

Mainframe-Rehosting mit Micro Focus' Enterprise Server

Adieu Dino

Oliver Müller

IBM-Mainframes sind Hochleistungssysteme. Ihre Stabilität, ihr Datendurchsatz, ihre Hochverfügbarkeit und ihre Zuverlässigkeit sind legendär. Sie haben jedoch einen eklatanten Nachteil: Sie sind extrem teuer! Rehosting auf Unix oder Windows verspricht hier erhebliches Einsparpotenzial.



Neben den hohen monatlichen Wartungs- und Lizenzkosten für Hard- und Software schlägt vor allem der Betrieb eines Großrechners deutlich zu Buche. Sind die Systeme doch wenig intuitiv und höchst kryptisch zu bedienen – gerade im Operating. Viele teure, hochspezialisierte Fachkräfte sind nötig, um ein Big Iron seine Arbeit verrichten zu lassen. Fachkräfte, die zudem immer rarer werden, da der Nachwuchs fehlt.

Weg vom Mainframe und hin zum offenen System verspricht daher ein hohes Einsparungspotenzial. Gerade seit dem Siegeszug von Unix/Linux und den dramatisch gefallenem Hardwarepreisen drängt sich eine berechtigte Frage auf: Können solche Systeme nicht die gleiche Last stemmen wie ein Mainframe? Wenn nicht allein, dann sicherlich als Cluster.

Ein IBM-Mainframe z10 oder z196 mit seinen MIPS-Daten (Million Instructions per Second) im deutlich fünfstelligen Bereich sind leistungsmäßig nicht leicht zu schlagen. Aber auch das Unix-Umfeld kennt inzwischen Systeme, die am Thron des oft als Dinosaurier der IT bezeichneten Mainframe rütteln. Außerdem sind für viele Anwendungen keine Mainframes mit

fünfstelligen MIPS-Werten notwendig. Oft genügen den beziehungsweise nutzen die Anwendungen nur Bruchteile der Leistung eines z10/z196-Boliden für einzelne Anwendungen und Kundeninstallationen.

Selbst der Mainframe-übliche hohe Datendurchsatz ist heute mit modernem SAN-Storage adäquat zu bewältigen. Der Wechsel vom Mainframe auf Unix ist daher ein naheliegender Schritt.

Umziehen statt neu bauen

Ein Ansatz wäre die Portierung oder Reimplementierung auf die neue Plattform. Der Kasten „Stolperfallen“ beschreibt, wo hier die Krux liegt. Daneben bietet sich das Rehosting der Anwendung inklusive seiner Systemumgebung an. Die jeweilige Rehosting-Umgebung läuft unter Unix, Linux oder Windows. Sie bildet dort neben dem betreffenden Transaktionsmonitor, beispielsweise CICS (siehe „Glossar“), die notwendige Unterstützung für die alten Mainframe-Jobs ab. Um die Jobs ohne signifikante Änderungen vom Host übernehmen zu können, stellen Rehosting-Produkte einen Ersatz für das Job

Entry System JES2 oder JES3 bereit und bilden selbst in JCL benötigte Dienstprogramme wie *IDCAMS*, *IEBGENER* und *IEBCOPY* nach.

Eine solche Umgebung ist die Enterprise Edition des Servers von Micro Focus. Sie stellt auf Unix, Linux und Windows eine Mainframe-kompatible Laufzeitumgebung für CICS, IMS und JES2/3 für Programme in Cobol, PL/1 und JCL bereit. Mit dem Produkt hat der Hersteller schon Mainframe-Installationen im vierstelligen MIPS-Bereich erfolgreich abgelöst. Den fünfstelligen Bereich hat Micro Focus nach eigenen Angaben noch nicht erreicht. Dies jedoch weniger aufgrund der mangelnden Leistungsfähigkeit des Produkts. Vielmehr gab es diese Dimension bei Kundenmigrationen noch nicht.

Ein Blick hinter die technischen Kulissen

Micro Focus ist seit Langem als Hersteller von Cobol-Compilern bekannt. Die Firma war einer der ersten Compiler-Hersteller, der die Entwicklung vom Host auf den PC verlagerte. Dort entwickelte Cobol-Programme lassen sich 1:1 auf dem Mainframe portieren und umgekehrt. Seit der Übernahme von Liant durch Micro Focus lassen sich auch PL/1-Programme umlagern.

