



Keine Zukunft für den Mainframe

Zeilentransport

Jens Köhlert

Die Euler Hermes Group mit Sitz in Paris hat ihre IT-Infrastruktur konsolidiert, ihre Mainframes nach AIX/Unix migriert und so die Betriebskosten gesenkt. Das zweijährige Migrationsprojekt musste allerdings etliche Stolpersteine überwinden.

Die Migration vom Mainframe zu Unix/Linux ist nach den Erfahrungen aus vielen Projekten technisch inzwischen gut beherrschbar. Dabei gilt allerdings: Je größer der Mainframe und umso komplexer die teils über Jahrzehnte gewachsenen Strukturen, desto schwieri-

ger ist der Umstellungsprozess. Das Migrationsprojekt bei der Euler Hermes Deutschland AG gehörte zu den größeren und komplizierteren. Die drei wichtigsten Vorgaben für die etwa zweijährige Migration waren „Zentralisierung“, „Umstellung auf Unix“ und „Kostenreduzierung“.

Nach der Fusion von Euler und Hermes im Jahr 2002 galt es, diverse IT-Standorte mit unterschiedlichsten Systemen und Applikationen auf eine einheitliche, zentrale und zukunftsfähige IT-Infrastruktur zu konsolidieren („Consolidation of Application“). Der Mainframe der deutschen Euler Hermes gehörte nach der ersten Konsolidierungsphase zu den noch verbliebenen zu migrierenden Legacy-Systemen. Zum einen gab die Gruppenphilosophie AIX-/Unix-Systeme vor. Zum anderen waren die Lizenz- und Wartungskosten des Mainframes für die verbliebenen Applikationen hoch. Schließlich war es auch immer schwieriger geworden, qualifiziertes Personal für den Betrieb des Mainframes zu finden.

Die Konsolidierung birgt große Herausforderungen

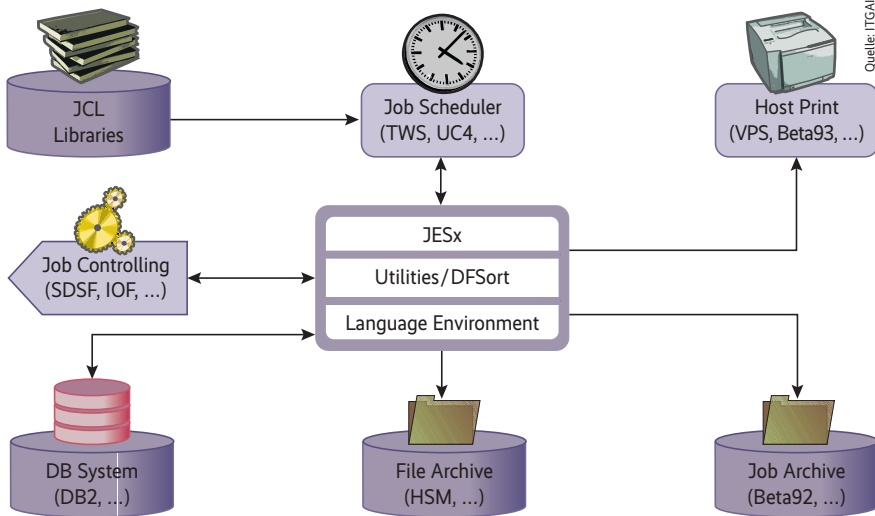
Eine Machbarkeitsstudie der Migrationspezialisten von der ITGAIN GmbH aus Hamburg bestätigte die Umsetzbarkeit. Sie empfahl für die Umstellung ein Re-Hosting-Konzept, bei dem man die Mainframe-Anwendungen inhaltlich nicht verändern muss. Sie laufen unter Unix in einer „Host-Umgebung“, die den Programmen einen Mainframe vortäuscht. Das verringert Projektlaufzeit, Aufwand und Risiken, da die Anwendungen mit gleicher Logik weiterlaufen können wie bisher. Es musste allerdings auch mit etlichen Stolpersteinen gerechnet werden.

