AI on IBM zSystems

Accelerating AI inference for high volume transactional workloads on IBM z16

Andrew M. Sica
STSM, AI on IBM zSystems
andrewsi@us.ibm.com

Agenda

01 The modern mainframe

02 OLTP workload Primer

03 IBM Telum

04 Software ecosystem

05 Putting it all together

IBM zSystems is at the core of the world's businesses

72% of the customer-facing applications are backed by IBM zSystems applications and data

70% of business transactions run on IBM zSystems (including 90% of all credit card transactions)

... and this means 30B transactions per day

Who runs their businesses on IBM zSystems?

67 of the Fortune 100

45 of the world's top 50 banks

8 of the top 10 insurers

4 of the top 5 airlines

7 of the top 10 global retailers

8 of the top 10 telco's

24 of the top 25 countries by GDP*

IBM zSystems - the backbone of critical business processes and the world's economy

… for unmatched reliability and security across 70%+ of Fortune 500 companies

Banking

92 of the top 100 banks

- ATM transaction processing
- Financial services transactions
- Clearing and statutory reporting
- Mobile banking apps and web payment platforms

Insurance

8 of the top 10 insurance providers

- Insurance claims processing
- Customer information management
- Billing and payments systems of record

Retail

7 of the top 10 global retailers

- Customer transaction processing with 24x7 uptime during seasonal peaks
- Secure and compliant store for customer data (GDPR, CCPA, etc.)

Healthcare

8 of the top 10 global healthcare providers

- Pervasive encryption for electronic health records systems
- Regulation compliant analytics for Protected Health Information (PHI) workloads

What is at Stake for IBM zSystems clients?

DAILY

30 billion transactions

400 million retail transactions

1 million hotel nights

ANNUALLY

29 billion ATM transactions

87% of all credit card transactions

90% of all airline reservations

The biggest customers use IBM zSystems because…

- it is reliable → The famous “7 nines” availability (99.99999%)
  - 3.2 secs of downtime per year and a Mean Time Between Failures (MTBF) measured in decades!
- it runs mixed workloads → OLTP, databases, analytics and batch on the same machine with 90%+ CPU utilization
- it is secure → With technologies like Pervasive Encryption data from the IBM zSystems remains encrypted at every point of its life cycle
- it is highly scalable → 1 – 200 computing cores, >10,000 Linux VMs on a single server
- it is cost-efficient → IBM zSystems houses > 70% of enterprise data assets but account for <10% of total IT costs
- it is modern → zSystems runs Linux, OpenShift, Python, Blockchain, talks REST APIs and makes a perfect link of the DevOps chain

Two faces of IBM zSystems

Mostly “foundational” workload…

Db2

CICS

COBOL

MQ

IMS

Java

zSystems

zOS, Linux, zVM, KVM, zTPF, zVSE

LinuxONE

Linux, KVM, zVM

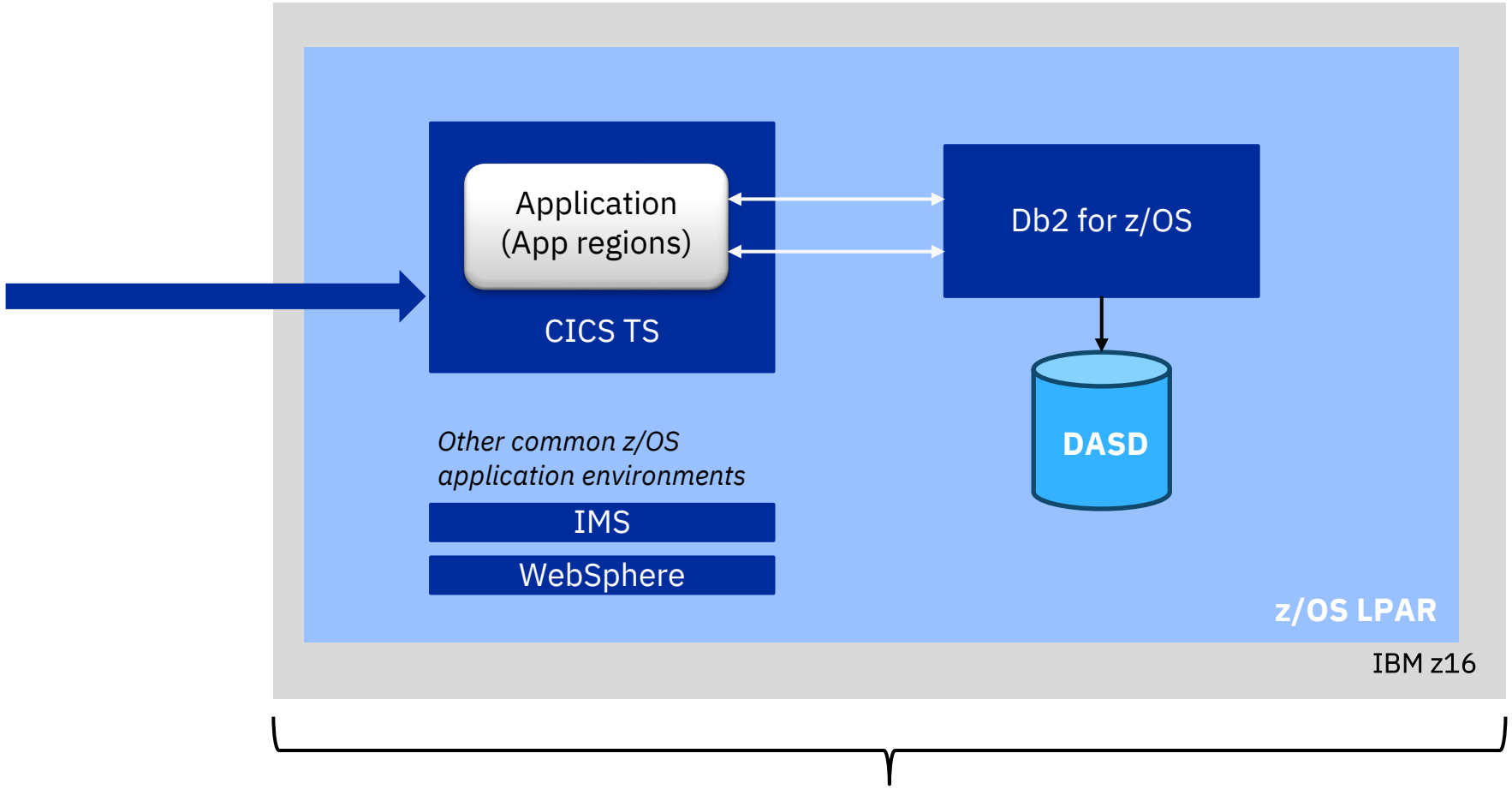
The majority of the open-source products

Anatomy of a typical z/OS transaction workload environment

Core business systems and databases co-located in a logical partition.

Reliability, availability, serviceability core considerations clients rely on.

Parallel Sysplex (not shown) provides for tightly coupled multi-system management with ongoing data consistency (not “eventual consistency”)



SLA = Service level agreement

Transaction processing workloads are often high volume, low latency workloads with tight SLA requirements.

