October 2020

Ready for the Cloud with IBM z/OS Container Extensions

How zCX containerization brings z/OS resilience and security to Linux applications

The next generation of applications

Whether large or small, businesses see value in moving to the cloud. Yet 80% of enterprise workloads still need to be modernized into the next generation of cloud-native and microservices-based solutions.[1] A key challenge for most organizations is how to maintain security and high availability while modernizing business-critical workloads.[1]

For enterprises using IBM Z®, the ability to keep mission-critical workloads available and secure for business is paramount. Of the top 20 global banks, 75% have acquired an IBM z15™.[2] IBM Z is designed to run vast amounts of mission critical workloads efficiently and reliably. IBM z15 can run up to 19 billion encrypted transactions a day[3] - the equivalent of two years of Nasdaq stock market trading[4] within a single day. With forecasts estimating mobile transactions to be in the trillions per day by 2025, compute capability is essential for any enterprise. IBM Z is an enterprise-proven platform for fast, secure and reliable transaction computing for mobile and cloud environments.

z/OS benefits for Linux workloads

Linux® workloads can also benefit from the advantages of IBM Z and z/OS® attributes with containerization using IBM z/OS Container Extensions (zCX). zCX enable Linux applications to be deployed in a Docker container in z/OS without requiring a separately provisioned and managed Linux server.[5] One or multiple zCX instances or address spaces can be deployed within a z/OS system (see figure 1). zCX enables Linux applications to communicate with native z/OS applications over high speed virtual IP network for fast, reliable operations.

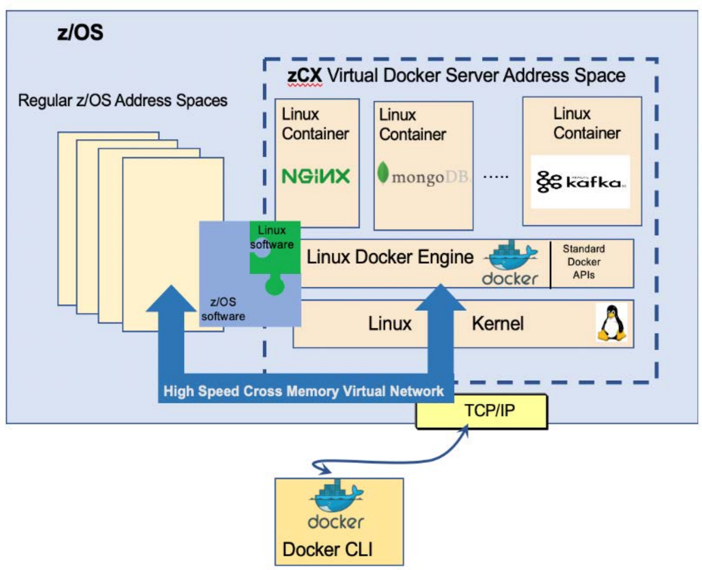

Figure 1: Overview of z/OS Container Extensions with multiple Docker containers within z/OS

[1] https://www.ibm.com/blogs/cloud-computing/2019/03/05/20-percent-cloud-transformation/
[2] Data is based on internal IBM z15 client shipped systems data up until and including 30th of June 2020. Top 20 banks according to Rel Banks 2019 list based on market capitalization - https://www.relbanks.com/worlds-top-banks/market-cap-2019
[3] https://www.ibm.com/it-infrastructure/z/capabilities/transaction-processing
[4] Based on over two years, or 787 days, of Nasdaq trading using average of 24 million trades per day during week of 9/22/2020 to 9/28/2020, http://www.nasdaqtrader.com/Trader.aspx?id=DailyMarketSummary
[5] https://www.youtube.com/watch?v=5o1r2EPMMUc

Since Linux applications deployed with zCX reside within z/OS, they inherit z/OS environment attributes, providing a highly secure, resilient and scalable environment. No root access is allowed, and access is defined by Docker interfaces. zCX enables direct support for Linux workloads that have an affinity to z/OS, expanding the z/OS software ecosystem.

Additionally, no IBM Z skills are required to manage Linux workloads within zCX. Linux developers and system administrators familiar with containerization can easily leverage IBM Z security and high availability capabilities through zCX.

