



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Fakultät Informatik

**Automatisierte
Provisionierungsmechanismen für
Laufzeitumgebungen von Legacy z/OS
Anwendungen mit „IBM Cloud
Provisioning and Management for z/OS“
am Beispiel der „Rechnungsschreibung“
bei DATEV eG**

Bachelorarbeit im Studiengang Informatik

vorgelegt von

David Krug

Matrikelnummer 3036355

Erstgutachter: Prof. Dr. Korbinian Riedhammer

Zweitgutachter: Prof. Dr. Friedhelm Stappert

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: _____ Vorname: _____ Matrikel-Nr.: _____

Fakultät: _____ Studiengang: _____

Semester: _____

Titel der Abschlussarbeit:

Ich versichere, dass ich die Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

Die Entscheidung über die vollständige oder auszugsweise Veröffentlichung der Abschlussarbeit liegt grundsätzlich erst einmal allein in der Zuständigkeit der/des studentischen Verfasserin/Verfassers. Nach dem Urheberrechtsgesetz (UrhG) erwirbt die Verfasserin/der Verfasser einer Abschlussarbeit mit Anfertigung ihrer/seiner Arbeit das alleinige Urheberrecht und grundsätzlich auch die hieraus resultierenden Nutzungsrechte wie z.B. Erstveröffentlichung (§ 12 UrhG), Verbreitung (§ 17 UrhG), Vervielfältigung (§ 16 UrhG), Online-Nutzung usw., also alle Rechte, die die nicht-kommerzielle oder kommerzielle Verwertung betreffen.

Die Hochschule und deren Beschäftigte werden Abschlussarbeiten oder Teile davon nicht ohne Zustimmung der/des studentischen Verfasserin/Verfassers veröffentlichen, insbesondere nicht öffentlich zugänglich in die Bibliothek der Hochschule einstellen.

Hiermit ☐ genehmige ich, wenn und soweit keine entgegenstehenden
Vereinbarungen mit Dritten getroffen worden sind,
☐ genehmige ich nicht,

dass die oben genannte Abschlussarbeit durch die Technische Hochschule Nürnberg Georg Simon Ohm, ggf. nach Ablauf einer mittels eines auf der Abschlussarbeit aufgebrachten Sperrvermerks kenntlich gemachten Sperrfrist

von _____ Jahren (0 - 5 Jahren ab Datum der Abgabe der Arbeit),

der Öffentlichkeit zugänglich gemacht wird. Im Falle der Genehmigung erfolgt diese unwiderruflich; hierzu wird der Abschlussarbeit ein Exemplar im digitalisierten PDF-Format auf einem Datenträger beigelegt. Bestimmungen der jeweils geltenden Studien- und Prüfungsordnung über Art und Umfang der im Rahmen der Arbeit abzugebenden Exemplare und Materialien werden hierdurch nicht berührt.

Ort, Datum, Unterschrift Studierende/Studierender

Kurzdarstellung

Deutsche Kurzzusammenfassung Zitattest 1 [[Kuhn 19](#)] Zitattest 2 [[Roge 11](#)]

Abstract

english translation of ‘kurzzusammenfassung‘

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	5
2.1	Customer Information Control System	5
2.2	IBM Cloud Provisioning and Management for z/OS	5
3	Vorgehensweise	7
4	Analyse	9
4.1	Rechnungsschreibung	9
4.1.1	Beschreibung	9
4.1.2	Architektur	12
4.2	Aktueller Bereitstellungsprozess	12
5	Realisierung	13
5.1	Testplex	13
5.2	Entwicklungsumgebung	13
5.3	Nutzwertanalyse	13
6	Ausblick	15
7	Zusammenfassung	17
	Abbildungsverzeichnis	19
	Tabellenverzeichnis	21
	Quellcodeverzeichnis	23
	Literaturverzeichnis	25

Kapitel 1

Einleitung

Vor mehr als fünfzig Jahren wurde der allererste Großrechner, auch Mainframe genannt, vorgestellt. Seit dieser Zeit setzen sich die monolithisch aufgebauten Systemen im Bezug auf Leistungsfähigkeit und Zuverlässigkeit gegenüber andere Systeme ab. Obwohl die Systeme immer weniger Platz brauchten, anfangs waren es ganze Gebäudestockwerke, heute sind es ungefähr die Ausmaße eines großen Kleiderschranks. Und weiteren Verbesserung bei der Handhabbarkeit, von reinen Druckausgaben über text-basierenden Terminals bis hin zu benutzerfreundlichen GUI's. Auch hat sich die Weise, wie Programme entwickelt werden verändert. Zu Beginn mussten diese noch auf Lochkarten (ABBILDUNG !!) gestanzt und umständlich über ein Lesegerät eingelesen werden. Heute stehen dem Entwickler moderne IDE's zur Verfügung. Trotz dieser Veränderungen verlor der Mainframe durch die Dezentralisierung der IT hinzu Client-Server-Umgebungen in den 1990-er Jahren an Bedeutung. Dieser Prozess führte soweit, dass in den frühen 1990-er Jahren bereits Vorhersagen über die Abschaltung des letzten Mainframes getroffen wurden. ¹ [Ceru 03]