Large credit card processor example:
- 60,000 transaction per second
- <10ms per request

Challenges to leverage AI in this environment

80%

of respondents agreed that real-time insights are important[1]

49%

getting insights where and when they are needed is a big challenge[2]

(difficulty with applying AI to business-critical workloads

without impacting SLAs)

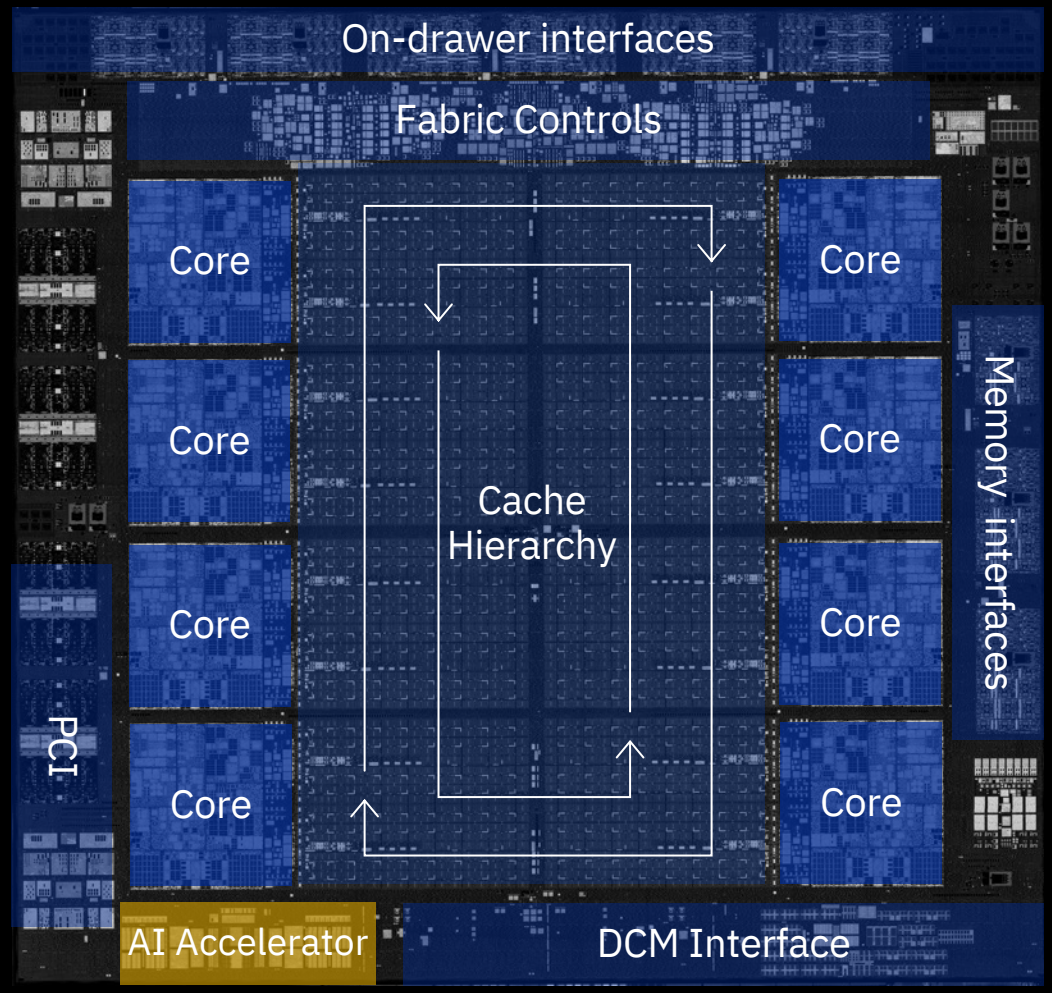
10%

of transactions in high volume enterprise workloads go through real-time AI screening[3]

Telum Processor and IBM z16

- Telum: Next generation IBM Z processor optimized to run enterprise workloads with embedded real time AI insights
  - 7nm design with 8 cores per chip @5.2 GHz
  - 40% per socket performance growth
  - Quantum-safe cryptography
  - On-chip low latency AI acceleration
- IBM z16 is announced earlier this year with up to 32 Telum processors and 40 TB of memory

ISCA'2022 – Session 4A: Industry Track



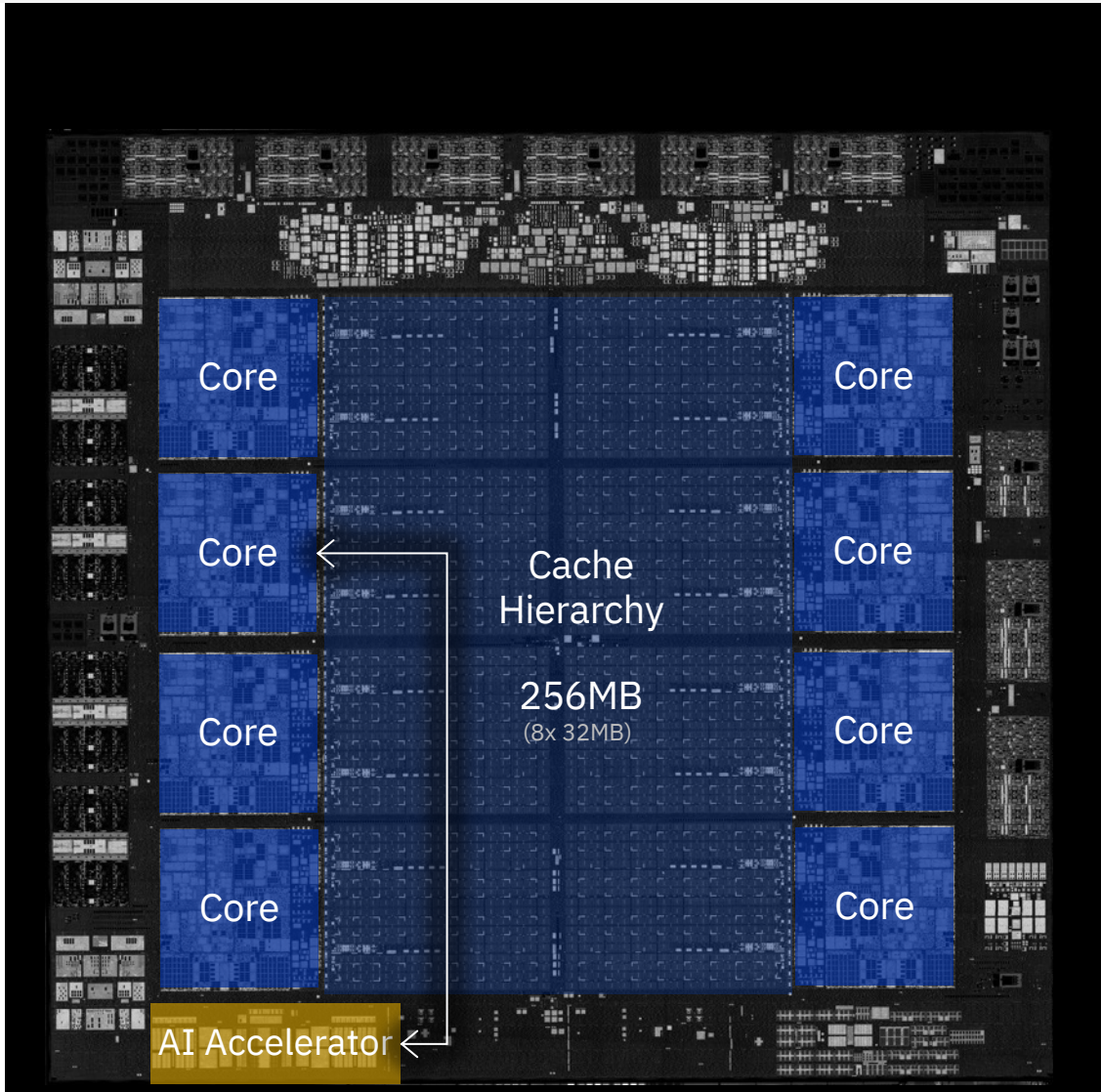
AI Inference with Central Low-latency Accelerator

