

# Application Development Tools and Virtualization Advantages for Linux on System z

*Timothy Hahn  
Distinguished Engineer, Rational  
IBM, Raleigh*

*Laurence England  
STSM – Application Development Tool Architect  
IBM, Silicon Valley*

*Gerald Hosch  
Marketing Linux on IBM System z  
IBM, Boeblingen*

*This paper describes application development tools for Linux on IBM System z and the strengths of the flexible virtualization environment available for Linux on IBM System z.*

**Table of Contents**

- Introduction ..... 2
- An emerging model of development..... 3
- Source location ..... 4
  - IBM Tools*..... 5
  - Open Source Tools*..... 5
- Edit Capabilities ..... 5
  - IBM tools*..... 6
  - Open Source tools*..... 6
- Building/Constructing the Application ..... 6
  - IBM Tools*..... 7
  - Open Source Tools*..... 7
- Testing the Application ..... 7
  - IBM Tools*..... 7
  - Open Source Tools*..... 8
- Strengths of the virtual Linux environment on IBM System z ..... 8
  - Software developer requirements* ..... 8
  - Matching the development requirements* ..... 8
  - Leveraging the advantages* ..... 9
- Conclusion ..... 10
- Resources ..... 12
- About the authors ..... 12

# Application Development Tools and Virtualization Advantages for Linux on System z

## Introduction

Linux<sup>®</sup> is used increasingly for application development. The reason for this growth is threefold. First, the availability of integrated development environments that run on Linux has been greatly expanded. Second, development on Linux is considerably easier and safer than development on other operating environments, when the deployment environment is Linux. Third, Linux provides a secure, reliable and flexible platform for software development.<sup>1</sup>

IBM provides a sufficiently comprehensive, well-integrated development platform for Linux with the IBM Rational<sup>®</sup> Software Development Platform. IBM also provides a highly virtualized, very flexible and Dynamic Infrastructure<sup>®</sup> with the IBM System z<sup>®</sup> virtualization capabilities provided by z/VM<sup>®</sup>.

When considering application development tools for Linux on IBM System z, one must consider a number of aspects, including:

- Requirements and analysis tools
- Design, development (authorship, editing) and construction (build) tools
- Testing and software quality tools
- Software configuration management tools

When creating an overall development environment for application developers, there are options concerning where certain development tasks can be performed. At a high level, the questions to be answered are:

- Where shall I store my source code?
- Where and how shall I create and edit the source code elements for my application?
- Where and how should I perform a build to produce an executable? This applies for both individual developers as well as team-level builds, which accumulate a number of changes from a group of developers.
- Where and how should I perform test, and where is my test data? This also includes questions of using interactive debuggers by developers who are unit testing their changes, and functional and system testers who are testing a team-level build using a more comprehensive set of tests.

This paper will describe some of the tools commercially and freely available for accomplishing these tasks, with applications running on Linux-based systems as the target runtime environment, and the strengths of the very flexible virtualization environment available for Linux on IBM System z.

# Application Development Tools and Virtualization Advantages for Linux on System z

## An emerging model of development

Today, many organizations use a complex set of systems to provide their computing infrastructure. Linux utilization in these systems, both as desktop/workstation systems and as “enterprise server” systems on IBM System z, is increasing. Indeed, Linux has been used as a host environment for Web-style applications, be they Apache-based, IBM WebSphere®-based, or based on other Web application runtime environments, for quite some time now.

Aside from user-interactive Web-based applications, Linux on IBM System z systems enjoys increased utilization as:

- Data serving and data warehousing servers (for example, Cognos® BI, DB2®, Oracle, Informix®)
- Application servers in several areas (for example, SAP for ERP, Domino® for e-mail and collaboration)
- Network infrastructure servers (for example, DNS, FTP, NFS, firewall servers).

With the wide acceptance of Linux-based systems in the IT organization, there is an increasing requirement for application development tools for Linux. These tools must support and enable application developers to design, write, manage, compile/link, debug, and test applications that are meant to run on systems that are physically or virtually separated from the systems that developers are using. This is even more apparent as virtual system images and so-called “cloud computing” systems are considered.

