

Why should I upgrade to the latest compilers?

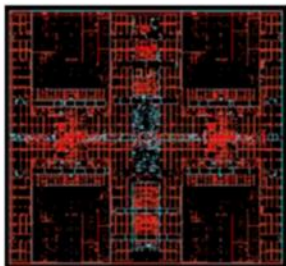


Why IBM Compilers ?

- Reduce CPU utilization by improving performance of applications
 - Exploitation of z/Architecture
 - Advanced Optimization
- Support next-generation Applications
 - Modernize business critical applications to support new applications (e.g. web and mobile applications)
 - Work with latest IBM Middleware (CICS, IMS, DB2...)
- Improve Productivity
 - New programming features
 - New problem determination features

z Systems - Processor Roadmap

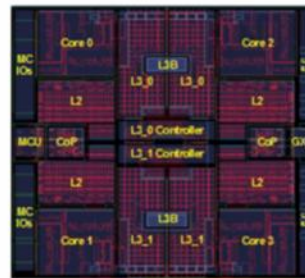
z10
2/2008



Workload Consolidation and Integration Engine for CPU Intensive Workloads

- Decimal FP
- Infiniband
- 64-CP Image
- Large Pages
- Shared Memory

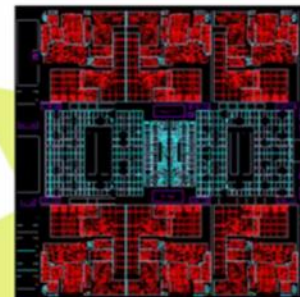
z196
9/2010



Top Tier Single Thread Performance, System Capacity

- Accelerator Integration
- Out of Order Execution
- Water Cooling
- PCIe I/O Fabric
- RAIM
- Enhanced Energy Management

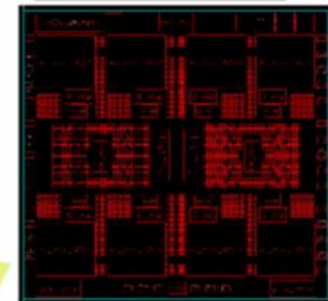
zEC12
8/2012



Leadership Single Thread, Enhanced Throughput

- Improved out-of-order
- Transactional Memory
- Dynamic Optimization
- 2 GB page support
- Step Function in System Capacity

z13
1/2015

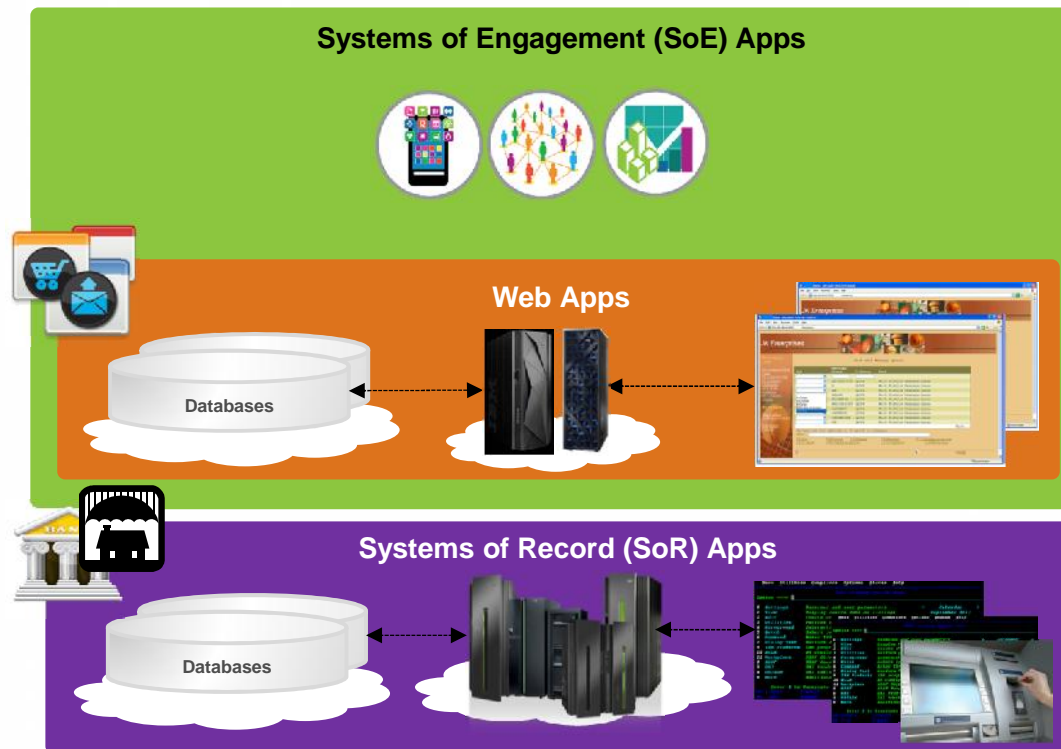


Leadership System Capacity and Performance

- Modularity & Scalability
- Dynamic SMT
- Supports two instruction threads
- SIMD
- PCIe attached accelerators (XML)
- Business Analytics Optimized

Developing next-generation Applications

Extend Business Critical (SoR) Applications to support new System of Engagement (SoE) Applications; Increase performance and increase efficiency and quality of software delivery with new compiler technologies



