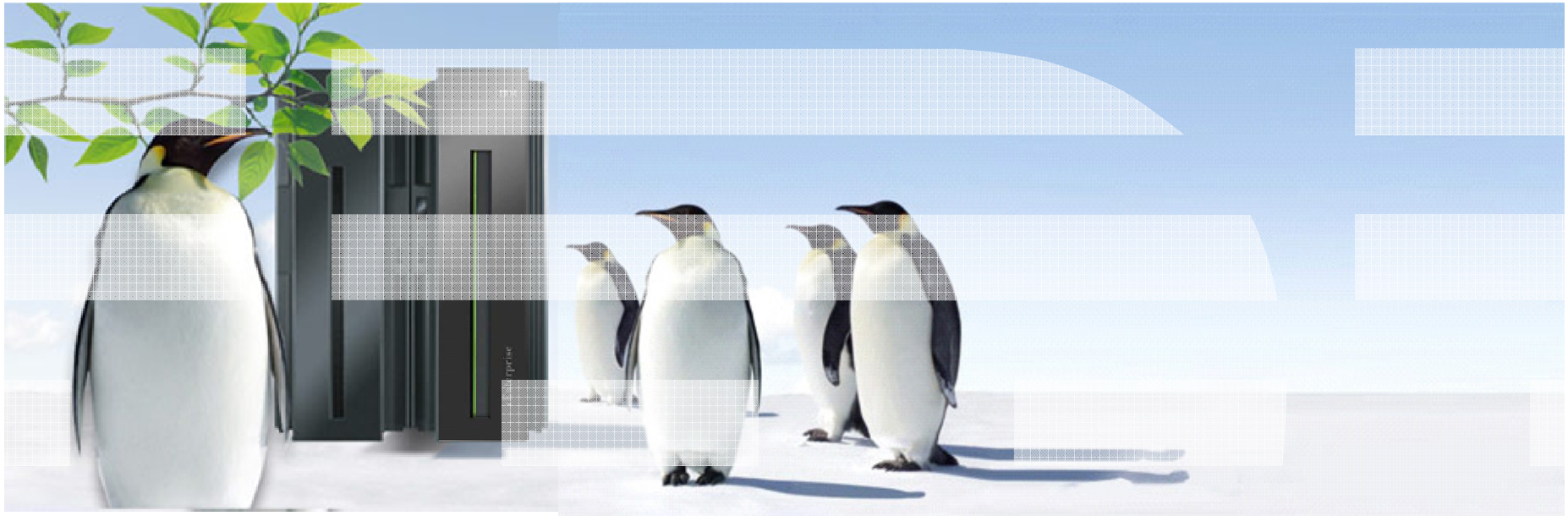
Porting Java applications to System z

Linux on System z Live Virtual Class (Jan 15th / 16th)

Marc Beyerle (marc.beyerle@de.ibm.com)
System z Specialist, Senior Java Performance Engineer



Trademarks

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

AlphaBlox*
APPN*
CICS*
Cool Blue
DB2*
DFSMS
DFSMShsm
DFSMSrmm
DFSORT*
DirMaint
DRDA*
DS6000
DS8000
ECKD
ESCON*
FICON*
FlashCopy*
GDPS*
HiperSockets
HyperSwap
IBM*
IBM eServer
IBM logo*
IMS
Language Environment*
Lotus*
Multiprise*
MVS
OMEGAMON*
Parallel Sysplex*
Performance Toolkit for VM
POWER6
PowerPC*
PR/SM
Processor Resource/Systems Manager
RACF*
Redbooks*
Resource Link
RETAIN*
REXX
RMF
S/370
S/390*
Scalable Architecture for Financial Reporting
Sysplex Timer*
Systems Director Active Energy Manager
System p*
System Storage
System x
System z
System z9*
System z10
System/30
Tivoli*
Tivoli Storage Manager
TotalStorage*
VSE/ESA
VTAM*
WebSphere*
xSeries*
z9*
z10
z10 BC
z10 EC
z/Architecture*
z/OS*
z/VM*
z/VSE
zEnterprise*
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

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.
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 other products may be trademarks or registered trademarks of their respective 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.