Trotz dieser Vorhersagen verarbeiten heutzutage Großrechner weltweit circa 1,2 Millionen CICS² Transaktionen pro Sekunde.³ Im Vergleich hierzu werden 63.000 Google Suchanfragen pro Sekunde abgesetzt.⁴ Wie hat es diese schon seit den frühen 1990-er Jahren totgesagte Technologie geschafft auch heute noch diese Relevanz zu haben? Hier kommen die klar definierte Vorteile und Use-Cases des Mainframes zum tragen. Zunächst ist 'RAS'⁵ zu nennen. Dies beschreibt grundsätzlich, die Stabilität eines Hard- und Softwaresystems. Hierzu zählt vor allem das Verhalten bei einem Hardware-/Softwaredefekt und möglichst automatische Erkennung und möglichst effektive Behebung von diesem. Zusätzlich sollte dies keinen oder nur selten einen kompletten Systemausfall zur Folge haben. Abbildung 1.1 zeigt die ungeplante Server Ausfallzeit in Minuten pro Server im Jahr 2019. Wie zu sehen ist, schneidet der IBM z Systems w/Linux oder z/OS, das Mainframesystem der IBM, am besten ab.

¹[Also 93]

²Begriffserklärung zu CICS in 2.1

³[IBM 19]

⁴[Sull 16]

⁵reliability, availability and serviceability



Abbildung 1.1: Annual amount of unplanned server downtime worldwide in 2019, by hardware platform

Hinzu kommen spezielle Sicherheitsmechanismen und Skalierbarkeit. All dies verbunden mit der durch (HIER SPECS EINFÜGEN) gewährleisteten Performance, ermöglicht spezielle Use-Cases. Unter anderem Massendatenverarbeitung, die dazugehörige Ressourcenverwaltung und Breitband Kommunikation. Das macht den Mainframe vor allem für Banken, das Gesundheitswesen, Versicherungen, Fluggesellschaften usw. attraktiv. Zu diesen Unternehmen zählt auch die DATEV eG. [IBM 14]

Die DATEV eG wurde am 14.02.1966 von 65 Steuerbevollmächtigten gegründet. Sie verfolgten mit der Gründung das Ziel Buchführungsaufgaben mit Hilfe der EDV zu bewältigen. Aufgrund hohen Mitgliederwachstums wurde hierfür 1969 in einen firmeneigenen IBM-Großrechner investiert.[DATE 17] Heute umfasst das Leistungsspektrum der DATEV eG unter anderen das Rechnungswesen, Personalwirtschaft, Consulting, IT-Sicherheit, Weiter-

bildung. Ein nicht unbeträchtlicher Teil (PROZENTSATZ ?) der betriebswirtschaftlichen Anwendungen laufen bis heute auf einem IBM Großrechner. So werden pro Tag circa 150.000 Batch Jobs und circa 90 Millionen CICS-Transaktionen verarbeitet. Hierfür stehen dem System 114.000 MIPS an CPU-Kapazität zur Verfügung. Diese Last wird von circa 14.000 aktiven Modulen erzeugt. Wie in der Abbildung ?? zu sehen ist, ist COBOL mit XXX Prozent die am häufigsten verwendete Programmiersprache am Großrechner bei der DATEV eG. Durch diese Module werden unter anderem im Monat circa 13 Millionen Lohnabrechnungen erstellt und circa 1 Millionen Umsatzsteuer-Voranmeldungen durchgeführt.

Die Risiken die sich für die DATEV eG durch die Nutzung eines IBM Großrechners ergeben, werden im Folgenden dargestellt.

Zunächst ist zu nennen, dass die Verfügbarkeit von Skills im Mainframebereich immer schlechter wird. Die aktuellen Wissenträger fallen durch ihr Alter langsam aus. Durch die geringe Beliebtheit und wenig Präsenz an Universitäten und Hochschulen sind junge Nachfolger nur schwer zu finden. So ist zum Beispiel die Programmiersprache auf den TIOBE Index Platz 28.⁶ Zum anderen gibt es in Deutschland nur XXX Universitäten und Hochschulen die einen Studiengang mit Schwerpunkt Mainframe anbieten.[?]

