Mainframe Rehosting

Elmar Klausmeier, Steria Mummert Consulting AG

14. August 2011

Zusammenfassung

Das Mainframe Rehosting hat zum Ziel den bestehenden Mainframe durch einen Verbund von leistungsstarken UNIX Rechnern zu ersetzen. Damit sind erhebliche Kosteneinsparungen möglich. Dieser Beitrag zeigt, wann die Gewinnschwelle erreicht wird, welche Hersteller sich in diesem Markt bewegen und welches Vorgehen anzusetzen ist bevor man mit einem solchen Projekt startet.

1 Einleitung

Banken und Finanzdienstleister beschäftigen sich intensiv mit der Renditeoptimierung ihrer Anlagen am Finanzmarkt. Die Darlehnsvergabe bei Banken ist ebenfalls stark renditeorientiert. Der Rendite wird allerdings beim Betrieb des eigenen Rechenzentrums nicht die Aufmerksamkeit geschenkt, wie anderen Finanzanlagen. Dieser Beitrag zeigt, dass die Ablösung der Rechenanlagen vom Monopolanbieter erhebliche Einsparungen erbringt und zu ansehnlichen Renditen führt.

Beim Mainframe Rehosting wird der Mainframe komplett durch einen Verbund aus UNIX (ggf. Windows) Rechnern ersetzt. Für die Endanwender ändert sich nichts. Die Entwicklerproduktivität steigt, da die dezentralen Plattformen mit modernen Entwicklungs- und Testwerkzeugen aufwarten. Die Kosten werden signifikant gesenkt. Die Performance der Anwendungen wird stark verbessert. Die Auswahl an käuflichen und freien Software Produkten ist im UNIX und Windows Bereich erheblich umfangreicher als im Monopolbereich.

Statt von Mainframe Rehosting wird auch von Replattforming, Mainframe Migration oder Mainframe Replacement gesprochen.

Der besondere Charme von Mainframe Rehosting ist, dass im Gegensatz zu einer Neuentwicklung keine aufwendige Konzeptphase vonnöten ist. Das Rehosting Projekt startet direkt bei der Phase Implementierung und kürzt diese durch Verwendung des bisherigen Quellcodes stark ab. Die Integration mit modernen Technologien (C++, Java, PHP, SOA, etc.) ist nach dem Rehosting besonders einfach. Rehosting bietet auch eine Lösung für das Problem, dass erfahrenes und qualifiziertes Mainframe Personal langsam knapp wird.

2 Kostenbetrachtung

Es seien $K_U > 0$ die Kosten für UNIX und zwar für Hard- und Software und für das Migrationsprojekt. Es seien $W_H > 0$ die jährlichen Wartungskosten für den Host und zwar Hard- und Software (Lizenzkosten), sowie einkalkuliert der Verkaufserlös der Host Anlage nach Projektende. Ferner bezeichne $W_U \geq 0$ die jährlichen Wartungskosten für UNIX, die nach Projektende anfallen. Von Interesse ist der Fall $W_U < W_H$. Der Fall $W_U = 0$ träte beispielsweise bei ausschließlicher Nutzung von freier Software (open source) ein. Mit T werde die Projektdauer in Jahren bezeichnet. Beispielsweise bedeutet $T = 1.5$ eine Projektlaufzeit von anderthalb Jahren. Die Variable t sei die Zeit.

Die Momentankosten für UNIX sind dann

$$u(t) = \begin{cases} K_U/T + W_H, & \text{für } 0 \leq t < T, \\ W_U, & \text{für } t \geq T, \end{cases}$$

und die Momentankosten für den Host sind

$$h(t) = W_H, \qquad \text{für } t \geq 0.$$

Die UNIX Kosten summieren sich auf zu

$$\begin{aligned} U(t) &= \int_0^t u(x)dx \\ &= \begin{cases} (K_U/T + W_H)t, & 0 \leq t < T, \\ K_U + W_H T + W_U \cdot (t - T), & t \geq T, \end{cases} \end{aligned}$$

und die Host Kosten summieren sich auf zu

$$H(t) = \int_0^t h(x)dx = W_H t.$$

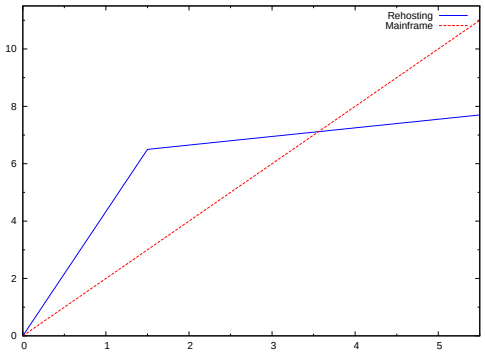
Die Gewinnschwelle t_B (break-even) wird nur für $t_B > T$ erreicht, da $K_U/T > 0$. Die Gewinnschwelle ergibt sich aus

$$U(t_B) = H(t_B).$$

Auflösung nach t_B liefert

$$t_B = T + \frac{K_U}{W_H - W_U}.$$

Die nachfolgende Zeichnung zeigt die Graphen von $U(t)$ und $H(t)$.