New System z Compilers

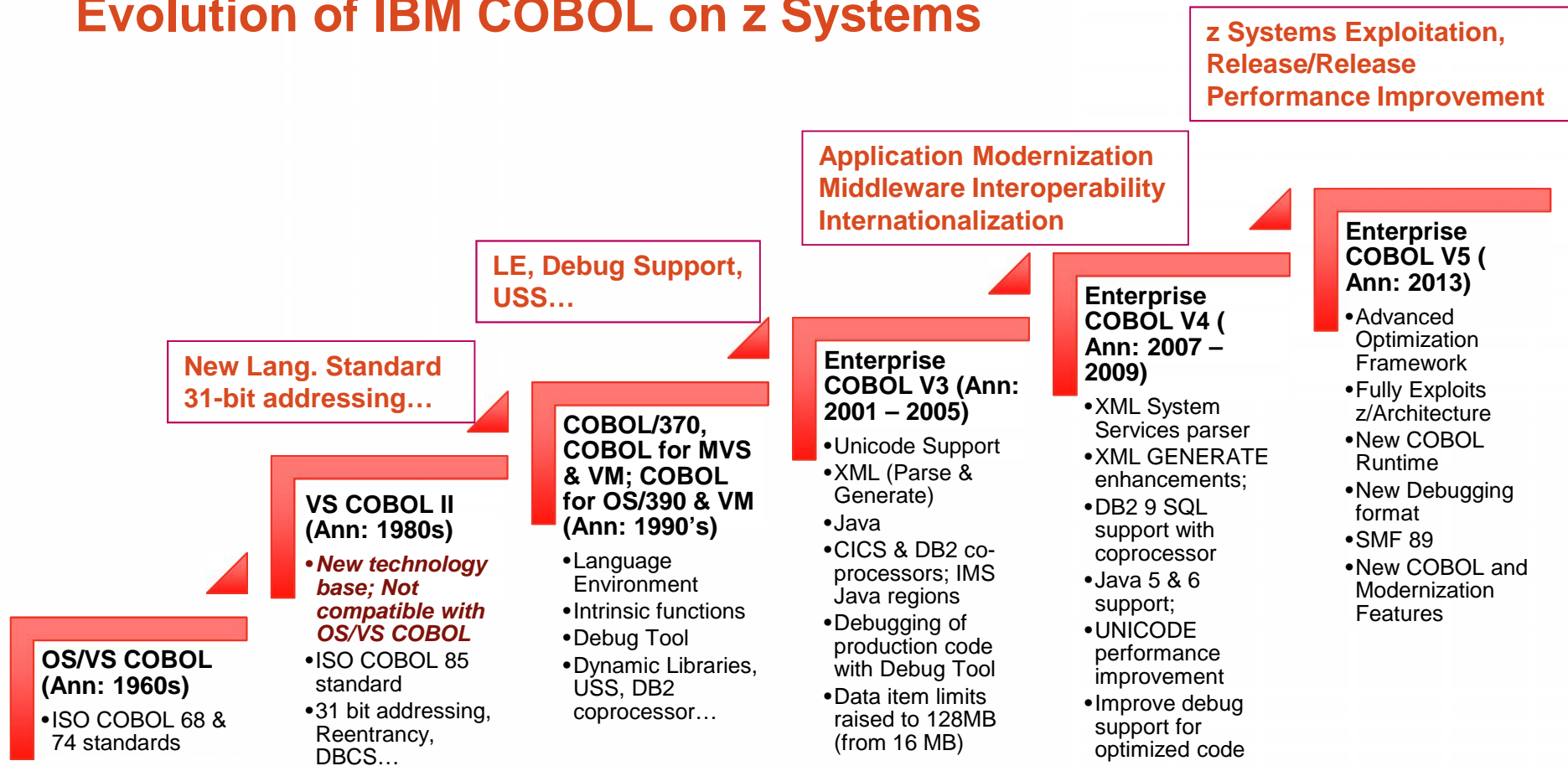
- Enterprise COBOL for z/OS v5.2
- XL C/C++ Compiler for Linux on z Systems v1.1
- Enterprise PL/I for z/OS v4.5
- z/OS XL C/C++ V2R1M1



New System z Compilers

- Enterprise COBOL for z/OS v5.2
- Enterprise PL/I for z/OS v4.5
- XL C/C++ Compiler for Linux on z Systems
- z/OS XL C/C++ V2R1M1

Evolution of IBM COBOL on z Systems



Enterprise COBOL for z/OS V5

- New optimization framework
- New COBOL Runtime
- New Object and Debugging formats
- V5.1 GA'd on Jun 21, 2013
 - Delivered greater than 10% performance improvement over Enterprise COBOL v4 for well structured, CPU-intensive batch applications¹
 - Lays solid groundwork for delivering release-to-release performance improvement roadmap for COBOL on z Systems

Migrating to COBOL V5 requires advanced planning and more work than migrating for COBOL V3 to V4

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Enterprise COBOL for z/OS V5

- Maintains source and binary compatibility
 - Correct COBOL programs will compile and execute without changes and produce the same results
 - “Old” and “new” code can be mixed within an application
 - Removed some old language extensions and options
- Supports the ecosystem of programming tools supplied by IBM and ISVs.



¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Things you should know when upgrading to V5

- H/W and S/W Pre-requisites
- PDSE load libraries
- Dataset and Memory Requirements & SMF
- Old “OS/VS COBOL” and “VS COBOL II NORES” Code



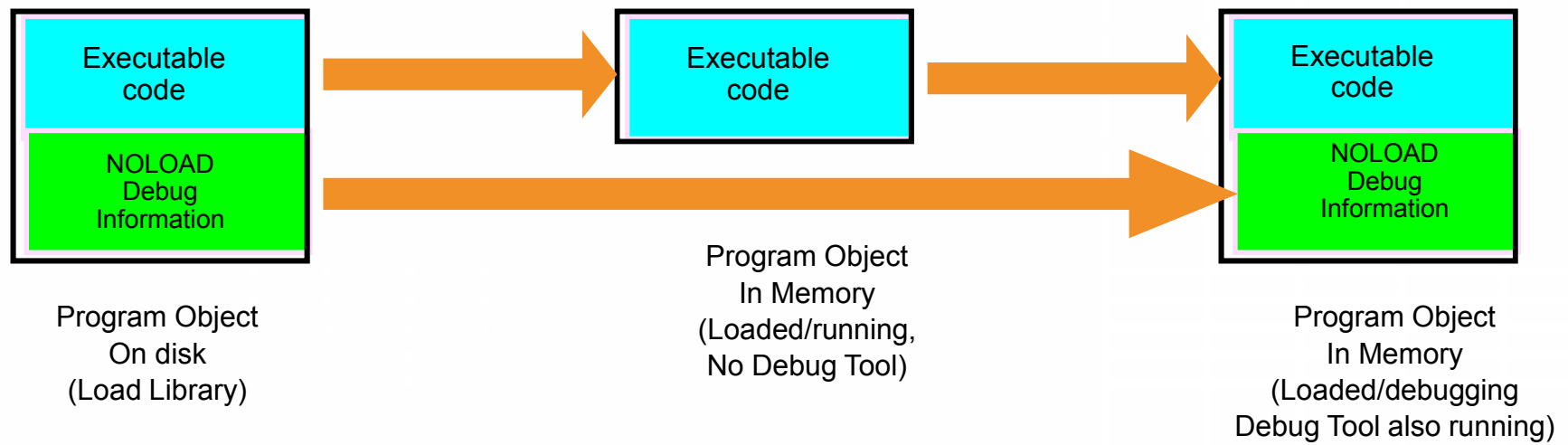
*developerWorks Article: [Prepare to upgrade to Enterprise COBOL for z/OS V5](#)
[Enterprise COBOL for z/OS V5.1.1 Migration Guide](#)*

Pre-requisites

- Hardware
 - Minimum hardware requirement (V5.1: z990/z890; V5.2: z9).
 - Programs compiled with COBOL v5.1 will not run on machines that pre-date minimum hardware requirement
- Software
 - z/OS V1R13 or V2
 - CICS Transaction Server V3 or later
 - IBM DB2 V9 or later
 - IBM IMS V11 or later
 - PD tools V12 or later (Debug Tool, Fault Analyzer Application Performance Analyzer (V13))
 - Rational Developer for System z V9
- PTFs
 - Use **SMP/E FIXCAT** to identify all required PTFs
 - e.g. **SMP/E MISSINGFIX** command w/keyword **Enterprise-COBOL.V5R2**

PDSE

- Enterprise COBOL V5 executable are Program Objects
 - Must reside in PDSE datasets
 - Must move load modules in PDS datasets to PDSE datasets before running COBOL V5 programs
- Requires some advanced planning if using PDS load libraries
 - Moving load modules from PDS to PSDE is fairly straight forward



Dataset and Memory Requirements & SMF

- Dataset and Memory
 - COBOL V5 requires more datasets and memory at compile time
 - Performs more aggressive optimizations to increase the performance of COBOL programs.
 - In addition to the work datasets SYSUT1-SYSUT7, Enterprise COBOL v5.1 requires SYSUT8-SYSUT15 and SYSMDECK
 - Recommended region size: 200MB
- SMF
 - Automatically generates SMF89 records
 - Used by the Sub-capacity Reporting Tool (SCRT) V21.2.0.
 - No need to manually track usage of the COBOL compiler.

Old COBOL Code

- LE
 - Complete migration to LE
- OS/VS COBOL
 - Cannot mix OS/VS COBOL programs with programs compiled with Enterprise COBOL V5.1
 - Must migrate to Enterprise COBOL
- VS COBOL II NORES
 - Recompile VS COBOL II NORES programs with Enterprise COBOL to work with V5.1



Enterprise COBOL for z/OS V5.1 – Since GA

Continuously Delivered Improvements based on customers' feedback via PTF stream

- COBOL V3/V4 features to improve migration
 - AMODE 24, XMLPARSE(COMPAT), VLR(COMPAT), MAP(HEX), ZONEDATA(MIG)
- New features
 - Support for IMS V13 EXEC SQLIMS
- Performance Improvements
 - Execution Performance: Working Storage, Procedure Pointer calls Unstring...
 - Compile time Performance

Complete Fix list for Enterprise COBOL for z/OS <http://www-01.ibm.com/support/docview.wss?uid=swg27041164>

