

[www.linux-magazin.de/technical-review](http://www.linux-magazin.de/technical-review)

Ausgabe 2009

Alles über:

# Migration

**Das Motiv: Linux ist besser und billiger**

## Strategien

Wie man Wirtschaftlichkeit berechnet

Wechsel juristisch bedenkenlos

Migrationsleitfaden gibt detaillierte Hilfestellung

Linux – das ERP-Betriebssystem der Zukunft

## Technik

So gelingt der Office-Umstieg

Wine, Terminalserver, Virtualisierung:

Windows-Applikationen unter Linux

Infrastruktur migrieren


**Vom Mainframe zu Linux-**

ein Beitrag der  
**ITGAIN GmbH**

ITGAIN



# Vom Mainframe zu Linux



Lohnt es sich, über Alternativen zum zentralen Mainframe nachzudenken? Die Antwort darauf fällt je nach Ausgangslage unterschiedlich aus. Wer sich jedoch Vorteile ausrechnen kann, der hat gute Karten: Er kann von den Erfahrungen vieler Linux-Migrationsprojekte profitieren, die erfolgreich Großrechner abgelöst haben.

Thomas Kalb, Joachim Roll

**Die Migration** vom Mainframe zu Linux ist inzwischen technisch gut beherrschbar. Ohne eine intensive Vorbereitung gerät sie dennoch schnell zum risikoreichen und teuren Abenteuer. Die Senkung der Betriebskosten ist meistens das Hauptmotiv für die Abwanderung vom Mainframe. Allerdings nennen immer mehr Anwender auch die mangelnde Einsetzbarkeit von Open-Source-Technologien, die große Abhängigkeit von einem Hersteller oder zunehmende Schwierigkeiten, geeignetes Personal für den Betrieb eines Großrechners zu finden.

Vermutlich hätten sich bereits viel mehr Unternehmen vom Mainframe getrennt, wenn nicht so viele, über lange Jahre gewachsene Altanwendungen betroffen wären. Der Migrationsaufwand dafür erscheint oft zu hoch und es gibt auch eine weit verbreitete Skepsis, ob verteilte Lösungen in puncto Leistung und Verfügbarkeit mit dem Großrechner mithalten können. Die Erfahrungen zeigen jedoch, dass die Verlagerung einzelner Anwendungssysteme oder sogar der gesamten System- und Anwendungslandschaft von einem z/OS-System auf eine Linux-Plattform nicht nur machbar ist, sondern dass bei einem solchen Plattformwechsel ein stabiles, performantes und gut administrierbares Gesamtsystem entstehen kann. Voraussetzung hierfür ist eine sorgfältige Planung und die Wahl erprobter Methoden und Verfahren.

## Der Mainframe

Der Mainframe verkörpert ein extrem zentralistisches Rechnerkonzept, bei dem sich so unterschiedliche Aufgaben wie die von Datenbank-, Kommunikations- oder Applikations-Servern (Batch und Online) mit ihren widersprüchlichen Anforderungen (Workloads) eine Hardware teilen. Bei der Konfiguration des Mainframes sind deshalb immer Kompromisse beim Verwalten von I/O, CPU und Memory notwendig. Dies erschwert häufig eine Optimierung.

Der Vorteil des zentralistischen Betriebes liegt aufseiten der Administration: Beim Mainframe ist die zentrale Konsole der Dreh- und Angelpunkt des Systems, was für den Betrieb viele Vorteile bringt, zum Beispiel bessere Kontrolle oder einfachere Datensicherung.