Beispielsweise liegt für $T = 1.5$, $K_U = 3.5$, $W_U = 0.3$ und $W_H = 2$ der break-even bei $t_B = 3.5588\ldots$. M.a.W., nach weniger als vier Jahren ist die Gewinnschwelle erreicht, wenn die jährlichen Host Leasing- und Wartungsgebühren zwei Millionen Euro betragen, und das Projekt nach anderthalb Jahren mit einer Investitionssumme von dreieinhalb Millionen Euro abschließt und anschließend 300 Tausend Euro jährliche Wartungsgebühren im UNIX Umfeld fällig sind. Verschiebt sich die Produktivsetzung um ein halbes Jahr, dann verschiebt sich der Break-Even Zeitpunkt um die gleiche Zeitspanne, wenn man voraussetzt, dass keine höheren Kosten K_U anfallen. Dies tritt auf, wenn man einem anderen Projekt Vorrang beim Produktionseinsatz lassen muß.

Der Endwert (future value) des Differenzenzahlungsstroms $\big(h(t) - u(t)\big)$ mit dem Zins r ist

$$F(r,t) = \int_0^t \big(h(x) - u(x)\big)e^{r\cdot(t-x)}dx.$$

Da $t_B > T$, die Gewinnschwelle nach Projektende T liegt, errechnet man

$$F(r,t) = \frac{1}{r}\left[\left(\frac{K_U}{T} + W_H - W_U\right)e^{r(t-T)} - \frac{K_U}{T}e^{rt} - \big(W_H - W_U\big)\right].$$

Gesucht ist der interne Zinsfuß $r(t)$ (internal rate of return), man vergleiche [BL], für den gilt

$$F(r(t),t) = 0.$$

Da ein einfacher formelmäßiger Zusammenhang zwischen r und t gesucht ist, wird die Exponentialfunktion nur bis zum quadratischen Glied benutzt:

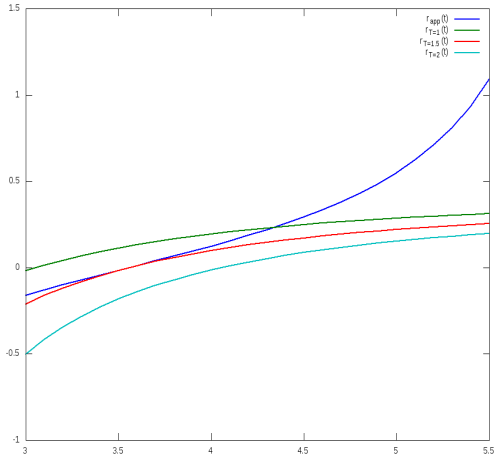
$$e^x = 1 + x + \frac{x^2}{2!} + \cdots$$

Mit Hilfe dieser Vereinfachung findet man

$$r(t) \approx 2\frac{(t-T)(W_H - W_U) - K_U}{(2t-T)K_U - (t-T)^2(W_H - W_U)}.$$

Die angenäherte Renditefunktion hat damit einen hyperbolischen Verlauf. Man erkennt, dass trotz Näherung weiterhin gilt $r(t_B) = 0$. Eine Näherungen kubischer Ordnung liesse sich analog konstruieren, [MX].

Das nachfolgende Bild zeigt den Renditeverlauf für $K_U = 3.5$, $W_H = 2$ und $W_U = 0.3$ für die Projektlaufzeiten $T = 1$, $T = 1.5$ und $T = 2$. Man erkennt, dass die quadratische Approximation (im Bild nur für $T = 1.5$) die positive Rendite überschätzt und die negative unterschätzt.



In der Zeichnung kann man erkennen, dass nach 4 Jahren für $T = 1.5$ die exakte Rendite 9.8%, nach 5 Jahren 22% beträgt.