Application development now requires developers to interact with potentially separate systems for source configuration management (SCM) support, build systems, and test systems (see Figure 1).

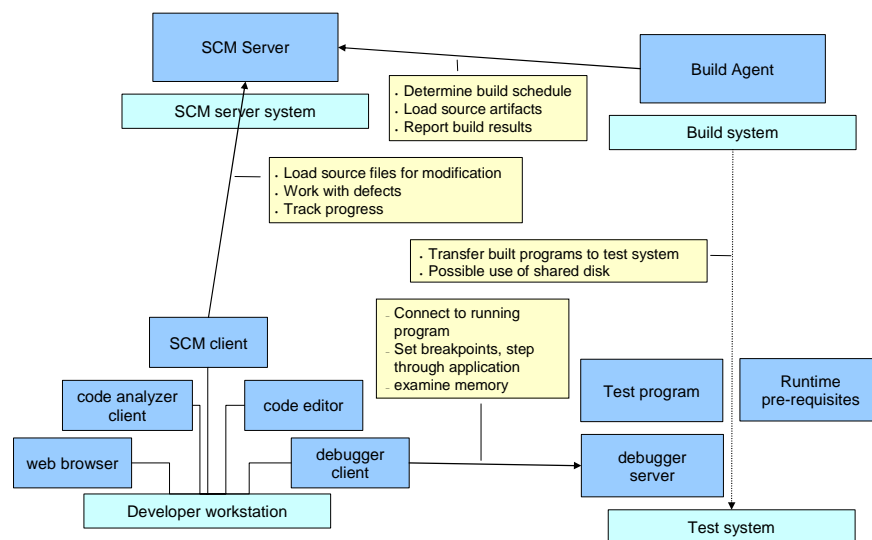


Figure 1. Set of systems used for a development environment

## Application Development Tools and Virtualization Advantages for Linux on System z

To develop or modify an application, the developer and his colleagues must perform these tasks:

- Connect to a SCM server to obtain an editable version of the source code for the project.
- Analyze, edit, and otherwise work on the source code that must be modified for the feature that they are adding or defect that they are correcting.
- Schedule a build of the necessary source code and affected application binaries by invoking or calling for a build to be performed on a system where the necessary toolkits and runtime libraries are present. Note that differences in target system architecture may require builds to be performed on systems other than the desktop/workstation system. While cross-compilers may be used in some cases, this is not always available or appropriate.
- Move, copy, or deploy the built artifacts onto a system that can be used for “unit testing”/debug of the changes that the application developer has made. This can involve yet another system in a distributed environment, separate from the development and build systems.
- Connect to an SCM server to check-in changes to the source code.

Each of these functions: source configuration management, edit/analyze, build and debug are likely to be performed on separate systems. For larger applications, it is typical for multiple systems to be involved for both test and production environments, with the application component relying on services running on other test systems in order for even unit testing/debug to be done. The flexible virtual Linux environment on IBM System z provides major advantages, described later in this document.

It should be noted here that when using physically distributed and dispersed servers, there is a heavy reliance on network connectivity between the various elements in the environment. Depending on the products and technologies used and the locations of the systems being employed, often network infrastructure teams need to be involved to ensure that the required level of connectivity is maintained. By using virtual Linux servers, this can be avoided. These servers can benefit from the very fast and secure internal network of the IBM System z servers.

### Source location

Application artifacts may be held in a source configuration management (SCM) system that resides on a remote system possibly on a different platform.

Considerations such as file encoding, transfer time to move the artifacts from the repository to the edit location or the build location, and end-of-line markers must be addressed.

Depending on where the source code resides, it might need to be moved between systems in order to edit and to build the application. This can be an impact to network bandwidth and reliability.

Source encoding may be a consideration for the SCM. For example, is the SCM capable of managing source code that contains Bi-Directional (BiDi) characters or Double-Byte Character Set (DBCS)? Is it possible to store EBCDIC encoded source in a base-plus-delta format? What

## Application Development Tools and Virtualization Advantages for Linux on System z

file encoding transforms will be performed by the SCM? By the editor? Unicode based SCMs provide a significant amount of versatility.

