Reducing CPU Usage for Critical Applications with IBM’s Cutting-Edge COBOL Offerings

Roland Koo, Offering Manager, COBOL, ABO and Node.js on z/OS

DevOps for IBM Z Virtual Conference
July 25 – 27
Disclaimer

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
Agenda

- Application Performance
  - where does it come from?

- Latest IBM COBOL Offerings
  - COBOL V6.2
  - ABO V1.3

- Q & A
Where does application performance come from?

Before

Now and Future

To improve application performance, you’ll need to
- Recompile
- Optimize
- Refactor ...

Applications
Software Stack
Hardware

Applications
Software Stack
Hardware

Effort
HIGH
MEDIUM
LOW
Value of IBM COBOL Compilers on IBM Z

IBM’s continual investment on COBOL compilation technology

Reduce CPU utilization by improving performance of applications with no source changes
  – Exploitation of z/Architecture (z10, z196, zEC12, zBC12, z13, z14)
  – Advanced Optimization

Support next-generation Applications
  – Modernize business critical System of Record applications to support new System of Engagement applications (e.g. web, mobile, cloud…)
  – Support latest IBM Middleware (CICS, IMS, DB2…)

Improve Productivity
  – New programming features
  – New problem determination features
Old binaries cannot take advantage of new H/W features

- **z10EC/z10BC ARCH(8):**
  - Decimal Floating Point (DFP)
  - Wider immediate data

- **z196/z114 ARCH(9):**
  - Distinct Operand
  - Conditional Load
  - High Word Instructions

- **zEC12/zBC12 ARCH(10):**
  - Decimal-Floating-Point Zoned-Conversion Facility…

- **z13/z13s ARCH(11):**
  - Decimal-Floating-Point Packed-Conversion Facility
  - SIMD

- **z14 ARCH(12):**
  - Vector Packed Decimal Facility
  - SIMD

Default ARCH level in Enterprise COBOL v6.1 is now ARCH(7) - IBM System z9EC (2094-xxx models) IBM System z9 BC (2096-xxx models)
COBOL on IBM Z
2 complementary offerings

Designed to improve performance, and programming capabilities of business critical COBOL applications

- Enterprise COBOL for z/OS
  - Compile and Optimize COBOL applications

- Automatic Binary Optimizer for z/OS
  - Optimize existing program modules
Enterprise COBOL v6.2
Ann: 7/17/17
GA: 09/08/17
COBOL V5/V6
Introduced major infrastructure changes

• Advanced Optimization Framework
• COBOL Runtime
• Object and Debug formats
  • Program Objects - Requires PDSE
  • DWARF debug format
• Maintains source and binary compatibility
  • Correct COBOL programs will compile and execute without changes and will produce same results
  • “Old” and “new” code can be mixed within an application
  • Removed some old language extensions and options
• Supports IBM DevOps tools
  • Application Delivery Foundation for z (IBM Developer for z Systems, IBM Debug, Fault Analyzer, Application Performance Analyzer), ADDI (Application Discovery and Delivery Intelligence)
• Supports the ecosystem of programming tools supplied by ISVs.
## COBOL Offerings Through Time

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Front end</th>
<th>Back End</th>
<th>ABO Eligibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS/VS COBOL</td>
<td>74 Std</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; generation</td>
<td></td>
</tr>
<tr>
<td>VS COBOL II</td>
<td>85 Std (new)</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; generation (new)</td>
<td>ABO 1.2 (must be LE enabled) or later</td>
</tr>
<tr>
<td>COBOL/370</td>
<td>85 Std (same)</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; generation (same)</td>
<td>ABO 1.2 or later</td>
</tr>
<tr>
<td>COBOL for OS/390 V2</td>
<td>85 Std (same)</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; generation (same)</td>
<td>ABO 1.2 or later</td>
</tr>
<tr>
<td>COBOL for z/OS V3</td>
<td>85 Std (same)</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; generation (same)</td>
<td>ABO 1.1 or later</td>
</tr>
<tr>
<td>COBOL for z/OS V4</td>
<td>85 Std (same)</td>
<td>2&lt;sup&gt;nd&lt;/sup&gt; generation (same)</td>
<td>ABO 1.1 or later</td>
</tr>
<tr>
<td>COBOL for z/OS V5</td>
<td>85 Std (same)</td>
<td>3&lt;sup&gt;rd&lt;/sup&gt; generation (new)</td>
<td></td>
</tr>
<tr>
<td>COBOL for z/OS V6</td>
<td>85 Std (same)</td>
<td>3&lt;sup&gt;rd&lt;/sup&gt; generation (same)</td>
<td></td>
</tr>
</tbody>
</table>

