z86VM – a technology overview

A New Evolution in Smart Computing for an Enterprise

Jim Porell

Jim.porell@mantissa.com

04/29/2014

z86VM from Mantissa

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

What is it?

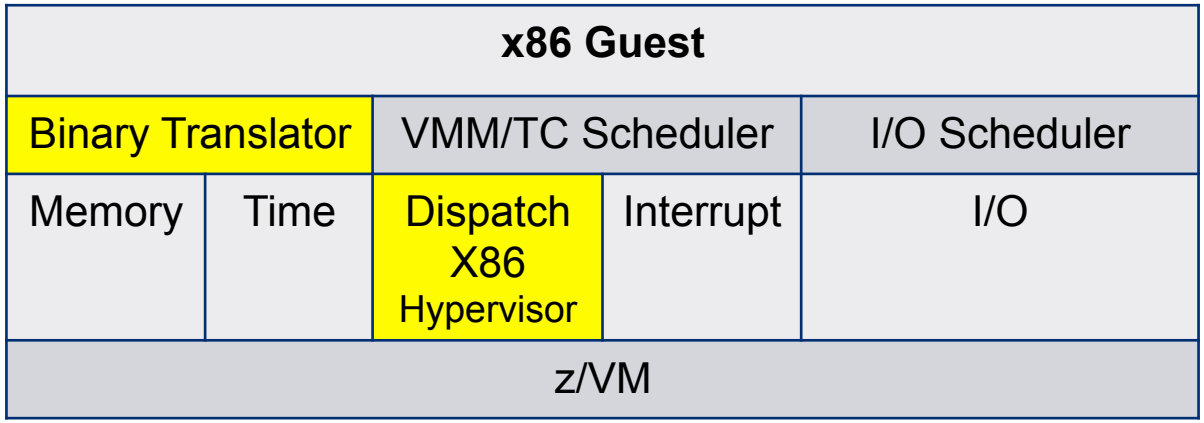
- x86 infrastructure on IBM Mainframe
- Leverages IBM hypervisor, z/VM and System z mainframe hardware for virtualization layer
- 32 bit x86 implementation
- Beta 2 supports SME Linux (a Centos distro)
  - Available today
- Desktop and Server consolidation
- "cut and paste" applications to z

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

© 2014 Mantissa Corp

IP and Component overview



- 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.