Bei der Migration vom Mainframe zu Unix stellten sich fünf große Herausforderungen. So existiert für in Assembler geschriebene Mainframe-Anwendungen unter Unix keine „Host-Umgebung“. Die Programme waren daher noch auf dem Host in COBOL zu konvertieren. Auch die IMS-Datenbanken haben keinen Unix-Gegenpart. Sie mussten noch auf dem Host zu DB2 z/OS umziehen, bevor an den Übergang zu DB2 LUW (AIX) zu denken war. Mit 40 000 externen Nutzern und täglich 250 000 Online-Transaktionen war die Onlineproduktion besonders sensibel und sollte auch in der Umsetzungsphase idealerweise beeinträchtigungsfrei laufen.

Die Batch-Produktion war in über 30 Jahren Betrieb kontinuierlich gewachsen und die Job Control Language (JCL) für die Stapelverarbeitung über diese lange Zeit sehr unterschiedlich und variabel zum Einsatz gekommen. Deshalb bedurfte es eines extrem leistungsfähigen JCL-Interpreters, der diese Batch-Vielfalt auch unter Unix fehlerfrei ausführen konnte.

Zudem erforderten die selbst noch während des Migrationsprozesses konti-

Batch-Produktion (z/OS)



Ein Batch-Produktionssystem besteht aus dem Job Entry Subsystem (JES2 oder JES3), das die einzelnen JCL-Jobs prüft und ausführt, sowie einer Anzahl von „Satellitensystemen“, die spezielle Aufgaben eines Produktionsbetriebs übernehmen (Abb. 1).

nuierlich veränderten Programme und Datenbanken ein Management der regelmäßigen Synchronisation von Test- und Host-Umgebung. Schließlich gab die Kostenplanung nur ein begrenztes Zeitfenster vor, weil die Wartungsverträge des Mainframes auslaufen sollten.

Projektstrategie ohne Big Bang

Es galt, die großen Umstellungsrisiken eines komplexen Mainframe-/Unix-Projekts beherrschbar zu halten. Da es sich um ein komplexes internationales Projekt handelt, spiegelte sich das in der Struktur wider, und die Leitung wurde mit einer Doppelspitze, bestehend aus Bertrand Vigna (Frankreich) und Jens Köhler (Deutschland), besetzt. Dies war eine wichtige Voraussetzung, die Zusammenführung der ehemals eigenständigen IT-Abteilungen parallel voranzutreiben. Die daraus resultierende internationale Zusammenarbeit war ein weiteres Projektergebnis.

Technisch ging es darum, einen großen risikoreichen Big Bang zu vermeiden, indem wichtige Teil-Migrationen schon auf dem Host erfolgten. Weiterhin sollte es keine Vermengung des Migrationsprozesses mit Weiterentwicklungen des Systems geben, weil das das Projekt unbeherrschbar gemacht hätte. Schließlich stellte eine sehr enge Projektführung die ununterbrochene Kontrolle über das Verfahren sicher. Der Projektleitungsausschuss aus IT und Fachbereichen traf sich anfangs monatlich, um später je nach Bedarf zweiwöchentlich oder wöchentlich zu tagen. Laufend aktualisierte Fehlerreports und zeitnah erstellte und kontrollierte Aktionspläne erwiesen sich als gute Steuerungsinstrumente.

Schrittweise Migration

Als Zielarchitektur war gemäß der Richtlinie der Euler Hermes Group eine AIX-Architektur vorgegeben. Zum Einsatz kamen Micro Focus Enterprise Server und das COBOL Runtime Environment.



- Als „Spätfolge“ der Fusion stand die Euler Hermes Group vor der Aufgabe, ihre bestehende, auf Mainframes setzende IT-Infrastruktur auf Unix zu migrieren.
- Im Rahmen einer Vorstudie fiel die Entscheidung zugunsten eines Re-Hosting-Ansatzes.
- Dank intensiver Vorbereitung, straffer Projektführung sowie Abstimmung mit den Fachbereichen konnte die Migration ohne „Big Bang“ erfolgen.

