

BETRIEBSKOSTEN REDUZIEREN

DURCH UMSTIEG AUF JAVA-BASIERTE z/OS-ANWENDUNGEN



PPI Financial Systems ging mit großem Engagement ins Testlabor der IBM nach Böblingen. Im Gepäck das Produkt HBCI-Dienste und das Nachfolgeprodukt TRAVIC-Retail. Drei Fragen sollten die aufwändigen Last- und Performance-Tests zweier Online-Banking-Anwendungen beantworten:

Wie verhalten sich eine in ANSI-C programmierte Anwendung (HBCI-Dienste) und eine in moderner Architektur entwickelte J2EE-Anwendung (Nachfolgeprodukt TRAVIC-Retail) in einer direkten Vergleichsmessung auf einem z/OS-System mit identischer Hardware und identischen Testreihen?

Welchen Einfluss haben die aktuellen Middleware-Entwicklungen für Java-Anwendungen? Konkret: Welchen Effekt hat der neueste WebSphere Application Server (WAS) for z/OS Version V6.1 gegenüber seinem Vorgänger V6.0?

Wie verhält sich die WebSphere-Infrastruktur auf einem z/OS-System mit DB2-Anbindung über Type-2-Treiber (bei lokaler Datenbank) und unter Berücksichtigung der Nutzung von Application-Assist-Prozessoren (zAAPs)?

Die Konkurrenten

Die HBCI-Dienste sind ein Online-Banking-System für die HBCI-Protokolle V2.1 und V2.2 mit Chipkarte oder PIN/TAN. Die zu verarbeitenden Datenströme basieren auf einer Trennzeichensyntax. Die Anwendung ist in der Programmiersprache ANSI-C entwickelt. Als Datenbank kommt DB2 zum Einsatz. Die Anwendung läuft unter z/OS im Transaktionsmonitor CICS mit lokaler Datenbank. Die HBCI-Dienste sind seit vielen Jahren bei mehreren Rechenzentren im Einsatz und bis hin zur aktuellen Version V2.2 R3 in Verbrauch und Performance konsequent optimiert worden.

Das Produkt TRAVIC-Retail ist eine J2EE-Applikation in moderner Architektur, die ebenfalls die Datenbank DB2 nutzt. Verarbeitet wird im Kern das XML-basierte Protokoll FinTS 4.0. TRAVIC-Retail ist darüber hinaus mehrprotokollfähig, d. h. es kann gleichzeitig sowohl per HBCI V2.2 als auch per FinTS 4.0 eingereichte Aufträge verarbeiten. HBCI V2.2 wird von einer Vorkomponente (Servlet) auf FinTS 4.0 konvertiert.

Die IBM hat TRAVIC-Retail bereits in der Konzeptphase mit besonderem Augenmerk auf Performance und Verbrauch begleitet. Im gesamten Entwicklungsverlauf hat die Anwendung konsequent Lasttests absolviert. TRAVIC-Retail ist seit über einem Jahr in Produktion; die aktuelle Version V1.3 arbeitet in einem Rechenzentrum und zwei weiteren Banken.

Java sticht

Die Verbrauchstests bei der IBM umfassten mehrere Testreihen auf identischer Hardware mit gleicher Systemsoftware, in denen die Parallelität stetig erhöht wurde. Dauer und Umfang der Tests und der Testdaten waren auf den Systemen identisch. Basis war das von beiden Anwendungen unterstützte Protokoll HBCI V2.2

mit PIN/TAN. Getestet wurden die typischen HBCI-Geschäftsvorfälle Umsatzabfrage, Überweisung und Kontostandsabfrage. Die Testergebnisse von TRAVIC-Retail beinhalten auch die dort zusätzlich notwendige XML-Verarbeitung.