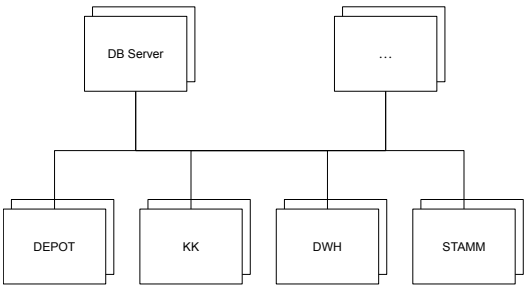
In der Praxis sind Projektlaufzeiten von 6 bis 18 Monaten typisch und realistisch, d.h. $T = 0.5$, $T = 1$ oder $T = 1.5$. Weitere Preisbeispiele findet man in [ME].

3 Rechenanlage

Von den großen Hardware Herstellern gibt es leistungsstarke Rechenmaschinen, die im Verbund ein Mehrfaches der benötigten Rechenleistung erbringen, wie sie der Mainframe bereitstellen kann. Als Beispiele für Rechner auf Basis von Linux (Intel/AMD), die sich gut in einem Verbund zusammen betreiben lassen, seien genannt:

1. HP ProLiant DL980 Server [HP]
2. Oracle (Sun) Fire X4800 Server [Ora]
3. Fujitsu Siemens PRIMERGY RX900 Server [FJS]
4. IBM System x3755 [IBM]

Das nachfolgende Bild skizziert eine Architektur.



Hierbei werden die Rechner entweder paarweise oder in größeren Gruppen zu einem Cluster zusammengeschlossen. Alternativ lassen sich natürlich auch große, monolithische Systeme verwenden, wie z.B. Sun M9000 oder HP 9000 Superdome.

Im obigen Schaubild wurden die Aufgaben nach Sachgebiet und Systemen verteilt. Im Beispiel sind dies die Systeme für Wertpapier und Depot, für Kontokorrent und Zahlungsverkehr, auswertende Systeme auf Basis von Datawarehouse Lösungen, sowie die Stammdatensysteme (Kunden, Wertpapier, Filialorganisation). Aufgrund der hohen Last für die zentrale Datenhaltung wurde die Datenbank auf einen eigenen, leistungsstarken Rechner gelegt.

4 Werkzeuge

Im Folgenden sollen namhafte Hersteller und Werkzeuge aufgelistet werden, welche bei der Migration benötigt werden. Die Aufzählung ist nicht vollständig, repräsentiert aber sicherlich die wichtigsten Produkte und Hersteller.
An COBOL Compilern für dezentrale Plattformen seien genannt:

1. Micro Focus COBOL [MF]
2. AcuCOBOL [MF]
3. Liant RM/COBOL (jetzt auch Micro Focus) [MF]
4. Fujitsu NetCOBOL [NC]
5. OpenCOBOL (open source) [OC]
6. COBOL-IT (basiert auf OpenCOBOL) [CI]
7. Veryant isCOBOL [Vy]

Die Auswahl an PL/I Compilern ist erheblich kleiner:

1. Micro Focus Open PL/I (früher Liant Open PL/I) [MF]
2. IBM PL/I for AIX [IBM]
3. Iron Spring PL/I for OS/2 and Linux [IS]

An CICS Produkten für UNIX und Windows seien genannt:

1. Oracle Tuxedo Application Runtime for CICS and Batch [Ora]
2. Micro Focus Enterprise Edition [MF]
3. Clerity UniKix TPE [CY]
4. HTWC XFRAME [HT]
5. Fujitsu NeoKicks (nur für Windows) [NC]

Alternativ kann man den CICS BMS Datenstrom umwandeln in JSP/HTML/JavaScript, man vergleiche dazu [LW], ferner wird der Transaktionsmonitor CICS durch Tuxedo [Ora] ersetzt.
Der IBM Transaktionsmonitor IMS/DC und die IBM IMS/DB Datenbank können durch eine der folgenden Produkte übernommen werden:

1. Micro Focus Enterprise Edition [MF]
2. Clerity UniKix TPE mit H-RDB [CY]
3. HTWC H2R [HT]

Bei den genannten Produkt bleiben die COBOL und PL/I Programme, die auf IMS zugreifen, unverändert. [Mue] beschreibt die Micro Focus Produkte. Ist nur der IMS Transaktionsmonitor für die Maskensteuerung zu ersetzen, so kann dieser vergleichsweise einfach selber nachprogrammiert werden, in dem man sich die Benutzersteuerung und Lastverteilung durch einen Web-Server zunutze macht.