Enterprise COBOL for z/OS V5.2

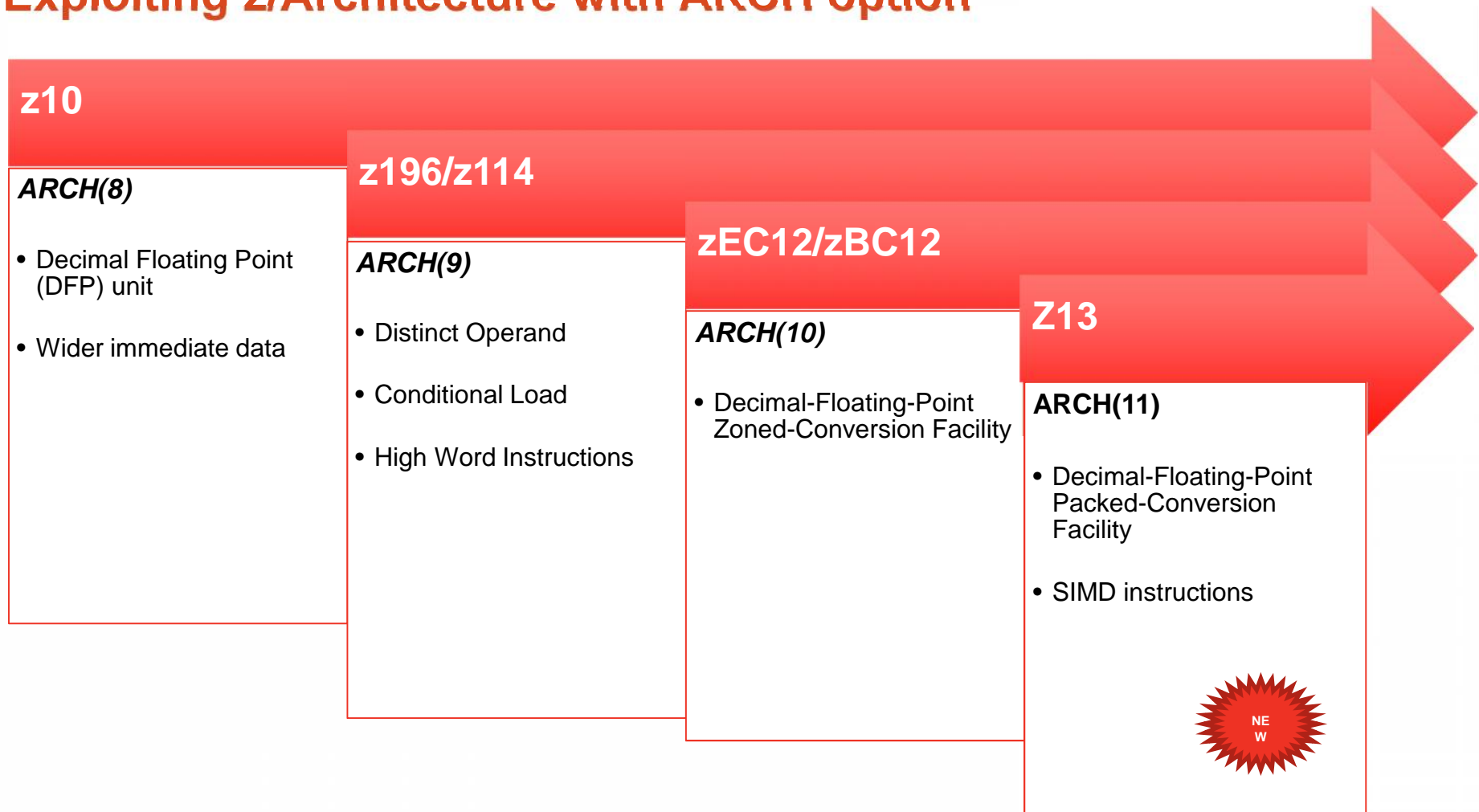
Ann: Jan 14, 2015; GA Feb 27, 2015

2nd release of the new COBOL roadmap

- Provides easy migration from COBOL V5.1
- Includes all enhancements delivered in COBOL V5.1 PTFs
 - Restored migration features, performance features, new programming features
- Provides Day 1 support for z13 processor
 - Instruction scheduler tuned to new micro architecture at ARCH 11
 - Expanded use of Decimal Floating Point for PACKED-DECIMAL data
 - Uses new SIMD instructions for INSPECT TALLYING or REPLACING statements
- Improves Application Performance
 - Well-structured, compute intensive batch applications running on z13 (compiled with Enterprise COBOL V5.2) have shown CPU time reduction of up to 14% over the same applications running on zEC12 (compiled with the GA release of Enterprise COBOL V5.1) ¹

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Exploiting z/Architecture with ARCH option



ARCH 6 is no longer supported in COBOL V5.2

Default ARCH level is now ARCH 7 - IBM System z9EC (2094-xxx models) IBM System z9 BC (2096-xxx models)

Exploiting z13 – Example

PACKED-DECIMAL (COMP-3) faster with Decimal Floating Point

WORKING-STORAGE SECTION.

01 VARS.

02 A PIC S9(25) COMP-3 VALUE +1234567890123456789012345.

02 B PIC S9(25) COMP-3 VALUE +2468097531246809753124680.

02 C PIC S9(25) COMP-3 VALUE 0.

PROCEDURE DIVISION.

PERFORM 100000000 TIMES

DIVIDE A BY B GIVING C

END-PERFORM

V5.2 – ARCH11

```

CXPT    FP0:FP2,152(13,R8),0x8
CXPT    FP1:FP3,165(13,R8),0x8
DXTR    FP4:FP6,FP0:FP2,FP1:FP3
FIXTR   FP0:FP2,9,FP4:FP6
CPXT    FP0:FP2,178(13,R8),0x9
AHI     R2,0xffff
CIJ     R2,L0034,0,HT(mask=0x2),
  
```

V5.1 – ARCH10

```

XGR     R0,R0
ICMH    R0,X'1',152(,R8)      # A
L       R0,153(,R8)          # A
LG      R1,157(,R8)          # A
CXSTR   FP0,R0
XGR     R0,R0
ICMH    R0,X'1',165(,R8)      # B
L       R0,166(,R8)          # B
LG      R1,170(,R8)          # B
CXSTR   FP1,R0
DXTR    FP4:FP6,FP0:FP2,FP1:FP3
FIXTR   FP0:FP2,9,FP4:FP6
CSXTR   R0:R1,0,FP0:FP2
STCMH   R0,X'1',178(,R8)      # C
ST      R0,179(,R8)          # C
STG     R1,183(,R8)          # C
ZAP     178(13,R8),178(13,R8) # C,
  
```

Performance Comparison

Timing – (100 Million times in a loop)

V5.1 : 2.53 cpu seconds

V5.2 : 1.64 cpu seconds (**35% faster**)

Exploiting z13 – Example

INSPECT ... TALLYING faster with SIMD

WORKING-STORAGE SECTION.