Anzeige

Die Software-Matrix von z/OS und Unix		
Komponenten	z/OS	Unix/Linux
System	z/OS	SLES 9/10, RHEL 5.x
Datenbanken	DB2 z/OS IMS DB	DB2 LUW, Oracle et cetera Micro Focus Server™ Enterprise Edition TWINSOFT (CONVEX, DATCON, DB-HDL)
TRX-Systeme	IMS/TM CICS	Micro Focus Enterprise Server with the Mainframe Transaction Option (ES/MTO) Micro Focus Enterprise Server with the Mainframe Transaction Option (ES/MTO) IBM TXSeries for Multiplatforms
Security-Systeme	RACF, ACF2, TOP Secret	LDAP Server
System-Management	MVS Console (SMF, RMF)	Nagios, TEC (Tivoli Enterprise Console)
Storage-System	DFHSM	Tivoli TSM 360
COBOL	IBM COBOL for z/OS	Micro Focus Server Express Fujitsu NetCOBOL for Linux OpenCOBOL
Assembler	360 Assembler	Konvertierung nach COBOL oder C
PL/1	Enterprise PL/1 for z/OS	Open PL/1
SNA-Protokoll	SNA (LU 6.2)	IBM Communications Server for Linux
JCL-Runtime	JES2/JES3	J2U
Job-Scheduler	TWS (OPC), UC4 ...	UC4, TWS, Control-M, Zeke ...
Sort Utility	DFSORT	Ahlsort
Host-Print	VPS, Beta93	VPSX, BetaUX
Job-Archiv	Beta92	BetaUX

Auf dem alten Host waren die Umwandlung der IMS-Datenbanken in DB2 und die Migration der Assembler- zu COBOL-Modulen die beiden zentralen Vorbereitungsschritte. Die Migration in Etappen hatte wesentliche Vorteile: Sie entzerrte das gesamte Migrationsprogramm und man konnte die umgestellten Module zunächst noch auf dem Host testen und in Produktion gehen lassen. So konnte das Projektteam einige kritische Meilensteine bereits endgültig abschließen und potenzielle Fehlerquellen frühzeitig beseitigen.

Von den 14 000 COBOL-Modulen ließen sich 9500 ohne Änderung maschinell einfach durch erneutes Übersetzen von z/OS nach AIX umziehen. 4500 Module bearbeitete der Dienstleister ITGAIN vor dem Rekompilieren mit hauseigenen Migrationskripten. Es verblieben lediglich 800 Module, die die Entwickler manuell „anfassen“ mussten. Ein Schwerpunkt lag darin, dass sie die in den Programmen fest codierten Hex-Konstanten analysieren und entsprechend der Ziel-Codepage umstellen sowie mehrfach genutzte Indexe in COBOL-Tabellen gegebenenfalls dupli-

zieren mussten. Darüber hinaus galt es, mehrere in einer Quelldatei enthaltene auf einzelne Dateien aufzuteilen und den Aufruf von Stored Procedures anzupassen.

Datenbankumzug in mehreren Etappen

Für den Umzug von DB/2 z/OS nach DB2 LUW standen die operativen Datenbanken mit circa 700 GByte sowie rund 1,4 TByte Daten des Data Warehouse mit 1800 operativen und 900 Data-Warehouse-Tabellen zur Migration an. Um diese Datenmengen zu bewältigen, wurden die Tabellen zuerst auf dem Host exportiert, dann auf die AIX-Umgebung übertragen und dort in DB2 LUW importiert. Die Codepage-Konvertierung fand dabei implizit beim FTP-Transfer statt. Dieses Verfahren funktioniert nicht für Spalten mit Binärdaten, da die nicht konvertiert werden dürfen. Hierfür musste man auf einen sogenannten Crossload, das heißt den direkten Load der z/OS-Datenbank zum LUW DB2 zurückgreifen, obwohl dieses Verfahren um einiges langsamer ist als ein Unload/Load.