Workload Manager support

Containers running within the zCX environment inherit the operational control of z/OS Workload Manager (WLM)[6] allowing access to WLM resource management and high throughput benefits, such as WLM monitoring to determine how many resources should be given to each container. WLM also allocates each virtual CPU as a dispatchable thread within the address space, enabling isolation of applications, different performance priorities and capping of resources that are allocated to workloads. WLM allows zCX to own and manage system, storage, network, and memory resources.

Enhanced disaster recovery options

zCX offers the ability to fully integrate containers into disaster recovery and planned outage scenarios.[7] Storage and disk access are provided through z/OS owned and managed Virtual Storage Access Method (VSAM) data sets, allowing zCX to leverage the latest I/O enhancements, host-based encryption, replication, and HyperSwap® technologies. Single points of failure can be eliminated due to zCX's ability to exploit Parallel Sysplex® clusters[8] and shared storage. If needed, a zCX instance can be seamlessly restarted in another system in the Sysplex. Cross memory services provide high-performance network access across z/OS applications and Linux Docker containers enable non-disruptive changes, failover, and dynamic movement of workloads. All these inherited z/OS and IBM Z infrastructure benefits provide Docker containers the best of both worlds - a secure and scalable hardware platform for Linux container workloads alongside traditional z/OS workloads.

zCX versus x86 Docker containers

To examine how zCX Docker containers perform on IBM Z compared to Docker containers with SSL on x86, two client simulated banking applications, a monolithic banking application, and a micro-services version of the same application, were

[6] https://www.ibm.com/it-infrastructure/z/zos-workload-management
[7] https://www.ibm.com/support/z-content-solutions/container-extensions/
[8] https://www.ibm.com/it-infrastructure/z/technologies/parallel-sysplex

evaluated on IBM Z and x86. JMeter was used to drive transactions in a 50,000 user simulated Linux retail banking environment accessing data from a z/OS Db2® database and transactions per second were measured to assess throughput.

Microservices application scenario

The microservices application was composed of seven microservices that connected to an IBM Db2 v12 database on z/OS via Java™ Database Connectivity (JDBC) and IBM WebSphere®. The database resided in one LPAR configured with three CPs, 12 zIIPs and 128GB of memory on a z15 T01. The z/OS Container Extensions environment that ran the microservices application was collocated in the same LPAR as the database (see figure 2).

For the x86 scenario, the microservice application ran on a 32-core Haswell x86 server with 512GB of memory running in three Docker containers with Secure Socket Layer, and accessed data from the z/OS Db2 database. IBM testing found that the Java microservices banking application simulation using z/OS data on a z15 T01 in an IBM z/OS Container Extensions environment could deliver on average 54% more transactions per second per core versus a compared x86 Docker container SSL environment using z/OS data.[9]