## Der Linux-Cluster

Bei einem dezentralen Linux-Cluster übernehmen dedizierte Rechner (real oder virtualisiert) spezielle Aufgaben wie Datenbank-, HTTP-

oder Batch-Server. Die unmittelbaren Vorteile sind hier:

- Jeder Server lässt sich entsprechend seiner Aufgabe und seiner Workload-Charakteristik optimal konfigurieren.
- Performance-Engpässe sind durch zusätzliche Hardware behebbar. Die jeweilige Server-Komponente skaliert häufig linear.
- Die nachteilige Zersplitterung der Informationen aufgrund der Dezentralisierung lässt sich durch die Implementierung eines zentralen Monitoring- und Alert-Systems entschärfen. Die systemrelevanten Informationen der einzelnen Server sammelt dabei ein zentraler Log-Server. Er übernimmt dann die Aufgaben der MVS-Konsole.

Für die Überwachung komplexer IT-Infrastrukturen eignet sich das Open-Source-Produkt Nagios oder die Tivoli Enterprise Console der Firma IBM. Für beide Produkte gilt allerdings, dass sie zunächst nur einen Werkzeugkasten bereitstellen.

## Die Migrationsstrategie

Beim Plattformwechsel vom Mainframe auf eine verteilte Unix-/Linux-Umgebung kommen zwei prinzipiell unterschiedliche Strategien in Betracht: das Re-Engineering und das Re-Hosting. Sie unterscheiden sich wie folgt: ▶

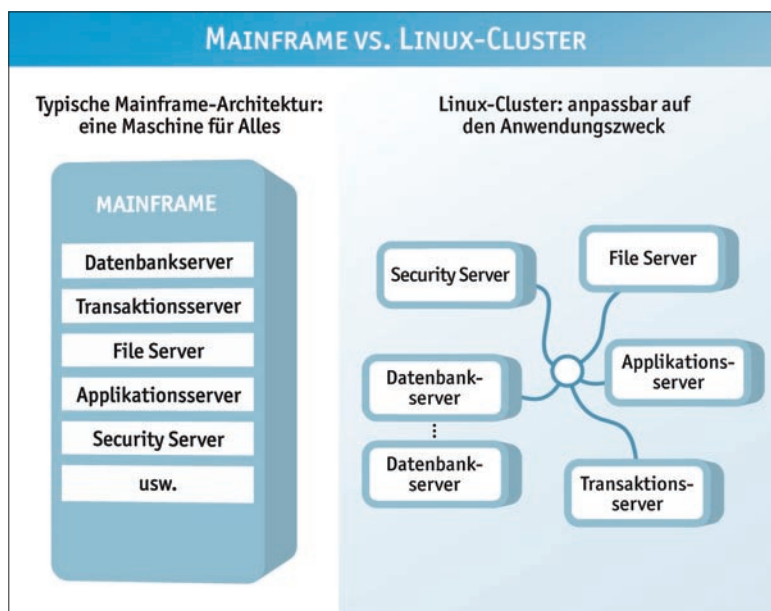


Abbildung 1: Während der Mainframe zu Kompromissen gezwungen ist, weil er Aufgaben mit widersprüchlichen Anforderungen auf einer Hardware vereint, stellt der dezentrale Linux-Cluster dedizierte Rechner für spezielle Aufgaben bereit.

**Re-Engineering:** Dieses Verfahren verlangt die gezielte Neuentwicklung aller Anwendungen für die Zielplattform. Prinzipieller Vorteil: sauber angepasste und auf die Wünsche der Fachabteilung zugeschnittene Anwendungen. Prinzipieller Nachteil: hoher Aufwand durch Neuentwicklung (Abstimmung, Neukonzeption, Testaufwand jeder Anwendung).

**Re-Hosting:** Bei diesem Verfahren laufen die Anwendungen auf der Zielplattform in einer Umgebung, die ihnen ihren gewohnten Host vortäuscht. Nur ein sehr kleiner Teil der Anwendungen ist anzupassen. Prinzipieller Vorteil: geringerer Gesamtaufwand als beim Re-Engineering (kein Abstimm-, wenig Testaufwand), dadurch geringere Projektlaufzeit; hohe Sicherheit, da die Anwendungen mit gleicher Logik laufen wie bisher. Prinzipieller Nachteil: gleiches Look-and-Feel der Anwendungen, keine neuen Produkte. Nicht für jedes z/OS-Produkt gibt es ein Pendant unter Linux.