Auch für das Rehosting bilden die Compiler den Kern des Systems. Sie übersetzen den Quellcode der Mainframe-Anwendungen in dynamisch gelinkte Bibliotheken, die die Enterprise Edition auf der Rehosting-Platt-

Stolperfallen beim Umgang mit Mainframes

Abseits der reinen Betrachtung von Rechenleistung und I/O-Durchsatz lauern die Schwierigkeiten eines Umstiegs vom Mainframe auf andere Plattformen an unerwarteter Stelle: Großrechnersysteme arbeiten nach einem völlig anderen Prinzip als Unix und Windows. Eine Systemumgebung auf einem IBM-Mainframe unter z/OS oder z/VSE ist geprägt von Cobol oder PL/1 sowie CICS oder IMS und bisweilen garniert mit direktem VSAM-Zugriffen. Das Steuern der Verarbeitung übernehmen kryptische Jobdefinition in der Programmiersprache JCL. Das Batch-Processing bestimmt das Tagesgeschäft in dieser Umgebung, wo selbst der Dialogbetrieb unter TSO eigentlich ein Batch-Job ist.

Eine in diesem andersartigen Umfeld implementierte Geschäftslogik in der Welt der offenen Systeme nativ oder unter Java EE

abzubilden gleicht einem Gewaltakt. Portierungen resultieren ganz oder teilweise in einer Neuimplementierung. Für Transaktionsmonitore wie CICS oder IMS/TM gilt es Alternativen zu finden, die in der Regel mit dem ursprünglichen Produkt inkompatibel sind. Die oft hohen Investitionen in die Entwicklung des alten Systems ebenso wie der über die Jahrzehnte bewährte und gehärtete Code bleiben dabei nicht selten auf der Strecke.

Doch abseits von Programmen und Systemplattformen lauert meist eine größere Schwierigkeit: die Daten des Altsystems. Einerseits ist das Datenvolumen nach oft Jahrzehnten des Betriebs einer Anwendung sehr hoch. Nicht selten bewegt es sich in den Terabyte-Bereich hinein. Einen solchen Umfang bei Migrationen, zeitweisem Parallelbetrieb in der Umstellungsphase und im Zuge von Anonymisie-

rungen für den System- und Abnahmetest zu verwalten, ist keine triviale Aufgabe.

Als anderer Stolperstein erweist sich oft die Datenkonvertierung. Häufig liefern historisch gewachsene Applikationen auf dem Host einen Mix aus relationalen oder hierarchischen Datenbanken, VSAM und sequenziellen Datasets. Diese Daten in ihren unterschiedlichen Formaten von EBCDIC nach ASCII zu konvertieren ist selten ein Kinderspiel. Sowohl EBCDIC als auch ASCII kennen viele Code-Pages, von UTF-8 ganz zu schweigen. Mainframe-Anbinder können daher von verschwundenen Umlauten sowie durch Fragezeichen und Grinsenmännchen ersetzten Sonderzeichen ein leidvolles Lied singen können. Diese Kombination aus Zeichensatzkonvertierung, Aus-/Einlesen alter Datenformate und hohem Datenvolumen macht dieses Problem so anspruchsvoll.

form analog zu Load-Modules beziehungsweise RPL lädt und ausführt – es erfolgt also keine Emulation eines Mainframe. Die Programme müssen im Quellcode vorliegen und sind auf der Rehosting-Plattform komplett neu zu übersetzen.

Nach dem gleichen Prinzip geht Micro Focus auch beim Übertragen von JCL vor. Statt die JCL-Anweisungen ähnlich einer Skript-Sprache zur Laufzeit zu interpretieren, setzt man hier auf einen Compile- und Link-Zyklus. Eine JCL-Jobdefinition endet beim Rehosting ebenfalls in einer dynamischen Link-Library. Ein direkter Submit an die Jobverwaltung JES wie beim IBM-

Mainframe funktioniert daher ohne Build-Zyklus nicht.

Für das Bereitstellen kompatibler Transaktionsmonitore für CICS und IMS/TM programmierte Micro Focus die IBM-Spezifikationen nach. Ebenso geht der Hersteller bei IMS/DB, dem hierarchischen Datenbanksystem, und einer zu JES-kompatiblen Jobverarbeitung vor. Es entsteht somit ein Produkt, das sich ebenso verhält, wie es die IBM-Dokumentationen beschreibt.