01 VARS.

02 STR PIC X(255).

02 C PIC 9(5) COMP-5 VALUE 0.

PROCEDURE DIVISION.

MOVE ALL 'abc def ghi jkl ' TO STR

PERFORM 100000000 TIMES

INSPECT STR TALLYING C FOR ALL ''

END-PERFORM

STOP RUN.

V5.1 – ARCH10

```

                LHI      R0,0xff
                XR       R1,R1
                LA       R12,152(,R8)      #
STR
L0064: EQU        *
                CLI     0(,R12),X'40'    #
                JNOP    L0066
                LA       R1,1(,R1)       #
L0066: EQU        *
                LA       R12,1(,R12)     #
                BRCT    R0,L0064
                A        R1,407(,R8)     # C
                ST       R1,407(,R8)     # C
    
```

V5.2 – ARCH11

```

LHI      R0,0xfe
XR       R1,R1
LA       R12,152(,R8)      # STR
VREPIB  VRF27,0x40
VGBM    VRF25,0x0
L0066:  EQU        *
VLL     VRF24,R0,0(,R12)  #
AHI     R12,0x10
VCEQB   VRF24,VRF24,VRF27
AHI     R0,0xffff0
VLCB    VRF24,VRF24
VAB     VRF25,VRF25,VRF24
JNL     L0066
VGBM    VRF26,0x0
VSUMB   VRF25,VRF25,VRF26
VSUMQF  VRF25,VRF25,VRF26
VLGVG   R1,VRF25,1(,R1)   #
A       R1,407(,R8)       # C
ST      R1,407(,R8)       # C
    
```

Performance Comparison

Timing – (100 Million times in a loop)

V5.1 : 46.63 cpu seconds

V5.2 : 1.54 cpu seconds (**30X faster !**)

Advanced Optimizations

- Provide multiple levels of optimization
- Debugging of optimized code is supported with OPT + TEST options

OPT(0)

- Minimum Optimization

OPT(1)

- Increased Optimization
 - e.g. Inline PERFORM statement
 - Commoning sub-expressions in a block
 - Sequential constant store simplification...

OPT(2)

- Maximum Optimization
 - e.g. Eliminating a stored value that is never re-used anywhere in the program
 - Global view of register assignment...
 - Instruction scheduling to exploit micro-architecture...

Longer compile time
Reduced debugging
Faster Executing Code

Advanced Optimization – Example

Instruction Scheduling for Performance

```

1 z7v2a pic s9(7)v9(2).
1 z7v2b pic s9(7)v9(2).
1 z7v2c pic s9(7)v9(2).
...
ADD 1 TO z7v2a z7v2b z7v2c

```

COBOL V4 – each PACK/AP/ZAP instruction group for each receiver is generated in order.

V4 – OPTIMIZE

- Instructions appear in original order and subject to hardware read after write penalties

PACK	344(5,13),0(9,2)
AP	344(5,13),51(2,10)
ZAP	344(5,13),344(5,13)
UNPK	0(9,2),344(5,13)
PACK	344(5,13),16(9,2)
AP	344(5,13),51(2,10)
ZAP	344(5,13),344(5,13)
UNPK	16(9,2),344(5,13)
PACK	344(5,13),32(9,2)
AP	344(5,13),51(2,10)
ZAP	344(5,13),344(5,13)
UNPK	32(9,2),344(5,13)

COBOL V5 - at OPT(2) low level instruction scheduling is performed to reduce data dependencies, avoid hardware penalties and to best take advantage of the micro-architecture.

V5 – OPT(2)

- Independent operations are grouped to reduce read after write hardware penalties
- ARCH(8)**

PACK	352(5,R13),152(9,R8)
PACK	344(5,R13),168(9,R8)
PACK	336(5,R13),184(9,R8)
AP	352(5,R13),416(2,R3)
AP	344(5,R13),416(2,R3)
AP	336(5,R13),416(2,R3)
ZAP	352(5,R13),352(5,R13)
ZAP	344(5,R13),344(5,R13)
ZAP	336(5,R13),336(5,R13)
UNPK	152(9,R8),352(5,R13)
UNPK	168(9,R8),344(5,R13)
UNPK	184(9,R8),336(5,R13)

Performance Comparison

Timing – (100 million in a loop)

V5 : 2.35 cpu seconds
V4 : 2.50 cpu seconds **(6% faster)**

Enterprise COBOL for z/OS V5.2 - New features

- Access to z/OS JSON services * from COBOL.
 - Provide mobile (i.e. system of engagement) applications easy access to data and the processing they need from business critical enterprise (i.e. system of record) applications written in COBOL
- Improved XML GENERATE
 - More powerful SUPPRESS capabilities
- RULES option to help programmers write better code
- COPYRIGHT and SERVICE options to better manage applications
- New VOLATILE attribute and SERVICE LABEL functionality
 - Enable full optimization of User-Written condition handlers
- Accessing VSAM data sets with extended addressability attribute is now supported
- Some New COBOL 2002 features