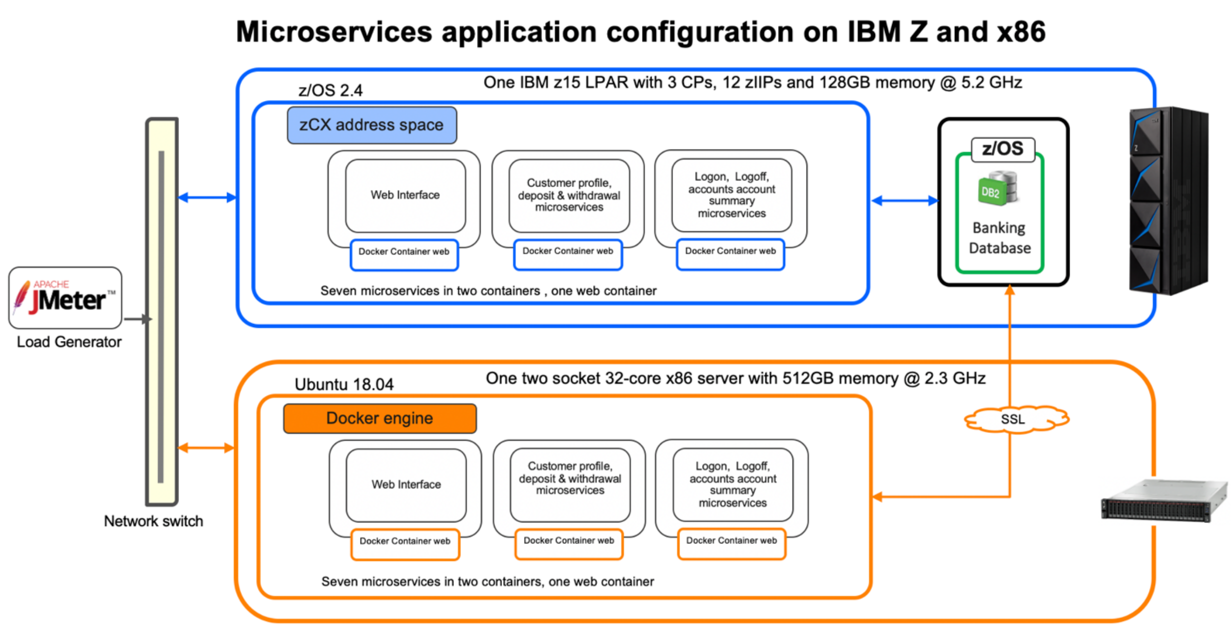

Figure 2: Overview of banking microservices application on IBM Z and x86

[9] Results were achieved using JMeter to drive transactions in a 50,000 user simulated Linux Java microservices retail banking environment in an IBM z/OS Container Extensions environment that was colocated with Db2 v12 in one LPAR configured with three CPs, 12 zIIP and 128GB of memory on a z15 T01 versus a 32-core Haswell x86 server with 512GB of memory running in three Docker containers with Secure Socket Layer to access data from the z/OS Db2 database on the z15 T01. Testing was performed in an IBM lab. Results will vary.

Monolithic application scenario

The monolithic version of the client simulated banking application was also configured with the application residing in the same LPAR as the IBM Db2 v12 database but with fewer zIIPs (six versus 12) and CPs (two versus three), and 128GB of memory on a z15 T01. The x86 environment was also configured on a 32-core Haswell x86 server with 512GB of memory but with only two (versus three) Docker containers with Secure Socket Layer to access data from the z/OS Db2 database on the z15 T01. IBM testing found that a Java monolithic banking application simulation using z/OS data on a z15 T01 in an IBM z/OS Container Extensions environment could deliver on average 44% more transactions per second per core versus a compared x86 Docker container SSL environment using z/OS data.[10]

Other findings

Both application scenarios found higher on average transactions per second on IBM Z[7,8] as a result of zCX inheritance of z/OS workload resource management and colocation with z/OS data (see figure 3).

In addition to high throughput benefits, the IBM Z environment provided encryption for the entire dataset. For both scenarios pervasive encryption was active with practically zero overhead[11] allowing applications to run in a secure environment without incremental charges for general processor computing.

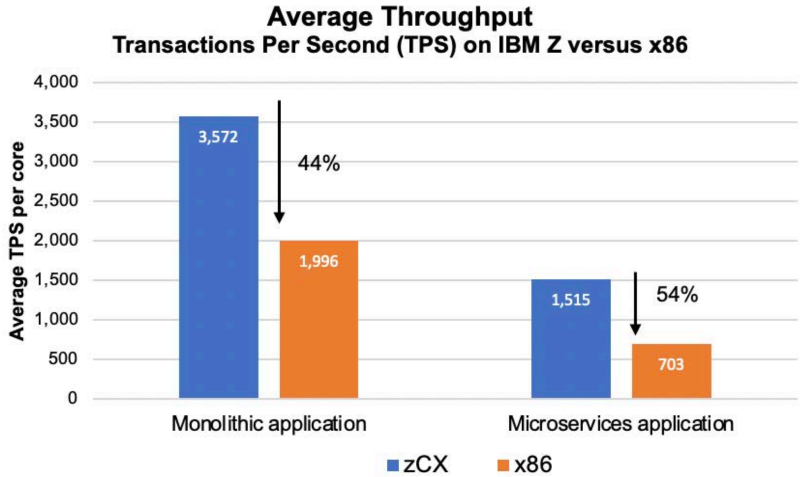

Figure 3: Average throughput per core comparison of monolithic and microservices applications deployed in z/OS with zCX versus Docker containers on x86 with SSL

