Solution White Paper

Manage Java for Optimal Performance on IBM® z/OS®

Gain insight into how Java impacts mainframe performance with MainView for Java Environments

Table of Contents

Executive Summary

In order to meet the faster, bigger, and always available demands of digital business, enterprises have turned to Java on IBM® z/OS® in an attempt to reduce costs, increase productivity, and respond faster to the needs of the business. Java applications can be developed and deployed quickly and cost effectively. According to a recent BMC survey of more than 1,200 mainframe professionals, **93 percent of respondents say that Java usage on the mainframe is growing or steady and 70 percent say their Java growth is driven by new applications written in Java.**[1]

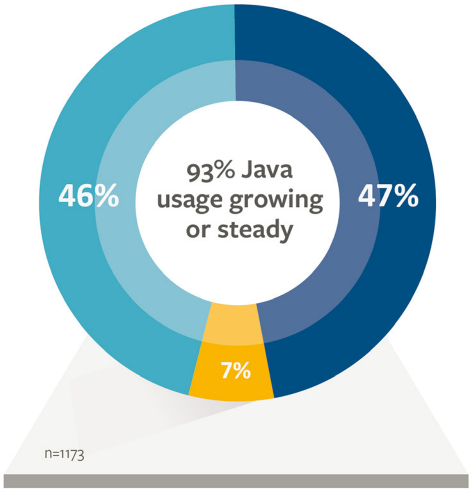
Despite the improved speed and agility, managing Java in mainframe environments is challenging. Java is platform agnostic, runs in its own environment called a Java Virtual Machine (JVM), and manages its own resources without knowledge of other workloads. In order to meet fast-paced business needs, application development teams can develop and deploy JVMs without involving IT operations. IT may be unaware of the true scope of JVMs, or even their existence, until a JVM negatively impacts the performance and availability of another workload.

To tackle the challenges that Java presents, IT teams need an integrated solution with the visibility to find and resolve Java performance issues before they affect the business. MainView for Java Environments is an integrated performance management solution that provides insight into how Java is consuming mainframe resources and their impact on mainframe application performance. This paper discusses how these distinct capabilities ensure IT is able to meet the demands of digital business:

- Decreased mean time to repair (MTTR)
- Reduced costs
- Increased end user satisfaction
- Enhanced staff productivity

New and Rewritten Apps Fuel Java Growth

Q. How has Java on the mainframe usage changed at your site?

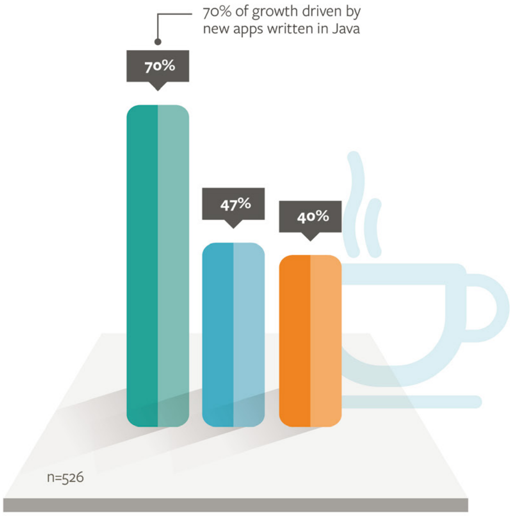


46% say Java usage is increasing 10% or more

47% say Java usage is either flat or steady

7% say Java usage is shrinking

Q. Why is your Java on the mainframe usage increasing?
Asked to those responding Java usage increasing.



● New apps in Java ● Rewriting exising apps ● Existing app growth

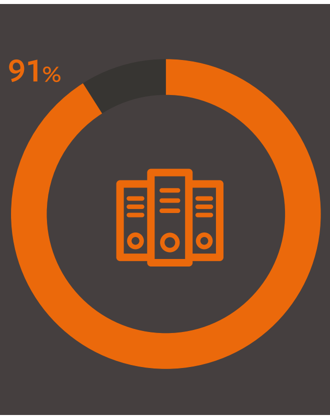
[1] 2015 BMC Annual Mainframe Research Results

THE IMPACT OF JAVA ON THE MAINFRAME

It's critical to monitor the impact of Java

Digital business has dramatically changed the way people interact with technology. Consumers have high expectations for services and demand applications and devices that know who they are, where they are, and what they want. Businesses must offer instantaneous personalized services, anytime and anywhere. Many businesses are meeting their customers' expectations with high-performing, always-available applications. This demand for personalized service has resulted in the need for the rapid deployment of new applications. Ninety-one percent of all new applications require a mainframe to complete their transactions, making the mainframe a strategic enabler of digital business.[2]

Mobile technologies and analytics generate high volumes of transactions as consumers use applications in new and expanding ways—to make purchases, order services, get information, compare prices, research options, check statuses, and receive notifications and alerts. Consumers also expect instantaneous responses to their digital requests—even if the interaction does not result in more revenue for the business.