Dieses Vorgehen ermöglicht bestmögliche Kompatibilität, stellt aber zugleich den größten Schwachpunkt der Software dar. Denn im Umkehrschluss heißt das: Was nicht dokumentiert ist,

lässt sich auch nicht nachprogrammieren. Wann immer also Code auf dem Mainframe undokumentierte Features und „Grauzonen“ bewusst oder unbewusst ausnutzt, kann die Rehosting-Plattform im Vergleich zum originalen Mainframe abweichend reagieren. An dieser Stelle gibt es zwei Optionen: Entweder man passt den zu portierenden Code an oder kontaktiert Micro Focus. Der Hersteller ist bemüht, solche Situationen abzufangen. Wenn in Zusammenarbeit mit dem Kunden möglich, beseitigt man die Differenzen zwischen dem Verhalten auf dem echten Großrechner und dem „Nachbau“. Auf diese Weise tastet sich Micro Focus nach und nach an das Verhalten des Originals heran.



- Micro Focus kann mit der Enterprise Edition seines Servers eine kostengünstigere Alternative zum echten Mainframe bieten.
- Die Rehosting-Plattform verfügt über Subsysteme für CICS, IMS und JES auf Unix, Linux und Windows.
- Alte Cobol- und PL/1-Anwendungen lassen sich damit auf offenen Systemen betreiben.

Flexibilität bei der Datenspeicherung

Aus dem Micro Focus Server heraus lässt sich die alte DB2 oder IMS/DB auf dem Host ansprechen. Das ist sinnvoll, wenn Datenbankanwendungen auf dem Host entweder dauerhaft oder während der sukzessiven Migration

verbleiben. Andernfalls lassen sich andere Datenbanken wie DB2 UDB, MS SQL Server oder Oracle über separate Präprozessoren anbinden. Die Aufrufe *EXEC SQL* beziehungsweise *EXEC DLI* bildet die Software intern auf die jeweils angebundene Datasource ab. Auf diese Weise lassen sich die Großrechnerdatenbanken durch kostengünstigere SQL-Datenbanken auf offenen Systemen ersetzen.

Zum Abbilden von Datasets bietet die Rehosting-Plattform einen eigenen File-Manager. Er bildet Dateien der offenen Welt auf Datasets des Großrechners ab. Micro Focus führt ein eigenes Katalogsystem (siehe Abb. 1) und hält zu jedem Dataset Informationen über den DCB sowie die Datenorganisation bereit. Die so emulierten Datasets lassen sich aus den portierten Anwendungen und Jobs wie auf dem Mainframe ansprechen, verarbeiten und anlegen.

Das Übertragen der Datasets gestaltet sich reibungslos. Da Micro Focus Daten auch in EBCDIC halten kann, reicht ein einfacher binärer Transfer via FTP, FTPS oder SFTP aus. Den DCB und die Datenorganisation kann man bequem über die grafische Oberfläche im Browser definieren. Sogar VSAM lässt sich problemlos übertragen. Auf dem Großrechner exportiert *IDCAMS* VSAM in sequenzielle Datasets, die binär auf den Rehosting-Server wandern. Da *IDCAMS* auch auf Micro Focus als Clone verfügbar ist, lassen sich mit einem einfachen Job daraus wieder VSAM-Datasets zaubern. Einzig, wer den Komfort von Tools wie CA FAVER

nutzt, muss sich hier auf Bordmittel wie *IDCAMS* besinnen.

Für jede Systemumgebung stellt Micro Focus einen eigenen Application-Server bereit. Die Software kapselt in ihren Servern das, was auf dem Mainframe in separaten Load-Libraries und Address-Spaces getrennt vorliegt. Jeder Server verfügt wahlweise über Subsysteme für JES, CICS oder IMS. Optional bietet er einen TN3270-Server an, über den migrierte 3270-Anwendungen wie gewohnt im 3270-Emulator laufen können. Für den Anwender ändert sich gegenüber dem Mainframe an dieser Stelle nichts. Selbstverständlich stehen die üblichen Konnektoren für Java EE in den Subsystemen für CICS und IMS ebenfalls zur Verfügung.