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Agenda

- **Java Workloads on System z**
  - Porting aspects
  - Consolidation aspects
  - Performance aspects

- **Field report: porting a real-life, large Java application to Linux on System z**

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Porting aspects

- **Problem: people think there is little or *no porting required***
  - Think about getting porting support right from the project start (and not only when the project is already about to fail)

- **Experience shows there are *subtle differences* between the different JVM™s**
  - ***Very important key point***: the IBM® Java® SDK is not a "special" version of Java, it is 100% pure Java, as it passes all compatibility tests from Sun™

- **Differences fall into *2 categories*:**
  - ***Infrastructure related*** differences (mostly Java command line parameter differences, for example: garbage collection settings)
  - ***Coding related*** differences (for example: Java class library implementation differences)

Source: http://www.smscs.com

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Porting aspects, *continued*

- **Question: Why are there such differences at all?**

- **Infrastructure related: many of the command line parameters are *non-standard* (those starting with `-X`…) and can be added / changed by the JVM vendor without notice**
  - Many customers heavily tweak their command line parameters for one particular JVM

- **Coding related: the Java API specification sometimes *leaves room for interpretation*. Specific example: `java.nio.channels.SelectionKey`**
  - *"… Exactly how this synchronization is performed is implementation-dependent: … reading or writing the interest set **may** block indefinitely if a selection operation is already in progress…"*
  - Developers' responsibility to ensure application's portability across JDK™s

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Porting aspects, *continued*

- **Next *important key point*: all of the problems arising out of those subtle differences *can be fixed***

- **Infrastructure related: some of the command line parameters can be adjusted / translated easily, some of them require additional analysis of the application**
  - There is ***no simple translation table*** since every application behaves differently

- **Coding related: sometimes, small changes (really just a few lines) in the Java application code are required**
  - Problem: there are only ***very few people*** able (and available) to perform the required Java debugging and coding

Source: http://www.smscs.com

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Porting aspects, *continued*

- ***Best practice / strong recommendation*: try to evaluate the to-be-ported application with the IBM Java SDK on any other platform (for example Intel® x86), before going for System z®**
  - Most of the porting related issues are related to the mentioned subtle differences in the various JVMs and not System z
  - Following this best practice, the problems can be addressed where they belong to (which is either the application or the IBM Java SDK, but not System z)

- **Elements / patterns that are *known to cause trouble*:**
  - Heavy usage of platform native libraries / *Java Native Interface* (JNI)
  - Hard-coded path names (happens mostly with Java applications that were developed on / developed for Microsoft® Windows®)
  - Using vendor-specific APIs (for example Java packages starting with `com.sun`)

- **Additional problem (project management related): running a *large scale stress test* for the first time as part of the porting**
  - Issues in the application that are ***not related to the actual porting*** will surface

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Agenda

- Java Workloads on System z
  - Porting aspects
  - Consolidation aspects
  - Performance aspects

- Field report: porting a real-life, large Java application to Linux on System z

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Consolidation aspects

- **Problem: both real-world and benchmark type of Java applications tend to be both *CPU and memory intensive***
  - This is a general statement only – it does not hold true for all applications

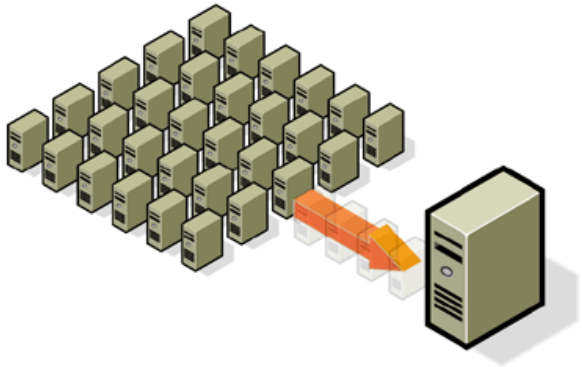
- **Many reasons for this:**
  - Popular approach in software projects: reduce development cycle by ***re-using*** as much existing technology as possible (both Open Source and proprietary libraries)
  - Non-optimal code quality seems to be ***generally accepted***
  - Developers: ***performance follows functionality*** (not a specific Java issue)

- **However, Java is still the *de facto standard* for new applications *for good reasons*, so there is no need to go back to assembler et al.**
  - Broad industry / educational support
  - Excellent documentation (freely) available
  - High performance is really achievable if code is implemented wisely

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Consolidation aspects, *continued*

- **Question: Can Java workloads be consolidated *at all*?**



Source: http://www.maindec.com

- **Answer: *Yes they can*!**
  - See this and the next slides for some thoughts…

- **To cite an example: one customer in Austria that I supported *started with ~50 Java applications* (on WebSphere® for z/OS®), now after some years and after some performance tuning sessions, they are running *way more than 100* Java applications on just a couple of z114 processors**

