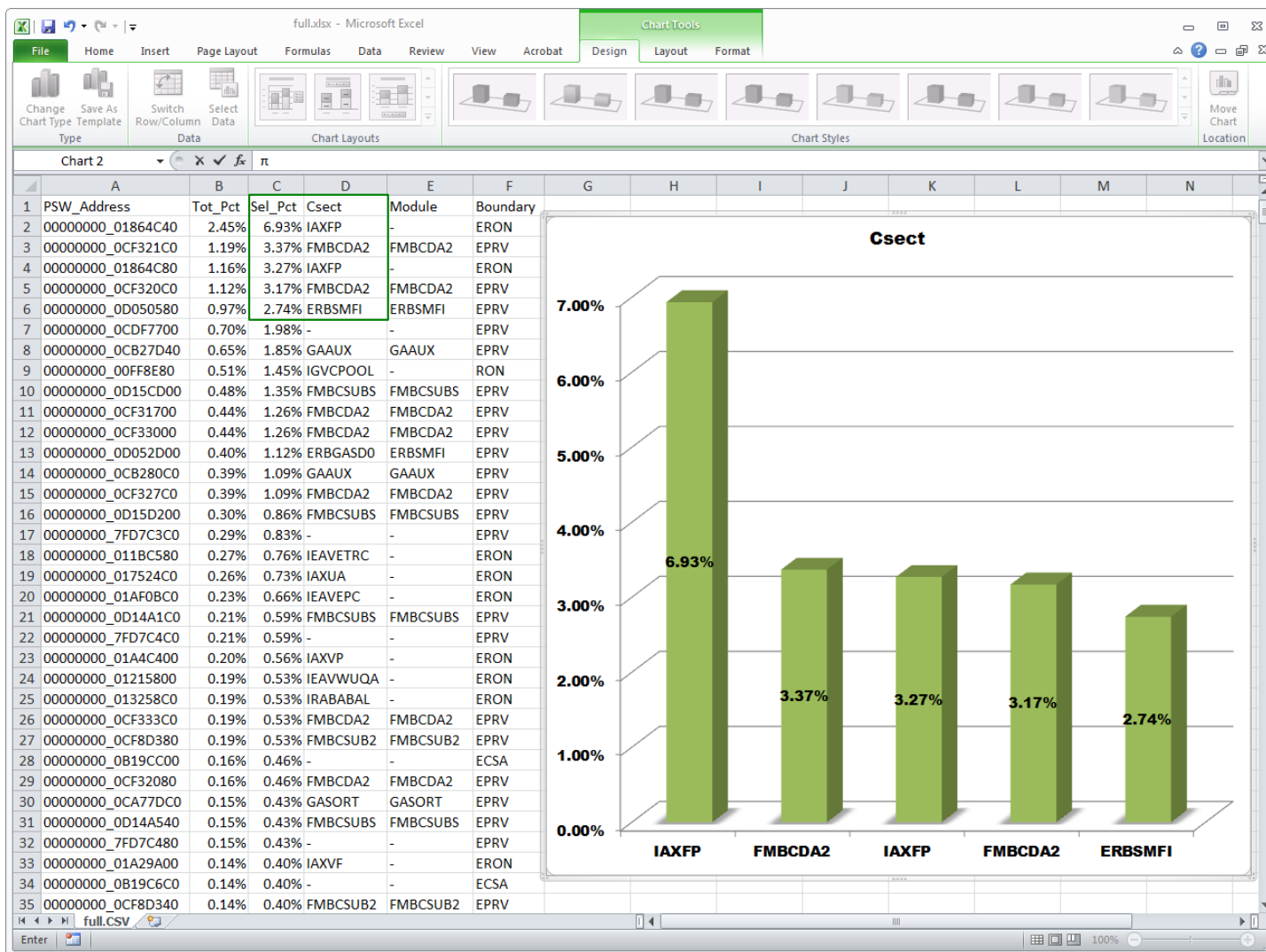
Output Member Name _____

Output Path Name _____

Character encoding ASCII

1=Help 3=End 4=Print 5=Save 7=Export 11=Clear

Import CSV File into Your Spreadsheet



What's New?

- 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.

Questions?