Centralized On-chip accelerator shared by all cores

- Very low and consistent inference latency
- Compute capacity and bandwidth for utilization at scale
- Enterprise-grade memory virtualization and protection

Neural Network Processing Assist instruction

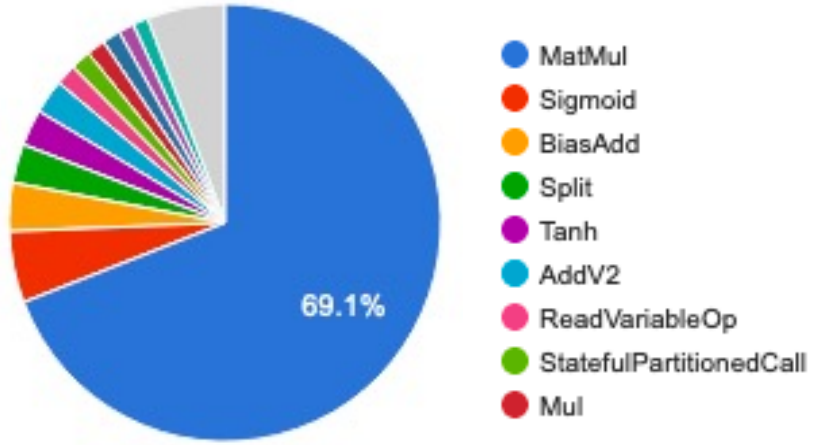
- Memory-to-memory core instruction
- Operate directly on tensors in user space
- Direct memory access with all protection mechanisms
- No data duplication and copying



AI Accelerator Supported functions

- AI Functions/Macros abstracted via NNPA instruction
  - Elementwise, Activation
  - Normalization, Pooling
  - Matrix-multiplication
  - Convolution
  - Conv+Scale+Activate
  - MatMul+Compare/Activate
  - RNN activation

Function group	#	Function support in GA1
Elementwise ops	0x10	NNPA_EL_ADD
	0x11	NNPA_EL_SUB
	0x12	NNPA_EL_MUL
	0x13	NNPA_EL_DIV
	0x14	NNPA_EL_MIN
	0x15	NNPA_EL_MAX
Activation ops	0x20	NNPA_LOG
	0x21	NNPA_EXP
	0x31	NNPA_RELU
	0x32	NNPA_TANH
	0x33	NNPA_SIGMOID
Norm op.	0x34	NNPA_SOFTMAX
	0x40	NNPA_BATCHNORM
Pooling	0x50	NNPA_AVGPOOL2D
	0x51	NNPA_MAXPOOL2D
Systolic ops	0x70	NNPA_CONVOLUTION
	0x71	NNPA_MATMUL_OP
	0x72	NNPA_MATMUL_OP_BCAST23
RNN	0x60	NNPA_LSTMACT
	0x61	NNPA_GRUACT
	0x00	NNPA_QAF



RNN model

- New functions can be added via firmware update

AI Ecosystem on zSystems

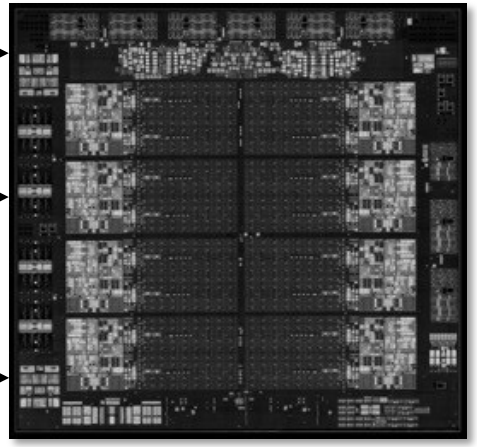
Build & Train anywhere

python
jupyter
Snap ML
dmlc XGBoost
learn
LightGBM

Keras
PyTorch
SAS
MATLAB
Chainer
TensorFlow

ONNX
ONNX

TensorFlow
Snap ML
Deep Learning Compiler



Deploy AI at scale on IBM zSystems

Applications

Banking
Finance
Insurance
Retail
Hospitality
Transport
Healthcare
Government
...