- **Basically, look for *"small" applications* (with "small" I mean not consuming a lot of CPU and memory) for a successful consolidation project**
  - Same approach compared to other types of applications

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Consolidation aspects, *continued*

- **Being *honest* is important: you cannot have extremely high consolidation ratios and extremely high performance *at the same time***
  - This statement holds true for all types of applications, not just Java
  - It just doesn't work to have both – it will always be a trade-off

- **Comparable to Oracle® workloads, many Java workloads are extremely *oversized from a memory* point of view**
  - Tools available for determining how much memory a Java application really requires
  - However, besides the tools, expert knowledge is also required…

- **If you think in z/VM® terms, *right-sizing* the memory is absolutely key**
  - The Java heap can hardly be paged out by z/VM at all, so you definitely won't be able to achieve high memory overcommitment ratios

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Consolidation aspects, *continued*

- **From an infrastructure (z/VM, Linux®, JVM, WebSphere) point of view, there are possibilities to *reduce the CPU "noise"* caused by idle guests**
  - Steve Wehr – Dealing with Noisy Guests
  - WebSphere Application Server – Idle Server Tuning: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101894

- **Most of the Java applications can be (more or less ☺) easily tuned in order to *save both CPU cycles and memory***
  - See the performance part of this presentation for a more detailed discussion

- **There is an IBM answer to the Apache® Tomcat® / JBoss® *Open Source* discussion**
  - WebSphere Liberty Profile

Source: http://josh-jarvis.com

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Agenda

- Java Workloads on System z
  - Porting aspects
  - Consolidation aspects
  - Performance aspects
- Field report: porting a real-life, large Java application to Linux on System z

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Performance aspects

- **Problem: many customers think that Java on System z (in general) is *slower than on other platforms***
  - Simply not true – I did my own experiments and we are ***actually faster*** in a very popular CPU-intensive Java benchmark (tests were done on a IBM zEnterprise® 196 (z196) compared to latest / greatest Nehalem back then)

- **Next problem: once you are in a performance discussion, people start thinking *it's all about performance***
  - Performance is important, but it is not the one and only aspect of a "good" server
  - There is ***much more*** than just performance when it comes to System z
  - Fit-for-Purpose discussions help to get the broader picture

- **A *holistic approach* is required for improving performance**
  - All layers (complete stack: hardware, hypervisor, operating system, middleware, application) have to be checked for bottlenecks / tuning potentials
  - Not an easy task – requires a team of experts in many cases

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Performance aspects, *continued*

- **Personally, I prefer the *bottom-up approach* for analyzing the stack**
  - Many real-world examples for improvements in all layers
  - Check for the easy things first (LPAR configuration, # of virtual CPUs, etc.)

- **The ultimate tuning discipline (when all else fails) is *application-level tuning***
  - Should only be used as a last resort if really everything else did not lead to significant improvements
  - Make use of standard offerings before (healthcheck, etc.)

- **It is a *very powerful instrument* and should be treated as such**
  - Can be used ***pro-actively*** (retain customer satisfaction with System z, accelerate new System z deals)
  - Can also be used ***re-actively*** (CritSit support)
  - Not a standard IBM offering

Source: http://www.appian.com

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Performance aspects, *continued*

- **How does an *application tuning session* work?**
  - Step 1: explain to the customer how to install Jinsight and how to take application traces
  - Step 2: customer has to identify (with IBM support if required) the "hot spots" in the application – a whole application simply cannot be analyzed and tuned due to the massive amount of data that gets produced; in all cases up to now, tuning the most important parts was ***way enough*** (80/20 rule)
  - Step 3: customer takes the traces and sends them to IBM
  - Step 4: remote analysis of the traces
  - Step 5: on-site presentation and discussion of the observations

- **What *people from the customer* do I need for this?**
  - Step 1-3 and 5: infrastructure people (systems programmers and / or WebSphere administrators)
  - Step 4: usually, customer support is not required
  - Step 5 (and sometimes step 2 also): application developers and management

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Performance aspects, *continued*

- **What can you *expect* from such an engagement?**
  - Experience shows that ***20 – 30% reduction in CPU consumption*** (which is equal to an increase in throughput of 25 – 43%) is achievable in virtually any engagement
  - Very often, CPU consumption can be ***reduced to 50%*** of the original value (which is equal to 2 x throughput compared to before)
  - Sometimes, even more is possible (up to a factor of 17 in throughput increase, or 5.9% of the original CPU consumption)