From a total cost of ownership perspective, IBM Z provided savings. Acquisition and operating costs were examined to estimate IBM Z and x86 hardware, software, floor space, energy, networking and disaster recovery costs for a client environment requiring 8,000 transactions per second (TPS). In a three year cost comparison model, a Java microservices banking application using z/OS data on a z15 T01 in an IBM z/OS

[10] Results were achieved using JMeter to drive transactions in a 50,000 user simulated Linux Java monolithic banking environment in an IBM z/OS Container Extensions environment that was colocated with Db2 v12 in one LPAR configured with two CPs, six zIIP and 128GB of memory on a z15 T01 versus a 32-core Haswell x86 server with 512GB of memory running in two Docker containers with Secure Socket Layer to access data from the z/OS Db2 database on the z15 T01. Testing was performed in an IBM lab. Results will vary.

[11] IBM testing with an OLTP banking application simulation running on z15 T01 showed less than a 0.5% change in performance with encryption and compression enabled on a 3 GB database . Results were achieved using Rational Performance Tester (RPT) to drive transactions in a 50,000 user simulated retail banking environment using a CICS v5.4 application that was colocated with Db2 v12 in one LPAR configured with 8 CPs, 1 zIIP and 32GB of memory on a z15 T01. Encryption and compression of Db2 were enabled (3 GB size database) and disabled (3.7 GB size database) from the IBM Data Facility Storage Management Subsystem (DFSMS) for the tests. Unencrypted tests resulted in an average of 10,108 TPS per utilized processor while the encrypted test resulted in an average of 10,059 TPS per utilized processor for a difference of 0.49%. Testing was performed in an IBM lab. Results will vary.

Container Extensions environment delivers a cumulative 43% reduction in total ownership costs versus the compared x86 Docker container SSL environment using z/OS data.[12] And in a three year cost comparison model, a Java monolithic banking application using z/OS data on a z15 T01 in an IBM z/OS Container Extensions environment delivers a cumulative 69% reduction in total ownership costs versus the compared x86 Docker container SSL environment using z/OS data (see figure 4).[13]