Percentage of all new applications that require a mainframe to complete their transactions

Balancing the demands of unpredictable workloads with the need to develop and deploy applications faster is a huge challenge for IT organizations that are also being pressured to manage costs and optimize productivity. They are increasingly turning to Java for application development and deployment speed and agility, and to reduce operating costs through z Systems™ Integrated Information Processor (zIIP) offloading.

But Java applications have their own challenges. They run in Java virtual machines (JVMs) that manage their own resources and can **consume memory and other system resources**. This behavior can lead to resource constraints and issues that can cause dramatic performance degradation and limit availability of mission-critical applications. Each JVM allocates resources based on how it is configured, without knowledge of or consideration to what resources are available or what other applications may need them. Mainframes can have thousands of JVMs running at any given time, and they can execute anywhere across mainframe subsystems: IBM® CICS®, IBM® DB2®, IBM® IMS™, and batch. Java applications communicate with existing application code, so **a problem in a single JVM can impact multiple subsystems and applications**. When a JVM runs out of memory, it will automatically start a garbage collection (GC) process. This removes any data objects that are no longer needed from the JVM data area, freeing up memory for further use. This GC process consumes CPU resources, and an inefficient garbage collection process can cause transactions to run extremely slowly or even be "paused" while the GC process completes.

Without a mainframe systems management solution that views Java along with all the other technologies, IT doesn't have insight into how JVMs are affecting other applications and transactions.

Today, very few mainframe systems management solutions include the ability to discover and manage Java environments. In the always-available, always-on culture, a Java application that negatively impacts your customer experience could result in customers taking their business elsewhere.

That's why it is so important to have **powerful, proactive systems management that can address the entire mainframe technology stack**, including Java.

[2] Dennis, Valerie, IBM Systems Magazine - Mainframe Edition, October 2014.

REQUIREMENTS FOR MANAGING JAVA ON THE MAINFRAME

MainView for Java Environments extends the capabilities of MainView so that IT can manage the entire mainframe environment from one console. It is a simple, intelligent, and powerful solution that includes metrics and analytics capabilities to automatically discover JVMs and understand how they are impacting other workloads. It enables IT to proactively manage Java on the mainframe to find and resolve performance issues before they impact business operations. This solution will also show you how JVMs are utilizing zIIP specialty engines, so that IT can optimize available zIIP capacity to reduce costs.

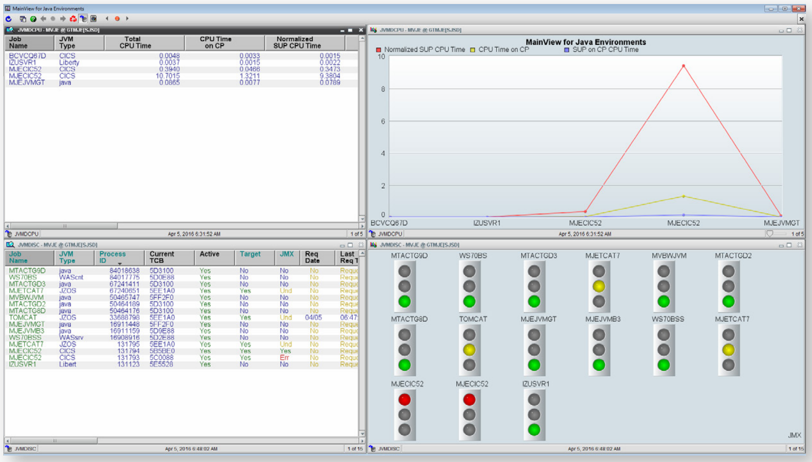
MainView for Java Environments provides visibility into JVMs across the mainframe environment, including:

- **Discovery:** automatically discovers JVMs across LPARS and subsystems
- **Monitoring:** provides visibility into real-time metrics for the z/OS Java runtime environment such as garbage collection effectiveness, memory usage, and heap trends
- **Dashboards:** creates views of Java runtime metrics to meet the needs of any audience
- **Single System Image:** provides a consolidated view of all mainframe subsystems to efficiently discover and analyze how Java is impacting other resources
- **Exception Analysis:** proactively monitors for abnormally running JVMs

Discovery

JVMs create their own environment and can run virtually anywhere on the mainframe. They can be deployed rapidly, sometimes without the assistance or knowledge of the data center. Dozens, hundreds, or even thousands of JVMs may be running at any given time. Without a systems management solution that includes Java environments, IT may not even be aware that JVMs are running in their data center.