* z/OS Client Web Enablement Toolkit

New features from COBOL 2002 Standard

- Access to z/OS JSON services from COBOL.
 - Format 2 of SORT – the table SORT statement
 - Arranges table elements in a user-specified sequence
 - New formats of EXIT statements
 - EXIT METHOD
 - EXIT PARAGRAPH
 - EXIT PERFORM
 - EXIT PERFORM CYCLE
 - Improved COPY REPLACING statement
 - LEADING and TRAILING phrases (better partial word replacement)
 - Improved REPLACE statement
 - LEADING and TRAILING phrases (better partial word replacement)

Developer Trial



- Zero cost evaluation license for 90 days
 - Does not initiate Single Version Charging (SVC)
- Assess the value that could be gained from upgrading to Enterprise COBOL V5.2
- Offer same functionalities as Enterprise COBOL for z/OS V5.2
 - Code compiled with Enterprise COBOL Developer Trial cannot be used for production
- Available as standard offering through ShopzSeries on March 27, 2015
 - Contact your IBM representative for ordering assistance

<http://www-03.ibm.com/software/products/ph/en/enterprise-cobol-developer-trial-for-zos>

New System z Compilers

- Enterprise COBOL for z/OS v5.2
- Enterprise PL/I for z/OS v4.5
- XL C/C++ Compiler for Linux on z Systems
- z/OS XL C/C++ V2R1M1

Enterprise PL/I

- Strategic Programming Language
 - Significant use in business applications but also in some scientific and engineering application
- Advanced optimization technology
 - Shares optimizing back-end technology with z/OS XL C/C++
 - Timely delivery of leading edge optimization and hardware exploitation to PL/I customers
- Time proven
 - First Enterprise PL/I product released in 2001 (Enterprise PL/I for z/OS and OS/390 v3.1)
 - Latest release of Enterprise PL/I for z/OS (v4.5) is based on same architecture
 - Provides easy migration
- Shipped new release every year since 1999
 - Addressed customer requirements
 - Improved optimization technology, z/Architecture exploitation, usability, middleware support, and application modernization features.



Enterprise PL/I for z/OS v4.5

Ann: Jan 14, 2015; GA: Feb 27,2015

- Advanced optimization and full z13 exploitation
 - New optimizations to improve performance for packed decimal
 - Use of SIMD instructions to improve code for SEARCH and VERIFY
 - Inlining of MOD and REM of large packed decimal and multiply
 - Much better code for SELECT of CHAR(2) and CHAR(4)
- Provides significant performance improvements over Enterprise PL/I V4.4
 - Compute intensive applications running on z13 (compiled with Enterprise PL/I V4.5) have shown CPU time reduction of up to 17% over the same applications running on zEC12 (compiled with Enterprise PL/I V4.4). Performance Improvement
- Addressed 28 RFEs

Enterprise PL/I for z/OS v4.5

Ann: Jan 14, 2015; GA: Feb 27,2015

- For example, this simple code tests if a UTF-16 string is numeric

```
wnum: proc( s );
```

```
    dcl s    wchar(*) var;  
    dcl n    wchar value( '0123456789' );  
    dcl sx   fixed bin(31);
```

```
    sx = verify( s, n );  
    if sx > 0 then ...
```

- With ARCH <= 10, it is implemented via an expensive library call
- With ARCH(11), it is inlined with vector instructions and runs 2X faster

Enterprise PL/I for z/OS v4.5

Ann: Jan 14, 2015; GA: Feb 27,2015

E700	E000	0006		VL	v0,+CONSTANT_AREA(,r14,0)
E740	E010	0006		VL	v4,+CONSTANT_AREA(,r14,16)
			@1L2	DS	0H
A74E	0010			CHI	r4,H'16'
4150	0010			LA	r5,16
B9F2	4054			LOCRL	r5,r4
B9FA	F0E2			ALRK	r14,r2,r15
E725	E000	0037		VLL	v2,r5,_shadow1(r14,0)
E722	0180	408A		VSTRC	v2,v2,v0,v4,b'0001',b'1000'
E7E2	0001	2021		VLGV	r14,v2,1,2
EC5E	000D	2076		CRJH	r5,r14,@1L3
A74A	FFF0			AHI	r4,H'-16'
A7FA	0010			AHI	r15,H'16'
EC4C	000E	007E		CIJNH	r4,H'0',@1L4
A7F4	FFE5			J	@1L2
			@1L3	DS	0H

Enterprise PL/I for z/OS v4.5 New Features

- Ships a JSON parser
 - Support Parse, Generate, and Validate with native PL/I language
 - Also works with z/OS JSON services*
 - Allows enterprise (i.e. system of record) applications written in PL/I to be extended to handle data access and processing requests from mobile (i.e. system of engagement) applications
- Improved middleware support
 - Faster code for CICS calls
 - Support for named constants as SQL host variables
 - Allow structures as SQL indicator variables
- New productivity features
 - Extend size of strings from 32K to 128M
 - New INLIST and BETWEEN built-in functions
 - New built-in functions to generate the store-clock hardware instructions
 - New REINIT statement
 - Added features to help enforce code quality

For more information, visit <http://www-03.ibm.com/software/products/en/plizos>

* z/OS Client Web Enablement Toolkit

New System z Compilers

- Enterprise COBOL for z/OS v5.2
- Enterprise PL/I for z/OS v4.5
- XL C/C++ Compiler for Linux on z Systems
- z/OS XL C/C++ V2R1M1