- **How much does the application *need to be changed* in order to achieve the above numbers?**
  - I am always trying to make the proposed changes as small as possible, in all cases so far usually a change of a couple of lines of code (10 or less) are enough
  - Important to remember is that this is not about completely rewriting the application

Source: http://techamt.com

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Agenda

- Java Workloads on System z
  - Porting aspects
  - Consolidation aspects
  - Performance aspects

- **Field report: porting a real-life, large Java application to Linux on System z**

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report

- **Customer is a *large European bank* that is active in many countries**
  - In their home country, they have one of the ***largest branch networks***

- **Project context: *Linux on System z* Proof of Concept**
  - Customer ***heard a lot*** about the z/VM / Linux on System z platform and wanted to get to know the environment from a technical perspective in order to ***evaluate*** whether it is a viable option for them
  - Prior to the project start, the customer had ***very limited z/VM expertise*** only

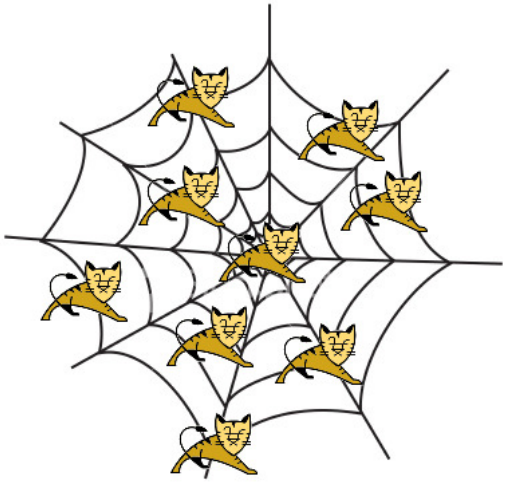
- **Project objectives / success criteria**
  - Successfully port one of their ***largest Java applications*** to Linux on System z
  - Run some ***basic performance tests*** with the ported application
  - On top of this, the goal was also to run a number of ***additional tests*** with some other workload (not part of this field report, though)

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **Application is a *typical banking application* as it is found in many other international banks**
  - Today, it is running on Intel x86 based servers
  - ***Possible candidate*** for Linux on System z in the future, once the customer has gained enough working knowledge with the environment

- **It consists of ~20-25 Apache Tomcat instances with 2-5 web applications deployed on each, so around *60-70 web applications* in total**
  - Largest porting I have ever done so far



Source: http://tomcat.apache.org

- **Java web applications are *heavily intertwined* with each other**
  - One end-user request touches a lot of different applications under the covers
  - If one component in this chain fails, it becomes visible immediately in the user interface

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **First problem that I encountered was a** `java.lang.NoClassDefFoundError: org.apache.harmony.security.fortress.Services$NormalServices (initialization failure)`
  - Obviously, this is ***related to security*** settings
  - Turned out to come from a ***security provider*** that the customer had written themselves
  - Core of the problem: ***hard-coded Sun class name*** `(sun.security.rsa.SunRsaSign)`

- **Workaround was to *change 2 existing* lines of Java code and *add 5 additional* lines of code**
  - In other words, the required changes were ***minimal***
  - From a systems programmer point of view, one `.jar` file had to be exchanged

- **Besides this, the `java.security` file had to be adjusted by changing 1 line**
  - Before: `security.provider.9=myProvider`
  - After: `security.provider.10=myProvider`

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **After fixing the first problem, most of the Tomcat servers could at least be *started successfully* (i.e. no errors in `catalina.out`)**
  - Be careful: a successful Tomcat startup doesn't mean that the application also started successfully

- **Second problem: `java.security.NoSuchAlgorithmException: MessageDigest MD5 implementation not found`**
  - Obviously, ***related to security*** again – one could think that the MD5 algorithm isn't supported by the runtime (which is not true, of course)

- **Turned out to be a Tomcat *configuration issue* (no coding changes required)**
  - The ***Java extension directories*** setting had to be extended to include `${JAVA_HOME}/jre/lib/ext`

Source: http://www.bobslocks.co.uk

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **After fixing the second problem, one of the Tomcat servers started to *consume 100% CPU continuously***
  - For obvious reasons, this looked like an ***infinite loop***

- **Problem determination approach: analyze Java stack traces (by producing Javadumps, also called `javacore` files)**
  - Very easy to realize: send `SIGQUIT` signal to the JVM in trouble