Erfolgt die Entscheidung für einen Plattformwechsel primär aus Kostengründen, ist immer der Re-Hosting-Ansatz zu bevorzugen, da diese

Methode eine hohe Sicherheit durch Beibehaltung der Geschäftslogik bei deutlich geringeren Projektkosten verspricht.

### Ablauf eines Re-Hosting-Projekts

Der erste Schritt in einem solchen Projekt ist die Erfassung des Ist-Zustands mithilfe spezialisierter Tools und Interviews. Im Ergebnis entsteht eine z/OS-Linux-Produktmatrix, die die prinzipielle Machbarkeit beschreibt. Es folgt eine detaillierte Analyse des Systems mit einem Proof of Concept (Prototyp). Am Ende steht ein grober Projektplan, der Aussagen zur Architektur sowie Sizing und Pricing des Zielsystems, eine Schätzung der Projektkosten und die Berechnung des Business Case enthält. Der Verlauf des eigentlichen Projekts unterscheidet sich dann nicht prinzipiell von dem eines Re-Engineering-Projektes. Wesentlicher Unterschied beider Vorgehensweisen ist die Tatsache, dass beim Re-Hosting ein großer Teil der Neuentwicklung samt der meisten Unit- und Integrationstests entfallen kann.

Tabelle 1: Die z/OS – Linux Software Matrix

Komponenten	z/OS	Linux
System	z/OS	SLES 9/10, RHEL 5.x
Datenbanken	DB2 z/OS IMS DB	DB2 LUW, Oracle etc. Micro Focus Server Enterprise Edition TWINISOFT (CONVEX, DATCON, DB-HDL)
TRX-Systeme	IMS/TM CICS	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-Konsole (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

## Migrationsanalyse

Jede Migration vom Mainframe zu Linux ist machbar. Offen sind nur die Kosten und die Risiken. Diese ermittelt eine sorgfältige Ist-Aufnahme. Je mehr Sorgfalt und Zeit man in diese Phase investiert, umso besser sind die Ergebnisse und umso geringer die Kosten im Verlauf der Migration.

Die Ist-Aufnahme erfolgt in mehreren Stufen. In der Regel erstellt man zunächst einen „Schnappschuss“, um eine erste Einschätzung der Machbarkeit, der Kosten und Risiken zu ermitteln. Die anschließende detaillierte Ist-Aufnahme profitiert theoretisch von einer guten System-Dokumentation. Die Praxis zeigt allerdings auch, dass die Programme selbst die verlässlichste Dokumentation darstellen. Mithilfe von Analysewerkzeugen lassen sie sich prüfen und ihre Abhängigkeiten und Aufruflogiken dokumentieren. Dann tragen Interviews mithilfe von Fragebögen Informationen zu der Systemlandschaft und den Besonderheiten zusammen.

Die Analyse der Anwendungen sammelt die entscheidenden Fakten für die spätere Migration: Auf welcher Technologie beruhen die vorhandenen Anwendungen? Welche Programmiersprachen sind im Einsatz? Generell gilt: Wenn Altanwendungen eine Programmiersprache wie Cobol verwenden, die es auch in der Linux-Welt gibt, dann ist eine weitgehende 1:1-Umsetzung möglich. Finden sich Programmiersprachen wie Assembler, die nicht auf das Zielsystem portierbar sind, muss man neu programmieren, was die Migration deutlich aufwendiger macht.

Die genaue Inventur der vorhandenen Programme beinhaltet eine Kategorisierung in solche, bei denen keine Eingriffe nötig sind, eine Liste der Programme mit Unverträglichkeiten und eine Kategorie für weitere Programme, die neu zu schreiben sind. Finden sich etwa in Cobol-Programmen hexadezimale Konstanten für die Druckersteuerung oder Übersetzungstabellen, dann ist an dieser Stelle Handarbeit angesagt.