### *IBM Tools*

*IBM's Rational ClearCase*<sup>®</sup> is a client/server, enterprise-level, source configuration management tool. The server side provides a repository of versioned artifacts and is supported on Linux on IBM System z.

Closely coupled to IBM Rational ClearCase is the defect tracking product *IBM Rational ClearQuest*<sup>®</sup>, which is capable of using *IBM DB2 9<sup>2</sup>* on Linux on IBM System z.

*IBM's Rational Team Concert*<sup>™</sup> (RTC) is a collaborative software delivery environment that simplifies the process of software delivery. With RTC, the software development team has an integrated set of work items/defects, a configurable development process, an integrated build capability, and a software configuration management (SCM) system. Rational Team Concert for IBM System z supports a client/server structure. The server component, along with the repository (for work items and the SCM – and more) is supported on Linux on IBM System z. See: [ibm.com/software/awdtools/rtc](http://ibm.com/software/awdtools/rtc).

### *Open Source Tools*

Two open source tools are available for source configuration management, named CVS and SubVersion. Both tools are popular software configuration management tools available today for Linux. SubVersion is supported on Red Hat Linux Versions 4 and 5. The support of CVS or SubVersion on Linux on IBM System z is left to the adopter.

- For information about CVS, see [www.nongnu.org/cvs](http://www.nongnu.org/cvs). Various Integrated development environments, including any Eclipse-based product and emacs, support access to CVS.
- For information about SubVersion, see [subversion.tigris.org](http://subversion.tigris.org).

An open source tool for tracking defects and features for software development projects is bugzilla, see [www.bugzilla.org](http://www.bugzilla.org).

## **Edit Capabilities**

Editing capabilities have improved significantly in recent years. Eclipse, an [open source](http://www.eclipse.org) integrated development environment (see [www.eclipse.org](http://www.eclipse.org)), in its support of Java<sup>™</sup>, has raised the expectations with regards to editing of software source. The enhanced support includes various refactoring capabilities such as renaming variables as well as method names or class names, and source formatting. This is accomplished through the creation of an in-memory model of the source code, which must be robust and tolerant of incomplete and incorrect source code (which is the case while the source is actively being edited).

The Edit Location describes the best location to perform sophisticated editing functions. These functions must provide the basic editing capabilities, and they must also be able to support refactoring capabilities, which can range from simple renaming of a variable to the extraction of logic for the creation of a service. Languages with pre-processors and macro support also add another level of complexity.

## Application Development Tools and Virtualization Advantages for Linux on System z

Some of the current editing characteristics expected by the user include:

- Source code syntax highlighting/coloring based upon language type
- Syntax verification or validation
- Content assist, also known as completion assist, for language constructs as well as declared variables
- Context sensitive help for the programming language

### *IBM tools*

*IBM's Rational Application Developer (RAD)* supports the development of J2EE applications that will be deployed to a J2EE runtime container such as *IBM WebSphere Application Server* on Linux on IBM System z. The sophisticated editing capabilities for Java are found in RAD.

*IBM's Rational Developer for System z (RDz)*, provides editing support for COBOL, PL/I, HLASM, C/C++, shell scripts, and additional text-based file types. Access to files on remote systems is provided through the Remote System Explorer (RSE) view. RSE provides the ability to view, edit, and manipulate the file system of remote Linux on IBM System z systems from an Integrated Development Environment (IDE) running on a Windows system, as if the remote file system resides locally.

### *Open Source tools*

Some of the popular open source tools for editing are:

- vim – an advanced editor with the capabilities of the de-facto-standard UNIX<sup>®</sup> editor vi. See [www.vim.org](http://www.vim.org).
- emacs – another popular text editor that is configurable with advanced features such as syntax highlighting/coloring. See [www.gnu.org/software/emacs](http://www.gnu.org/software/emacs).
- kdevelop - an integrated development environment for K Desktop Environment (KDE) users. See [www.kdevelop.org](http://www.kdevelop.org).