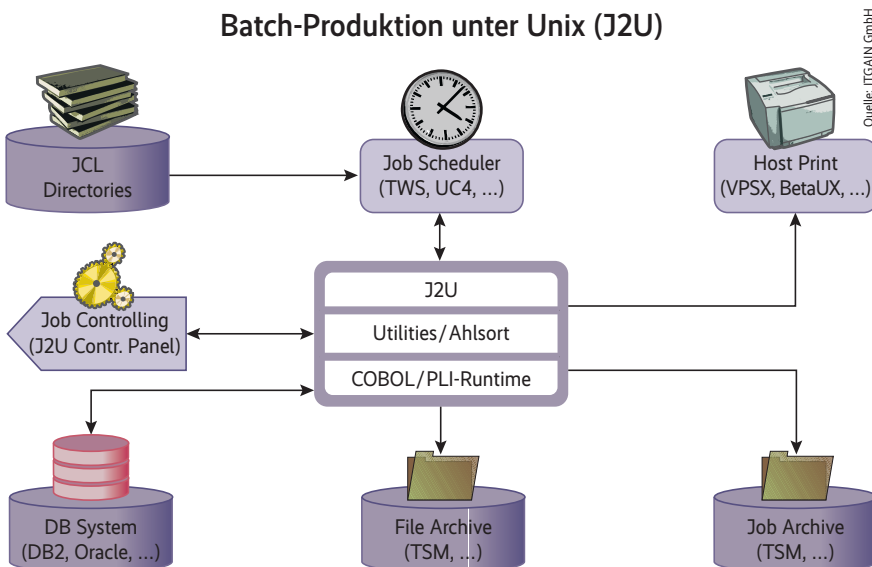
„Für gewachsene und sehr individuelle Mainframe-Umgebungen hat es sich als hilfreich bewährt, individuelle und zielgerichtete Skripte zu entwickeln. So war immer genau bekannt, was an welcher Stelle warum geschieht. Verbunden mit dem Re-Hosting-Konzept ist das eine sichere und gleichzeitig kostensparende Migrationsstrategie“, bestätigt Kai Mares von ITGAIN, Teilprojektleiter für die Anwendungsseite.

Als weitere Besonderheiten galt es zu berücksichtigen, dass DB2 z/OS eine größere nutzbare Zeilenlänge in einem Tabellenraum unterstützt als das AIX-Gegenstück. War diese unter z/OS voll genutzt, mussten die Verantwortlichen nach einer Datenanalyse über eine Verkleinerung der Tabelle entscheiden. Glücklicherweise trat dieser Fall nur bei rund zehn Tabellen auf.

Einige unter z/OS vorhandene UDF (User Defined Function) für Datumsberechnungen, beispielsweise *LAST_DAY*, mussten die Entwickler unter LUW neu erstellen. Außerdem war partitionierter Tabellenraum unter z/OS zu partitionierten Tabellen unter LUW umzuwandeln. Parallel dazu erfolgte die Migration der Security-Profile. Die User-Authentifizierung des Mainframes („TOP Secret“) lief unter Unix ebenfalls nicht. Skripte überführten die Userprofile in LDAP-Attribute.

Die Entwicklungsumgebung wurde komplett neu aufgebaut. Das Migrations-

Batch-Produktion unter Unix (J2U)



Unter Unix übernimmt der JCL-Interpreter J2U die Aufgabe des Job Entry Subsystems. In dieser Umgebung interpretiert er die Job Control Language nach den bisherigen Regeln von z/OS (Abb. 2).