Für die Portierung von DFSORT und SyncSort Anweisungen kommt man in vielen Fällen mit dem klassischen UNIX sort aus, andernfalls empfiehlt sich eine Prüfung des Ahlsort Produktes, [AS] und [IG].
An JCL/JES Emulatoren seien genannt:

1. Micro Focus Enterprise Edition [MF]
2. Clerity UniKix BPE [CY]
3. HTWC XFRAME (XEBE) [HT]
4. Fujitsu NeoBatch (nur Windows) [NC]
5. IT-gain J2U [IG]

Mithilfe dieser Batch Systeme werden die JCL und OPC Abläufe übernommen. Die eigentliche Job Planung und Ausführung wird üblicherweise durch eine der klassischen Scheduling Systeme erledigt, z.B. UC4, siehe [UC4].

Die eigentliche Migration wird in hohem Maße automatisiert ablaufen. Die zu übernehmenden Programme werden nicht etwa händisch auf die neue Plattform umgearbeitet, sondern mittels Scripten umgestellt. Hier kommt häufig die Scriptsprache Perl zum Einsatz [PE]. Damit ist sichergestellt, dass parallel laufende Projekte bis kurz vor der Schlußabnahme noch Source Code zum Rehosting einliefern können.

5 Vorgehen

Vor dem eigentlichen Mainframe Rehosting ist eine Vorstudie aufzusetzen, die sich mit der Informationsbeschaffung, den Alternativen, den Stolpersteinen und lokalen Gegebenheiten beschäftigt. Eine Beschreibung aus der Praxis findet man beispielsweise bei [BS], bei der die Migration von Bull GCOS8 zu AIX beschrieben wird.
Die Bestandsaufnahme umfaßt:

1. Eigenerstelle Programme
2. Source code
3. Fremdprogramme ohne Quellcode
4. Job Abläufe
5. Netzwerkschnittstellen zwischen Mainframe und anderen Systemen (Messaging, File Transfer)
6. Datenmodell und performance-kritische Tabellen
7. Dokumentation und bisherige Testergebnisse
8. Nicht mehr benötigte Programme
9. Geplante Produktionseinsätze anderer Projekte

Nach der Bestandsaufnahme ist das Inventar zu bewerten: Welche Programme sind zu übernehmen, welche können entfallen, gibt es vom Fremdhersteller auf der neuen Plattform entsprechende Lösungen, ist der Source Code vollständig? Die Vorstudie skizziert das Cluster- und Disaster Recovery Konzept. Ein besonderer Punkt ist die Migration von Assembler Programmen. Die Umwandlung von Assembler nach COBOL kann zwar automatisiert werden, jedoch ist hier stets manuelle Nacharbeit erforderlich. Umso wichtiger ist es im Rahmen der Vorstudie festzustellen, welche Assembler Programme erst gar nicht mehr migriert werden müssen. Die Vorstudie ergibt, ob es zweckmäßig ist den vorhanden Source Code Tool-gestützt zu restrukturieren, das sind Ausschluß toter Code, GOTO Eliminierung und Verringerung, z.B. mit Hilfe der in [SSV] angegebenen Verfahren.

Die Vorstudie sollte mit Hilfe eines Prototyps (proof of concept) den Nachweis führen, dass kritische Prozesse grundsätzlich migrierbar sind. Anhand einer performance-kritischen Datenbanktabelle und einzelnen, damit zusammenhängenden Programmen wird exemplarisch eine Migration durchgeführt und der Nachweis erbracht, dass auf einer dezentralen Plattform grundsätzlich eine ausreichende Geschwindigkeit erzielbar ist. Dies ist wichtig, um im Unternehmen auch gegenüber Zweiflern eine Handhabe zu besitzen. Gleichzeitig wird eine Empfehlung für die zu verwendenden Werkzeuge gegeben.

Die lokalen Gegebenheiten berücksichtigen das Sicherheitskonzept des Unternehmens, Versionsführung, Projektorganisation und Berichtswesen. Schließlich liefert die Vorstudie die voraussichtlichen Kosten K_U, die Projektlaufzeit T für die Migration und benennt die zu erwartenden Wartungsgebühren W_U. Mit diesen Daten kann man das Vorhaben entweder ausschreiben oder vom hauseigenen Dienstleister durchführen lassen. Es sei betont, dass die Größen K_U, W_U und T auch schon vor der Vorstudie geschätzt werden können.

6 Erfahrungen und Ausblick

Hunderte Unternehmen und Behörden aus allen Branchen, Banken, Versicherungen, Automobil, Bildung, Stahl u.s.w. haben ihre Mainframe Anwendungen rehostet bzw. planen dies in naher Zukunft zu tun.