Praktischer Einsatz mit Hindernissen

Für den eingefleischten „Hostie“ hält die Konfiguration des Micro-Focus-Systems jedoch das eine oder andere Rätsel bereit. Diese betreffen jedoch primär die Konfiguration der CICS- und IMS-kompatiblen Subsysteme. Ein markantes Beispiel ist die Ressourcen-Definition für CICS. Sie erfolgt in Micro Focus über eine einfach zu bedienende Weboberfläche im Browser. Zusätzlich lässt sich die CSD aber auch aus den ISAM-Files von Micro Focus auslesen. Danach kann man die so erhaltene Konfiguration in einem Texteditor anpassen und die Änderungen zurück ins ISAM spielen. Gerade für

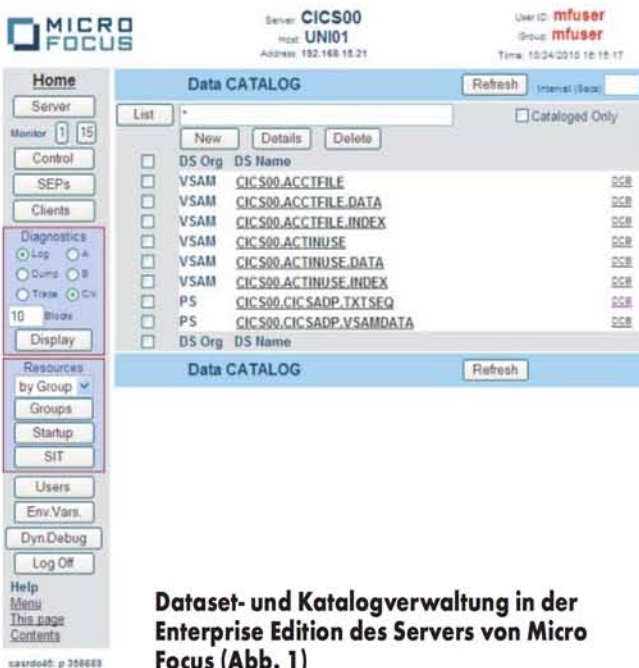
das initiale Anlegen von Anwendungen ist das vor allem bei Unattended-Installationen ein hilfreiches Feature.

Obwohl spätestens bei der CSD viel Ähnlichkeit zu CICS herrscht, verwundert die eine oder andere Einschränkung. Sowohl innerhalb einer Ressourcengruppe im Browser-Interface als auch im ausgelesenen CSD finden sich beispielsweise keine Mapset-Definitionen für BMS. Auf derartige Definitionen verzichtet Micro Focus und setzt hier auf das dynamische Lookup im Dateisystem. Micro Focus sucht im vom CICS-Subsystem definierten *Map Path* nach Load-Modules, die das betreffende Mapset enthalten. Die *EXEC CICS-Zeile*

```
EXEC CICS SEND MAP('MAP01')
MAPSET('MSET01') ...
```

veranlasst den Micro Focus Server dazu, nach einem Load-Module *MSET01.mod* im *Map Path* zu suchen und das gefundene Modul zu laden.

Konsequenterweise muss das mit der BMS-Anweisung *DFHMSD* definierte Mapset folglich in einer *.mod*-Datei mit gleichem Namen liegen. Idealerweise befindet sich das definierte Mapset in einer Quelltextdatei mit diesem Namen. Im Beispiel wäre dies *MSET01.bms*. Gerade die Integration von Micro Focus in Visual Studio gestattet es leider, dass der Name der *.bms*-Datei anders als der des Mapsets lautet. Per Default heißt dann die *.mod*-Datei ebenso wie das *.bms*-Pendant. Schwierigkeiten ergibt das beim Build keine. Zur Laufzeit beim Rehosting



Dataset- und Katalogverwaltung in der Enterprise Edition des Servers von Micro Focus (Abb. 1)