Euler Hermes Group

Die Euler Hermes Group ist Weltmarktführer im Kreditversicherungsgeschäft und eine der führenden Gesellschaften in den Bereichen Inkasso und Kautionsversicherung. Mit mehr als 6000 Mitarbeitern in über 50 Ländern bietet Euler Hermes ein komplettes Spektrum an Dienstleistungen für das Forderungsmanagement an. Der Umsatz 2011 betrug 2,27 Mrd. Euro.

Der Konzern analysiert die Bonität von über 40 Millionen Unternehmen und versicherte Ende Dezember 2011 weltweit Transaktionen im Wert von 702 Mrd. Euro. Euler Hermes, Tochtergesellschaft der Allianz, ist an der Euronext Paris notiert. Die Gruppe und ihre wichtigsten Tochtergesellschaften im Bereich Kreditversicherung erhielten von Standard & Poor's das Rating AA.

team implementierte das früher für die COBOL-Entwicklung eingesetzte ChangeMan unter AIX/Unix auf Eclipse-Basis neu und integrierte es in die Delivery-Prozesse von Euler Hermes.

Die Herausforderung: Das Batch-Mengengerüst

In einer normalen Tagesproduktion führt die IT bei Euler Hermes pro Nacht bis zu 330 Applikationen mit bis zu 2500 Jobs aus. Allein die Dateiübertragung der 112 000 Batch-Dateien mit einem Volumen von rund 9,5 TByte vom Host auf AIX dauerte zehn Monate. Zunächst wurde die zu konvertierende Menge um alle reinen Archivdaten reduziert, sodass die

künftige Batch-Produktion nur noch 130 sequenzielle Dateien bearbeiten muss.

Die nächste Herausforderung bestand im Konvertieren des EBCDIC-Zeichensatzes des Hosts in die Latin-1-Zeichen für AIX. Hier war darauf zu achten, die Klartextfelder vom binären Teil zu unterscheiden, da Binärdaten unverändert bleiben mussten. Erst jetzt konnte der entscheidende Re-Hosting-Schritt der Batch-Produktion beginnen.

Im Rahmen einer Vorstudie hatte das Projektteam infrage kommende Produkte identifiziert und bewertet. Mit dem System J2U von ITGAIN hatte man sich für eine Laufzeitumgebung, in der sich alle Mainframe-Jobs auch unter Unix unverändert ausführen lassen. Das hatte den Vorteil, dass sich die bewährte Verarbeitungslogik der Jobs auf der neuen Zielplattform nicht änderte. Dabei übernimmt J2U unter Unix die Aufgabe des Job Entry Subsystems indem es einen JCL-Interpreter und Utilities, insbesondere das DFSORT-kompatible Sort-Utility Ahsort bereitstellt.

In dieser Umgebung interpretiert das System die Job Control Language nach den bisherigen Regeln von z/OS. J2U bindet transparent jede benötigte Laufzeitumgebung für die in den Jobs aufgerufenen Programme (COBOL, PL/1, C) ein und stellt Schnittstellen für die unter Unix verfügbaren „Satellitensysteme“ bereit. Die übernehmen spezielle Aufgaben eines Produktionsbetriebs, beispielsweise das Job-Scheduling zum Planen und Kontrollieren der Produktionsabläufe oder die Job-Archivierung zum revisions-sicheren Speichern der Jobprotokolle (siehe Abbildung 2). In der rund sechs Monate dauernden Testphase konnte Euler Hermes

```
dwtest@:/data1/dr/dwtest/jobs>sub J2U0001
J2U - VERSION v3.7
Copyright (C) 2004-2009 itgain Consulting GmbH
J2U INFO: JOB J2U0001 (JOB 100554) SUBMITTED
J2U INFO: JOB J2U0001 (JOB 100554) ACTIVATED
J2U INFO: JOB 100554 ENDED WITH HIGHEST RC 0
```

Durch Einsatz von J2U bleibt die Verarbeitungslogik der Mainframe-Batch-Jobs auf der Zielplattform unverändert (Abb. 3).