Offerings

IBM Watson

IT Operations

INSTANA an IBM Company
turbonomic
Cloud Pak for AIOps

Frameworks

Keras
PyTorch
Apache Spark
scikit learn
ANACONDA.
ONNX

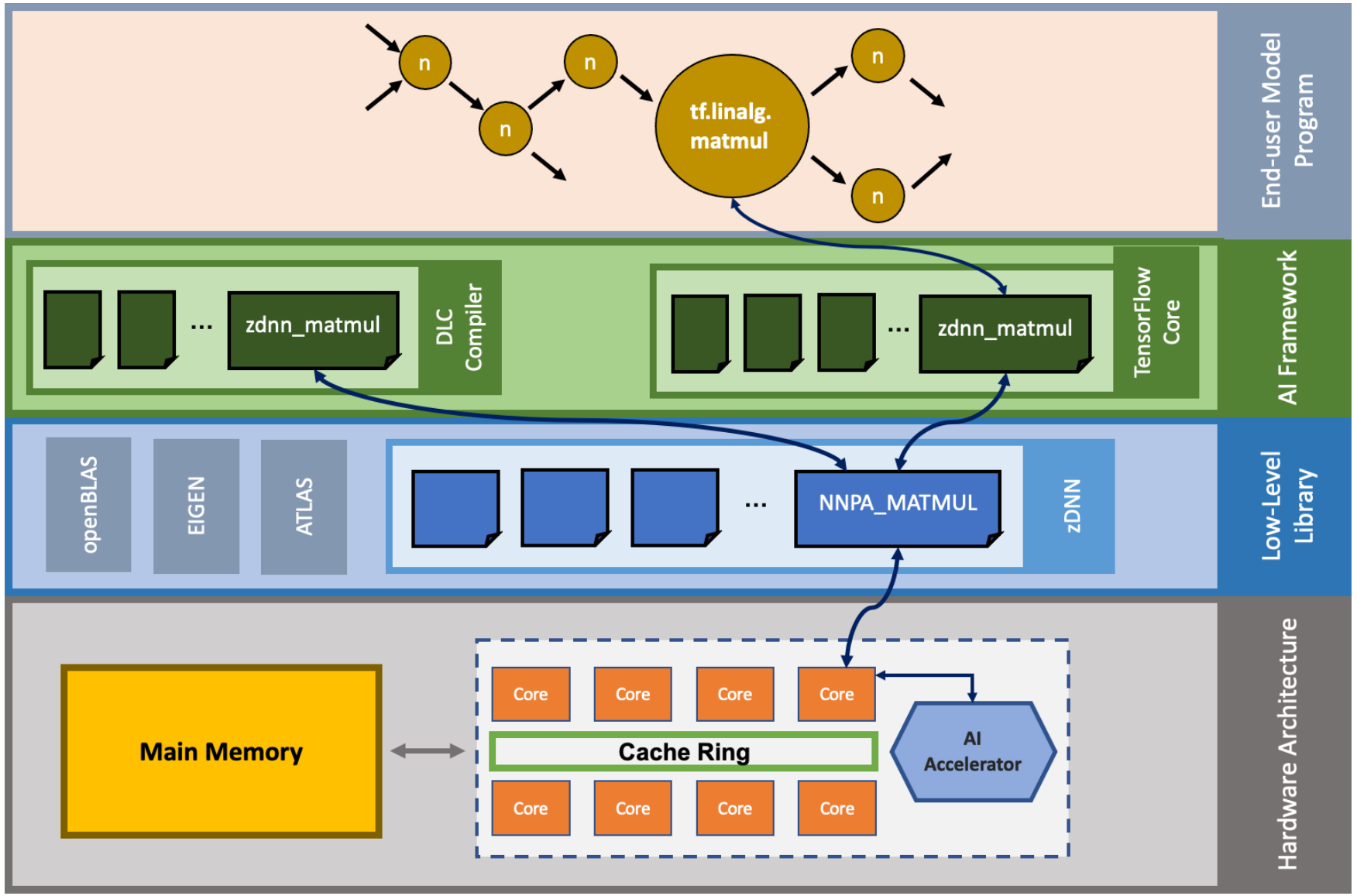
Libraries, Compilers

python
Java
GO
GCC
C++
OpenBLAS
OpenAS
Deep Learning Compiler

OS, Virtualization

z/OS
Ubuntu
SUSE
OPENSHIFT

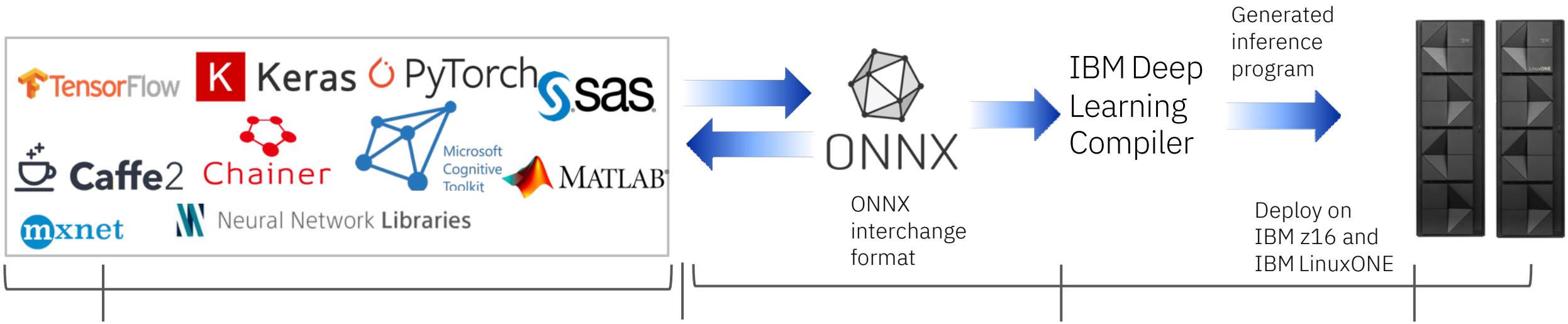
Vertical integration with IBM Telum AI accelerator



- AI models developed and trained anywhere can transparently leverage the z16 accelerator when deployed on IBM z16.
- Placement occurs via AI frameworks on zSystems instrumented to leverage IBM zDNN
- IBM zDNN is an open-source toolkit enabling developers to more simply target the z16 accelerator.

AI Ecosystem:
Seamlessly leverage AI accelerator on IBM z16

- Bring machine learning & deep learning models to IBM z16 with ONNX/DLC
- Exploit IBM Integrated Accelerator for AI for best inference performance
- Embedded within Watson Machine Learning z/OS



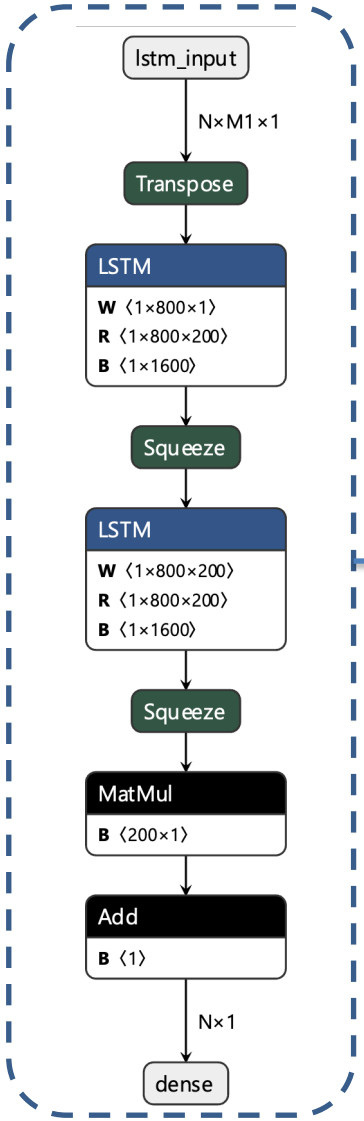
Build and train model in any popular framework on any platform of your choice

Use ONNX, an opensource tool for framework interoperability Models are converted to the ONNX interchange format Leverage zCX and run on zIIP engines

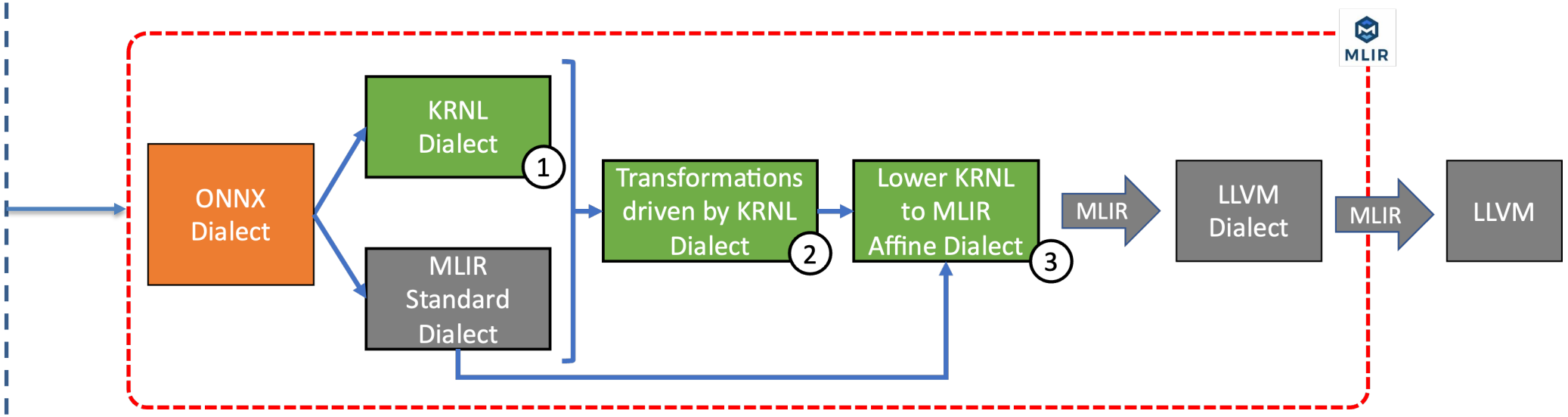
The IBM DLC (Deep Learning Compiler), optimized for performance and new libraries, generates a program from the model for execution on z/OS or Linux® on IBM z16

Deploy on IBM z16 and IBM LinuxONE and infuse model into workload application

onnx-mlir (IBM Z Deep Learning Compiler)



The ONNX model (dialect) is lowered and transformed through multiple phases of IR to a dialect that can be processed by an LLVM compiler.