Die Ergebnisse der CPU-SU-Verbräuche von den HBCI-Diensten und TRAVIC-Retail in den vier verschiedenen Setups sprechen eine deutliche Sprache:

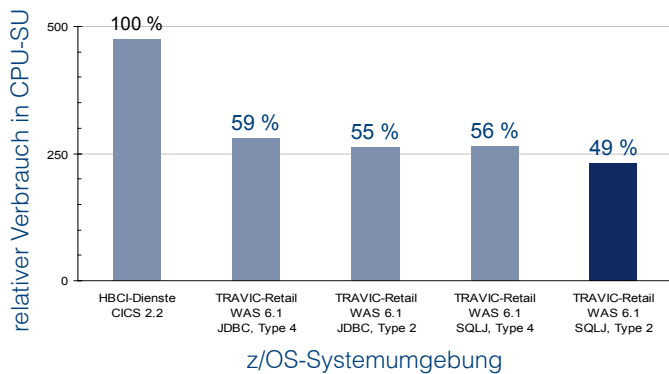


Abbildung 1: Kostensenkungspotenziale von TRAVIC-Retail

Die CPU-SU-Verbräuche wurden zwischen den CICS-basierten HBCI-Dienste unter z/OS 1.7 und dem aktuellen Nachfolgeprodukt TRAVIC-Retail unter WebSphere 6.1 auf z/OS 1.7 verglichen. Die HBCI-Dienste bilden den Referenzwert, dem verschiedene Konfigurationen von TRAVIC-Retail gegenüberstehen.

Die Balken stellen jeweils den Verbrauch in CPU-Service-Units pro Geschäftsvorfall dar und enthalten die prozentuale Einsparung gegenüber den HBCI-Diensten. Die verschiedenen Techniken und Treiber der Datenbankanbindung zeigen verschiedene mögliche Betriebskonstellationen (Type-4-Datenbanktreiber ermöglichen den Betrieb verteilter Datenbanken). Die zu den HBCI-Diensten vergleichbare Konstellation ist die ganz rechts in der Grafik dargestellte TRAVIC-Retail-Konfiguration (SQLJ mit lokaler Datenbankanbindung Type 2).

Die folgende Tabelle zeigt die Gewichtung der Parameter, die in die Berechnung der CPU-Service-Units eingeflossen sind.

Parameter	Erklärung	Wert
CPU	Prozessor	1
SRB	Service-Request	1

Potentiale bei WebSphere

Neben dem oben gezeigten Vergleich der Altanwendung mit dem aktuellen Produkt wurde auch die Verbrauchsreduzierung von WebSphere 6.0 mit JDK 1.4 zu WebSphere 6.1 mit JDK 1.5 gemessen.

Allein der Umstieg von WebSphere 6.0 auf WebSphere 6.1 reduzierte die CPU-Service-Units von TRAVIC-Retail um 20 %, so dass sich der Einsatz von WebSphere 6.1 mit JDK 1.5 in jedem Fall schnell rechnen wird.

Konsequenzen für z/OS

Bei genauerer Betrachtung der Ergebnisse lassen sich für z/OS folgende Aspekte zusammenfassen:

- Die Verwendung lokaler Konnektoren (Type 2) im DB2 spart deutlich Ressourcen.
- Dieser Ressourcenverbrauch könnte auch bei einer JDBC-Verbindung mit Type 4 zu einem gewissen Anteil (< 40 %) auf einem zIIP (Integrated Information Processor) abgearbeitet werden, was weitere Betriebskostenvorteile bedeuten würde.
- Die Verwendung lokaler DB2-Konnektoren bringt neben dem Ressourcen sparenden Verhalten noch weitere Vorteile:
 - einfache Infrastruktur
 - integrierte Security (RACF)
 - schnelles und verlässliches transaktionales Verhalten
 - integriertes Workload-Management
- Das größte Potential für die Einsparung von CPU-Service-Units wird jedoch durch den Einsatz von zAAPs erreicht. Laut den Messergebnissen nutzt TRAVIC-Retail den zAAP zu etwa 85 % aus. Außerdem läuft die WebSphere-Infrastruktur selbst

(Server- und Network-Deployment-Infrastruktur) zu mehr als 90 % auf dem zAAP. In dieser Betriebsart muss für eine Betriebskostenrechnung der einmalige Kaufpreis des zAAP gegengerechnet werden.

Die folgende Grafik zeigt diesen Vergleich eindrucksvoll. Die Balken stellen den Verbrauch in CPU-Service-Units (CPU-SU) pro Geschäftsvorfall auf den General Processors (GP) und dem von der IBM ausgewiesenen zAAP-Verbrauch dar. Als Datenbankanbindung kam SQLJ mit Type 2 zum Einsatz.