13 000 JCLs übernehmen und damit die Batch-Tagesläufe sowie die Wochen- und Monatsproduktionen simulieren. Anhand der Fehlermeldungen ließen sich nicht unterstützte Systemprogramme ermitteln und der JCL-Interpreter passend erweitern.

Von der Laborumgebung zum Go-live

Nach den ersten grundlegenden Arbeiten mit Musterdaten in einer Laborumgebung (AIX-Sandbox) setzten die Mitglieder des Projektteams die Ideen zunächst prototypisch um, migrierten exemplarisch Module und sammelten Erfahrungen. In der nachfolgend aufgebauten formalisierten Entwicklungsumgebung realisierten sie für unterschiedliche Produkte eigene Abnahmeumgebungen. 500 sorgfältig ausgewählte Testfälle aus dem Online- und Batch-Betrieb repräsentierten die wichtigsten Businessfälle. Auf dieser Basis entwickelte das Team die Skripte, testete und optimierte sie. Dabei spielte man alle Migrationschritte vom Host zur Unix-Umgebung immer wieder durch, bis keine Fehler mehr auftraten – im Durchschnitt zehn Mal je Testfall.

Zuletzt begann mit den Gesamtabnahmetests der kritischste Zeitabschnitt, da die Projektleitung entscheiden musste, ob der Projektstatus die Gesamtumstellung rechtfertigt. Schließlich legte man ein definitives Go-live-Wochenende fest. Der Donnerstag war der letzte Produktionstag. Bis Freitagabend wurden die Datenbanken migriert und die Zielumgebung nochmals mit Daten versorgt. Am Samstag war die letzte Gelegenheit für die abschließenden IT-Tests, und am Sonntag folgten die entscheidenden Abnahmetests durch die jeweiligen Fachabteilungen. Abends war alles klar: Die Umstellung war erfolgreich und das neue AIX-/Unix-System konnte am Montag in Produktion gehen.

Nach der Umstellung trat bislang nur ein einziger Datenmigrationsfehler auf. Die sorgfältige Vorstudie und die Entscheidung für das Re-Hosting-Konzept hatten sich als das richtige Vorgehen zum

Inkompatibilitäten zwischen Mainframe und AIX/Unix

SET :DATUM = Current Date ist im Zusammenspiel mit DB2 LUW keine gültige Anweisung. Man muss sie entsprechend zu *SELECT Current Date INTO :Datum FROM SYSIBM.SYSDUMMY1* umformen.

LUW akzeptiert doppelte Anführungszeichen nicht als Begrenzung von Literalen. Das heißt Anweisungen wie *SELECT NAME FROM T1 WHERE ORT = "Hamburg"* sind in *SELECT NAME FROM T1 WHERE ORT = 'Hamburg'* zu wandeln.

Der Compiler ignoriert die Spalten 1 bis 6 zur Nummerierung nicht vollständig. Unter z/OS würde er eine Anweisung wie **ABC IF ABC = "EFG"* berücksichtigen; der Microfocus Compiler ignoriert sie aber als Kommentar.

Manche COBOL-Unterprogramme führen kein explizites *GOBACK* für den Rücksprung ins aufrufende Programm aus. Unter Microfocus bleiben solche Anwendungen am Ende eines Unterprogramms stehen.

Eine Besonderheit, die sich nicht mit Skripten automatisieren lässt, ist die Verwendung von HEX-Konstanten in den Programm-Sources, beispielsweise:

```
05 HEX-SPC PIC X VALUE X"40"
```

Hier mussten die Entwickler analysieren, ob der HEX-Wert ein darstellbares EBCDIC-Zeichen repräsentiert (im obigen Beispiel: Space), und in den Latin-1-Wert umzuwandeln ist (hier also *VALUE X"20"*).