The output of LLVM compilation and build is a shared library object that can be deployed

Key Features of Watson Machine Learning for z/OS v3.1 – **Enterprise Edition**

GUI Configuration

Web-based Configuration Tool for single instance and HA configuration

Model training tool

- Integrated Jupyter notebook server for model training on Z
- Leverage IBM Z Spark 3.2 and Python AI Toolkit for training and scoring

Scoring Engines

- Online scoring for SparkML, PMML, ONNX, Python and Watson Core time series models
- Leverage z16 on-chip AI accelerator for ONNX model scoring

Integrated Scoring

In-transaction scoring through native CICS and WOLA interface for CICS, IMS and BATCH COBOL applications

UI Dashboard

Web-based UI for WMLz environment and end to end model lifecycle management

Explainability

Model explainability - Integration with OpenScale on Cloud Pak for Data on IBM Z

CICS COBOL Application Calls WMLz Scoring

- The WMLz scoring service integrated in a CICS region as a program called **ALNSCORE**
- Use the CICS **LINK** command in your CICS COBOL application to call ALNSCORE for online scoring for SparkML, PMML, and ONNX models
- The call uses special containers to transfer the scoring input and output between the COBOL application and the ALNSCORE program

Container name	Type	Format
ALN_DEPLOY_ID	String	Deployment ID
ALN_INPUT_DATA	Structure	A data structure holds the input record to ALNSCORE. It is generated by the DFHJS2LS utility.
ALN_INPUT_CLASS	String	The class name is specified by the user when using the ALNJCGEN JCL to create the Java class.
ALN_OUTPUT_DATA	Structure	A data structure holds the scoring output from ALNSCORE. It is generated by the DFHJS2LS utility.
ALN_OUTPUT_CLASS	String	The class name is specified by the user when using the ALNJCGEN JCL to create the Java class.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. MODELPGM.
DATA DIVISION.
WORKING-STORAGE SECTION.

01  MODELIN.
          06 COUNTRY-length              PIC S9999 COMP-5 SYNC.
          06 COUNTRY                     PIC X(255).
          06 GENDER-length               PIC S9999 COMP-5 SYNC.
          06 GENDER                      PIC X(255).
          06 AGE                         PIC S9(18) COMP-5 SYNC.
          06 MARITAL-STATUS-length       PIC S9999 COMP-5 SYNC.
          06 MARITAL-STATUS              PIC X(255).
          06 PROFESSION-length           PIC S9999 COMP-5 SYNC.
          06 PROFESSION                  PIC X(255).
          06 NATIONAL-ID-length          PIC S9999 COMP-5 SYNC.
          06 NATIONAL-ID                 PIC X(255).
          06 CUSTOMER-ID                 PIC S9(18) COMP-5 SYNC.

01  MODELOUT.
          06 PREDICTION                    COMP-2 SYNC.
          06 PROBABILITY OCCURS 2          COMP-2 SYNC.

01 I PIC 9(2) VALUE 1.

PROCEDURE DIVISION.
          MOVE 'M'       TO GENDER.
          MOVE 1         TO GENDER-length.
          MOVE 19        TO AGE.
          MOVE 'Single'  TO MARITAL-STATUS.
          MOVE 6         TO MARITAL-STATUS-length.
          MOVE 'Student' TO PROFESSION.
          MOVE 7         TO PROFESSION-length.
          MOVE 10        TO CUSTOMER-ID.
          MOVE 'USA'     TO COUNTRY.
          MOVE 4         TO COUNTRY-length.
          MOVE 'XXXX'    TO NATIONAL-ID.
          MOVE 4         TO NATIONAL-ID-length.

          DISPLAY 'GENDER         :' GENDER.
          DISPLAY 'COUNTRY        :' COUNTRY.
          DISPLAY 'MARITAL-STATUS :' MARITAL-STATUS.
          DISPLAY 'NATIONAL-ID    :' NATIONAL-ID.
          DISPLAY 'CUSTOMER-ID    :' CUSTOMER-ID.
```

```
EXEC CICS PUT CONTAINER('ALN_DEPLOY_ID') CHANNEL('CHAN')
     CHAR
     FROM('29439127-f77c-472c-a851-188ca2d4c78d')
     END-EXEC.

EXEC CICS PUT CONTAINER('ALN_INPUT_CLASS') CHANNEL('CHAN')
     CHAR FROM('ModelInWrapper')
     END-EXEC.

EXEC CICS PUT CONTAINER('ALN_INPUT_DATA') CHANNEL('CHAN')
     FROM(MODELIN) BIT END-EXEC.

EXEC CICS PUT CONTAINER('ALN_OUTPUT_CLASS') CHANNEL('CHAN')
     CHAR FROM('ModelOutWrapper')
     END-EXEC.

EXEC CICS LINK PROGRAM('ALNSCORE') CHANNEL('CHAN')
     END-EXEC.

EXEC CICS GET CONTAINER('ALN_OUTPUT_DATA') CHANNEL('CHAN')
     INTO(MODELOUT) END-EXEC.

DISPLAY 'PREDICTION     :' PREDICTION.
DISPLAY 'PROBABILITY    :'.