Eine sorgfältige Analyse findet immer auch einige Anwendungen, die nicht mehr nötig sind. Das spart Zeit und Kosten, zumal es sich dabei häufig um ältere Anwendungen ohne Dokumentation und mit vielen Unverträglichkeiten handelt, die Probleme bereiten würden.

Das Ergebnis der Ist-Aufnahme ist im Wesentlichen eine Software-Matrix, in der jedem auf dem Ursprungssystem eingesetzten Programm

ein entsprechendes, unter Linux verfügbares Produkt gegenübersteht (**Tabelle 1**).

Innerhalb der auf die Ist-Aufnahme folgenden Feinanalyse sollte man nach Möglichkeit ein Proof of Concept, also einen Prototypen implementieren. Hier geht es zum einen darum, Techniken zu entwickeln, die in späteren Projektphasen erforderlich sind, und zum anderen um den praktischen Nachweis der Machbarkeit.

## Die Zielarchitektur

Ein besonders wichtiger Teil der Ist-Aufnahme und der nachfolgenden Analyse ist die Sammlung von Daten und Fakten für Konzeption und Sizing der Zielarchitektur. Dabei gibt es einiges zu beachten:

- Die Einrichtung eines zentralen Loggings und Alertings und die daraus resultierende zentrale Konsole für die Administration des Systems ist ein Muss.
- Die Lizenzbedingungen von Software sind genau zu prüfen, um bei der Hardware-/Software-Planung eine optimale Kombination zu erreichen.

## Migration der Batch-Umgebung

Für die Migration der Batch-Produktion mit dem Re-Hosting-Ansatz bietet sich das System J2U von itgain als Host-Umgebung an. J2U ist eine Laufzeitumgebung, die alle Jobs unverändert ausführt. Das hat den Vorteil, dass sich die bewährte Verarbeitungslogik der Jobs auch auf der neuen Zielplattform nicht ändert.

Ein Batch-Produktionssystem besteht aus dem Job Entry Subsystem, das die einzelnen JCL-Jobs prüft und ausführt, und einer Anzahl von Satellitensystemen, die spezielle Aufgaben übernehmen, wie zum Beispiel das Job-Scheduling zum Planen und Kontrollieren der Produktionsabläufe oder die Job-Archivierung zum revisionssicheren Speichern der abgearbeiteten Job-Protokolle.

Das Job Entry Subsystems stellt einen JCL-Interpreter und Utilities wie das DFSort-kompatible Sort-Utility Ahlsort bereit. Diese Umgebung interpretiert die Job Control Language (JCL) nach den bisherigen Regeln von z/OS. Das System bindet transparent jede benötigte Laufzeitumgebung für die in den Jobs aufgerufenen Programme (Cobol, PL/1, C) ein und stellt Schnittstellen für die unter Linux verfügbaren Satelliten bereit. ▶



## Re-Hosting sorgt für effiziente Migration

Das Job Entry Subsystem offenbart die Vorteile des Re-Hosting-Ansatzes: An den Geschäftsregeln der Batch-Produktion, also der Logik und dem Verhalten der einzelnen Steps innerhalb von Jobs und der einzelnen Jobs innerhalb von Jobnetzen, ändert sich nichts. Die Batch-Produktion läuft genauso ab wie bisher, nur auf einer anderen Plattform.

Es entfallen alle der im Rahmen eines Migrationsprojektes normalerweise erforderlichen Maßnahmen zum Migrieren und vor allem zum Testen jedes einzelnen Jobs. Damit verringern sich unmittelbar die Kosten und die Laufzeit des gesamten Projektes.