Extra work is needed when migrating across the thick lines because of infrastructure changes:

- **OS/VS COBOL migration new generations is the most difficult**
- **Migration from 2<sup>nd</sup> generation to 3<sup>rd</sup> generation is difficult**
- **Migration within the generation, is very easy, including from v5 to v6**

**Compiler Terms:**

- Front End handles parsing and language rules
- Back End handles optimization and code generation
Client Experiences

• Strong adoption
  • >550 customers

• Positive feedback on performance of batch applications
  • CPU reduction of 10% to 20%
  • Best case: CPU reduction over 60% (Elapsed time reduction over 70%) reported by a major FSS customer in US
Modernization of COBOL environment

BCEE – Luxembourg’s oldest bank – has successfully migrated to the latest generation of the IBM Enterprise COBOL compiler.

ibm.com/zsystems
Migration to COBOL V5: BCEE in Luxembourg

BCEE had been using the V3 COBOL compiler for many years.

When IBM announced end of service support for the V3 COBOL compiler, concern arose within BCEE.

BCEE performed preliminary migration testing on a subset of production programs in order to help determine what the best course of action should be.

As a result of the migration testing, BCEE felt the best course of action was to move forward with a COBOL modernization initiative.

Migrating to the COBOL V5 compiler would provide BCEE with the latest in COBOL compiler technology and continued IBM support.

Collaborating with IBM, BCEE developed a sound, thorough, COBOL compiler migration plan that included significant planning, testing and execution.

A key component of BCEE’s migration plan was the implementation of a user driven testing initiative.

New features to aid with the COBOL migration process, such as INITCHECK, resulted from this collaboration.

Migration to the V5 COBOL compiler was considered a success!

BCEE successfully recompiled and tested roughly 9000 production programs.

BCEE are very happy with the outcome of the V5 COBOL compiler migration and as a result, are working with IBM on new mainframe solutions.

The message? Confidently upgrade to the latest COBOL compiler, leveraging IBM’s support to overcome migration challenges in a timely and efficient manner.

*IBM, IBM logo, are trademarks or registered trademarks of the International Business Machines Corporation.
All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

For further information contact rab@ca.ibm.com
DATEV, one of Germany’s leading software and IT service providers, modernized their payroll accounting process. Now, their accounting process runs 10 – 15% faster.
DATEV had a critical accounting process that created more than 10 million payroll transactions per month, which resulted in the processing of batch jobs nearly 24 hours a day.

In order to optimize performance, DATEV had upgraded to IBM z13™.

When IBM announced the release of the latest generation of Enterprise COBOL compilers, many at DATEV saw the potential to further optimize their performance capabilities by upgrading their COBOL compiler.

DATEV installed and tested the Enterprise COBOL V5 compiler. Due to the significant performance improvements that resulted from migrating to Enterprise COBOL V5, DATEV decided to migrate to Enterprise COBOL V6.

The migration to Enterprise COBOL V5.2 was considered a success! As a result of the COBOL V5 compiler upgrade, DATEV’s accounting process now runs 10-15% faster, and, one production process in particular (XML) reduced its CPU usage by more than 3X!

The migration to Enterprise COBOL V6 is on going.

The message? By upgrading to the latest generation of Enterprise COBOL compiler, you could significantly reduce your CPU usage!
Modernization of a COBOL Environment

UBS - a leading player in the global financial industry - has successfully recompiled a portion of their production programs with the Enterprise COBOL V5 compiler. These programs have seen performance improvements of 12 - 13%.

ibm.com/zsystems
Migration to COBOL V5: UBS in Switzerland

**IN THE BEGINNING**
UBS had COBOL code that had been generated from different versions of the COBOL compiler. As a result, UBS was maintaining support for multiple versions of COBOL.