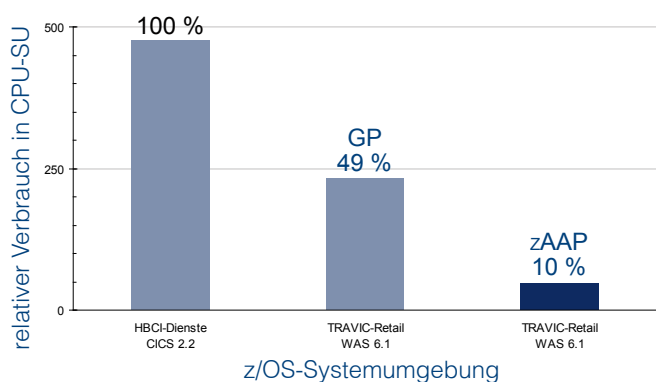


Abbildung 2: Wirkung zAAP-Prozessoren

Analyse im Detail

Eine detaillierte Sicht auf die einzelnen Verbrauchswerte in der WebSphere-Architektur zeigt weitere Optimierungsmöglichkeiten in den einzelnen Teilsystemen. Die ausgewiesenen Teilsysteme Server, Enclave und Controller können durch das zAAP-Modell kostengünstiger betrieben werden.

Um DDF-Verbräuche zu vermeiden, ist eine lokale Type-2-Datenbankanbindung die beste Wahl. Ist diese nicht möglich, weil JDBC und eine Type-4-Datenbankanbindung durch die Architektur gesetzt sind, ist die Verwendung eines zIIPs zu prüfen. Vergleicht man die

Messergebnisse zwischen JDBC/Type-4 und SQLJ/Type-4, verschieben sich abrechnungsrelevante Verbrauchskosten von DDF mit zIIP-Optimierungsoption zu Server/Enclave/Controller mit zAAP-Optimierungsoption. Je nach Architekturentscheidung kann hier ein optimiertes Modell gefunden werden.

Die Verbrauchswerte der Teilsysteme TCP/IP und DB2 können nicht über zAAP- und zIIP-Modelle optimiert werden. Da sie aber sehr geringe Anteile am Gesamtverbrauch haben, werden Optimierungsmöglichkeiten für diese beiden Anteile nicht weiter betrachtet.

Vergleicht man die Gesamtsummen der Verbrauchskosten (281, 267 und 227 SU/GV), liegt die maximal erzielbare Reduzierung bei 54 SU/GV. Dies ist jedoch nur dann darstellbar, wenn alle Architekturentscheidungen in Betracht gezogen werden können. Prozentual entspricht dies einer Einsparung von rund 20 % zwischen der verbrauchsintensivsten und der verbrauchsgünstigsten Konfiguration – im Massentransaktionsgeschäft eine relevante Größe.

Der Einsatz eines zAAP kann laut IBM in der Spitze 80 % der Verbrauchskosten von Server, Enclave und Controller reduzieren, also von den ausgewiesenen 227 SU/GV auf 46 SU/GV (bei SQLJ/Type-2). Für den DDF mit zIIP (bei JDBC/Type-4) ergeben sich laut IBM weitere Einsparungen von ca. 40 %, also von den ausgewiesenen 44 SU/GV auf 26 SU/GV – in der Summe aller möglichen Reduzierungen von 281 SU/GV auf 78 SU/GV. Dies entspricht einer Gesamtreduktion der Verbrauchskosten und damit einer Gesamteinsparung von Transaktionskosten in Höhe von 72 %.

Die Analyse der Messergebnisse zeigt, dass durch die geeignete Wahl der Techniken für die Produktion die Transaktionskosten signifikant gesenkt werden können, ohne die Applikation verändern zu müssen.