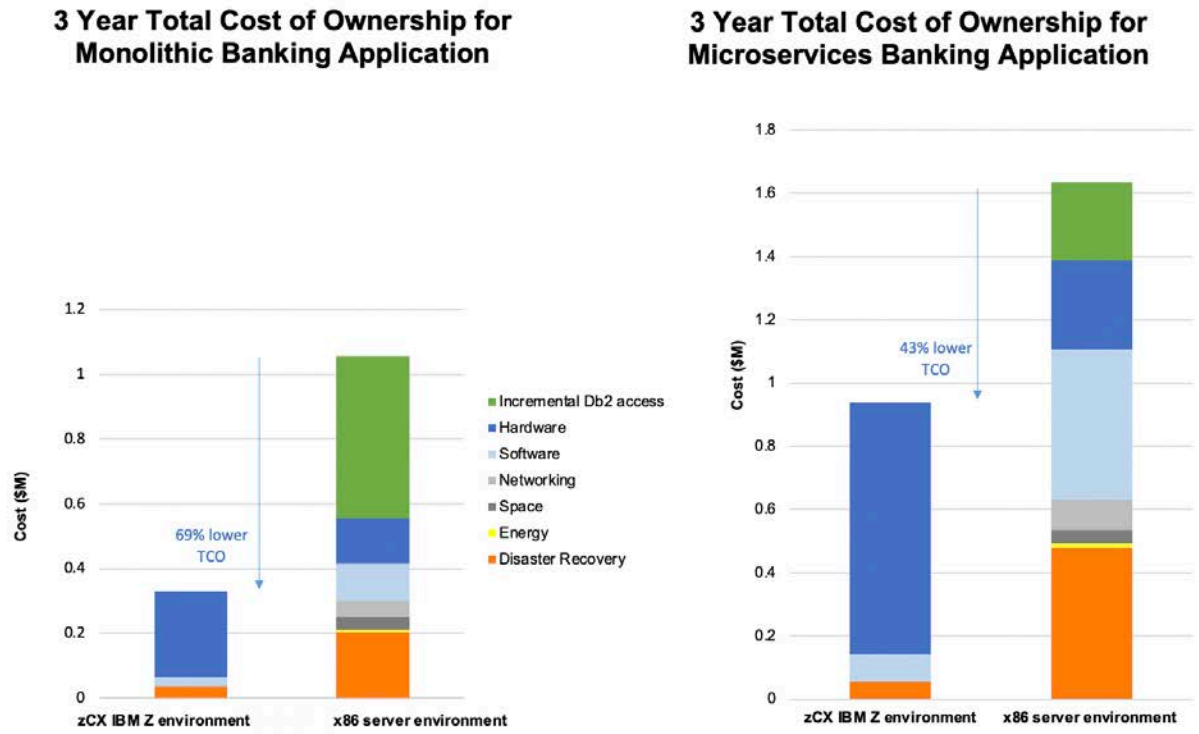

Figure 4: Three year total cost of ownership by cost category for client simulated monolithic and microservices banking applications deployed in zCX on IBM Z versus Docker containers on x86

Cost factors

While total cost of ownership for both types of applications was higher on x86, individual cost drivers varied.

[12] An IBM IT Economics total cost of ownership three year model was used to examine IBM Z and x86 hardware, software, floor space, energy, networking and disaster recovery costs for a Java microservices banking application simulation using z/OS data on a z15 T01 in an IBM z/OS Container Extensions environment versus a x86 Docker container SSL environment using z/OS data. The IBM z/OS Container Extensions environment was colocated with Db2 v12 in one LPAR configured with three CPs, 12 zIIPs and 128GB of memory on a z15 T01 configured to run an average of 1,515 TPS per core in the IBM z/OS Container Extensions environment. The x86 Java microservices application was configured to run an average of 702 TPS per core on a 32-core Haswell x86 server with 512GB of memory in three Docker containers with Secure Socket Layer. Both IBM Z and x86 scenarios accessed data from the z/OS Db2 database on the z15 T01. The IBM Z environment used IBM z/OS Container Extensions in one z/OS LPAR and the x86 environment used Docker containers and virtualization software. The cost model assumes no growth for an 8,000 TPS user scenario production environment within an existing 10,000 MIPS customer environment over three years. An additional 100% of production capacity was projected for development, test, and QA environments, and an additional 100% of production capacity was projected on a remote site for disaster recovery for x86. For IBM Z an additional 25% of production capacity was projected to provision development, test, and QA environments on the same system, and an additional 100% of production capacity was projected on a remote site for disaster recovery. Both cost cases include Ubuntu, WebSphere Liberty, and Tivoli system management tools. The x86 case includes the incremental cost of x86 remote access to Db2 on z/OS. IBM Z SW, central processor and zIIPs costs, where applicable, are based on U.S. prices as of February 2020 from client data in IT Economics assessments. x86 distributed SW is also based on U.S. prices from client data with a 20% discount. x86 hardware pricing is based on IBM analysis of U.S. prices as of 06/10/2020 from IDC with a 30% discount. Data center costs (floor space, energy, networking) for x86 are based client data in IT Economics assessments. For additional information on the cost model contact the IBM IT Economics team, IT.Economics@us.ibm.com..
[13] An IBM IT Economics total cost of ownership three year model was used to examine IBM Z and x86 hardware, software, floor space, energy, networking and disaster recovery costs for a Java monolithic banking application simulation using z/OS data on a z15 T01 in an IBM z/OS Container Extensions environment versus a x86 Docker container SSL environment using z/OS data. The IBM z/OS Container Extensions environment was colocated with Db2 v12 in one LPAR configured with two CPs, six zIIPs and 128GB of memory on a z15 T01 configured to run an average of 3,572 TPS per core in the IBM z/OS Container Extensions environment. The x86 Java monolithic application was configured to run an average of 1,996 TPS per core on a 32-core Haswell x86 server with 512GB of memory in two Docker containers with Secure Socket Layer. Both IBM Z and x86 scenarios accessed data from the z/OS Db2 database on the z15 T01. The IBM Z environment used IBM z/OS Container Extensions in one z/OS LPAR and the x86 environment used Docker containers and virtualization software. The cost model assumes no growth for an 8,000 TPS user scenario production environment within an existing 10,000 MIPS customer environment over three years. An additional 100% of production capacity was projected for development, test, and QA environments, and an additional 100% of production capacity was projected on a remote site for disaster recovery for x86. For IBM Z an additional 25% of production capacity was projected to provision development, test, and QA environments on the same system, and an additional 100% of production capacity was projected on a remote site for disaster recovery. Both cost cases include Ubuntu, WebSphere Liberty, and Tivoli system management tools. The x86 case includes the incremental cost of x86 remote access to Db2 on z/OS. IBM Z SW, central processor and zIIPs costs, where applicable, are based on U.S. prices as of February 2020 from client data in IT Economics assessments. x86 distributed SW is also based on U.S. prices from client data with a 20% discount. x86 hardware pricing is based on IBM analysis of U.S. prices as of 06/10/2020 from IDC with a 30% discount. Data center costs (floor space, energy, networking) for x86 are based client data in IT Economics assessments. For additional information on the cost model contact the IBM IT Economics team, IT.Economics@us.ibm.com.

