IBM Software Group

Leverage Compilation Technology to Optimize Investment on System z

Al Grega
Rational WW Sales Leader – Enterprise Modernization And Compilers
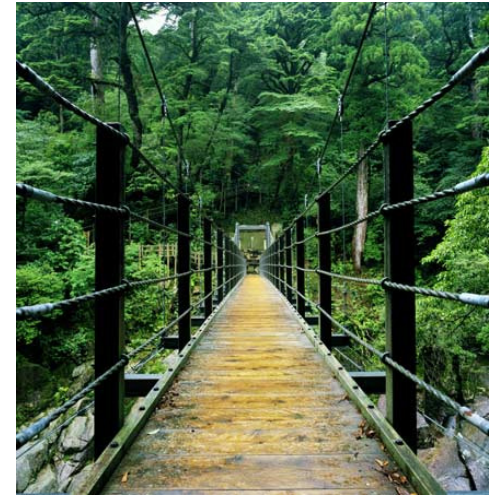
Roland Koo
Compiler Product Line Manager

What are Compilers?

"Compilers are a bridge between user applications and systems that run their business"

- Converts human readable source code to machine executable instructions
- Improve programmer's productivity
- Optimize application performance

Why IBM Compilers?

- **Designed to unleash full power of IBM processors**
  - System z, POWER, Blue Gene, Cell Broadband Engine, and future architectures

- **Industry leadership in advanced optimization technology**

- **Strong investment commitment for Compiler development**
  - COBOL, PL/I, C/C++, and Fortran

- **Highly skilled development and research teams**

IBM System z Compilers

- **Enterprise COBOL for z/OS, Enterprise PL/I for z/OS, and z/OS XL C/C++**

- **Target Commercial Application Development and deployment on z/OS**
  - Support Batch Processing and legacy file systems (ISAM, VSAM …)
  - Provide major programming interface for middleware subsystems
    - CICS, IMS, DB2

- **COBOL and PL/I run the world's most business critical applications**
  - COBOL processes 75% of the world's business data and 90% of the world's financial transactions (with CICS).
    - Estimate 200 Billion lines of Code
  - PL/I is used for scientific, engineering and commercial applications
    - Estimate ~5% of all mainframe applications are written in PL/I

- **C/C++ usage increasing as z/OS development language**
  - Dominant implementation language of portable applications and middleware
  - Provide System Programming capability with Metal C

History

- **40+ years in delivering leading-edge COBOL compiler**
- **30+ years in delivering leading-edge PL/I compiler**
- **20+ years in delivering leading-edge C/C++ compiler**

- **Grew with IBM hardware and middleware**
  - System/360 to System z, CICS, DB2, IMS
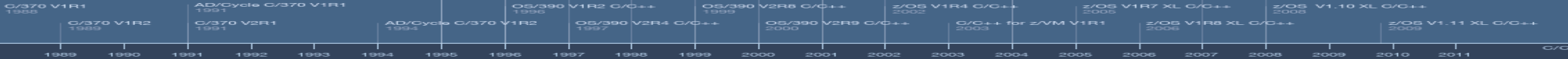- **Run the world's most critical business applications**

Modernizing Legacy Application
Leverage modern technology in legacy applications
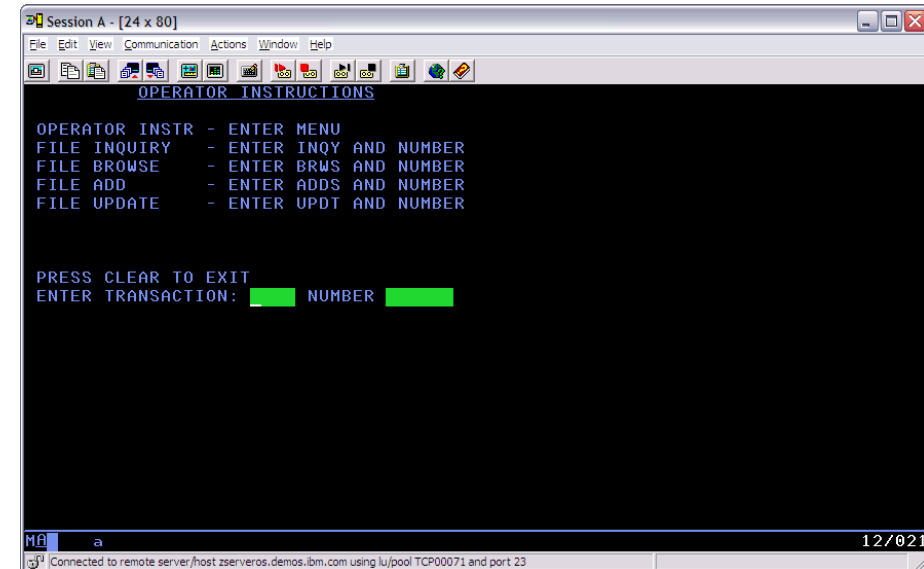
- **Legacy applications crucial to businesses**
  - Considerable resources have already been spent in development and enhancements