Alle Banken haben die Euro Einführung gemeistert, sie haben den Jahrtausendwechsel erfolgreich bewältigt und die WKN Umstellung durchgeführt. Bei allen diesen Großprojekten waren die vorhanden Systeme, der Source Code und die Schnittstellen zu analysieren. Im Rahmen dieser Projekte sind häufig Analysewerkzeuge und Inventarlisten erstellt worden. All diese Hilfsmittel können beim Rehosting erneut verwendet werden. Unter diesem Gesichtspunkt ist das Rehosting risikoärmer und einfacher als die vorgenannten Projekte. Gemeinsam ist allen genannten Projekten, dass ihr Ursprung technischer Natur ist, jedoch ein gutes fachliches Verständnis bei der Umsetzung und insbesondere beim Test erforderlich ist. Neben der fachlichen Expertise ist ein sorgfältiges Projektmanagement vonnöten, welches die zahlreichen IT Abteilungen, die Abhängigkeiten zu anderen Projekten und die fristgerechte Fertigstellung einzelner Lieferungen überwacht und steuert. Weiterhin ist die Unterstützung durch die oberste Führungsebene notwendig, um Ängsten und Behinderungen entgegenzutreten.

Bei Rehosting Projekten wird vielfach der Wunsch vorgebracht mit der neuen Plattform auch gleichzeitig eine andere Datenbank zu verwenden. Häufig wird der Wechsel von DB2 zu Oracle gewünscht. Gelegentlich besteht auch Interesse von DB2 zu MySQL zu wechseln. Das Rehosting selber erzwingt keinen Wechsel des Datenbankherstellers unter der Voraussetzung, dass der Hersteller die alte und neue Plattform entsprechend unterstützt. Natürlich hat der Wechsel auf MySQL erhebliche Auswirkungen auf die Wartungskosten W_U. Aber dieser Wechsel ist nicht ganz so einfach. Leichter ist es, wenn man seinen bisherigen Datenbankhersteller beibehält. Selbstverständlich kann man hier in zwei Schritten vorgehen: zuerst das Rehosting und anschließend der Wechsel der Datenbank.

Eine Schwierigkeit, die man beim Mainframe Rehosting überwinden muß, ist das den größeren Instituten innewohnende Verharrungsvermögen. In [MM] findet man die Aussage, dass das Mainframe Rehosting ähnlich wirkt, wie der Fall der Berliner Mauer, der Gewinn an Freiheit sei phänomenal.

7 Literatur

[AS] http://www.ahlsort.com/

[BL] Blohm, Hans und Lüder, Klaus: "Investition", 4. Auflage, Verlag Franz Vahlen, München, 1978, *XII*+298 S

[BP] http://www.bphx.com/

[BS] Bach, Johannes und Schulze, Martin: "Das Debeka-Projekt MiKe — Migration der Debeka-Kernanwendungen von Bull/GCOS8 auf AIX", 10. Workshop Software-Reengineering der GI-Fachgruppe Software Reengineering, Bad-Honnef, 05.–07. Mai 2008

[CO] http://www.cobug.com/

[**CI**] http://www.cobol-it.com/

[**CY**] http://www.clerity.com/

[**FJS**] http://www.fujitsu-siemens.de/

[**HP**] http://www.hp.com/

[**HT**] http://www.htwc.com/

[**IBM**] http://www.ibm.com/

[**IG**] http://www.itgain.de/

[**IS**] http://www.iron-spring.com/

[**LW**] Laszewski, Tom and Williamson, Jason: "Oracle Modernization Solutions", Packt Publishing (26. September 2008), Birmingham, 432 p.

[**ME**] META Group: "Mainframe Rehosting Market Evaluation: Tools and Relative Costs", by Corey Ferengul and William Snyder, May 2003, 20 p.

[**MF**] http://www.microfocus.com/

[**MM**] http://mainframemigration.org/

[**Mue**] Müller, Oliver: "Adieu Dino — Mainframe Rehosting mit Micro Focus' Enterprise Server", iX, März 2011, heise Verlag, S.92–97

[**MX**] http://maxima.sourceforge.net/

[**NC**] http://www.netcobol.com/

[**OC**] http://www.opencobol.org/

[**Ora**] http://www.oracle.com/

[**PE**] http://www.perl.org/

[**SSV**] Sellink, Alex and Sneed, Harry and Verhoef, Chris: "Restructuring of COBOL/CICS Legacy Systems", Computer Programming, Volume 45, Issue 2–3, p. 193–243

[**UC4**] http://www.uc4.com/

[**Vy**] http://www.veryant.com/