Glossar

BMS	Basic Mapping Support
CICS	Customer Information Control System
CSD	CICS System Definition
DCB	Data Control Block
EBCDIC	Extended Binary Coded Decimals Interchange Code
GDG	Generation Data Group
IDCAMS	Programm zum Anlegen und Verwalten von VSAM-, Non-VSAM-Dateien oder GDGs
IMS	Information Management System
IMS/DB	hierarchisches Datenbanksystem von IMS
IMS/TM	Transaktionsmonitor von IMS
ISAM	Index Sequential Access Method
JCL	Job Control Language (Programmiersprache)
JES	Job Entry System (Jobverwaltung)
MIPS	Million Instructions per Second
RPL	Relocatable Program Library
TN3270	Telnet für 3270-Systeme
TSO	Timesharing Option (Kommandozeileninterpreter)
VSAM	Virtual Storage Access Method
z/VSE	Virtual Storage Extended (Betriebssystem)

erntet man damit jedoch einen zunächst unerklärlichen *ABEND APCT* (Programm nicht gefunden oder deaktiviert). Die Micro-Focus-eigene Entwicklungsumgebung NetExpress hingegen benennt das Mapset immer nach der Quelldatei. Sie hilft damit, solche Hakenleien per se zu vermeiden.

Diese und andere Feinheiten gilt es zu beachten. Leider kristallisieren sich diese erst während der Arbeit mit dem Produkt in der Praxis heraus. Die Dokumentation von Micro Focus bietet leider nicht immer die notwendige Hilfestellung. Sie durchläuft ein tiefer Graben. Die Referenzen zu den Produkten bietet zwar viele Informationen, helfen jedoch nicht in jeder Situation weiter. Bestes Beispiel ist das zuvor geschilderte Mapset-Problem. Die Tutorials bieten nicht die notwendige Tiefe, um in das komplexe System einzusteigen. Vieles erschließt sich erst durch teilweise mühseliges Experimentieren.

Schwierige Aussagen zur Wirtschaftlichkeit

Beim Wechsel vom IBM-Mainframe auf eine Rehosting-Plattform von Micro Focus ist das Ermitteln des Einsparungspotenzials so komplex wie eine Systemumgebung selbst. Einen Standardpreis kann Micro Focus nicht nennen. Vielmehr fließen eine ganze Reihe von Faktoren in die Preisermittlung ein. Die Kosten variieren nach Aussage des Herstellers je nach Zielplattform, CPU-Leistung und den Besonderheiten der technischen Komponenten wie IMS, CICS oder JCL-Unterstützung.

Auf Nachfrage teilte Micro Focus zumindest als Vergleichsgrundlage mit, dass die Einstiegspreise für eine Laufzeitumgebung zwischen 36 000 und 92 000 Euro liegen. Das seien laut Micro Focus etwa 25 % der jährlichen Kosten für einen Mainframe vergleichbarer Leistung. Offen bleibt hier, was das heißt und ob man hier die Listenpreise von IBM zugrunde gelegt hat oder nicht. Faktisch bleibt bei der Betrachtung der Wirtschaftlichkeit lediglich das Anfordern eines zugeschnittenen Angebots beim Hersteller für das individuelle Migrationsszenario. Grundsätzlich verführerisch ist der Umstand, dass die Kosten für Micro Focus einmalig für den Erwerb der Lizenz anfallen. Monatliche Lizenzgebühren wie beim Mainframe fallen nicht an. Je länger die Anwendung also rehosted läuft, desto höher das Einsparpotenzial.

Nach Aussagen von Micro Focus lassen sich mit dem Cobol-Compiler im Schnitt 99 % des originalen Mainframe-Codes unverändert übersetzen. Das klingt nicht nur hoch, sondern es ist ein beachtlicher Wert, der seinesgleichen sucht. Dennoch bleibt ein Prozentsatz inkompatibler Zeilen übrig, die Programmierer während der Migration identifizieren und modifizieren müssen. Legt man diesen Wert beispielsweise einem größeren Projekt mit zwei Millionen Zeilen Code zugrunde, blieben rein statistisch 20 000 inkompatible Zeilen übrig. Wie gesagt: Rein statistisch – die Realität mag je nach Projekt besser oder schlimmer aussehen. Diese inkompatiblen Code-teile sind zu identifizieren und hinsichtlich ihrer Bedeutsamkeit zu bewerten. Damit man keinen Schiffbruch

erleidet ist es wichtig, dass in der Firma noch genügend Know-how zum alten Code vorhanden ist. „Mal eben schnell“ Code unbesehen auf die Plattform von Micro Focus zu bringen, ist zum Scheitern verurteilt.

