

Running foreign hardware architectures on IBM System z/VM - a dream becoming reality

z86VM – a technology overview *A New Evolution in Smart Computing for an Enterprise* **Session 844**

Jim Porell

jim.porell@mantissa.com

Cell: 914-474-1864

06/12/2014

z86VM from Mantissa

http://www.mantissa.com/?page_id=214

What is it?

- x86 infrastructure on IBM Mainframe
- Leverages IBM hypervisor, z/VM and System z mainframe hardware for virtualization layer
- 32 bit x86 implementation
- Beta 2 supports SME Linux (a Centos distro)
 - Available today
- Desktop and Server consolidation
- “cut and paste” applications to z

What value?

- Ultimate green platform
- Reduced desktop mgt labor
- Mainframe QOS for x86...Security, BR, Capacity, BPI, Storage
- x86 on Demand – turn on engines to add capacity
- z/VM competitively compares to VMWare, KVM, Xen, Hyper-V
- Different values based on
 - 10’s of x86
 - 100’s of x86
 - 1000’s of x86

IP and Component overview

x86 Guest				
Binary Translator		VMM/TC Scheduler		I/O Scheduler
Memory	Time	Dispatch X86 Hypervisor	Interrupt	I/O
z/VM				

- Eight components make up this virtualization effort
- Only the YELLOW boxes deal with foreign architecture
- Gray boxes are analogous to Unix System Services for z/VM

All boxes could be updated to improve performance

There is an associated testing infrastructure that complements this.