Ein wesentlicher Vorteil ist die volle Kompatibilität. Der JCL-Interpreter emuliert die z/OS-JCL sowie die Funktionen und Kontrollbefehle von IBM-Utilities. Die Logik der Dateibehandlung innerhalb der Batch-Jobs bleibt vollständig erhalten. Die Funktion des zentralen MVS-Katalogs, der Informationen über alle Dateien im gesamten System enthält, übernehmen Datenbanktabellen.

Der Re-Hosting-Ansatz ist auch für die Mitarbeiterakzeptanz vorteilhaft. Mitarbeiter mit langjähriger MVS-Erfahrung können auch unter Linux die von JES2 gewohnten Joblisten erzeugen, die Aufschluss über den Status und den

Fortgang von Jobs geben. Die Arbeitsvorbereitung kann die Produktion nach den gewohnten Regeln steuern und bekannte JCL-Anweisungen lassen sich weiter verwenden.

Eine interessante Variante für die Anwendung des JCL-Interpreters ist das Teil-Re-Hosting. Dabei migrieren beispielsweise nur die Batch-Anwendungen, die für die Leistungsspitzen am Monatsende verantwortlich sind. Auch weitgehend isolierbare Anwendungen wie das Data Warehouse lassen sich damit unter Linux betreiben, ohne den Mainframe vollständig abzuschaffen.

Die Kontrolle der Batch-Produktion, das Drucken von Listen und das Speichern von Job-Protokollen verbleiben bei dieser Lösung zentral auf dem Host. Neben den Erfahrungen mit dem Migrationsprozess gewinnt das Unternehmen auch wegen der teilweise deutlich geringeren Lizenzkosten für den weiterlaufenden Mainframe.

## Empfehlungen für das Re-Hosting

- Das Quellsystem (Mainframe) bestimmt die Leistung und das Verhalten des Linux-Zielsystems nach Abschluss des Re-Hostings. Hier sollte es weder zu Verlusten kommen, noch sollte man die Migration durch Zusatzforderung verkomplizieren. Ein typischer Fehler ist, schon lange existierende Missstände im Umfeld des Mainframes bei der Migration gleich mit beseitigen zu wollen, etwa eine Code-Page-Umstellung der Datenbank auf UTF 8. Am besten friert man die Anwendungen zu einem Stichtag ein und beschränkt sich danach auf Wartungsarbeiten.
- Assembler-Programme, die der Mainframe möglicherweise noch einsetzt, laufen unter Linux nicht. Trotzdem wäre die erforderliche Konvertierung zu Cobol während der Migration ein unnötiges Risiko. Diese Arbeiten führt man besser vorher noch auf dem Mainframe aus, um auftretende Fehler und Probleme dort in Ruhe bereinigen zu können. Dasselbe gilt für die Ablösung von IMS-Datenbanken (durch DB2) oder die Portierung von APPC-Kommunikation nach TCP/IP und Ähnliches.
- Die Datenbank eines Systems ist während des Re-Hosting-Projekts als Einheit zu behandeln. Während der Umzug von Prozessen schrittweise erfolgen kann, muss der Datenbankumzug in einem Schritt stattfinden.