FIGURE 1: With the click of a button, mainframe IT teams can now automatically discover JVMs across their environment.



JVMDCPU - MVJE @ GTMJE[SJSD]

Job Name	JVM Type	Total CPU Time	CPU Time on CP	Normalized SUP CPU Time
BCVCQ67D	CICS	0.0048	0.0033	0.0015
IZUSVR1	Liberty	0.0037	0.0015	0.0022
MJECIC52	CICS	0.3940	0.0466	0.3473
MJECIC52	CICS	10.7015	1.3211	9.3804
MJEJVMGT	java	0.0865	0.0077	0.0789

JVMDCPU Apr 5, 2016 6:31:52 AM 1 of 5

JVMDISC - MVJE @ GTMJE[SJSD]

Job Name	JVM Type	Process ID	Current TCB	Active	Target	JMX	Req Date	Last Req T
MTACTG9D	java	84018638	5D3100	Yes	No	No	No	Reque
WS70BS	WAScnt	84017775	5D0E88	Yes	No	No	No	Reque
MTACTGD3	java	67241411	5D3100	Yes	No	No	No	Reque
MJETCAT7	JZOS	67240651	5EE1A0	Yes	Yes	Und	No	Reque
MVBWJVM	java	50465747	5FF2F0	Yes	No	No	No	Reque
MTACTGD2	java	50464189	5D3100	Yes	No	No	No	Reque
MTACTG8D	java	50464176	5D3100	Yes	No	No	No	Reque
TOMCAT	JZOS	33688798	5EE1A0	Yes	Yes	Und	04/05	06:47
MJEJVMGT	java	16911448	5FF2F0	Yes	No	No	No	Reque
MJEJVMB3	java	16911159	5D9E88	Yes	No	No	No	Reque
WS70BSS	WASsrv	16908916	5D2E88	Yes	No	No	No	Reque
MJETCAT7	JZOS	131795	5EE1A0	Yes	Yes	Und	No	Reque
MJECIC52	CICS	131794	5B5BE0	Yes	Yes	Yes	No	Reque
MJECIC52	CICS	131793	5C0088	Yes	Yes	Err	No	Reque
IZUSVR1	Libert	131123	5E5528	Yes	No	No	No	Reque

JVMDISC Apr 5, 2016 6:48:02 AM 1 of 15

Monitoring

MainView for Java Environments is integrated with MainView, which enables IT to see the entire environment in a single graphical console. Consolidation simplifies the mainframe's complexity, so you can view all aspects of your systems in one place. You can also compare Java's real-time performance with historical trends.

MainView's out-of-the-box single-system views have the same look and feel across your mainframe systems. You can view key metrics such as process ID, CPU utilization, zIIP engines usage, and other activities on a simple, single display to understand Java's impact on application availability and performance. Using MainView's advanced alarming and alerting, IT can find and resolve problems before they impact service, customer experience, and eventually your business. MainView for Java Environments also leverages MainView's dynamic hyperlink capability combined with a single system image that enables the user to follow a problem wherever it leads across different subsystems. Problems are identified rapidly and the time to restore service is reduced. In a Tech Validate survey of over 130 MainView customers, 78 percent reported that MainView improved the time to detect and diagnose problems.[3]

[3] TechValidate survey of 134 users of BMC MainView, May 2015.

In a Tech Validate survey of over 130 MainView customers, 78 percent reported that MainView improved the time to detect and diagnose problems.[4]

Smart Dashboards

Dashboards visualize the performance of your mission-critical applications using easy-to-create customized views of Java runtime metrics. With a mouse click, you can create a custom dashboard for a variety of uses, which includes sharing with a line of business (LOB) or other key stakeholders. Custom views provide a clear understanding of how Java interacts with subsystems and enables swift recognition of performance issues.