UBS had migrated to the latest z hardware (z13) architecture in order to gain performance improvements.

**THE BIG IDEA**
When IBM announced the release of the latest – and most advanced - generation of COBOL compilers, UBS felt there was opportunity to further improve performance by upgrading to the Enterprise COBOL V5 compiler.

Upgrading to Enterprise COBOL V5 would not only enforce standardization and streamlining of the COBOL environment, but also, eliminate the need to maintain support for multiple versions of the COBOL compiler.

**TAKING ACTION**
Collaborative efforts with IBM led to:
1) the development of a systematic COBOL compiler/coprocessor migration plan that included significant compiler, LE, Debug Tool and ISV component planning & testing, &
2) timely and efficient issue detection and problem solving.

UBS’s migration plan included nightly build and testing capabilities where several hundred test use cases were performed every night.

**RESULT**
The migration to the V5 COBOL compiler is ongoing and considered a success!

UBS has successfully recompiled and tested roughly 10,500 programs with the goal of recompiling a total of 35,000 production programs by 2018.

Initial performance measurements have shown 12 – 13% improvement in COBOL code path execution.

UBS is very happy with the V5 COBOL compiler migration and the collaborative support that IBM has provided.

**The message?** By upgrading to the latest generation of Enterprise COBOL compiler, you could significantly reduce your CPU usage.

*IBM, IBM logo, are trademarks or registered trademarks of the International Business Machines Corporation. All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.*

For further information contact rab@ca.ibm.com

ZSC03356USEN-00
z14 – New Vector Packed Decimal Facility

• Key ARCH(12) z14 Vector Packed Decimal Facility summary
  • 22 new instructions added to z14 Instruction Set Architecture
  • Covers a wide range of the most common decimal operations in COBOL
  • All computations now done in hardware registers instead of in memory
    • Lower latency and better hardware performance

• No source changes required– just recompile with new ARCH Level

Benefits the following COBOL Source Statements

<table>
<thead>
<tr>
<th>Benefits the following COBOL Source Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal COMPUTEs/MOVEs</td>
</tr>
<tr>
<td>Mixed signed/unsigned operations</td>
</tr>
<tr>
<td>Many Intrinsic Functions</td>
</tr>
<tr>
<td>Complex mixed type computations</td>
</tr>
<tr>
<td>Large (&gt; 9 digit) binary COMPUTEs/MOVEs</td>
</tr>
</tbody>
</table>
Example: Unsigned Packed Decimal Add – 4.85x

Faster

01 WS-VAR-1 COMP-3 PIC 9(7)
01 WS-VAR-2 COMP-3 PIC 9(7).
01 WS-VAR-3 COMP-3 PIC 9(7).
ADD WS-VAR-1 TO WS-VAR-2 GIVING WS-VAR-3.

V4
ARCH(6|7|8|9|10)
- Use in memory instructions
- Explicit sign setting

MVC 168(4,R9),160(R9)
OI 171(,R9),X'0F'
MVC 352(4,R13),152(R9)
OI 355(,R13),X'0F'
AP 168(4,R9),352(4,R13)
OI 171(,R9),X'0F'

ARCH(11)
- Convert to DFP
- Conversion overhead

CDPT FP0,160(4,R9),0x9
CDPT FP1,152(4,R9),0x9
ADTR FP0,FP0,FP1
LPDFR FP0,FP0
CPDT FP0,168(4,R9),0xb

ARCH(12)
- Use new ARCH(12) facility
- No conversions, no explicit sign setting

VLRL VRF16,160(,R9),0x3
VLRL VRF17,152(,R9),0x3
VAP VRF16,VRF16,VRF17,0x7,14

ARCH(12) Speedup on z14

- Speedup
  - ARCH(12) 4.85
  - ARCH(11) 1.66
  - ARCH(10) & V4 1
Example: Large Decimal Divide – 135x Faster

```
01 WS-VAR-1 COMP-3 PIC s9(29).
01 WS-VAR-2 COMP-3 PIC s9(3).
01 WS-VAR-3 COMP-3 PIC s9(25)v9(6).
```