Bei gutem Know-how und Kenntnis der Legacy-Umgebungen kann eine Migration auf Micro Focus jedoch geringere Kosten verursachen als eine Neuimplementierung. Zum Nulltarif ist das freilich auch nicht zu haben, schon allein wegen des Aufwands für Code-Anpassungen und neuerliche Tests, insbesondere System- und Abnahmetest.

Kurzum: Die Investitionsrechnung für ein Migrationsprojekt vom Mainframe auf eine Rehosting-Plattform wie Micro Focus ist lang und individuell. Ein allgemeines Rezept lässt sich naturgemäß nicht ableiten. Die zu realisierenden Einsparungen, ebenso wie der ROI (Return on Investment) sind individuell von der bestehenden Umgebung und Anwendung sowie dem (noch) verfügbaren Know-how zum Legacy abhängig. Einsparungen im Bereich von 60 bis 70 % der Kosten für Hard- und Software sowie Betrieb gegenüber einem Mainframe sind laut Hersteller jedoch erreichbar und keine Seltenheit. Auch ein kurzer ROI ist – ausreichende Kenntnis der alten Umgebungen vorausgesetzt – praktisch realisierbar.

Fazit

Mit seiner Rehosting-Plattform kann Micro Focus den kostengünstigeren Betrieb alter Großrechneranwendungen

The screenshot displays the 'Enterprise Server Administration' web interface. At the top, it shows the status 'MDS00001 OK' and the date 'Sun Oct 24 20 56:16 2010'. The main area contains a table of servers with columns for Name, Status, License, Security, and Description. The 'CICS00' server is selected, and its details are shown in a pop-up window, including logs for server start and termination. The interface is designed for browser-based administration of Micro Focus servers.

Mit der komplett im Browser laufenden Administrationsoberfläche können Administratoren auf die Konsole weitgehend verzichten (Abb. 2).

Bericht von einem praktischen Einsatz

Ein Migrationsprojekt eines Versicherers, an dem der Autor beratend beteiligt ist, hat zum Ziel, Mainframe-Anwendungen in großen Teilen auf Unix mit Java EE neu zu implementieren. Es handelt sich um mehrere Cobol-CICS-Anwendungen mit Datenhaltung in VSAM-Datasets und DB2 auf z/OS. Gewisse Satellitenprogramme sind schon in Java realisiert und angebunden.

In der Legacy-Umgebung stand nur ein „bescheidenes“ Leistungsvolumen von rund 350 MIPS zur Verfügung. Dieses Potenzial reichte jedoch bei Weitem nicht aus, die konkurrierenden Ansprüche aus dem (Parallel-)Betrieb und der Wartung der Altanwendung, der Umgebungen für Integrations-, System- und Abnahmetest sowie der Aufbereitung der Daten für das neue Zielsystem zu befriedigen.

Für die Stufen benötigte die Neuanwendung „Echtdaten“ zum Testen. Dazu waren für die Tests zunächst die Daten aus dem Altsystem zu lesen, zu anonymisieren und im Neusystem konvertiert bereitzustellen. Eine häufige Schwierigkeit bei derartigen Migrationsprojekten ist, dass auf dem Mainframe keine zusätzlichen Ressourcen zur Verfügung stehen. Das Management, das vom Großrechner weg will, sieht es nicht gerne, wenn Projekte zusätzliche, kostspielige Mainframe-Ressourcen zur Datenaufbereitung anfordern.

Alternativ hätte man Leerlaufzeiten auf den bestehenden Großrechnerressourcen nutzen können. Das hieße jedoch bei großen Datenvolumina im Wesentlichen, Lückenfüller zu spielen. Am Tag im Dialogbetrieb regelt das Workload-Management die Datenaufbereitung zugunsten kurzer Antwortzeiten fast bis zum Stillstand herunter. Startet in der Nacht die Batch-Verarbeitung in ihren engen Zeitfenstern, hungert das Workload-Management die Datenaufbereitung wieder aus. Die pro Tag verbleibenden „MIPS-Krümel“ reichten für eine zeitgerechte Aufbereitung nicht aus. Wer das im Parallelbetrieb und während der Entwicklung mehrfach durchexerzieren muss, macht sich keine Freunde bei drängelnden Entwicklern, Testern und Rolloutern. Der Ruf nach weiteren Ressourcen ist vorprogrammiert.