## **Building/Constructing the Application**

The build location describes where builds can be accomplished. Java compiles can be performed on any platform, and the results moved to the runtime platform. In the absence of a cross-compiler, C/C++ builds must be built on the platform where the program will run.

In some large shops, rebuilding everything is prohibitive and costly. Therefore, you might want to have the build process re-build only the parts that are affected by a coding change. These 'smart' builds must understand the various dependencies of the components that comprise the application suite.

For security and management purposes, understanding which programmer requested the build and the constitution of the build (which parts at what level were included in the build) is mandated for some industries. Given a running application, you should be able to trace back to

## Application Development Tools and Virtualization Advantages for Linux on System z

all of the artifacts that contributed to the executable. Given a source artifact, you should be able to determine with confidence exactly where the artifact is used.

### *IBM Tools*

In conjunction with *IBM's Rational Team Concert*, *IBM's Rational Build Forge* delivers the key characteristics for a build infrastructure deployed on Linux on IBM System z.

Details about IBM Rational Build Forge can be found at [ibm.com/software/awdtools/buildforge](http://ibm.com/software/awdtools/buildforge).

IBM Rational Build Forge is capable of driving other build processing tools such as “Ant” and “Make”.

### *Open Source Tools*

The popular open source tools for “build” are Make (from gnu) and Ant (from Apache).

- Ant is a Java-based build tool that uses an XML markup file to describe build steps. This is an extensible structure allowing you to tailor the specific Ant tasks to perform as needed. The XML build script allows for conditional execution of tasks so that a ‘smart’ build can be coded in the XML build script. See [ant.apache.org](http://ant.apache.org).
- Make is another build tool from Gnu. The build script, called a make file, has a unique syntax and also provides for the conditional build of application parts depending on what parts have been changed. See [www.gnu.org/software/make](http://www.gnu.org/software/make). Makefiles are used to declare explicit dependencies between modules and therefore require a rebuild of whatever is necessary, based on which source files have been modified.

## Testing the Application

When the application is built, it must be moved to the platform where it will be tested, then installed/deployed into the runtime container, and then enabled for execution. Test data must also be available. Many times, the test data is a subset of the operational data, which may need to be obfuscated for privacy purposes.

There are many aspects of testing, including, but not limited to:

- Unit test
- Functional test
- Performance test
- Security and conformance testing

### *IBM Tools*

IBM Rational offers a wide variety of tools for numerous aspects of testing:

- IBM Rational Quality Manager



## Application Development Tools and Virtualization Advantages for Linux on System z

- IBM Rational Test Manager
- IBM Rational Policy Tester (formerly WatchFire®)
- IBM Rational AppScan

For more information, see [ibm.com/software/rational/offerings/quality](http://ibm.com/software/rational/offerings/quality).

### *Open Source Tools*

Widely used on Linux for testing are two frameworks that are named “JUnit” and “CppUnit”. The latter was based on the former, and they both have similar goals.

- JUnit is a simple framework to write repeatable tests for Java code. See [www.junit.org](http://www.junit.org).
- CppUnit is a framework similar to JUnit, except that it is targeted for C/C++. See [apps.sourceforge.net/mediawiki/cppunit/index.php?title=Main\\_Page](http://apps.sourceforge.net/mediawiki/cppunit/index.php?title=Main_Page).

## Strengths of the virtual Linux environment on IBM System z

### *Software developer requirements*

Software developers should be kept isolated from production environments. Software in development tends to be unstable and not always predictable. Therefore, it could compromise the business if not kept completely isolated from the production environment.

Developers should have shared access to tools, middleware and associated program development suites. Ideally, this should be in a common repository, to ensure that all developers are working with the same levels of software to avoid incompatibilities.

The development environment should be highly available. Compilers, tools and test environments should be readily available 24 x 7 to keep productivity high, especially for worldwide development teams.

Development errors should be corrected quickly during the test process. Therefore the development and test environment should be able to be built/rebuilt quickly.

### *Matching the development requirements*

#### **Isolation from production**

IBM System z servers are capable of totally isolating multiple execution environments. This is done through the use of hardware functions, such as Logical Partitioning (LPAR), as well as through the use of z/VM software virtualization to define virtual servers.