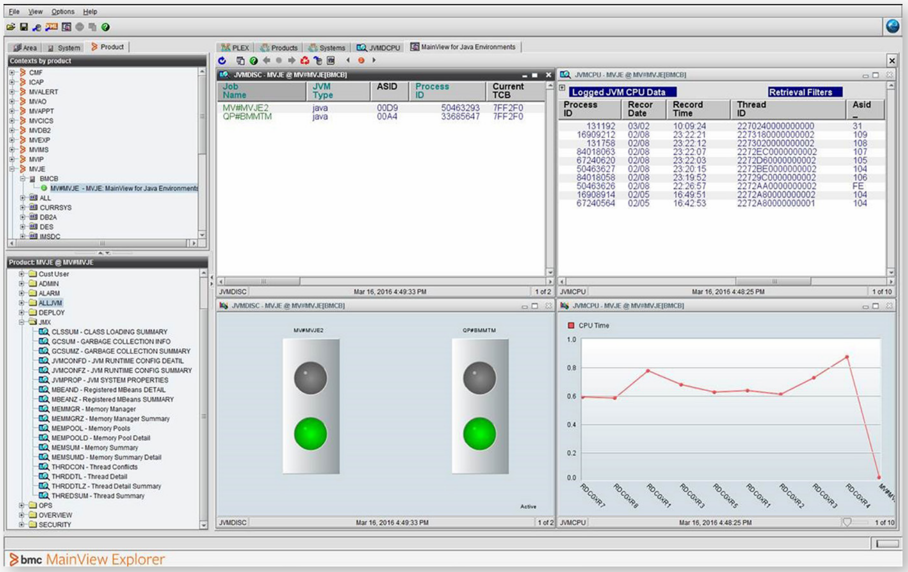
Single System Image

With the ability to monitor all JVMs from a single view, you can gain insight into Java processes to determine if they are running in batch, IBM® CICS®, IBM® IMS™, IBM® DB2®, IBM® WebSphere® Application Server Liberty Core, WebSphere® Application Server, or elsewhere. You can see the impact that Java is having on applications, performance, and availability so that you can take corrective action (redesign processes or reallocate resources, etc.).

Examples include:

- Determine if work is being offloaded to zIIP engines
- Identify a looping JVM that is consuming excessive resources
- Understand where processing is inefficient

FIGURE 2: With a consolidated view (via single system image), IT can quickly understand how Java runtime environments are utilizing resources.


File View Options Help
Area System Product
Contexts by product
CMF
ICAP
MVALERT
MVAO
MVAPPT
MVCICS
MVDB2
MVEXP
MVIMS
MVIP
MVJE
BMCB
MV#MVJE - MVJE: MainView for Java Environments
ALL
CURRSYS
DB2A
DES
IMSDC
Product: MVJE @ MV#MVJE
Cust User
ADMIN
ALARM
ALLJVM
DEPLOY
JMX
CLSSUM - CLASS LOADING SUMMARY
GCSUM - GARBAGE COLLECTION INFO
GCSUMZ - GARBAGE COLLECTION SUMMARY
JVMCONFD - JVM RUNTIME CONFIG DEATIL
JVMCONFZ - JVM RUNTIME CONFIG SUMMARY
JVMPROP - JVM SYSTEM PROPERTIES
MBEAND - Registered MBeans DETAIL
MBEANZ - Registered MBeans SUMMARY
MEMMGR - Memory Manager
MEMMGRZ - Memory Manager Summary
MEMPOOL - Memory Pools
MEMPOOLD - Memory Pool Detail
MEMSUM - Memory Summary
MEMSUMD - Memory Summary Detail
THRDCON - Thread Conflicts
THRDDTL - Thread Detail
THRDDTLZ - Thread Detail Summary
THREDSUM - Thread Summary
OPS
OVERVIEW
SECURITY
PLEX
Products
Systems
JVMDCPU
MainView for Java Environments
JVMDISC - MVJE @ MV#MVJE[BMCB]
MV#MVJE2
QP#BMMTM
Active
JVMDISC
Mar 16, 2016 4:49:33 PM
1 of 2
JVMCPU - MVJE @ MV#MVJE[BMCB]
CPU Time
1.0
0.8
0.6
0.4
0.2
0.0
RDCDIR7
RDCDIR8
RDCDIR1
RDCDIR3
RDCDIR5
RDCDIR1
RDCDIR2
RDCDIR3
RDCDIR4
JVMCPU
Mar 16, 2016 4:48:25 PM
1 of 10
bmc MainView Explorer

JVMDISC - MVJE @ MV#MVJE[BMCB]

Job Name	JVM Type	ASID	Process ID	Current TCB
MV#MVJE2	java	00D9	50463293	7FF2F0
QP#BMMTM	java	00A4	33685647	7FF2F0

JVMDISC Mar 16, 2016 4:49:33 PM 1 of 2

JVMCPU - MVJE @ MV#MVJE[BMCB]

Logged JVM CPU Data

Retrieval Filters

Process ID	Recor Date	Record Time	Thread ID	Asid
131192	03/02	10:09:24	2270240000000000	31
16909212	02/08	23:22:21	2273180000000002	109
131758	02/08	23:22:12	2273020000000002	108
84018063	02/08	23:22:07	2272EC0000000002	107
67240620	02/08	23:22:03	2272D60000000002	105
50463627	02/08	23:20:15	2272BE0000000002	104
84018058	02/08	23:19:52	22729C0000000002	106
50463626	02/08	22:26:57	2272AA0000000002	FE
16908914	02/05	16:49:51	2272A80000000002	104
67240564	02/05	16:42:53	2272A80000000001	104

JVMCPU Mar 16, 2016 4:48:25 PM 1 of 10

If you don't know what's happening in your environment, you won't be able to diagnose and resolve problems that are impacting applications performance and availability, so this level of visibility is critical. Depending on the complexity of the problem, it could take hours, days, or even weeks for resolution.