PERFORM UNTIL I=3
DISPLAY 'PROBABILITY-' I
DISPLAY PROBABILITY(I)
ADD 1 TO I
END-PERFORM.
EXEC CICS RETURN END-EXEC.
STOP RUN.
```

Deploy ID container

Input Class container

Input Data container

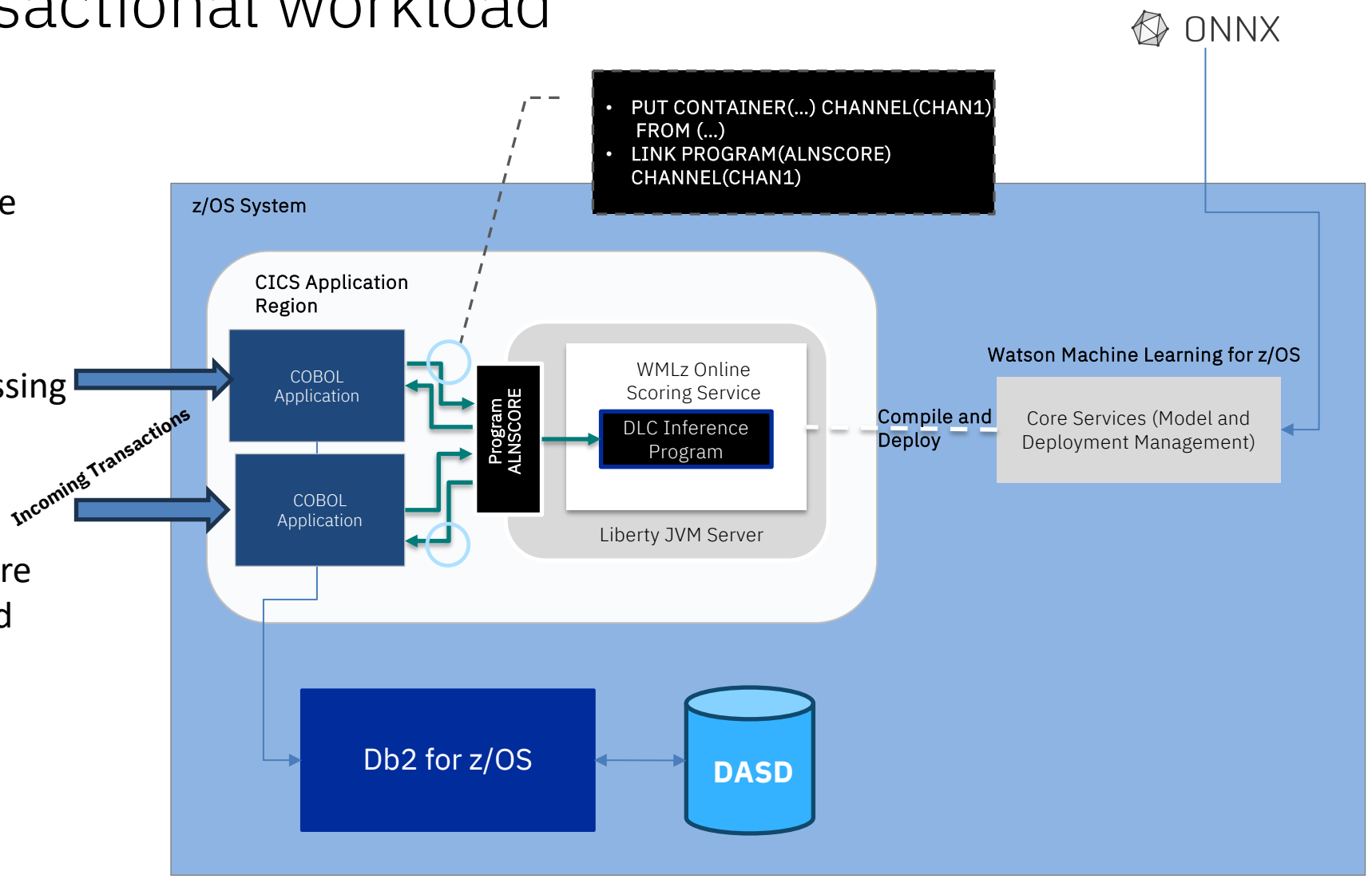
Output Class container

LINK to ALNSCORE

Output Data container

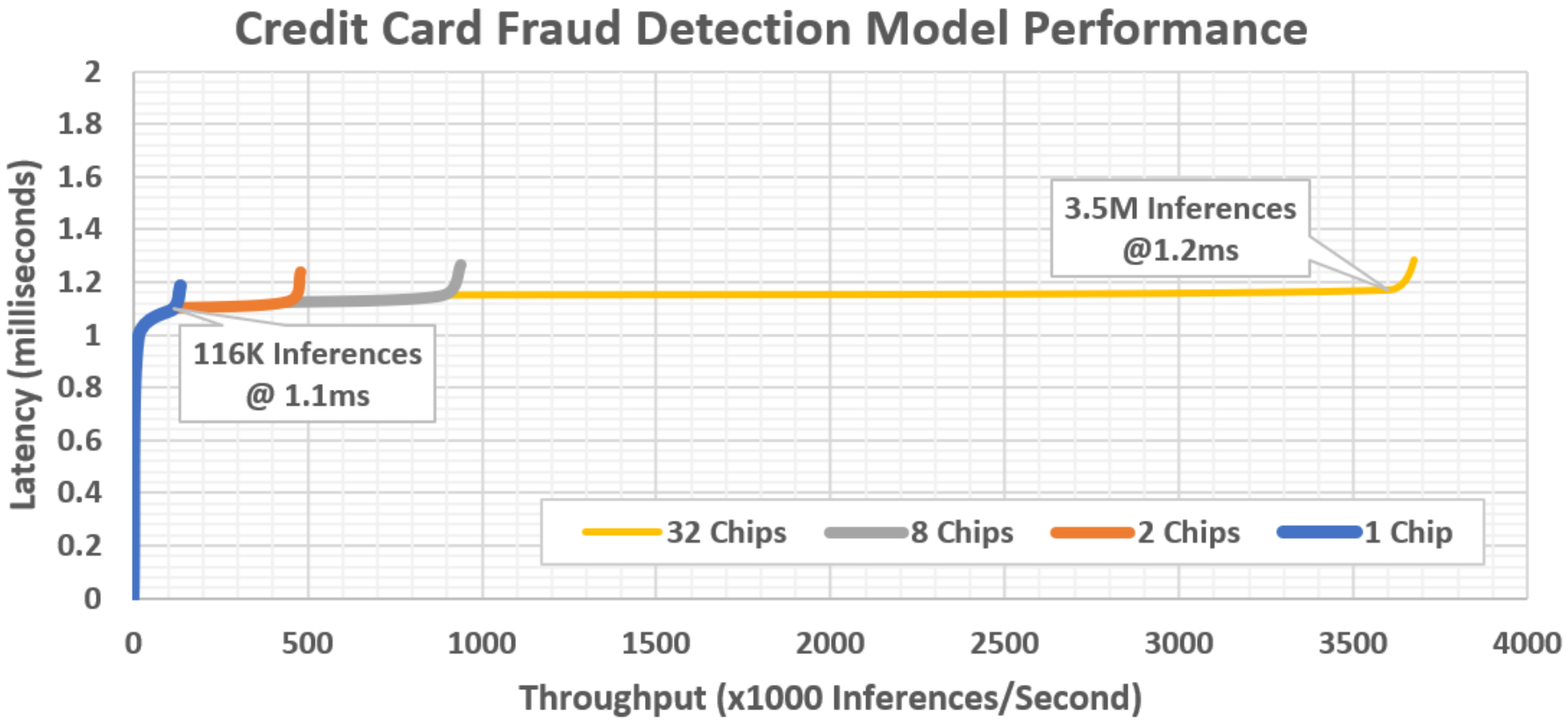
Deploying in a transactional workload

- Native language model inference library deployed in application address space.
- Utilize shared memory area, passing data in native format from/to Cobol application.
- Simplified implementation in core business applications, minimized overhead
- Transparently exploitation of hardware acceleration options



Stand-alone model execution results

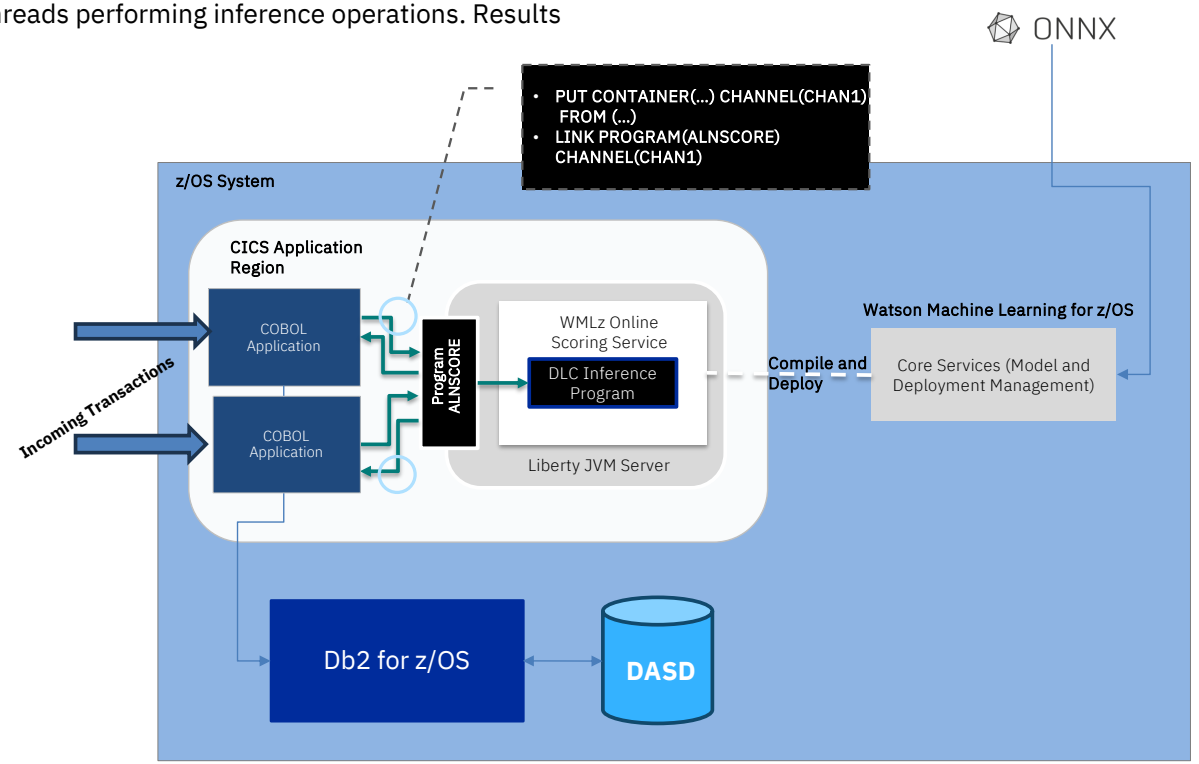
- Credit Card Fraud Detection Proxy Model
- Almost perfect scaling up to 32 chips in a system
- Input batch size of 128 (LSTM input is 7,128,204)



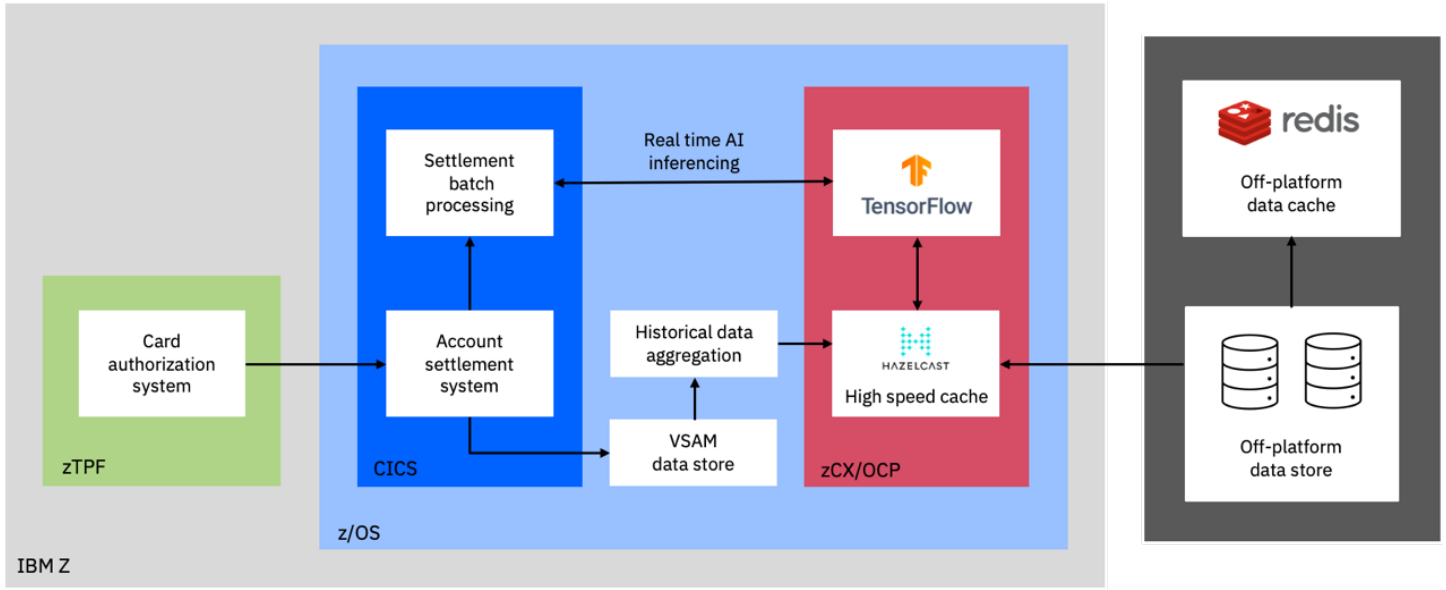
Results in transactional workload context

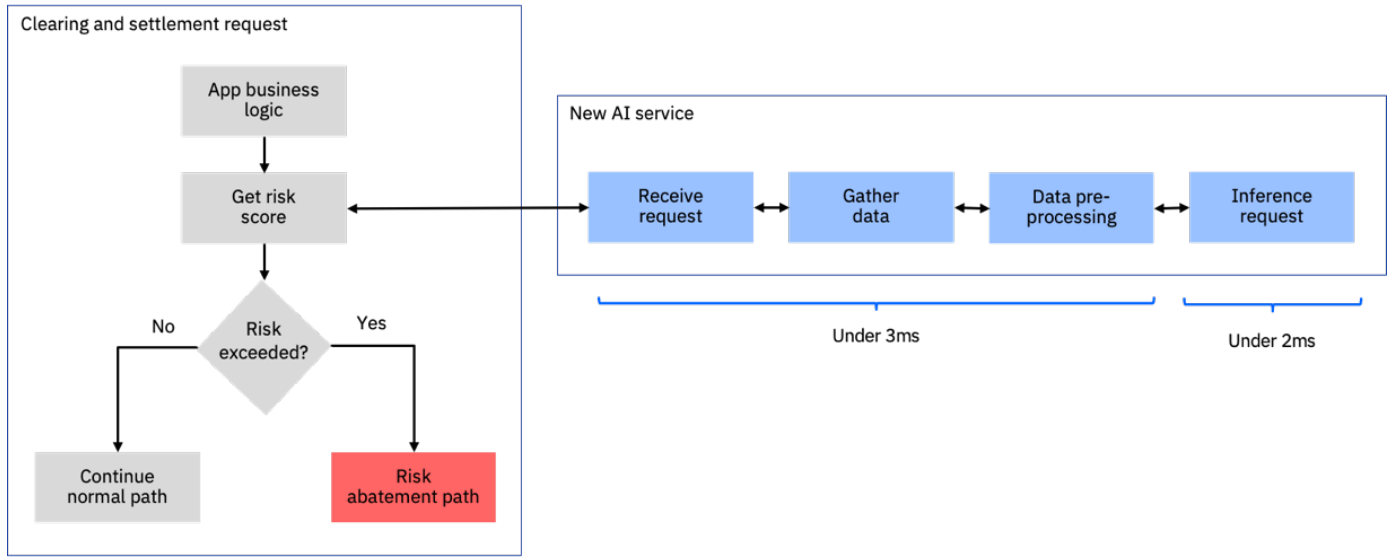
An IBM z16 system can process **up to 228K z/OS CICS credit card transactions per second with 6 ms response time**, each with an in-transaction fraud detection inference operation using a Deep Learning Model.

DISCLAIMER: Performance result is extrapolated from IBM internal tests running a CICS credit card transaction workload with inference operations on an IBM z16. A z/OS V2R4 LPAR configured with 6 CPs and 256 GB of memory was used. Inferencing was done with Watson Machine Learning for z/OS 2.4 running on Websphere Application Server Liberty 21.0.0.12, using a synthetic credit card fraud detection model (https://github.com/IBM/ai-on-z-fraud-detection) and the Integrated Accelerator for AI. Server-side batching was enabled on Watson Machine Learning for z/OS with a size of 8 inference operations. The benchmark was executed with 48 threads performing inference operations. Results represent a fully configured IBM z16 with 200 CPs and 40 TB storage. Results may vary.



