SIMD Processing on IBM z14, z13 and z13s

Background

Companies want to quickly and efficiently process huge amounts of information for analytics, mobile applications, data serving and more. To cost-effectively achieve these efficiencies new innovative technologies are needed. Vector processing and SIMD optimize performance by processing data in parallel on IBM Z®.

What is SIMD?

SIMD stands for Single Instruction Multiple Data. Unlike instructions which performs a single operation on a single data point, SIMD instructions can perform the same operation on multiple data points at once.

SIMD is an innovation that was delivered with IBM z13® (z13) and IBM z13s™ (z13s). With SIMD entire arrays of data can be processed by a single instruction enabling more efficient processing of data. In addition, the z13 and z13s superscalar processors feature 32 double-width (128 bit width) vector registers and 139 instructions to accelerate processing. Additional instructions, in support of decimal operations, were introduced with the new IBM z14™ (z14) and are designed to give a performance boost for traditional workloads using COBOL by tailoring the instructions to the specific need of financial applications. The z14 also enhanced the throughput of binary SIMD floating-point used in new applications like analytics.

Candidates for SIMD

Existing programs are excellent candidates for SIMD and can make use of SIMD by recompiling processing with integer and floating point data can all be accelerated using SIMD. Complex mathematical processing, algorithmic-intensive programs, image processing applications, and other applications can be efficiently processed with SIMD. SIMD also enables processing of a new class of applications on IBM Z using analytics, mathematical optimization without requiring an offload to specialized processors. Standard SIMD math and linear algebra libraries enable easier porting from X86 platforms.

Compilers

SIMD helps you achieve efficiencies using IBM z/OS® XL C/C++ V2.1.1, IBM Enterprise COBOL 5.2 and IBM Enterprise PL/I for z/OS V4.5, for applications with character or integer manipulation. IBM z/OS XL C/C++ V2.1.1 compilers offer new capabilities, such as architecture sections, inline assembly, and high optimizations which when combined with SIMD, provide a performance boost. The IBM z/OS XL C/C++ V2.1.1 compiler supports new data types and built-in functions to exploit SIMD instructions. Enterprise COBOL 5.2 uses SIMD to improve the processing of INSPECT...TALLYING or INSPECT... REPLACING statements and accelerates string operations.

Similarly, Enterprise PL/I for z/OS V4.5, also exploits SIMD, with expanded use of the Decimal Floating Point Facility for Packed Decimal calculations.

Analytics applications such as Apache Spark for z/OS benefits from SIMD improvements in decimal operations when processing highly iterative analysis on large volumes of data.

Public-Key Cryptographic Standards (PKCS) #11 is a cryptographic token interface standard. Select PKCS#11 clear key operations can take advantage of SIMD for performance gains.

Operating Systems

z/OS V2.3, and z/OS V2.2 or z/OS V2.1 with PTFs, can support SIMD instructions. VM V6.4 and VM 6.3 with PFT UM34752, enables guess support for SIMD, allowing Linux® to use SIMD. KVM executing on IBM Z also supports SIMD.

Java

Java® is a language of choice for next generation cloud and analytics workloads. IBM z14, z13s, z13, and IBM Java 8 drives Java performance to new levels with SIMD, simultaneous multithreading (SMT) and improved cryptographic processing.

Java applications, exploiting SMT-enabled zIIP specialty engines, benefit from many performance improvements due to Java 8 code optimizations and use of SIMD by Java string operations.

Java uses SIMD to provide dramatic acceleration of matrix operations, string processing and more. The Java JIT compiler offers native operations which work on Java byte arrays, eliminating the need for interim Java objects. Text accelerations include code page conversion and array compare operations as part of java.lang.String class operations.

Auto-vectorization is a Just-In-Time (JIT) compiler optimization in IBM Java 8 that accelerates simple scalar loops by leveraging vector operations on z13. Matrix multiplication operations were up to 60% faster when using the new Java 8 JIT in IBM lab measurements[1].

Candidate Applications for SIMD

- ✓ String and character array processing
- ✓ Loop processing.
- ✓ Binary floating point processing such as mathematical optimization & modeling
- ✓ Telematics applications
- ✓ Repetitive security calculations
- ✓ Graphics processing, video processing
- ✓ Java character and string processing
- ✓ XML System Services
- ✓ Other Java 8 SR1 applications

Using SIMD is Easy

With new vector processing in IBM Z, many existing programs need *only be recompiled* in-order to take advantage of SIMD. Programs or workloads using IBM Java will exploit SIMD *automatically* with the newest Java 8 version.

Summary

Software for analytics, business intelligence and database processing as well as traditional applications can easily get a boost in processing with SIMD on z14, z13s and z13, contributing to the value of IBM Z for processing your most data intensive workloads.

For More Information

- SIMD Business Analytics Acceleration on z Systems - Point of View, REDP-5145-00 http://www.redbooks.ibm.com/abstracts/redp5145.html?Open
- The SIMD accelerator for business analytics on the IBM z13 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7175128

[1] http://mainframeinsights.com/java-performance-ibm-z-systems-ibm-z13-ibm-java-sdk-8/