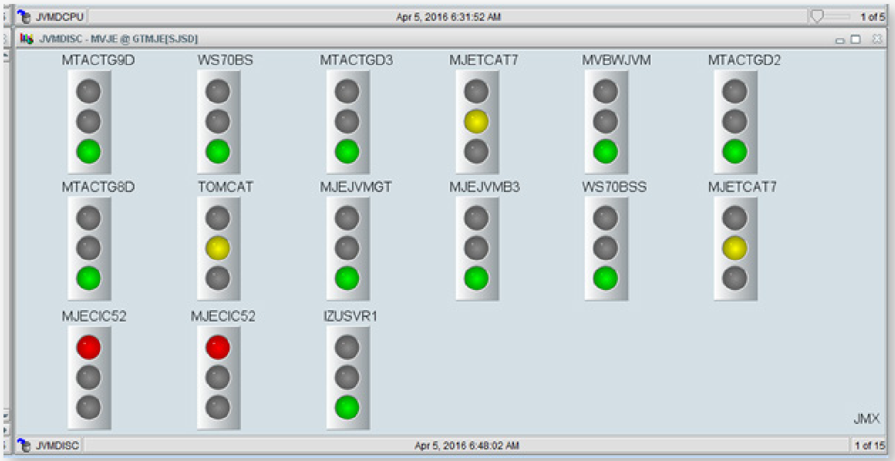
[4] TechValidate survey of 134 users of BMC MainView, May 2015.

Exception Analysis – Health Assessment

With MainView for Java Environments, IT can set proactive alarms on important metrics based on business criteria and service level agreements (SLAs). Alarms can be set on a variety of key metrics, like memory (heap) usage and the effectiveness of memory management done via garbage collection. Alarms can be set up ahead of time or in real-time. For example, if MainView for Java Environments highlights a problem with a JVM thread, an alarm can be set from the problem screen with the click of a button.

Criteria for alarms and notifications include identifying CPU usage by a JVM or a thread. Using a hyperlink in MainView for Java Environments, IT can follow a problem thread to CICS, IMS, DB2, and other subsystems to quickly find and resolve the cause of a poorly performing application. This enables IT to handle the volatility and increased velocity that are direct by-products of digital business.

FIGURE 3: An easy-to-understand screen displays the health of JVMs.



Cost-Effective Java Management

MainView for Java Environments can help IT reduce costs in two ways. MainView provides efficiency that reduces the cost of monitoring by up to 50 percent. Java workloads can run on lower-cost zIIPs, but that doesn't happen automatically. Monitoring zIIP offloading is imperative; if zIIP processing capacity is not available, Java will run on the more expensive general processor CPU and this situation could increase mainframe monthly license charge (MLC) costs for IBM software. Managing MLC costs is critical because it can represent 30 percent of a typical mainframe budget. By providing visibility into how Java affects workloads, IT can determine where the workloads are running and make sure they don't impact MLC costs.

CONCLUSION

The demands of digital business have changed the rules in how applications are developed, deployed, and monitored and has led to the growing use of Java on the mainframe. Java enables organizations to modernize applications, address skills concerns, and create platform agility. Because Java manages and consumes resources in a different way and without awareness or consideration of traditional applications running on the mainframe or other JVMs, Java can adversely affect performance across your environment.

MainView for Java Environments provides insight into how Java is consuming resources and affecting application performance, so you can decrease your MTTR and costs and increase staff productivity. This innovative solution will help IT manage the bigger, faster, unpredictable demands hitting the mainframe today—and help IT prepare for the unknown demands of tomorrow.

FOR MORE INFORMATION

To learn more about MainView for Java Environments, please visit **bmc.com/mainviewjava**

BMC is a global leader in innovative software solutions that enable businesses to transform into digital enterprises for the ultimate competitive advantage. Our Digital Enterprise Management solutions are designed to fast track digital business from mainframe to mobile to cloud and beyond.

BMC – Bring IT to Life

BMC digital IT transforms 82 percent of the Fortune 500.

480315