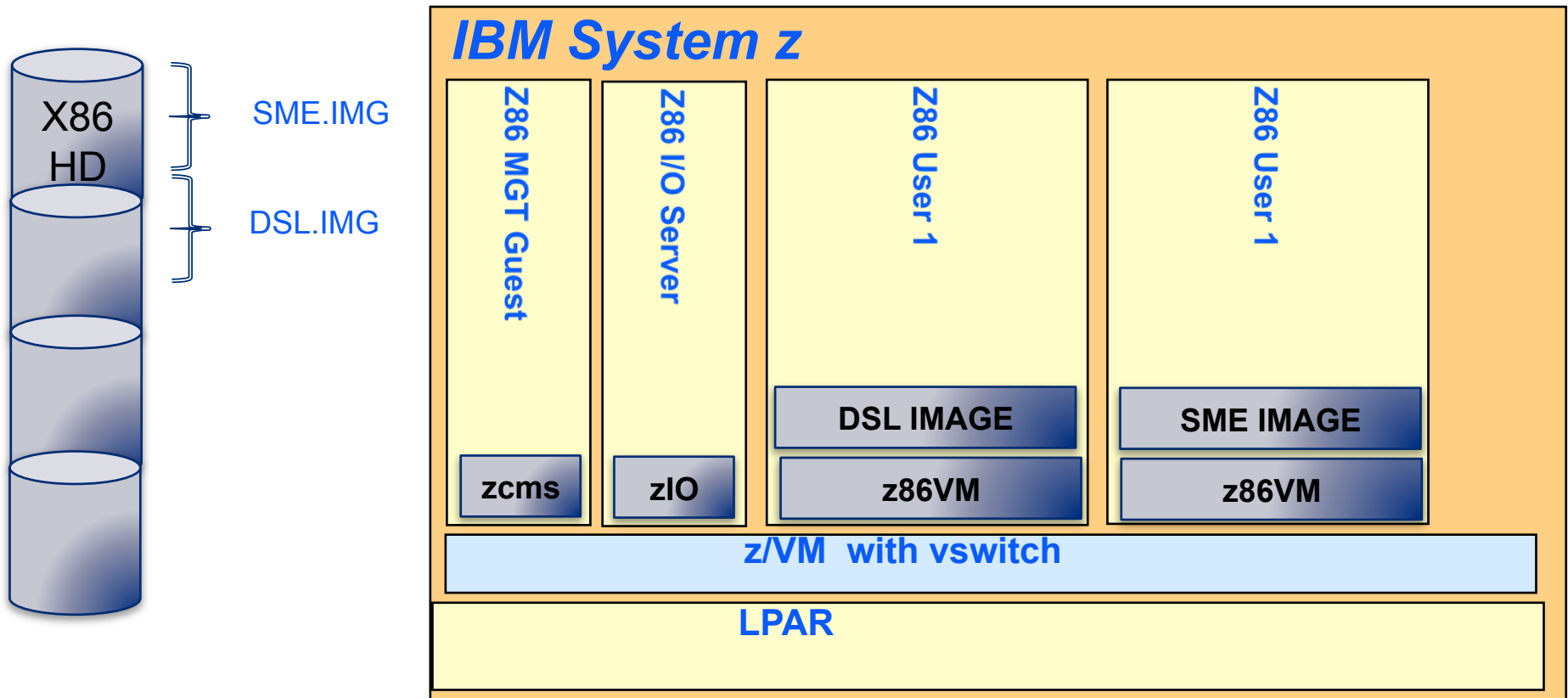
Performance History

- This has always been about price/performance
- Major function was achieved over a year ago.
- Performance has improved monthly and will continue

Date	Test Environment	Results	Notes
12/10/2010	System z10, z/VM 5.3	2,400 CPU seconds	First measurement
06/28/2012	AMD 1GHz 512MB	7 CPU seconds	Base case
	QEMU virtualization	16 seconds	Virtualization base
	System z10, z/VM 5.3	163 CPU seconds	4.4 Ghz processor
10/12/2012	System z196	56 CPU seconds	5.2 Ghz processor
11/29/2012	System z196	49 CPU seconds	
03/28/2013	System z114	39 CPU seconds	
05/01/2013	System z114	30 CPU Seconds	normalized to z196

z86VM Internals

- Z86 Management Guest – defines virtual PC hardware containers and I/O
 - Goal is to replace this with IBM Wave, if economical
- Z86 I/O server is a “daemon” that handles all user file and network I/Os
- Z86 User represents a Virtual x86 Hardware container
 - z86VM provide the x86 instruction set
 - IMAGE (DSL or SME and later Windows) is the Operating System and middleware image
 - Running on that OS image is a workload or application that solves a business problem



PROFILE Z86 – Tactical to be replaced by IBM Wave?

Define the IO Environment



Define the virtual PC

- Memory
- Boot device
- Boot image



```
07 zio_start = Z86DIO          /* Set the I
08 vswitch = VSW86           /* Name of
09 vnicadr = 0100           /* Virtual
10 boot = 0195 000001 00010 MANT01
11 module = ZI05 MODULE F
12 origin = 10000
13 /* Define the volumes for the I/O file system
14 /* Approximately 16 demodsl images per 3390-9
15 /*      |      |Start|#      |
16 /*      |Vadr |Cyl  |Cyls |Volser
17 x86hd = 0200 00001 10016 ZV0048
18 x86hd = 0201 00001 10016 ZV0051
19 zio_end = Z86DIO
```

```
z86_start = Z86D001
io_server= Z86DIO
boot = 0194 000011 00010 MANT01
module = V0R3C2M MODULE A
origin = 10000
bootdev = hd0
boottype = hd
vncport = 8700
pcimage = DSL.IMG
vmimage = DSL2G IMG A
memory = 256M
hdmemory= 2G
z86_end = Z86D001
```

```
z86_start = Z86D002
io_server= Z86DIO
boot = 0194 000021 00010 MANT01
module = V0R3C2M MODULE A
origin = 10000
bootdev = hd0
boottype = hd
vncport = 8701
pcimage = SME.img
vmimage = SME8110 DEMOWP A
memory = 900M
hdmemory= 7G
z86_end = Z86D002
```