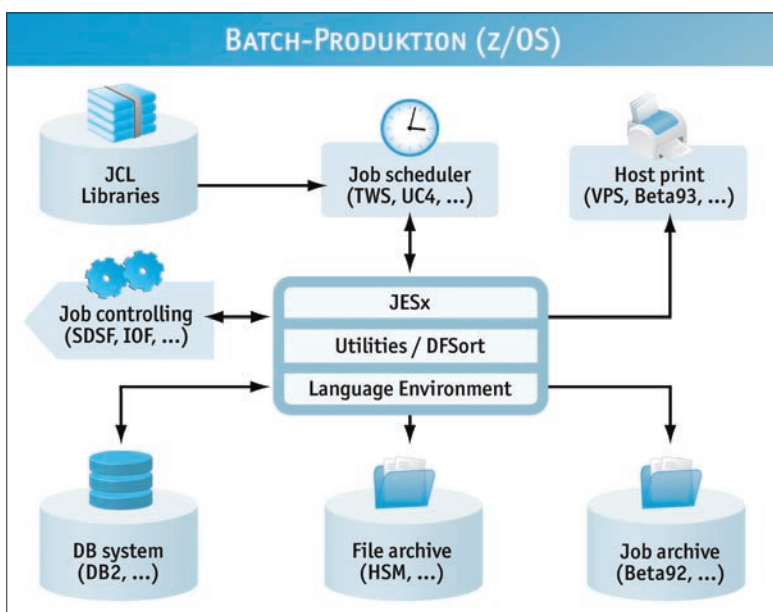


Abbildung 2: Ein Batch-Produktionssystem besteht aus dem Job Entry Subsystem, das die einzelnen JCL-Jobs ausführt, und Satellitensystemen für spezielle Aufgaben.

Einzelne Anwendungsprozesse können der Datenbank vorausgehen und danach über ein Gateway (etwa DB2 Connect) von Linux aus die Datenbank des Mainframes nutzen. Der Datenbankumzug schließt das Re-Hosting Projekt dann ab.

- Oft bestehen Bedenken hinsichtlich der Datenbank-Performance. Sie sind unbegründet. In gründlichen Mainframe-Linux-Performance-Tests anhand exemplarischer Workflows erwies sich Linux in 80 Prozent der Fälle als leistungsfähiger, und zwar mindestens um den Faktor zwei.
- Bestimmte Worst-Case-Szenarien sollte man vor Projektabschluss durchspielen: Was wäre zu tun, falls die Leistungskapazität des Linux-Systems nicht ausreicht? Gibt es eine Fallback-Strategie?

### Praxisbeispiel: Bürgel

Das Unternehmen Bürgel Wirtschaftsinformationen hat sein bisher Mainframe-basiertes System komplett auf ein Intel-basiertes SAN-System unter Linux umgestellt. Die Gesamtumstellung erfolgte in einem einzigen Prozess.

Bürgel ist eine der führenden Wirtschaftsauskunfteien Deutschlands und bietet als Tochtergesellschaft von Euler Hermes (Allianz-Gruppe) und EOS (Otto-Gruppe) Dienstleistungen in den Bereichen Kreditmanagement, Forderungsmanagement und Direktmarketingmanagement. Kunden greifen per Internet auf die Datenbanken bei Bürgel zu. Dabei erwarten sie eine performante Anwendung mit kurzen Antwortzeiten für Abfragen und Echtzeitaktualisierungen. Bis zur Umstellung leistete das eine DB2-Datenbank auf einem IBM-Mainframe mit Web-Gateway.

Das Unternehmen Bürgel expandiert stark mit massiv steigenden Anforderungen an die IT. Es war abzusehen, dass die Anwendungen an die Kapazitätsgrenzen des Großrechners stoßen würden. Als Ausweg kam eine Aufrüstung in Betracht. Die aber wäre mit hohen Kosten verbunden gewesen und trotzdem wäre die Hochverfügbarkeit ein Problem geblieben, denn der Host selbst war nicht redundant. Außerdem ließen sich auch die richtigen IT-Experten für den Mainframe nur noch schwer finden.

Diese Umstände motivierten den Wechsel und weil man bereits gute Erfahrungen mit einem Adress-Suchsystem auf Linux-Basis gesammelt hatte, prüfte man zuerst diese Alternative.

### Zwei Jahre Projektlaufzeit

Vorstudien und Analysen gingen Details auf den Grund und legten Schritte fest, um die Risiken der Umstellung zu minimieren. Das Projektteam erarbeitete Fall-Back-Mechanismen, um im Notfall mit möglichst geringen Problemen zur alten Systemumgebung zurückkehren zu können. Ein wichtiger Vorbereitungsschritt war der Performancetest Mainframe/Linux, nachdem bereits Expertisen von HP und IBM die generelle Machbarkeit des Projektes bestätigt hatten.