- **Desire to leverage modern technologies *without* requirement to re-write or re-host**
  - *Reduces risk, cost and time to market*
  - *Translates to bottom-line success*

- **COBOL and PL/I compilers include features to *extend* legacy applications**
  - Integrate with modern technologies (Web Services, Java, XML)
  - Support SOA to enable flexible reuse of core application assets across the enterprise

Modernizing Legacy Application
An Example:

CIO: *"We would like to enable our existing applications which are business critical, to work in our new web based infrastructure. Re-writing them from scratch takes too long and is too costly and risky"*

3270 terminal

(Green screen)

- CICS BMS UI code
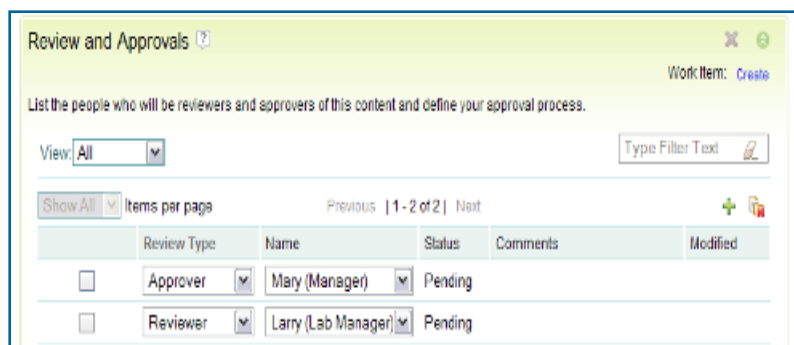- Business Logic
- z/OS
- COBOL/PLI
- CICS, IMS, DB2

Current Solution

- Field offices connected to main office via mainframe network

- "green screen" terminal user interface

Modernizing Legacy Application
The Solution

What is needed: Change to modern web-based user interface, consolidation of servers, address global marketplace.

Review and Approvals ⓘ
Work Item: Create
List the people who will be reviewers and approvers of this content and define your approval process.
View: All
Type Filter Text
Show All | Items per page   Previous | 1 - 2 of 2 | Next

	Review Type	Name	Status	Comments	Modified
☐	Approver	Mary (Manager)	Pending		
☐	Reviewer	Larry (Lab Manager)	Pending		

Windows Web client consisting of user interface code written in Java

Web server
- Business Logic
- WebSphere
- COBOL/PLI and/or Java

Convert XML to COBOL & PL/I data structures, EBCDIC output

Pass data to server using XML documents created by Java with different code pages