These technologies have reached the highest certification levels for their isolation function under the Common Criteria, the LPAR technology is certified at EAL5 and z/VM v5.3 is certified at EAL4+ 3.

If one developer's program terminates abnormally, loops, or adversely affects data, this behavior will not affect activities in other virtual machines or logical partitions. One or multiple

## Application Development Tools and Virtualization Advantages for Linux on System z

virtual Linux servers can be created for each developer that are totally isolated from the servers of other developers as well as the production servers.

### **Shared access**

Data and applications can be shared in a virtual Linux server environment. Tools, middleware, related program development and compilers can be placed in a common repository for use by the entire development team. z/VM also provides the developers with the ability to share tooling, related code, and other information through the use of shared mini-disks. Mini-disk is a logical subdivision (or all) of a direct access storage device, so that multiple virtual systems may mount and access file systems on these minidisks in order to enable shared access to common files.

Developers' systems can also communicate between one another easily through the use of a virtual LANs. Virtual LANs provide a more efficient communication path than an external network, while minimizing the requirement for networking hardware.

The "share everything" approach of IBM System z helps to avoid additional hardware purchase and setup time.

### **High availability and security**

IBM System z servers are uniquely positioned to provide technologies and services to help identify and remove single points of failure in critical components, and provide for security, disaster recovery and availability. Planned outages are a thing of the past; with IBM System z servers, your applications can stay up 24 x 7, including times when making concurrent changes to the environment.

### **Quick rebuild of development and test environment**

z/VM enables developers to build and provision virtual Linux servers in minutes. When a developer needs an additional server, or when a server or test environment needs to be created or recreated it can be done very quickly by using well established methods of cloning systems using z/VM or IBM Systems Director and the storage server.

### *Leveraging the advantages*

Before looking at the development phases in which the advantages of the virtual Linux environment on IBM System z can best be leveraged, a few important capabilities should be mentioned first, because they provide their values to developers during all development phases.

- The IBM System z internal network called HiperSockets™ is an in-memory implementation of a TCP/IP network. The dependency on the external network for code, data and communication between developers can be reduced dramatically by using HiperSockets. The advantages are improved efficiency and a highly-secure network environment. The efficiency is improved, since HiperSockets do not require cabling, routers, switches and hubs. Latency time associated with external networks is reduced as well. The internal network is simpler and easier to extend, modify and troubleshoot than an external network. A security advantage is provided, because all of the communication takes place within the IBM System z server; there can be no attacks on external cables or devices.

## Application Development Tools and Virtualization Advantages for Linux on System z

- IBM System z hardware, operating systems, and middleware elements have been designed to work together closely, providing an application environment with a high level of availability. The IBM System z environment approaches application availability with an integrated and cohesive strategy that encompasses single-server, multi-server, and multi-site environments. The IBM System z hardware itself is a highly available server. From its inception all of the hardware elements have always had an internal redundancy. Starting with the energy components and ending with the central processors, all of these redundant elements can be switched automatically in the event of a failure. As a result of this redundancy, it is possible to make fixes or changes to any element that is down without stopping the machine from working and providing support for the customers.

### Source location

In the source location phase, the IBM System z server can provide significant advantages to the development process due to its unique virtualization attributes. The IBM System z server may be used as a repository for sharing source code elements and program subroutines, as well as advanced development tools such as the IBM Rational Clear Case.

### Edit

In the edit stage, developers often like to use editors of their own choosing and other tools. However, the sharing of key subroutines, data definitions and other elements of the application can help to avoid re-coding due to the inclusion of routines or data elements at different levels. Again, the IBM System z servers allow sharing data and applications in the virtual Linux server environment.

### Build

In the build phase, code from many developers may be pulled together to form the entire application. This can be done readily and simply through the use of a virtual machine that accesses the developers' code in read-only mode, and builds the application in an automated fashion. This saves time and effort and ensures uniformity from build to build.

### Test

The IBM System z server offers an array of test tools that help developers quickly and completely test an application. Developers can also set up a virtual environment to isolate the testing environment. These capabilities provide a level of control which assists in the development of high quality applications.