Without ARCH(12)
- Call out to LE library routine
- Pre shifting operation
- Piecewise divide, call overhead

```
ZAP   336(16,13),16(2,2)
MVC   352(32,13),58(10)
MVC   366(15,13),0(2)
NI    380(13),X'F0'
MVN   383(1,13),14(2)
L     3,92(0,9)
L     15,180(0,3) V(IGZCXDI )
LA    1,180(0,10)
BASR  14,15
```

With ARCH(12)
- Use new ARCH(12) facility
- Inline hardware accelerated shift+divide

```
VLRL   VRF24, WSA[0x12c] 0,(R3),0xe
VLRL   VRF25, WSA[0x12c] 16,(R3),0x1
VSDP   VRF24, VRF24, VRF25, 0x6, 0
```

ARCH(12) Speedup on z14

- Speedup
  - ARCH(12): 135
  - Original: 1
Enterprise COBOL v6.2 – Performance on z14

- Compile time performance improvement over COBOL 6.1
  - Reduces CPU by
    - 6% at OPTIMIZE(1)
    - 18% at OPTIMIZE(2)

- Reduces application build time

* Performance results shown was obtained in a controlled, isolated environments using IBM internal test suite. Performance of other workload may vary
Enterprise COBOL v6.2 - Features

Parse JSON documents directly from COBOL
- Native support for JSON in COBOL - allow business critical functions to work with JSON objects and REST services

Improve reporting on invalid data and out of bounds checks
- Helps detect coding errors in existing source
- Improves code quality and migration from COBOL V4 & below

New NOSTGOPT implementation
- NO data division optimization done regardless of OPT-level
- “Eye-catchers” will always be preserved and initialized if they are defined with a VALUE clause

Condition Compilation Support
- COBOL 2002 feature
- Enables developers to create multiple variants of the same program without having to maintain separate source streams

Enhanced TEST options
- Consistent set of sub-options
- Added SEPARATE/NOSEPARATE

INLINE option and directives
- Provides fine grain control over which procedures to inline or not inline to optimize performance
COBOL Migration Assistant

- Provides step-by-step guidance on migrating to COBOL V5 or V6
- Allow users to log migration progress
- Cloud based - Hosted on Bluemix
  - IBMid required for free access
- Visit https://cobol-migration-assistant.mybluemix.net/

Consult Migration Guide http://www-01.ibm.com/support/docview.wss?uid=swg27036733 or ask your IBM rep. to arrange a Migration Workshop
Automatic Binary Optimizer v1.3
Ann: 7/17/17
GA: 09/08/17
Automatic Binary Optimizer (ABO)

- Optimizes performance of load modules compiled with COBOL v4 and below
  - Increases application performance
  - Reduces CPU usage and operating costs
- Requires no recompilation, or source migration
- Targets latest IBM Z mainframes (e.g. zEC12, zBC12, z13, z13s and, z14)
- Requires significantly less testing effort
- Complementary to Enterprise COBOL V6

Automatic Binary Optimizer (ABO)

**ABO**
- Unbind
- Binary $\rightarrow$ IR*

**Verify**
- Optimize
- Optimize
- Optimize...

**ABO**
- IR $\rightarrow$ Binary
- Rebind

**Done**

*IR - Intermediate Representation
ABO Compatibility and Testing

Binary As Input
ABO optimizes the original binary so exact behavior of the original program is known.

Options Unchanged
No compile options (performance or otherwise) are being changed.

Full Inter-operability
Optimized program would run faster and would behave the same as the original program.

Strong compatibility assurance for the modules produced by ABO

Testing
Requires System Verification and Performance Tests Only
If something goes wrong, revert to using original program

Users can optimize objects in test environment and to deploy executable to production environment
Tools & Middleware Support

IBM Application Delivery Foundation for z (ADFz)

Provides Day-1 support

- SYSDEBUG of original programs can be used along with ABO Listings to create new side files to work with
- Fault Analyzer (FA) - Provide ABEND analysis of optimized load modules
- Debug for z - Allows users to step through and debug optimized programs interactively
- Application Performance Analyzer (APA)
  - Allows users to compare the performance of original programs and optimized programs

ISVs Tools
- Macro 4
  - Knowledgbase article number: K73-4729
  - DumpMaster 8.300C or higher; Common Services 2.400C or higher; TraceMaster 6.600D or higher; TraceMaster CodeTrack 1.710C
  - Working with Compuware, and CA