Linux for z Systems

- Growing rapidly
 - Installed IFL MIPS increased 12% from 4Q'13 to 4Q'14.
 - As of 4Q14...
 - 27.3% of Total Installed MIPS run Linux
 - 39% of z Systems Customers have IFL's
- Only choice for C/C++ developers is GNU C/C++
- IBM middleware, Business Intelligence & Analytic, and ISV workloads are driving demand to increase performance



XL C/C++ for Linux on z Systems

Ann: Jan 14, 2015; GA: Feb 16, 2015

- Based on new technologies
 - Clang front end
 - C/C++ language support (Partial C11, C++11)
 - GNU C/C++ language and option compatibility
 - Allows easy migration from distributed Linux systems to Linux for z Systems
 - IBM optimization technology shipped in Enterprise COBOL and IBM Java
- Ships with high performance Math Libraries tuned for z Systems
 - MASS (Mathematical Acceleration Subsystem software) and ATLAS (Automatically Tuned Linear Algebra Software)
 - Provide elemental and basic linear algebra functions to simplify coding and improve application performance
- Runs on RHEL (6 & 7); and SLES (11 & 12)

XL C/C++ for Linux on z Systems

- Provides significant performance advantage over GNU C/C++
 - Advanced optimization and z/Architecture exploitation
 - CPU intensive applications compiled with XL C/C++ for Linux on z Systems V1.1 have shown up to 10% performance improvement over the same applications compiled with GNU C/C++ V4.4 ¹
 - 7% performance improvement over the same applications compiled with GNU C/C++ V4.7¹.
 - World class service and support by IBM

For more information, visit <http://www.ibm.com/software/products/en/czlinux>

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

New System z Compilers

- Enterprise COBOL for z/OS v5.2
- Enterprise PL/I for z/OS v4.5
- XL C/C++ Compiler for Linux on z Systems
- z/OS XL C/C++ V2R1M1

IBM z/OS XL C/C++

- Optionally priced feature of z/OS
 - Enables development of high performing business applications, system programs and low level C applications
- IBM has been delivering leading edge C/C++ compilers on z/OS for over 20 years
 - Every release sets new standard for performance
 - Includes advanced optimization technology originally designed for HPC applications, and innovations to improve programmer productivity
 - Improves support for C and C++ language standards
- Provides system programming capabilities with Metal C option
 - Allows developers to use C syntax to develop system programs and low level free standing applications on z/OS without coding in HLASM
 - Significantly shortens the learning curve
 - Leverage advanced optimization technology to generate high performance optimized code



z/OS XL C/C++ V2R1M1

Ann: Jan 14, 2015; GA: Feb 16, 2015

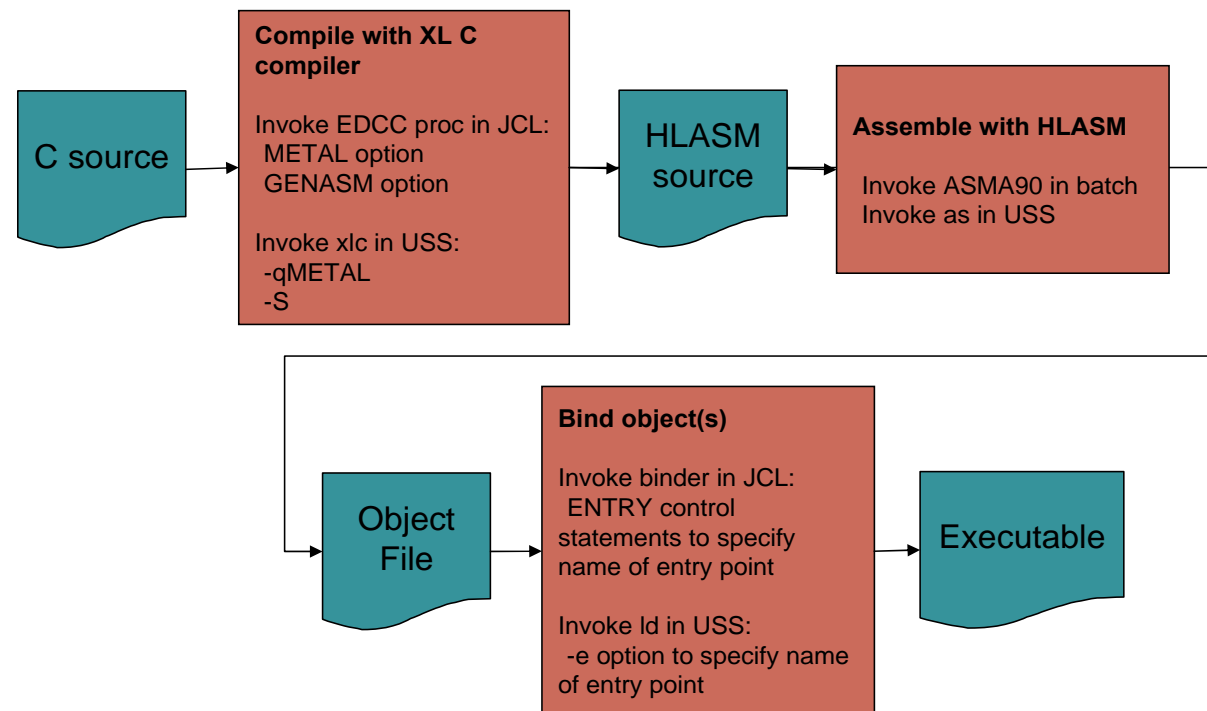
- Web deliverable for z/OS 2.1
 - Provide Day 1 support for z13
 - Target clients running z/OS V2.1 on new z13 Systems
 - Replaces z/OS XL C/C++ V2.1 after installation
- Support for Vector/SIMD
 - Compile option, datatype, and built-in functions
- Ships with High performance Math Libraries tuned for z13
 - MASS (Mathematical Acceleration Subsystem) library providing scalar, vector, and SIMD mathematical elemental functions
 - ATLAS (Automatically Tuned Linear Algebra Software) library providing basic linear algebra functions