Meistern der zu erwartenden Stolpersteine erwiesen. An anderer Stelle traten allerdings Verzögerungen ein, denn in einem komplexen Projekt dieser Größenordnung sind nicht alle Probleme vorhersehbar. Deren Ursachen lagen aber meist in Abhängigkeiten von parallel durchgeführten anderen Projekten.

Eine gute Vorbereitung, straffe Projektführung sowie die detaillierte Abstimmung mit den Fachbereichen sind deshalb so wichtig. So wurden Priorisierungen aus Sicht des Business getroffen und der Go-live-Zeitpunkt gemeinsam entschieden. ITGAIN steuerte Migrationserfahrungen und technisches Wissen bei.

Eine andere Philosophie für die IT

Euler Hermes hat seine Projektziele Zentralisierung, Umstellung auf Unix sowie eine zweistellige Reduzierung der Betriebskosten erreicht. Neben den geringeren Wartungskosten haben sich vor allem die bessere Ressourcennutzung und das gezieltere Aufrüsten von Prozessoren

oder Hauptspeicher rentiert. Für Euler Hermes sind darüber hinaus die moderne AIX-/Unix-Umgebung, die größere Flexibilität der Unix-Welt für zukünftige Weiterentwicklungen und die bessere Verfügbarkeit von Mitarbeitern mit Unix-Know-how von beträchtlichem Nutzen.

Dabei ist allerdings auch eine neue System-Philosophie zu berücksichtigen: Die Art zu arbeiten hat sich komplett verändert. Angefangen von der neuen Entwicklungsumgebung über die Security/User-Authentifizierung bis zum Monitoring – die gesamte Methodik ist umgestellt. Der Vorteil des Mainframe-Betriebs liegt in der Administration: Die zentrale Konsole ist der Dreh- und Angelpunkt des Systems. Das hat im Betrieb viele Vorteile, so sind die Kontrollmechanismen besser, die Datensicherungsmöglichkeiten einfacher und die Aufgabengliederung detaillierter. Der Nachteil des Mainframe-Betriebs liegt in der Konfiguration: Man muss so unterschiedliche Aufgaben wie die von Datenbank-, Kommunikations- oder Applikationsservern (Batch und Online) mit ihren widersprüchlichen Anforderungen (Workload) auf einer Hardware

(„in einem Blech“) betreiben. Das Konfigurieren des Mainframe-Systems erfordert deshalb immer Kompromisse beim Verwalten von I/O, CPU und Memory. Dies erschwert häufig eine Optimierung von Systemen und Anwendungen.

Beim Unix-/AIX-Konzept übernehmen dedizierte Rechner (real oder virtualisiert) die spezialisierten Aufgaben, beispielsweise Datenbank-, HTTP- oder Batch-Server. Dadurch lässt sich jeder Server entsprechend seiner Aufgabe und seiner Workload-Charakteristik optimal konfigurieren. Darüber hinaus kann man Performance-Engpässe durch den Einsatz zusätzlicher Komponenten oder weiterer Server gezielt beheben. Eine Skalierung der jeweiligen Server-Komponente ist häufig – zumindest annähernd – linear möglich.

Dem steht der prinzipielle Nachteil der „Zersplitterung“ der Informationen auf separate Server sowie das Fehlen eines Gesamt-Monitorings mit automatisiertem Reporting gegenüber. Dies lässt sich aber durch Einsatz eines zentralen Monitoring- und Alert-Systems entschärfen. Ein zentraler Log-Server sammelt dazu die systemrelevanten Informationen der einzelnen Server und übernimmt die Aufgaben der MVS-Konsole. Für die Überwachung komplexer IT-Infrastrukturen eignen sich beispielsweise das Open-Source-Tool Nagios oder IBMs Tivoli Enterprise Console. Für beide gilt aber, dass sie zunächst nur einen Werkzeugkasten bereitstellen, denn die Überwachung einer komplexen verteilten Umgebung ist eine anspruchsvolle konzeptionelle Aufgabe, die Wissen und Erfahrung erfordert. Die Zeit nach der Systemumstellung wird Euler Hermes daher nutzen, weitere Optimierungen und Konsolidierungen vorzunehmen sowie Erfahrungen zur Konfiguration von Betriebssystemen und Hardware zu sammeln.