Daher bot sich für den Versicherer Micro Focus für die Datenaufbereitung an. Einerseits kostete die Software im Vergleich zu einer äquivalenten z/OS-Umgebung nur einen Bruchteil. Gegenüber einer entsprechenden zusätzlichen, bereits rabattierten Umgebung auf dem IBM-Großrechner für ein Jahr lag die Micro Focus-Lizenz mit rund 35 % der Kosten wesentlich günstiger. Andererseits waren die Datenaufbereiter durch die Micro-Focus-Umgebung unabhängig von den schrumpfenden Mainframe-Ressourcen. Zumal für die Datenaufberei-

tung nur Teile der alten Anwendung auf den „Mainframe-Clone“ zu portieren waren, was recht unproblematisch vonstatten ging.

Im Falle des Projekts rechnete sich der Preis für Micro Focus allein schon für die Datenaufbereitung. Den Aufwand für das Bereitstellen der Programme zur Aufbereitung hätte der Kunde so oder so betreiben müssen. Das Umstellen von Cobol und JCL vom Mainframe auf den Micro-Focus-Server ging quasi im Grundrauschen der Entwicklung unter.

Nach diesem Erfolg drängt sich die Überlegung auf, gewisse Teile der Altumgebung zumindest für Tests auf Micro Focus zu portieren. Ließen sich doch damit mindestens rund 40 MIPS und die damit verbundenen monatlichen Kosten zugunsten einer Einmalzahlung einsparen. Sollte dies erfolgreich sein, ließe sich auch der durch die gleitende Migration notwendige Parallelbetrieb der Altanwendung vom Mainframe heben. Bei einer geplanten Restlaufzeit von über zwei Jahren ein reizvolles Rechenexempel. Dieser Part ist allerdings bislang eine reine Modellrechnung. Ein abgeschlossener Proof of Concept unterstreicht zwar die Rechnung. Fraglich ist nur, ob man wirklich ein zusätzliches Projekt aufmachen will. Allerdings könnte sich auch ein Schwenk bei der Migration von Java EE auf Micro Focus abzeichnen.

ermöglichen. Ein hohes Einsparpotenzial gegenüber dem Betrieb auf einem echten Mainframe lässt sich erreichen. Kurze oder zumindest überschaubare ROI sind realisierbar.

Der teure und von Nachwuchsmanager geprägte Betrieb von Mainframes ist nicht mehr zwingend notwendig. Die kryptische Operator-Console auf einem 3270-Bildschirm weicht einer modernen browsergestützten Administrationsoberfläche mit aktuellem Application-Server-Feeling (siehe Abb. 2). Teure Dienstleistungen, Monteur und Wartungsverträge für die Hardware eines System z sind ebenfalls passé; es weicht Komponenten des kostengünstigeren Massenmarktes.

Einzig das Know-how rund um CICS, IMS, Cobol oder PL/1 ist nach

wie vor notwendig. Auch haben Spezialisten mit JCL-Kenntnissen nicht ausgedient. Beim Umstieg von z/OS oder z/VSE auf Micro Focus bleiben die alten Sorgen um Nachwuchs in diesen Bereichen erhalten.

Selbst wenn Micro Focus erheblich günstiger als Big Blue mit seinem echten Mainframe ist, scheint eines bedenklich: So tauscht man als Anwender den proprietären Mainframe gegen das proprietäre Umfeld von Micro Focus und hier droht die nächste – womöglich kostentreibende – Abhängigkeit von einem Hersteller.

Doch die britische Firma Micro Focus kennt Konkurrenz. Jenseits des großen Teichs wartet Oracle mit seinem Tuxedo, das sich ebenfalls mit Laufzeitumgebungen für das Rehos-

ting von CICS und JES aufwerten lässt. In Frankreich stellt Metaware mit Refine ebenfalls eine Lösung zum Rehosting von CICS- und IMS-Anwendungen auf Unix/Linux bereit. Der Wechsel von Micro Focus auf Tuxedo oder Refine ist grundsätzlich realisierbar, wenn auch aufwendig. Außerdem werben auch die „Großen“ seit Jahren mit Rehosting: Bull, Fujitsu, HP Oracle/Sun, ... Das mag die Preise moderat halten und Monopole verhindern helfen. (avr)

OLIVER MÜLLER

ist Geschäftsführer der OMSE Software Engineering GmbH. Er leitet die Bereiche Software Engineering, Nearshoring und IT-Sicherheit.