Alle Anwendungen wurden stufenweise portiert. Erst im letzten Schritt kam die DB2-Datenbank des IBM-Mainframes an die Reihe. Alte Applikationen und bereits migrierte griffen in einer drei Monate dauernden Phase parallel auf den gleichen Datenbestand zu. Die Skripte waren vielfach getestet, alle Prozeduren automatisiert und die Aktion selbst wiederholt geprobt worden. So gab es mit der Umstellung erwartungsgemäß auch keine Probleme.

Wo vorher nur eine CPU im Spiel war, agieren in der neuen Umgebung viele CPUs mit konkurrierenden Prozessen. Sie arbeiten anders als ein Mainframe. Dafür ergeben sich nun etwa beim Datenbank-Tuning Möglichkeiten, die es auf dem Mainframe nicht gab. Das erfordert eine Neu-Justage dieser und anderer Komponenten. So arbeitet beispielsweise der

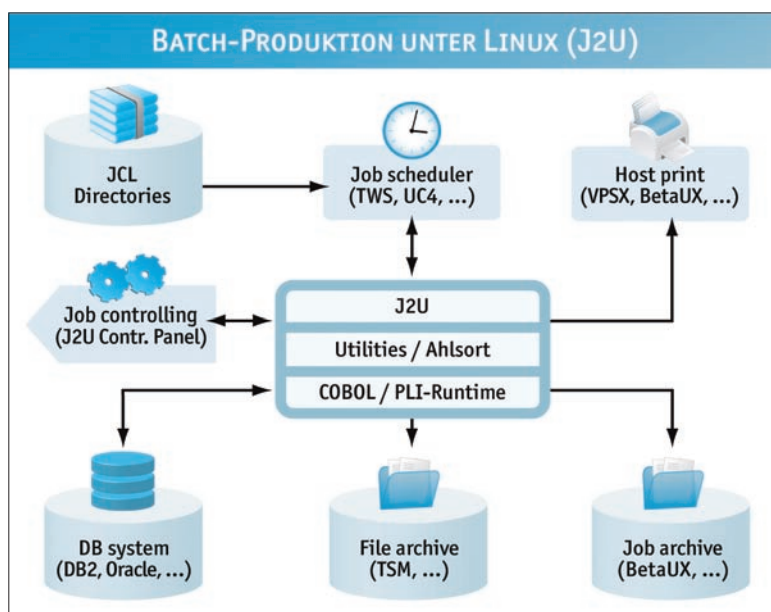


Abbildung 3: Batch-Produktion unter Linux: Das Job Entry Subsystem interpretiert Job Control Language nach den bisherigen Regeln von z/OS.

Microfocus-Cobol-Compiler restriktiver als der Host-Cobol-Compiler. Das machte es nötig, einige Programme anzupassen, was Skripte ohne manuellen Eingriff realisierten. Die eingesetzte Open-Source-System-Software) erwies sich als mindestens ebenbürtig gegenüber den Mainframe-Lösungen, ohne dass dafür die dort üblichen erheblichen Lizenzgebühren anfallen.

## Kosten und Nutzen

Eines der Hauptziele der Mainframe-Linux-Migration war, die Grund- und Betriebskosten zu senken. „Klar erreicht,“ sagt Achim Siebertz, Bereichsleiter EDV bei Bürgel. Die Kosten für Hardware und Software ließen sich deutlich senken. Insbesondere die bei Mainframes hohen Kosten für Software-Lizenzen reduzierte der Open-Source-Einsatz deutlich. Unterm Strich ist für das Unternehmen Bürgel in einem Markt mit sehr engen Margen die Wettbewerbsfähigkeit merklich gestiegen.