Data server
- Business Logic
- COBOL & PL/I code
- CICS, IMS, DB2

Process traditional data structures in EBCDIC

System z server

Enterprise COBOL and PL/I for z/OS
- Allows reuse of legacy COBOL code with a modern Java and XML based interface
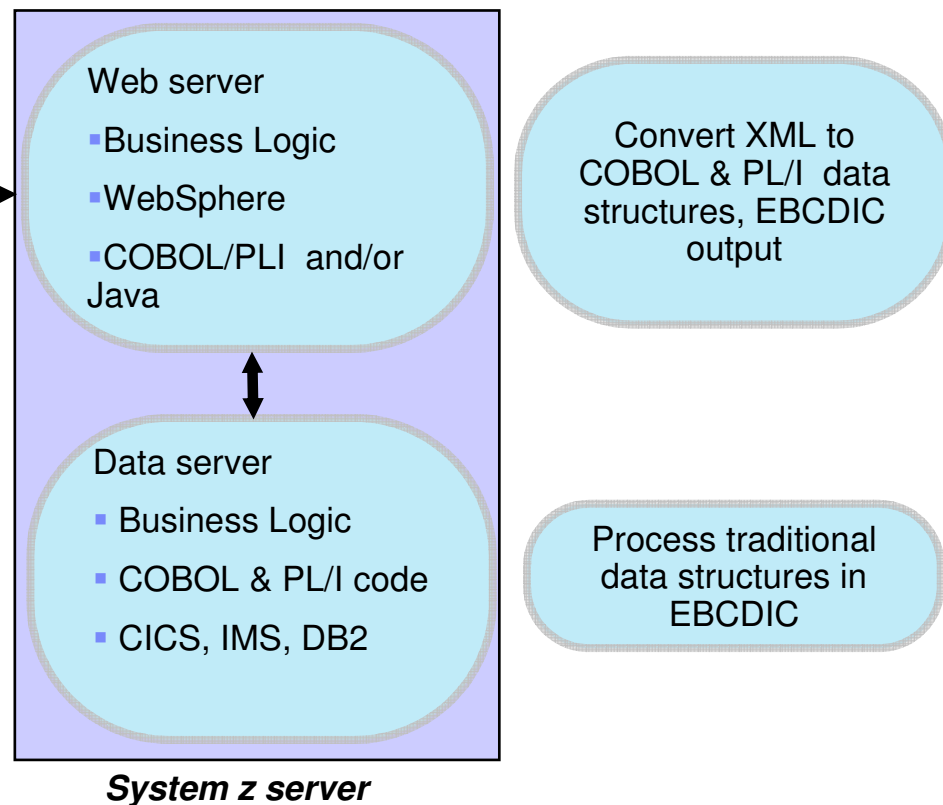
Comprehensive risk mitigation and reduced development cost by extending existing business critical applications to work with new web infrastructure.

Modernize your Development Environment

- **What is Rational Developer for z (RDz)**
  - Eclipse-based IDE speeding modern composite (SOA) application development

- **RDz supports Enterprise Modernization**
  - Links WAS and core system z processing
  - Supports common IDE for COBOL, PL/I, C, C++, HLASM, Java, and web services
  - Transforms UML to COBOL source code
  - Provides interactive access to z/OS for development, debug, job generation, submission, monitoring, command execution
  - Supports new and existing runtimes (CICS, IMS, Batch, USS, DB2 SP, WAS)

- **RDz supports SOA**
  - Enables CICS and IMS applications for web services and SOA
  - Supports for J2EE, JCA, XML, web services
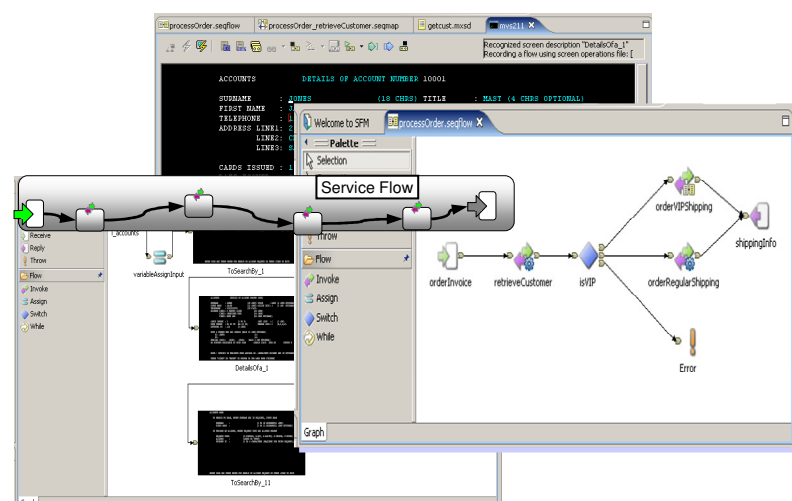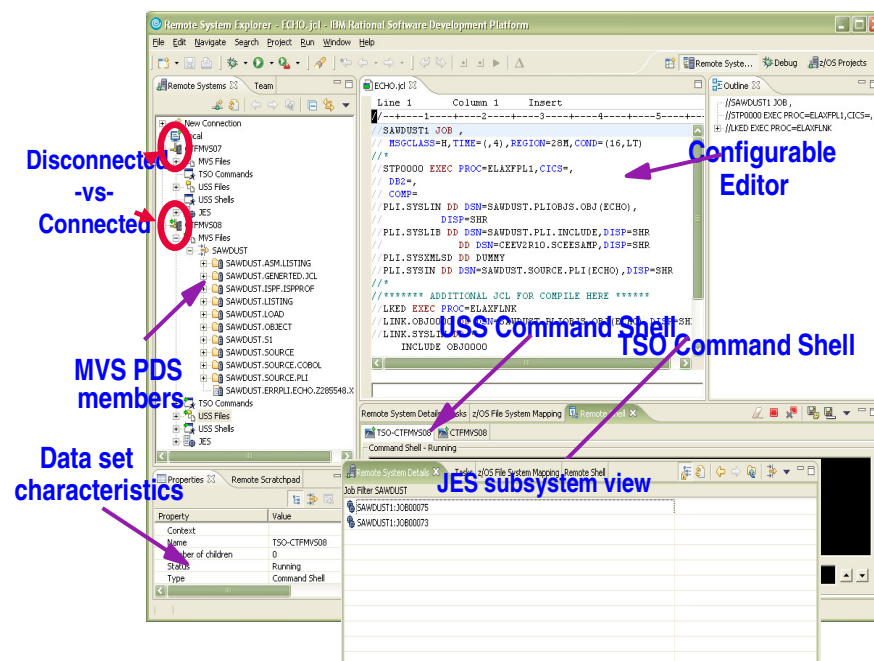
New Product Announcements