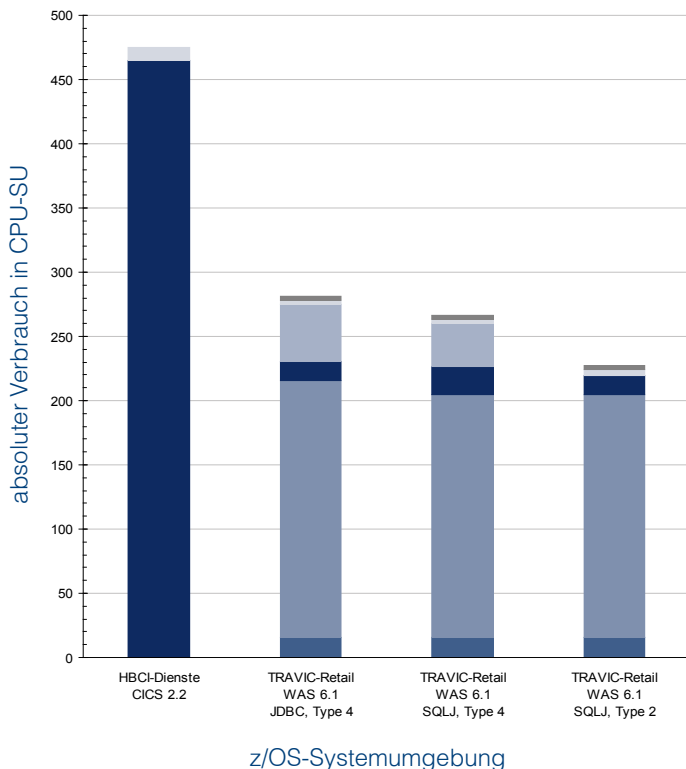


Abbildung 3: zIIP- und zAAP-Reduzierungspotential

	HBCI-Dienste CICS 2.2	TRAVIC-Retail WAS 6.1 JDBC, Type 4	TRAVIC-Retail WAS 6.1 SQLJ, Type 4	TRAVIC-Retail WAS 6.1 SQLJ, Type 2
Summe	475	281	267	227
■ TCP/IP	0	3	3	3
■ DB2	10	3	3	4
■ DDF	0	44	34	0
□ Server	465	15	22	15
■ Enclave	0	200	189	189
■ Controller	0	16	16	16

Fazit und Ausblick

Möglich wurden diese Ergebnisse durch die konsequente Ausrichtung der Anwendungsentwicklung von TRAVIC-Retail auf spätere Betriebskosten sowie durch die konsequente Weiterentwicklung der Prozessorfamilie durch die IBM-Labors.

Das neue Mainframe System z9™ ist optimal an die Nutzung von Java-Programmen angepasst. Die Messergebnisse zeigen, dass eine J2EE-Anwendung wie TRAVIC-Retail schon heute bessere Verbrauchswerte erzielen kann als die klassische CICS-Anwendung HBCI-Dienste.

- Ein Vergleich der aktuellsten WebSphere-Version 6.1 mit dem Vorgänger lässt weitere signifikante Verbrauchs- und Performanceverbesserungen allein durch die Weiterentwicklung auf Middle-ware-Ebene erwarten.
- Das integrierte Deployment von Java-Applikationen mit lokalen DB2-Konnektoren und der Ausnutzung von Spezialprozessoren (zAAP, zIIP) bringt zusätzlich deutliche Kostenvorteile.
- Mit TRAVIC-Retail sind beim Übergang auf das neue Protokoll FinTS 4.0 und dem konsequenten Einsatz der neuen Datagrammtechnik, die erstmalig von FinTS 4.0 angeboten wird, weitere Einsparungen von 40-50 % zu erwarten.

Die Ablösung der CICS-Anwendung HBCI-Dienste durch eine moderne WebSphere-Applikationen TRAVIC-Retail auf dem Mainframe rechnet sich schon nach kurzer Betriebszeit. Die Einsparungen vergrößern sich mittelfristig durch weitere zu erwartende Optimierungen im Bereich der Middleware und der Java-Plattform.



PPI FINANCIAL SYSTEMS

Bei Fragen und für weitere Informationen

PPI Financial Systems GmbH | Michael Schunk | Moorfuhrweg 13 | 22301 Hamburg
Tel.: +49 40 227433-315 | Fax: +49 40 227433-333 | michael.schunk@ppi.de | www.ppi.de