Auf der Nutzenseite nennt Siebertz vier Gründe, warum die Migration klare Vorteile gebracht hat: Im Vergleich zum Mainframe ist das Linux-geclusterte SAN-System um den Faktor zwei performanter. Die Antwortzeiten und die Laufzeiten der Programme im Batch sind für den Anwender spürbar besser. Als zweiter großer Vorteil hat sich erwiesen, dass die neue Umgebung auch technologisch flexibler ist. Aspekt Nummer drei: Die Ausfallsicherheit ist durch die redundante Auslegung mit zwei parallelen Datenbankrechnern und permanent gespiegelten Datenbanken besser gewährleistet. Vorteil Nummer vier: Weil Bürgel auch künftig erheblich expandieren will, schätzt man die gewonnene Skalierbarkeit des IT-Systems, wo vorher beim Mainframe sprunghafte Kostensteigerungen drohten.

Die Mainframe-Linux-Migration hat sich aus Sicht von Bürgel technisch, wirtschaftlich und für die Wettbewerbsfähigkeit klar gelohnt.

## Praxisbeispiel: Rechenzentrum

Auch das kommunale Rechenzentrum Minden-Ravensberg/Lippe hat den Mainframe komplett abgelöst und mit einem Re-Hosting-Ansatz die über viele Jahre gewachsenen Batch-Programme auf eine neue Plattform transferiert. Bis in die 70er- und 80er-Jahre dominierten auf dem Großrechner herkömmliche Fachverfahren der angeschlossenen Kommunen. Der zunehmende Trend zu Web-basierten Bürgerservices stieß einen Erneuerungsprozess dieser Anwendungen an. Die neuen Services arbeiteten nach dem Client-Server-Modell und ersetzten innerhalb weniger Jahre die bisherigen Großrechner-Applikationen.

Der Projektablauf orientierte sich an fünf Entwicklungsstufen, die jeweils mit drei Tests als Meilenstein abschlossen: Umgebungstests zur Integration aller neuen Hard- und Software-Komponenten; Lasttests mit definierten Workloads, um die Performance im Auge zu behalten und Tests zur Verfügbarkeit mit der Simulation von Rechnerausfällen und der Auswirkung auf die Lastverteilung und das Online-Verhalten. Abschließend erfolgten Anwendertests unter Produktionsbedingungen.

## Herausforderung JCL

Die große Herausforderung bei der Migration des Host-Systems auf die neue Systemplattform war die Übertragung der über viele Jahre aufgebauten produktiven Batch-Abläufe, ohne die bewährte Geschäftslogik zu gefährden. Das KRZ entschied sich für den JCL-Interpreter J2U, um alle Batch-Jobs ohne jede Änderung der JCL auch auf der neuen Plattform auszuführen. Ob und wann vielleicht einmal Linux-Skripte die JCL ablösen sollen, ist derzeit offen.

„Auch wenn das Niemand glauben wird – es ist uns gelungen, ein solch komplexes Projekt, wie die Ablösung eines Mainframe durch eine Client-Server-Lösung exakt in der geplanten Zeit und dem vereinbarten Budget zu realisieren“, so Reinhard Häcker, Projektleiter vom KRZ. Es habe zwar laufend kleine Änderungen und Verschiebungen gegeben, im Endergebnis habe dann aber doch alles gepasst.

(jcb)



## Die Autoren

Dipl.-Wirt.-Inform. Thomas Kalb Geschäftsführer von der ITGAIN GmbH und ITGAIN Consulting Hamburg. 15 Jahre Berufserfahrung als DB2-Experte für DB2 z/OS und DB2 LUW; DB2 Certified Advanced Database Administrator V8/V9; bei ITGAIN Beratung im DB2-Umfeld und verantwortlich für die Entwicklung von Speedgain for DB2.

Dipl.-Inform. Joachim Roll Geschäftsführer von ITGAIN Consulting Hamburg. Mehr als 20 Jahre Erfahrung im DB2-Bereich (Mainframe) und als DB2-Trainer für die IBM. Bei ITGAIN zuständig für die Beratung im DB2-Umfeld, Migration und Re-Hosting.