Incremental Db2 access

Both x86 configurations required secure SSL access to the Db2 z/OS database which resulted in a substantial increase in Db2 MIPS. Incremental database access costs were even greater for the monolithic application scenario in which zCX has a higher percent of zIIPs eligible workload, resulting in higher remote access costs for the x86 microservices application scenario.

Software

Both zCX scenarios had lower software costs. Since the zCX environments required fewer application cores than x86, fewer per core software licenses were needed for IBM WebSphere Liberty and system management tools such as IBM® Workload Automation, resulting in substantially lower zCX software costs. Additionally, Linux is provided with the Docker containers in zCX so no Linux costs were factored into the zCX TCOs. Conversely, Linux support was an additional software cost for the x86 TCOs.

Disaster recovery

Both x86 scenarios incurred significant disaster recovery costs. In order to replicate the production environment for disaster recovery, a second x86 server with fully configured software licensing was required. In contrast, the cost of the disaster recovery environment for the IBM Z scenarios was lower. Only incremental zIIP processors configured with Capacity Back Up (CBU) on the existing IBM Z disaster recovery system were needed to provide disaster recovery support for the zCX workloads.

Why zCX can help your organization

A key challenge for most organizations is how to maintain security and high availability while modernizing business-critical workloads. Using IBM Z with IBM Container Extensions creates a common operating model across DevOps and mitigates security concerns. zCX requires no z/OS skills to develop and deploy Docker containers, making it easy to use for non z/OS users, while achieving on average 44% - 54% more transactions per second per core compared x86. zCX can also provide a 43% - 69% reduction in total cost of ownership.

If your organization uses Linux and z/OS based applications and seeks to modernize to cloud-native and microservices-based solutions, z/OS Container Extensions can facilitate the journey to your next generation of applications. Contact the IBM IT Economics team at IT.Economics@us.ibm.com to learn more about zCX and how it can modernize and lower costs for your applications.

Author
Ruviano Martinez is the Program Director of the IBM IT Economic Consulting and Research organization. Ruviano manages the research team to create client scenarios and analyze the value of IBM technology versus alternative solutions. At the core of his methodology is understanding both strategic and tactical approaches to achieve efficient, cost-effective solutions. Ruviano consults across a range of industries, in particular financial services and manufacturing, to help clients find solutions that achieve their business objectives.

Contributors
Rakesh Krishnakumar
Senior Software and Systems Architect, IBM IT Economic Consulting and Research
e-mail: rkrishnm@sg.ibm.com

J.C. Yao, Ph.D.
Chief Strategist, IBM IT Economic Consulting and Research
e-mail: jcy@us.ibm.com

Tom Ambrosio
Competitive Research Manager, IBM IT Economic Consulting and Research
e-mail: tambrosi@us.ibm.com

Bill Lamastro
Infrastructure Architect, IBM IT Economic Consulting and Research
e-mail: wglamas@us.ibm.com

Susan Proietti Conti
Program Director, Worldwide IBM IT Economics Consulting & Research
IBM Executive Project Manager, PMP®
e-mail: sconti@us.ibm.com

90035790USEN-02