Other common patterns: open-source on Linux on Z





- Co-located with z/OS workload, on Linux on Z environment**.**
- **TensorFlow Serving** used to deploy a pre-trained TensorFlow model.
- **Hazelcast** (in-memory data store) utilized as feature store.
- Full end-to-end pipeline including transport achieving < 5ms latency at scale.

Considerations + Closing

- Focused strategy was critical to enabling real-time AI in high volume transaction workloads
  - Optimize inference; training not currently a high value target for platform.
  - Minimize ecosystem impacts; critical in a non-x86 architecture environment.

- Full business application context must be considered:
  - It’s not just about model optimization and acceleration… Must also consider:
    - Serving performance and scalability
    - Overhead in invoking APIs
  - Simplified methods of invoking model server – using native language constructs.
    - Important when an update is required to the businesses most critical application(s).

Thank you!

Interested in trying it out?

Free access to a LinuxONE (IBM Z) environment is available!

Register in LinuxONE Community Cloud – hosted by Marist College
- Instructions: https://ibm.biz/BdPcL8

Engage with us:

- aionz@us.ibm.com

- AI on IBM Z and LinuxONE Community

- https://ibm.github.io/ai-on-z-101/

- Contact us directly

Sites
Journey to AI on IBM Z Content Solution link
IBM Z and Cloud Mod Center AI Page link
Real-Time analytics and AI on the IBM mainframe link

Blogs
TensorFlow blog: link
ONNX blog: link

Demos
Watson Machine Learning Demo link
Anti-Money Laundering with AI on Z link
Fraud Detection Demo link

Redbooks
Optimized Inferencing and Integration with AI on IBM Z Introduction, Methodology, and Use Cases: link