z/OS XL C/C++ V2R1M1

- New Programming Features
 - New support for inline assembler (GNU compatibility)
 - New Debug support for Vector/SIMD data type
 - Header file cache optimization
- Provides significant performance improvements over z/OS XL C/C++ V2R1.
 - CPU intensive applications running on z13 (compiled with z/OS XL C/C++ V2R1M1) have shown up to 17% throughput improvement over the same applications running on zEC12 (compiled with z/OS XL C/C++ V2R1)

For more information, visit <http://www-03.ibm.com/software/products/en/czos>

Metal C Option

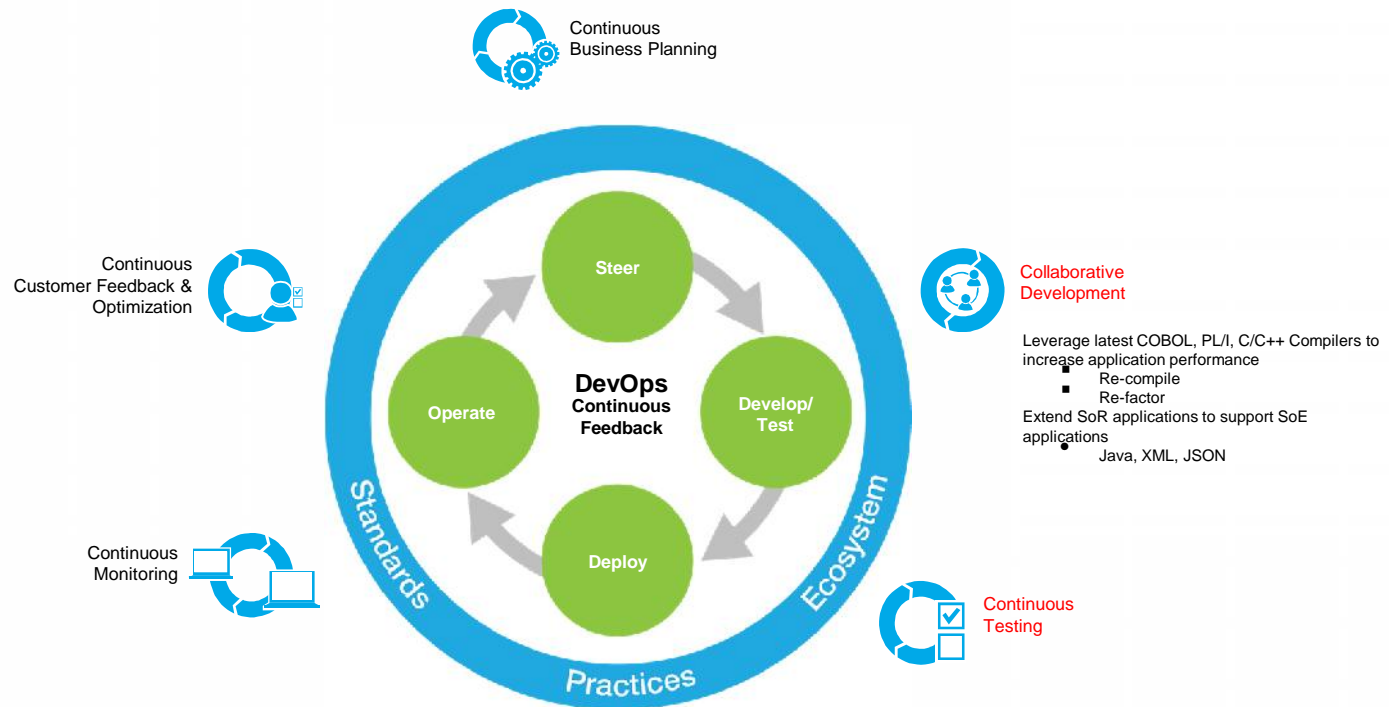


- Generates optimized ASM source code that is Language Environment independent
- Interoperates with existing HLASM programs
 - Uses MVS™ linkage conventions
- Supports AMODE 64 (allow mixing of AMODE 31 & AMODE 64 code)
- Provides a subset of C library functions.

Benefits of Metal C

- Provides high level language alternative to writing the program in assembly language.
- Allows developers to use C syntax to develop system programs
 - Significantly shortens the learning curve
 - No need to manage use of registers and developing the correct assembly instruction sequences
- Allows developers to develop low level free standing applications on z/OS
 - Supports embedding of HLASM source within C statements
 - Provide direct access to z/OS System services
- Leverage advanced optimization technology to generate high performance optimized code
- Develop the application once
 - Recompile to optimize for new z/Architecture

The DevOps Picture



Collaborative Development

Leverage latest COBOL, PL/I, C/C++ Compilers to increase application performance

- Re-compile
- Re-factor

Extend SoR applications to support SoE applications

- Java, XML, JSON



QUESTIONS

The word "QUESTIONS" is written in large, white, 3D-style letters. Each letter is filled with a different photograph of a diverse group of people, including men and women of various ethnicities, smiling or looking thoughtful. The letters have a slight shadow beneath them, giving them a floating appearance.



www.ibm.com/software/rational

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.