A development environment may often need to be reprovisioned. Data elements can be altered during the course of a test, and have to be restored to their original values for subsequent tests. IBM System z servers, when used in conjunction with IBM storage servers, offer a function called FlashCopy4. This function enables the rapid creation or duplication of data in seconds, as opposed to running disk to disk or tape to disk restores which could take several hours.

### Conclusion

In constructing a development environment, there are many options to choose from ranging from commercial offerings to open source offerings. Consideration needs to be given to the aspects of edit, source repository, build infrastructure, and testing infrastructure. The commercial offerings and the open source offerings can be used to compliment one another.

## Application Development Tools and Virtualization Advantages for Linux on System z

With a set of Linux systems as the target for applications to run on, regardless of whether the applications are Web application server-based, Linux daemon-style programs, batch applications, or a combination of these, a development and test environment is likely to be employed by the organization. As “cloud computing” becomes more popular, such environments will become even more commonplace.

Some of the new application development tool packages take a seamless view of the development cycle and the ability to collaborate and more easily share information. Especially interesting are the *Jazz* (see [jazz.net](http://jazz.net)) and *Rational Team Concert* (see [ibm.com/software/awdtools/rtc](http://ibm.com/software/awdtools/rtc)) tools, which should be considered.

And finally, the virtual Linux server environment on IBM System z offers capabilities that can help reduce the size and the complexity of a dynamic IT infrastructure, including application development.

# Application Development Tools and Virtualization Advantages for Linux on System z

## Resources

IBM Rational tools for Linux	<a href="http://ibm.com/software/rational/linux">ibm.com/software/rational/linux</a>
developerWorks® Rational software	<a href="http://ibm.com/developerworks/rational">ibm.com/developerworks/rational</a>
developerWorks WebSphere on IBM System z	<a href="http://ibm.com/developerworks/websphere/zones/systemz">ibm.com/developerworks/websphere/zones/systemz</a>
Linux on System z	<a href="http://ibm.com/systems/z/os/linux">ibm.com/systems/z/os/linux</a>
z/VM	<a href="http://www.vm.ibm.com">www.vm.ibm.com</a>

## About the authors

**“Jerry” Hosch** is a marketing expert working for the Linux on IBM System z development organization. He is working from Boeblingen, Germany. You can reach Jerry at [hosch@de.ibm.com](mailto:hosch@de.ibm.com).

**Larry England** is a Senior Technical Staff Member at IBM's Silicon Valley Lab in San Jose, California with architectural responsibilities for application development tools on IBM System z. When not working at IBM, he can be found running a trail in the California hills. You can reach England at [englandl@us.ibm.com](mailto:englandl@us.ibm.com).

**Tim Hahn** is a Distinguished Engineer in the Rational organization. He is Chief Architect for Enterprise Tools, working from Raleigh, NC. You can reach Tim at [hahnt@us.ibm.com](mailto:hahnt@us.ibm.com).

**Acknowledgement:** Many thanks to Theresa Elmendorf without her assistance this paper would not have been possible.

# Application Development Tools and Virtualization Advantages for Linux on System z



Copyright IBM Corporation 2009  
IBM Systems and Technology Group  
Route 100  
Somers, New York 10589  
U.S.A.  
Produced in the United States of America,  
04/2009  
All Rights Reserved

IBM, IBM (logo), ibm.com, ClearCase, ClearQuest, Cognos, DB2, developerWorks, Domino, Dynamic Infrastructure, FlashCopy, HiperSockets, Informix, Rational, Rational Team Concert, System z, WatchFire, WebSphere and z/VM are trademarks or registered trademarks of the International Business Machines Corporation.

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.

InfiniBand and InfiniBand Trade Association are registered trademarks of the InfiniBand Trade Association. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

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

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.

<sup>1</sup> IDC, White paper: Linux Application Development Ready for Prime Time, #05C4473

<sup>2</sup> IBM DB2 9 for Linux, UNIX and Windows

<sup>3</sup> [ibm.com/systems/z/os/linux/security/certification.html](http://ibm.com/systems/z/os/linux/security/certification.html)