Demystifying Data with AI on IBM Z –POV: link
Art of the Possible with AI on IBM zSystems link

Paper
IDC: The business value of the transformative mainframe link
Operationalizing Fraud Prevention on IBM z16: Reducing Losses in Banking, Cards, and Payments link

Open Source
IBM Z and LinuxONE container Image Registry: link
TensorFlow on IBM Z and LinuxONE container Image Registry: link
Anaconda Partnership link

Disclaimers

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.

Disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in this presentation (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM, IBM 8-bar Logo, ibm.com, and IBM Z

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.

Red Hat®, JBoss®, OpenShift®, Fedora®, Hibernate®, Ansible®, CloudForms®, RHCA®, RHCE®, RHCSA®, Ceph®, and Gluster® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

RStudio®, the RStudio logo and Shiny® are registered trademarks of RStudio, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Zowe™, the Zowe™ logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.

Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon 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 throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

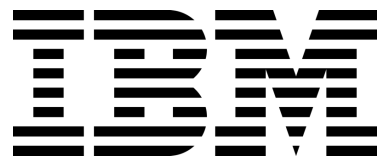
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

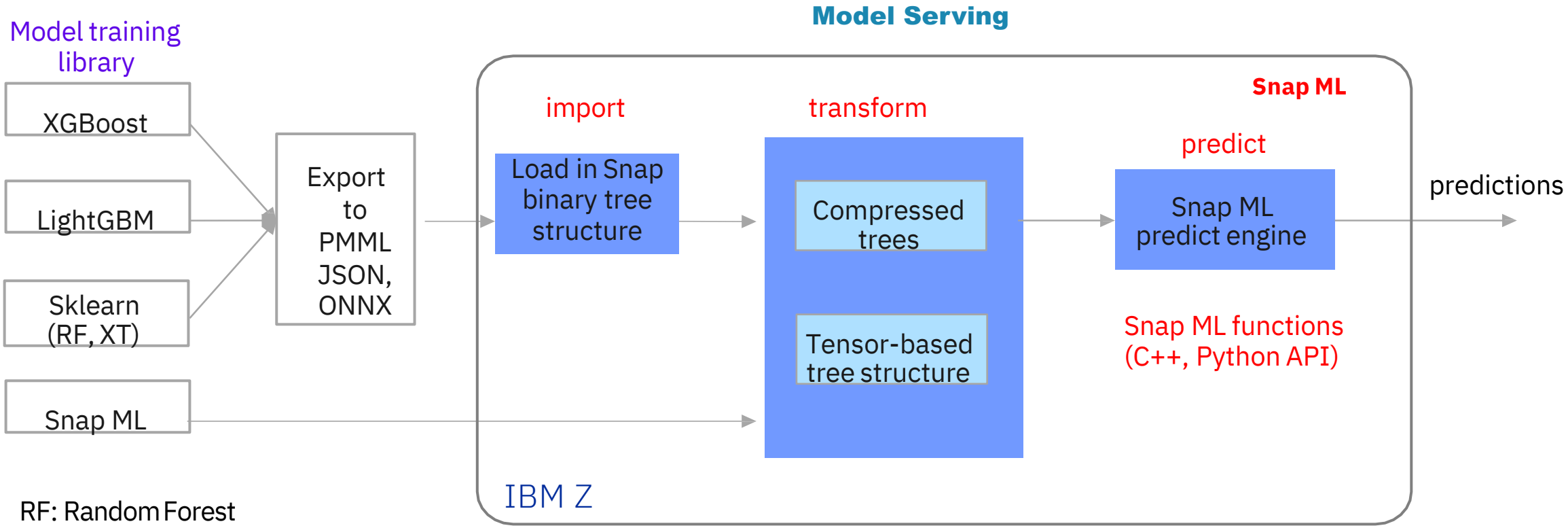
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.


IBM

IBM Snap ML for ML model execution



RF: Random Forest
XT: Extra Trees

Transformations are tailored to inference H/W:

Compressed Trees: Runs on the IBM Z CPU

Tensor-based Trees: Runs on the Integrated Accelerator for AI

Snap ML scoring pipeline is agnostic to the framework used to train the model

1. Save a model trained with scikit-learn, XGBoost, LightGBM into PMML/JSON/ONNX format
2. Import & transform the model
3. Score with Snap ML accelerated inference engine