Weitere Detailaufgaben ergeben sich aus dem unterschiedlichen Laufzeitverhalten einiger Unix-Transaktionen gegenüber dem früheren Host. COBOL-Programme, die bei vielen I/O-Zugriffen auf dem Mainframe eine gute Performance erreichten, zeigen unter Unix dagegen schlechte Antwortzeiten. Passt man diese Transaktionen gezielt an, lassen sich Verbesserungen um den Faktor zehn erreichen. Ähnliche Besonderheiten gibt es auch bei der Batch-Verarbeitung, sodass auch diese Prozesse Schritt für Schritt weitere Optimierungen erfahren werden. (avr)

Jens Köhlert

ist bei der Euler Hermes Group verantwortlich für den Bereich IT-Infrastrukturen.



Empfehlungen für das Re-Hosting (Best Practices)

Beim Dienstleister ITGAIN haben sich im Lauf der Jahre sieben Kernaspekte für Re-Hosting-Projekte herauskristallisiert.

- (1) Die Anforderung an das Zielsystem (Unix/Linux) respektive an das Ergebnis des Re-Hosting-Projekts ergeben sich aus dem IST-Zustand des Mainframes. Leistung und Verhalten sollten diesem entsprechen.
- (2) Die Migration des Mainframes sollte man nicht durch Zusatzanforderungen verkomplizieren. Ein typischer Fehler besteht darin, schon lange existierende Missstände im Mainframe-Umfeld bei der Migration mit beseitigen zu wollen, wie eine DB-Umstellung auf UTF-8.
- (3) Auch andere Modernisierungen oder Weiterentwicklungen, sollte man nicht mit dem Re-Hosting-Projekt verknüpfen. Besser ist es, die Anwendungen zu einem Stichtag „einzufrieren“. In dieser Phase dürfen nur produktionsrelevante Wartungen erfolgen. Dagegen sind geplante Applikationsverbesserungen oder neue Komponenten hier zu verbieten.
- (4) Kommen auf dem Mainframe noch Assembler-Programme zum Einsatz, lassen sich diese unter Unix/Linux nicht mehr ausführen. Trotzdem wäre die erforderliche Programmkonvertierung zu COBOL während der Migration ein unnötiges Risiko. Diese Arbeiten soll-

ten noch auf dem Mainframe erfolgen, um potenzielle Fehler und Probleme in Ruhe bereinigen zu können. Das Gleiche gilt für die Ablösung von IMS-Datenbanken (nach DB2), Portierung von APPC-Kommunikation nach TCP/IP und alle anderen Technikänderungen.

(5) Das Gedächtnis (die Datenbank) eines Systems sollte man beim Re-Hosting als Einheit behandeln. Während der Umzug von Prozessen schrittweise erfolgen kann, muss der Umzug der Datenbank zu einem Zeitpunkt als Einheit stattfinden. Einzelne Prozesse können bereits auf dem Unix/Linux-System arbeiten und über ein Gateway (wie DB2 Connect) weiterhin auf die Mainframe-Daten zugreifen. Hierbei gilt: Der Datenbank-Umzug schließt das Re-Hosting-Projekt ab – die Datenbank macht quasi das Licht aus.

(6) Bei bestimmten Ausnahmesituationen, die nach dem Umzug der Datenbank auftreten, sollte man diskutieren: Was könnte getan werden, falls die Leistungskapazität des Unix/Linux-Systems nicht ausreicht (Hardwareaufrüstung durch mehr CPUs, mehr RAM et cetera)? Hierzu gehört auch die Frage nach einer Fallback-Strategie.

(7) Generell gilt: Alles, was man im Vorfeld der Migration auf dem Mainframe abarbeiten kann, sollte man auch dort erledigen.