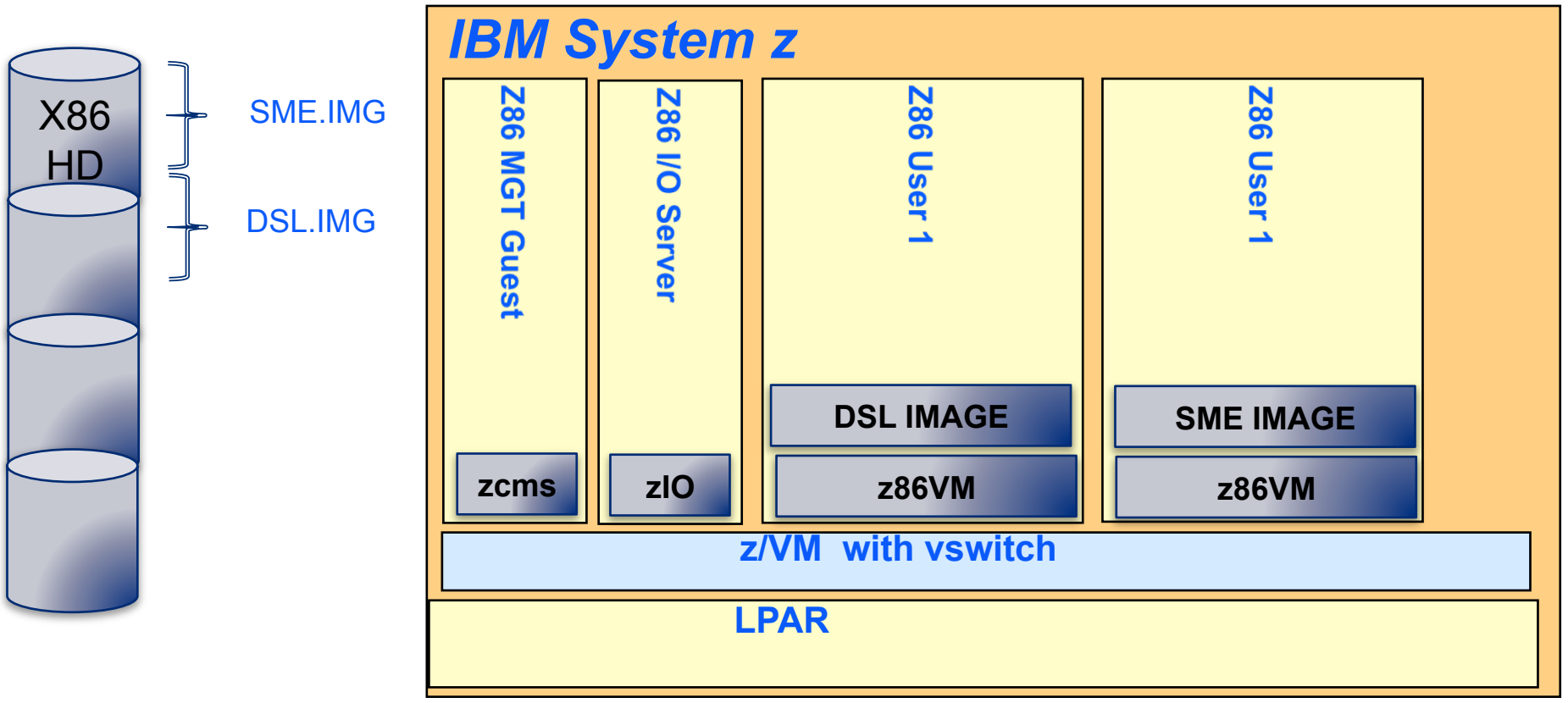
Performance History

- 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

```
zio_start = Z86DIO                    /* Set the I
 vswitch = VSW86                       /* Name of
 vnicadr = 0100                        /* Virtual
 boot = 0195 000001 00010 MANT01
 module = ZIO5 MODULE F
 origin = 10000
 /* Define the volumes for the I/O file system
 /* Approximately 16 demodsl images per 3390-9
 /*      |      |Start |#     |
 /*      |Vadr |Cyl    |Cyls |Volser
 x86hd = 0200 00001  10016 ZV0048
 x86hd = 0201 00001  10016 ZV0051
zio_end = Z86DIO
```

Define the virtual PC

- Memory
- Boot device
- Boot image

```
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 IPTYPCC         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:

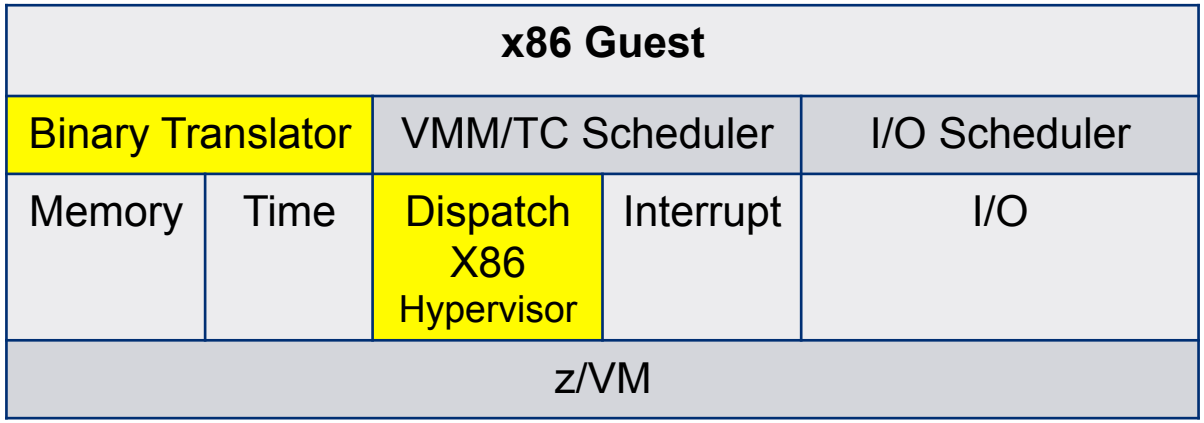
- Make the code more structured and easier to update and analyze
- Enable high performance programming through assembler

```
      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?

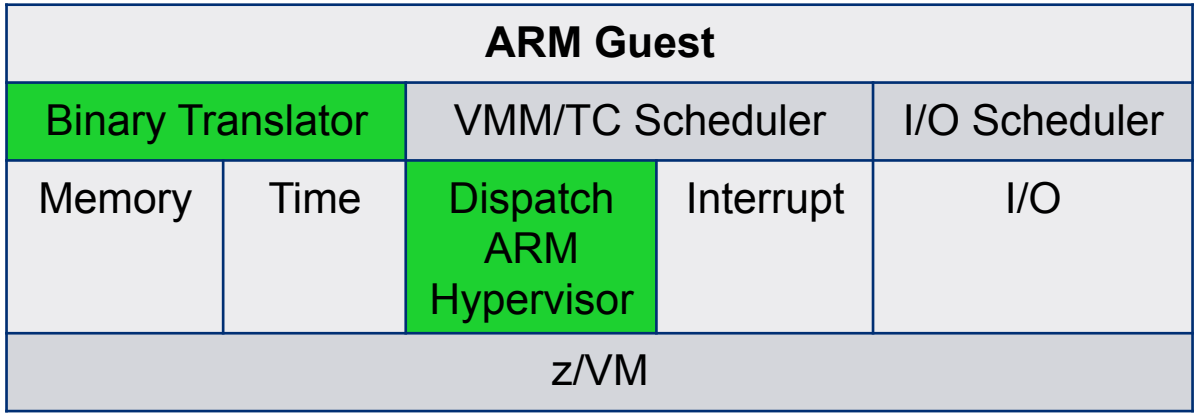
- Customer usage goals
  - 10's, 100's, 1000's
    - z/OS Management console host - 10's
    - App Dev desktop host – 10's
    - File, Print, Web Server – 100's
    - Desktops – 1000's
- Still lots of work to do before it is production ready
  - Code clean up
  - Benchmarks: Primitives, SMP
  - Pricing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

What could z86VM become? Our dreams



- 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



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.**