Middleware
- DB2, CICS, and IMS
ABO v1.3 - Performance on z14

- Fully exploits z14 architecture
  - Vector Packed Decimal Facility
- Optimize code every time you upgrade to new IBM Z
- Up to 2X Reduction in CPU and Memory usage over ABO v1.2

* Performance results shown was obtained in a controlled, isolated environments using IBM internal test suite. Performance of other workload may vary.
ABO and COBOL V6 are complementary

Recommendation: Use both to maximize time to value and ROI

<table>
<thead>
<tr>
<th>Use Case</th>
<th>ABO</th>
<th>COBOL V6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improve performance without recompile, migration or options tuning</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>No concerns for Interoperability/Legacy Compatibility – Load modules, program objects, OS/VS COBOL etc…</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>No need to be limited by ARCH setting of DR* machine. Optimize for deployment machine and run original binaries on DR</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Source code not available</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>New COBOL application development or modernize existing applications to work with new infrastructure (e.g. web, mobile…. )</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Need source changes (e.g. defect fixes and/or minor enhancements)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Maximize performance improvement</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

*DR → Disaster Recovery Machine → Usually older hardware since it is only used in emergency situation. This limits Compiler ARCH setting used to compile the application. As a result, the application cannot be fully optimized for deployment machine which usually is more current hardware.
Using COBOL V6 and ABO together

**COBOL V6**
- Defect Fixes
- Refactoring
- New programming features
- Modernization
- Recompile for Performance

**ABO**
- Increase performance of programs that have not been recompiled for a long time
  - No source risk
  - Less testing effort
- Reduce scope of V5/V6 migration
- Generate ROI to invest in business critical applications and adoption of DevOps

10%  
Additional recompile when upgrading to COBOL V6 to improve performance

20%  
Recompile due to Source Change

70%  
No recompile Plan
Economics $$

Increase performance of COBOL applications

- Reduces CPU usage and Processing Time
- Reduces Batch Window
- Lowers operating costs by driving down peak 4HRA workload
- Reclams capacity for other applications

MLC cost is calculated based on peak 4HRA workload (MSU)

No ABO
No COBOL V6

With ABO and/or COBOL V6

Increase performance of COBOL Applications lowers peak 4HRA workload (MSU). Lowers MLC cost for entire system

Reclaim Capacity

Increase performance of COBOL Applications lowers processing time and CPU but does not lower 4HRA. However, it allows customers to reclaim capacity for other workloads

© 2017 IBM Corporation
Trials

Enterprise COBOL V6
Automatic Binary Optimizer for z/OS

- Offers no charge evaluation license for up to 90 days
- Allows clients to assess the value of Offerings
  - Same function as GA product
  - Ships with latest PTF
  - Code processed by trial versions cannot be used for production
- Available as
  - Standard offering from IBM through ShopzSeries
  - Cloud Service https://optimizer.mybluemix.net
- Contact your IBM representative for ordering assistance
Quick Start Services

COBOL Migration (40 hours)
- Help customers migrate a small subset of COBOL programs to COBOL V6.
- Customers can use the hands-on experience to effectively plan their migration to COBOL V6 and take advantage of optimization technology to reduce CPU usage.

ABO (40 hours)
- Help customers identify, plan, and optimize a small subset of COBOL load modules compiled with Enterprise COBOL V4 or earlier.
- Customers can use the hands-on experience to effectively optimize their COBOL applications to reduce CPU usage.
Resources

- **Enterprise COBOL**
  - COBOL Migration Assistant: [https://cobol-migration-assistant.mybluemix.net](https://cobol-migration-assistant.mybluemix.net)
  - Link to Enterprise COBOL V6.2 Announcement Blog: [https://ibm.co/2vdZjrF](https://ibm.co/2vdZjrF)

- **ABO**
  - Trial on Cloud: [https://optimizer.mybluemix.net](https://optimizer.mybluemix.net)
  - ABO Demo: [https://youtu.be/xhnhdOQwwSQ](https://youtu.be/xhnhdOQwwSQ)
  - Link to ABO v1.3 Announcement Blog: [https://ibm.co/2vxMpUX](https://ibm.co/2vxMpUX)

- **RFE community:**
Thank You