- **Enterprise COBOL for z/OS v4.2**
  - GA: Aug 28, 2009
  - Product Webpage: http://www-01.ibm.com/software/awdtools/cobol/zos/

- **z/OS XL C/C++ v1.11**
  - GA: Sep. 25, 2009
  - Product Webpage: http://www-01.ibm.com/software/awdtools/czos/

- **Enterprise PL/I for z/OS v3.9**
  - GA: Oct 30, 2009
  - Product Webpage: http://www-01.ibm.com/software/awdtools/pli/plizos/

Enterprise COBOL for z/OS v4

- ***Enables the integration of existing applications with web applications***
  - Supports Java interoperability by object-oriented COBOL syntax
  - Supports access to enterprise beans that run on Websphere Application Server or J2EE-compliant EJB server.

- ***Support processing of XML documents***
  - Provides built-in language support for high speed XML parsing and generation. COBOL
  - XML parsing supports offloading to zAAP specialty processors

- ***Improved exploitation of z/Architecture***
  - Provides significant improvements to UNICODE performance
  - Supports new z10 architecture

- ***Support latest Middleware and tools***
  - Enhanced DB2 support through new SQL data types and new SQL syntax provided by DB2 9
  - Integrated CICS translator
  - Support Debug Tool

New! Enterprise COBOL for z/OS v4.2
What you need for running business-critical applications and modernizing legacy code

- **Improved support for processing of XML documents**
  - Enable validation of XML document against a schema while it processed by the COBOL application
  - Support offloading of XML parsing to *zAAP* specialty processors
  - Provide significant performance improvement when using z/OS XML System Services parser

- **Improved usability**
  - Underscore now supported in user-defined words and program name literals
  - New MSGEXIT allows customization of severity for Compiler messages
  - New BLOCK0 option to exploit system-determined blocksize for QSAM output files

- **Improved support for middleware and tools**
  - Support CICS v4.1
  - Capability to show CICS options in listings

- **More Java versions supported to integrate COBOL applications with Web applications**
 - Added support for Java 5 and Java 6

Enterprise PL/I for z/OS

- **_Exploit the new z10 processor_**
  - Full exploitation of z10 hardware with ARCH(8)/TUNE(8)
    - Exploit new Decimal Floating Point instructions
  - Exploits some 64-bit registers with 31-bit code to improve performance

- **_Enables the integration of existing PL/I applications with web applications and SOA_**
  - Support integration of PL/I and web-based business processes in Web services, XML, and Java™ applications

- **_Support processing of XML documents_**
  - Provides PL/I built-in subroutines for high speed XML parsing and generation
  - XML parsing supports offloading to zAAP specialty processors

- **_Support latest Middleware and tools_**
  - Integrated SQL and CICS translator
  - Support Debug Tool

New! Enterprise PL/I for z/OS v3.9

Improves performance of your business critical applications and accelerate modernization of legacy code
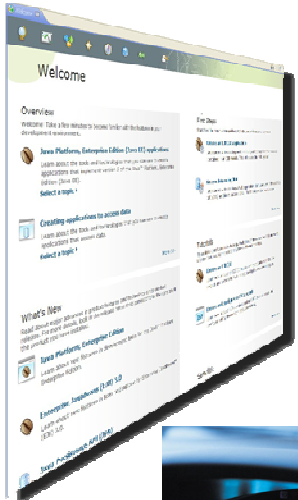
- **More exploitation of recent hardware**
  - Inline conversions amongst UTF-8, UTF-16 and UTF-32
  - Improved support for Internationalization
  - Improved null pointer usage with compare-and-trap instruction