- **Stack trace revealed that the Java thread got stuck in the *ajax4jsf* framework**
  - Internet research revealed that this was an ***already documented*** bug
  - See here: https://issues.jboss.org/browse/JBPAPP-4153

- ***Easy workaround*: use an alternative cache factory implementation**
  - From a systems programmer point of view, ***one additional `.jar` file*** (`oscache.jar`) and ***one additional Java command line option*** were required:
    `-Dorg.ajax4jsf.cache.CacheFactory=org.ajax4jsf.cache.OSCacheCacheFactory`

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **After fixing the third problem, the *biggest issue* surfaced:**
  `javax.xml.transform.TransformerConfigurationException: Could not compile stylesheet`
  - This problem required ***a lot of time*** and debugging
  - Turned out to be a ***complex class loading issue***, based on mixing old and new versions of the same Apache XML® related classes
  - In addition, the XML / XSLT "transformer factory" was ***hard-coded in the customer code***, not using the Java SDK default setting

- **In the user interface, the problem manifested itself in *empty panels***
  - We ***had to solve*** this issue, there was no way to circumvent it

Source: http://www.recruitment-specialist.de

- **Basically, there are *2 solutions* available**
  - Change the class loading order, so that the old versions of the Apache classes are preferred (by using Java's ***Endorsed Standards Override Mechanism***)
  - Change the application code in a way that the ***default Java SDK implementation of XSLT*** is used for the XSL transformation (preferred solution)

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Field report, *continued*

- **Since using the "endorsed standards" mechanism *did not require any code changes*, we chose this approach in order to prove that the application runs successfully on Linux on System z**
  - From a systems programmer point of view, the ***only required change*** to the configuration is to extend the `-Djava.endorsed.dirs` setting to include the directory containing the to-be-overridden `.jar` files
  - ***Easiest and quickest*** workaround

- **Nevertheless (after the end of the Proof of Concept), I gave the customer's developers *all the required code changes* in order to run the application with the Java SDK's built in XSLT implementation**
  - Rationale: the ***new XSLT is much faster*** compared to the old implementation

- **The customer's regression test team ran a *full range of test cases* and the application did not exhibit a single issue anymore -> success!**
  - Internally at the customer, this "officially proved" that the application runs on Linux on System z (and the IBM Java SDK)

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Summary of field report

- **Basically, the problems found fall into *2 categories* (again):**
  - ***Infrastructure related*** (Java command line adjustments for the IBM Java SDK)
  - ***Coding related*** (hard coded class names)

- **Very hard or even *impossible to predict* what exactly will surface when you start your own Linux on System z Proof of Concept**
  - In almost all of the cases (including this field report), the issues found were related to the mentioned ***subtle differences in the JVMs***
  - ***None*** of the issues found was related to Linux on System z

- ***Techniques*** **used for debugging:**
  - Interpreting Java ***stack traces***
  - Reading Java ***source code***
  - Generating ***Javadumps*** (and again interpreting stack traces)
  - ***Replicating*** the problem on my workstation
  - Implement workarounds (***requires coding***) and test them

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Summary of field report, *continued*

- **Linux on System z *performed very nicely*, even with very limited resources (both CPU and memory)**
  - Nice little side story: during one of the additional tests, we ran with ***2 capped CPs at 10% weight*** and didn't even notice it until we stressed the environment really hard

- **Admittedly, this was a *very complex* porting**
  - Don't expect this many problems when porting
  - Even for this complex project, the time required for the actual porting of the application (not including infrastructure setup, etc.) was ***~10 working days***
  - In many cases, Java applications work ***out of the box*** when moving them from one platform to another

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Thank you

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab

Resources

- **IBM Client Center – Systems and Software, IBM Germany Lab**
  - Part of the IBM Development Lab in Boeblingen, Germany
  - External homepage: http://www.ibm.com/de/entwicklung/clientcenter/index_en.html
  - IBM Intranet: http://clientcenter.de.ibm.com
  - Email: clientcenter@de.ibm.com
- **IBM developer kits: http://www.ibm.com/developerworks/java/jdk**
- **Java on z/OS: http://www.ibm.com/systems/z/os/zos/tools/java**
- **Java Diagnostics Guide: http://www.ibm.com/developerworks/java/jdk/diagnosis**
- **WASdev community (incl. Liberty Profile downloads): http://www.wasdev.net**
- **WebSphere Liberty Profile for z/OS Quick Start Guide on Techdocs: http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102110**

Marc Beyerle (marc.beyerle@de.ibm.com) – IBM Client Center, IBM Germany Lab