Modern Programming with Assembler

```
      USING ZIOU,R8                      ADDRESSABILITY
SWITCH IPTYPE,LEN=1    TEST IPTYPE
  CASE IPTYPPC        PENDING CONNECTION
    FGO ACCEPT       ISSUE ACCEPT
    BREAK
  CASE IPTYPC        CONNECTION COMPLETE
**   WTO 'IUCV CONNECTION COMPLETE'
    IF (TM,ZIOUMSGSTAT,ZIOUPEND,ONES),AND, IF PENDING CONN X
      (CLC,ZIOUMSGPATH,IPPATHID,EQ)  IF *MSG SERVICE
        MVI ZIOUMSGSTAT,ZIOUCONN    SET CONNECTED
    ENDIF
    BREAK
**   CASE IPTYPRP        PRIORITY INCOMING REPLY
    WTO 'IUCV PRIORITY INCOMING REPLY'
    BREAK
  CASE IPTYPRNP        INCOMING REPLY
    LLGF R1,IPMSGTAG    LOAD ECB ADDRESS
    POST ECB=(R1),JUMP=YES,SVC=NO  POST ECB
    BREAK
  CASE IPTYPSV        SEVERED CONNECTION
    FGO SEVER
    BREAK
  CASE IPTYPMNP        INCOMING MSG
    FGO INCOMING       RETRIEVE INCOMING MESSAGE
    BREAK
  CASE OTHER
    MVC MSG99H,=H'80'
    MVC MSG99,=CL80'XX <--- UNKNOWN IPTYPE FOR IUCV'
    SR R1,R1                      IUC00140
    IC R1,IPTYPE                  IUC00150
    PBTOD MSG99,2                 IUC00160
    WTO MSG99H
ENDSW
```

Mantissa has developed a set of IBM assembler macros that:

- Make the code more structured and easier to update and analyze
- Enable high performance programming through assembler

```
      USING IUCVL,R2                      ADDRESSABILITY
DO WHILE,(LTGR,R2,R2,NZ)    UNTIL ALL DONE
  IF (CLC,IUCVLPATH,IPPATHID,EQ)    IF SAME PATH
    MVC SAVEUSER,IUCVLUSER    COPY USERID FOR DISC MSG
    MVC SAVEFLAG2,IUCVLFLAG2  SAVE FLAG
    IF (TM,IUCVLFLAG1,IUCVLUSED,ONES) IF IUCVL IN USE
      OI IUCVLFLAG1,IUCVLTERM  FLAG TASK FOR TERMINATION
      MVI ACTFLAG,HEXFF    SET PATH ACTIVE FLAG
    ELSE
      XC IUCVLFLAG1,IUCVLFLAG1  CLEAR FLAG
    ENDIF
  ENDIF
  LG R2,IUCVLPTR                NEXT IUCVL
ENDDO
```

What's next for z86VM?

- Customer usage goals
 - 10's, 100's, 1000's
 - z/OS Management console host - 10's
 - App Dev desktop host – 100's
 - File, Print, Web Server – 100's
 - Desktops – 1000's
- Still lots of work to do before it is production ready
 - Code clean up
 - Benchmarks: Primitives, SMP
 - Pricing
 - More Customer use cases

What could schools do with this?

- Help build exploitation examples
 - Install console products and exploit them
 - DB2 Control Center, Tivoli Enterprise Portal, CA, ASG and BMC offerings
 - Demonstrate that any device can be a remote presentation device, but is now more secure and manageable.
 - Build file servers, security servers, print servers
 - Once Windows is running, run RDz inside the mainframe in a Cloud with direct access to zVM for unit test
- Let students have remote access to systems without having customized software on their desktops/smart devices
 - Go beyond the green screen 3270 into the world of graphics and simplicity of management

What could z86VM become? Our dreams

x86 Guest				
Binary Translator		VMM/TC Scheduler		I/O Scheduler
Memory	Time	Dispatch X86 Hypervisor	Interrupt	I/O
z/VM				

- Eight components make up this virtualization effort
- Only the YELLOW boxes deal with foreign architecture
- Gray boxes are analogous to Unix System Services for z/VM
- Only the GREEN boxes need to be updated for ARM functionality

ARM Guest				
Binary Translator		VMM/TC Scheduler		I/O Scheduler
Memory	Time	Dispatch ARM Hypervisor	Interrupt	I/O
z/VM				

All boxes could be updated to improve performance

Could then enable Android, Linux, Windows RT, and iOS applications to run on z.

No commitment here, **just our dreams.**

For More Information please contact...

Len Santalucia, CTO & Business Development Manager

Vicom Infinity, Inc.

One Penn Plaza – Suite 2010

New York, NY 10119

212-799-9375 office

917-856-4493 mobile

lsantalucia@vicominfinity.com

About Vicom Infinity

Account Presence Since Late 1990's

IBM Premier Business Partner

Reseller of IBM Hardware, Software, and Maintenance

Vendor Source for the Last 8 Generations of Mainframes/IBM Storage

Professional and IT Architectural Services

Vicom Family of Companies Also Offer Leasing & Financing, Computer Services, and IT Staffing & IT Project Management