- **Improved I/O capability**
  - Support of LRECL=X for QSAM files with records > 32K
  - Support of dynamic file allocation

- **Improved support for latest Middleware and tools**
  - Block scoping added to the integrated SQL and CICS preprocessors
  - Enhanced IPCS support for easier and faster problem resolution

- **Addressed many user requirements, including**
  - Improved performance in array and bit assignments
  - Enabled feature to allow easy compiling out of code with comments (i.e. %DO SKIP)
  - Provided more RULES suboptions to enforce coding standards
  - Compiler message to identify expensive statements to help tune application performance
  - Compiler option to suppress PUT and/or DISPLAY statements to ease switching from test code to production code.
  - Added compiler messages to flag poor and/or dangerous code to improve maintainability
  - Support for additional common European date formats
  - Provide new attributes for better code and messages (INONLY, INOUT and OUTONLY)
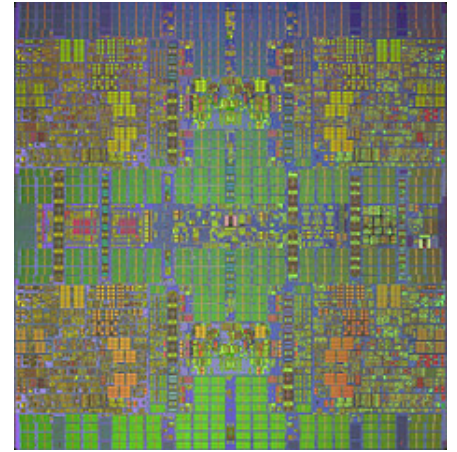
z/OS XL C/C++

- ***Exploit the new z10 processor***
  - Full exploitation of z10 hardware
    - New Decimal Floating Point instructions
    - Supports 31-bit and 64-bit

- ***Performing aggressive optimizations to C/C++ programs***
  - Loop optimizations, whole program optimization, profile-directed feedback, and memory hierarchy optimization…
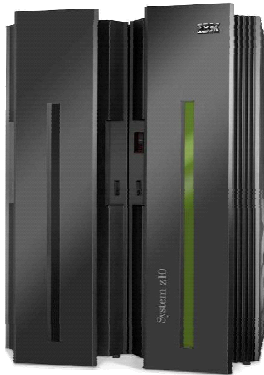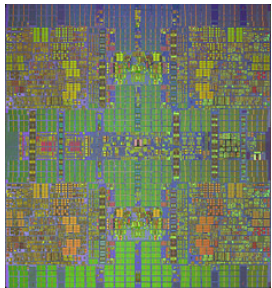
- ***Provide system programming capabilities***
  - METAL C Compiler option allows users to use z/OS C/C++ in place of assembler for system programming

- ***Support latest Middleware and tools***
  - Integrated SQL and CICS translator
  - Support Debug Tool, RDz

New! z/OS XL C/C++ v1.11

Improves portability & performance of your business-critical applications and accelerate system programming

- **Improved Performance**
  - Improved overall SPECint2006 performance by 8.7%
  - Improved libquantum[1] performance by ~15% with improved memory optimization
  - Improved a customer application ~20% with new optimization feature to inline C++ destructor

- **Language Standard**
  - Implemented Phase 1 of the new C++ language standard (C++0x):

- **Provide system programming capabilities**
  - METAL C Compiler option **received good review from customers**

- **Support latest Middleware and tools**
  - Improved readability of listings, which are frequently used on z/OS (qskipsrc)
  - Support debug when original source is not available (e.g. on production systems)
  - Produce pseudo-C listing to aid debugging of optimized code, performance tuning code and options used for compiling and linking (-qreport)

SPEC® and the benchmark names SPECint® and SPECfp® are registered trademarks of the Standard Performance Evaluation Corporation. Benchmark results stated above are estimates. For the latest information on SPEC®CPU2000 and SPEC®CPU2006, visit www.spec.org .

[1] libquantum is one of the constituent benchmarks of SPECCPU2006

16

© 2009 IBM Corporation

Improving Programmer Productivity
Programming in assembler vs <u>Metal C</u> on System z

```
        USING CTXT,R9              ESTABLISH BASE FOR CTXT
        CLC   CTXTACRN,=CL4'CTXT'  A CTXT?
        BNE   RETURN               SOMETHING IS WRONG IF NOT
        ICM   R3,B'1111',CTXTTXPN  A MINOR LINE?
        BNZ   RETURN               YES. NO GO
        ICM   R3,B'1111',CTXTTXPJ  PICK UP TEXT STRUCTURE POINTER
        BZ    RETURN               SHOULD NEVER BE THE CASE
        USING CTXTATTR,R3          ESTABLISH BASE FOR CTXTATTR
        SPACE 1
        CLC   IEE362A,CTXTTMSG     IEE362A?
        BNE   RETURN               DO NOTHING IF NOT
        CLC   SMFDSN,CTXTTMSG+L'SKIP DSNAME OK?
        BNE   RETURN               DO NOTHING IF NOT
        SPACE 1
        TIME  DEC                  GET DATE IN 00YYDDDF
        STCM  R1,B'0011',DDD       PICK UP DDD PORTION
        DP    PACKDDD,=PL1'7'      WHICH WEEK OF THE YEAR?
        CP    REM1,=P'0'           IS THIS THE 7TH DAY?
        BE    NOADD1               B. IF YES
        AP    QUO1,=P'1'
        SPACE 1
NOADD1  DS    0H
        MVC   WEEK,=CL4'EVEN'      ASSUME EVEN
        DP    QUO1,=PL1'2'         DETERMINE EVEN/ODD
        CP    REM2,=PL1'0'         EVEN?
        BE    NOCHANGE
        MVC   WEEK,=CL4'ODD'       NO. THEN IS ODD
        SPACE 1
NOCHANGE DS   0H
        MVC   MGCRTEXT(L'CMDTEXT),CMDTEXT COPY COMMAND INTO BUFFER
        MVC   MGCRTEXT+L'CMDTEXT-8(1),CTXTTMSG+L'IEE362A GET SUFFIX
        MVC   MGCRTEXT+L'CMDTEXT-4(4),WEEK SAY EVEN OR ODD
        LA    R0,MGCRTEXT-MGCRPL+L'CMDTEXT FIGURE OUT BUFFER LENGTH
        STC   R0,MGCRLGTH            INSERT EXACT BUFFER LENGTH
        SPACE 1
        SLR   R0,R0                CLEAR R0 FOR MGCR
        MGCR  MGCRPL               ISSUE START COMMAND
```

Platform: IBM System z (z/OS)
Problem: *System Programming -*
  user exit to handle the SMF
  dataset message (IEE362A) when
  a particular SMF dataset is full

```
void XIEE362A(struct ctxt * ctxtp) {
  struct ctxtattr * ctxtattrp;
  struct mgcrpl wmgcrpl;

  if (!memcmp(ctxtp->ctxtacrn,"CTXT",4)) return;
  if (ctxtp->ctxttxpn) return;
  ctxtattrp = ctxtp->ctxttxpj;
  if (ctxtattrp==NULL) return;
  if (!memcmp(ctxtattrp->ctxttmsg, "IEE362A SMF ENTER DUMP FOR ",
      27)) return;
  if (!memcmp(ctxtattrp->ctxttmsg+27, "SYS1.MAN", 8)) return;
  memcpy(wmgcrpl.mgcrtext, "START CLRSMF,N=x", 16);
  memcpy(wmgcrpl.mgcrtext+15, ctxtattrp->ctxttmsg+35, 1);
  wmgcrpl.mgcrlgth = 16;
  __asm("  XR 0,0\n MGCR %0"::"m"(wmgcrpl));
}
```

Code with
HLASM

Code with Metal C
z/OS C/C++

Improving Programmer Productivity

Leveraging Metal C to generate optimized assembly code

"... the Metal C optimization levels are just incredible. We're really impressed, there is no way that any of us assembler programmers can write code as efficient and keep it readable and do it in a reasonable amount of time. I'm going to require that all new development, unless there is a significant reason not to, be done in Metal C."

Gregg Willhoit, DataDirect

Metal C Generated Assembler (Optimized)

```
      if (!memcmp(ctxtp->ctxtacrn,"CTXT",4)) return;            000010
          USING @@PARMD@2,1                                     000010
          L     14,@2ctxtp                                      000010
          LARL  15,@@CONST@AREA@@                               000000
          CLC   0(4,14),17(15)                                  000010
          BRE   @2L7                                            000010
*     if (ctxtp->ctxttxpn) return;                              000011
          ICM   0,B'1111',12(14)         (*)ctxt.ctxt.ctxttxpn  000011
          BRNE  @2L7                                            000011
*     ctxtattrp = ctxtp->ctxttxpj;                              000012
          ICM   14,B'1111',8(14)         (*)ctxt.ctxt.ctxttxpj  000012
*     if (ctxtattrp==NULL) return;                              000013
          BRE   @2L7                                            000013
*     if (!memcmp(ctxtattrp->ctxttmsg, "IEE362A SMF ENTER DUMP FOR ", 27)  000014
          CLC   6(27,14),22(15)                                 000014
          BRE   @2L7                                            000014
*     if (!memcmp(ctxtattrp->ctxttmsg+27, "SYS1.MAN", 8)) return;  000015
          CLC   33(8,14),50(15)                                 000015
          BRE   @2L7                                            000015
*     memcpy(wmgcrpl.mgcrtext, "START CLRSMF,N=x", 16);         000016
          MVC   @12wmgcrpl+4(16),0(15)                          000016
*     memcpy(wmgcrpl.mgcrtext+15, ctxtattrp->ctxttmsg+35, 1);   000017
          IC    14,41(,14)               (*)Cvoid              000017
          STC   14,171(,13)              _a2_d171_l1_          000017
*     wmgcrpl.mgcrlgth = 16;                                    000018
          MVI   153(13),16                                      000018
*     __asm("  XR 0,0\n MGCR %0"::"m"(wmgcrpl));                000019
          LA    2,@12wmgcrpl                                    000019
          XR    0,0                                             000019
          MGCR  0(2)                                            000019
```

```
void XIEE362A(struct ctxt * ctxtp) {
  struct ctxtattr * ctxtattrp;
  struct mgcrpl wmgcrpl;

  if (!memcmp(ctxtp->ctxtacrn,"CTXT",4)) return;
  if (ctxtp->ctxttxpn) return;
  ctxtattrp = ctxtp->ctxttxpj;
  if (ctxtattrp==NULL) return;
  if (!memcmp(ctxtattrp->ctxttmsg, "IEE362A SMF ENTER DUMP FOR ", 27))
      return;
  if (!memcmp(ctxtattrp->ctxttmsg+27, "SYS1.MAN", 8)) return;
  memcpy(wmgcrpl.mgcrtext, "START CLRSMF,N=x", 16);
  memcpy(wmgcrpl.mgcrtext+15, ctxtattrp->ctxttmsg+35, 1);
  wmgcrpl.mgcrlgth = 16;
  __asm("  XR 0,0\n MGCR %0"::"m"(wmgcrpl));
}
```

Metal C

Maximizing application performance on new servers
## *Exploit Decimal Floating Point (DFP) Unit*

- **DFP** available in **new z10** processors

- *55%* **of numeric data are decimal** in commercial databases (e.g. DB2)

- **Binary floating point arithmetic is not sufficient** for commercial application
  - Can lead to unexpected results for values that are inherently decimal (e.g. monetary values and percentages)

- **Decimal arithmetic is carried out mostly in software**
  - Much slower than hardware (by 100X to 1000X)

- *Latest C/C++ and PL/I compilers support and exploit DFP*

Complex Applications with Multiple Languages
Leverage the best language for the task

- **100% support mixed language environment**
  - Intermix COBOL, PL/I, C, C++, and Fortran within the same application

- **Leverage the strengths of different programming languages within the same application**

- **Support Language Environment (LE) on z/OS**
  - Provides a common foundation to run programs written using different programming languages
  - Architected from ground up to facilitate interoperation between different languages
  - COBOL, PL/I, C, C++, assembler

    programs work together under LE

C/C++

Assembler

COBOL

PL/I
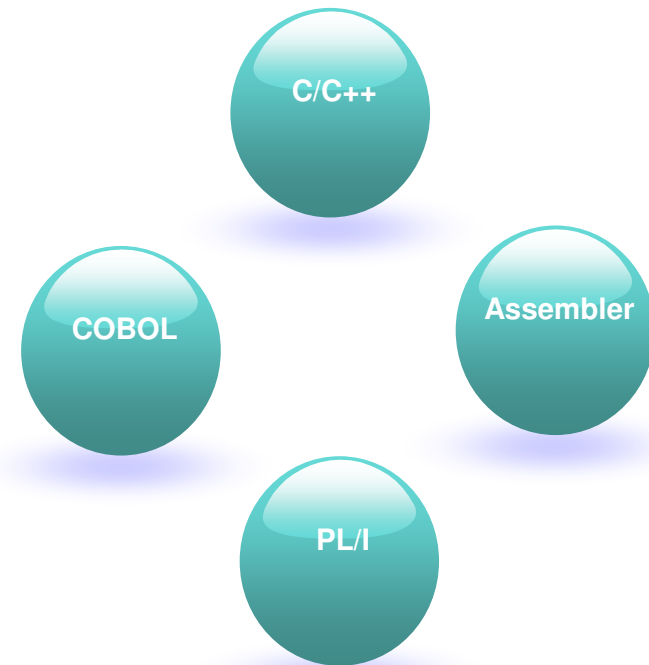
Summary

- **IBM compilers are designed to exploit …**
  - Advances in hardware architectures
  - Advances in middleware (CICS, DB2, IMS, …)

- **IBM compilers provide capability to ….**
  - Modernize legacy applications leveraging features like Java interoperability and XML support in COBOL and PL/I
  - Easily port applications to IBM platforms by supporting industry standards and extensions
  - Reduce programming complexity of hardware architectures

- **Rational is committed to delivering leading-edge compilation technology to ….**
  - *Maximize application performance and increase capacity*
  - *Improve programmer productivity and shorten development cycle*
  - *Lower TCO and increase return on IT investment*

ibm.com/rational/cafe

Join a community of industry leaders in modernization technology

Download trials of new technology

Learn to take full advantage of the latest tools

Share knowledge through forums, blogs and wiki documents
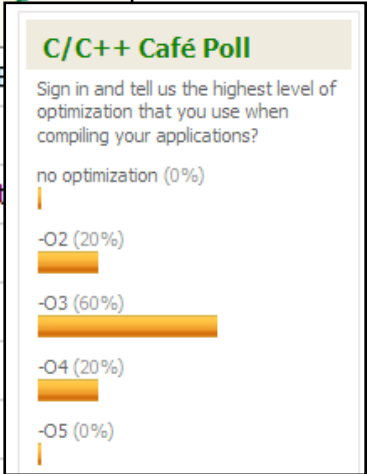
EGL

- Partners
- John's Corner - Learn
- Rich UI
- for IBM i
- Migration/Transformation
- Japan Street

RPG

- Partners
- RDi
- RDi SOA
- RTCi

Resource Library

- Articles, Presentations, and E...
- EGL JAM 2009
- Case Studies and Success St...
- Certification
- Demos
- Documentation Corner
- Downloads
- FAQ's
- Goings On
- Jon's Corner - Learn
- Samples
- Sand Box - i
- Sand Box - z
- Support
- Videos

C/C++ Café Poll

Sign in and tell us the highest level of optimization that you use when compiling your applications?

- no optimization (0%)
- -O2 (20%)
- -O3 (60%)
- -O4 (20%)
- -O5 (0%)

C/C++

- Commercial Computing
- Parallel and Multi-Core Computing
- Scientific Computing
- Standards

COBOL

- Partners
- RDz
- RDz with EGL
- RTCz
- HATS
- Standard
- Migrations

*Content subject to change

Resources

- **Main Product Line Pages:**
  - COBOL http://www.ibm.com/software/awdtools/cobol
  - PL/I http://www.ibm.com/software/awdtools/pli
  - C/C++ http://www.ibm.com/software/awdtools/ccompilers

- **Documentation**

  - **Enterprise COBOL for z/OS**
    - http://www-306.ibm.com/software/awdtools/cobol/zos/library/?S_CMP=rnav
  - **Enterprise PL/1 for z/OS**
    - http://www-306.ibm.com/software/awdtools/pli/plizos/library/?S_CMP=rnav
  - **z/OS XL C/C++**
    - http://www-306.ibm.com/software/awdtools/czos/library/?S_CMP=rnav

- **Key Contacts**

  - **Product Management Team**
    Roland Koo rkoo@ca.ibm.com 905-413-4469 (T/L 313-4469)
    Dwayne Moore dwayne@ca.ibm.com 905-413-3358 (T/L 313-3358)

  - **Worldwide Sales**
    Al Grega algrega@us.ibm.com 631-590-5037 (T/L 930-4764)

© 2009 IBM Corporation