Als nächstes ist die Herstellerabhängigkeit von IBM zu erwähnen. Die DATEV eG ist nicht nur in der Wahl des Betriebssystems eingeschränkt, sondern auch einem Datenbanksystem oder einer Messaging Lösung. Außerdem hat die IBM eine Quasi-Monopolstellung[?] im Mainframebereich, so ist die DATEV eG auch in ihrer Preisverhandlungspolitik eingeschränkt. Hinzu kommt die Abhängigkeit von der IBM Mainframe Strategie, also ob die IBM selbst noch weitere Ressourcen in ihren Großrechnerbereich investiert. Dies wird durch eine sinkende Kundenzahl am Markt verstärkt.⁷

Für die DATEV eG ist trotz der Risiken eine Ablösung der Mainframe Bestandsanwendungen durch cloud-native Anwendungen aktuell nicht absehbar. Ein weiterer Punkt ist, dass die Sicherung des Bestandsgeschäfts über einen Zeitraum von Jahren oder Jahrzehnten notwendig sein wird. Aufgrund der Stärken des Großrechners will man bei Modernisierungsprojekten eine Alternative für die cloud-native Entwicklung anbieten. Um dieses Ziel zu erreichen wird bei der DATEV eG in den letzten Jahren der Entwicklungsprozess für Mainframe-Projekte überarbeitet und diesen an übliche Entwicklungsstandards anzupassen. So wird auf eine von der IBM entwickelten auf eclipse-basierende Entwicklungsumgebung gesetzt. Somit ist die DATEV eG auch in diesem Bereich von der IBM abhängig. Außerdem läuft seit einiger Zeit die Umstellung auf git für die Verwaltung von COBOL und IBM-Assembler Sourcen.

Zum Entwicklungsprozess zählt jedoch nicht nur das Erzeugen von Code, sondern auch die Bereitstellung der dazugehörigen Laufzeitumgebung. So gewinnen bei der DATEV eG,

⁶[[TIOB 19](#)]

⁷weltweite installierte MIPS-Zahl sei laut IBM steigend

außerhalb des Mainframeumfelds, PaaS (Plattform as a Service) Ansätze immer mehr an Bedeutung. Ein Vorteil dabei ist die unkomplizierte, automatisierte Provisionierung von Laufzeitumgebungen. Dadurch wird die Entwicklungsgeschwindigkeit erhöht und die Bereitstellung von isolierten Testumgebungen vereinfacht. Außerdem können während eines laufenden Entwicklungsprozesses Komponenten, wie zum Beispiel ein Datenbanksystem, hinzugefügt oder ausgetauscht werden. Um sich auch in diesem Bereich der großrechnerfremden Entwicklung anzupassen, sucht die DATEV eG hier nach einer Lösung. Für die automatisierte Provisionierung von Laufzeitumgebungen im Mainframeumfeld stellt die IBM seit dem Jahr 2019 das Tool ‘IBM Cloud Provisioning and Management for z/OS’⁸ zur Verfügung. Anhand des Beispiels der Rechnungsschreibung⁹ wird in dieser Arbeit untersucht, ob und wie es möglich ist, ähnlich wie bei einem PaaS Ansatz, solche Provisionierungsmechanismen für z/OS Anwendungen zu automatisieren.

⁸Beschreibung in Absatz 2.2

⁹Beschreibung im Kapitel 4.1.1

Kapitel 2

Grundlagen

2.1 Customer Information Control System

2.2 IBM Cloud Provisioning and Management for z/OS

Technische Begriffe erklären / Mainframe Begriffe erklären

Kapitel 3

Vorgehensweise

Einarbeitung in die Thematik Analyse Ist-Zustand (inkl. Beschreibung der Anwendung) (evtl. erster Workshop erwähnen) Tool im Testplex (Testumgebung der Admins) zuerst nur die Laufzeitumgebung, dann die Anforderungen der Anwendung nach und nach mit einbauen (HIER KEINE DATEN VORHANDEN), wirklich nur die Umgebung Tool in der Entwicklungsumgebung (Testumgebung der Entwickler) wie auf dem Testplex versuchen und dann noch die Daten und die eigentliche Anwendung mit einbeziehen Am Ende Nutzwertanalyse mit Admins und Entwicklern (Evtl. noch einen Workshop bzw. Vorstellung der Ergebnisse)

Kapitel 4

Analyse

Im Folgendem erfolgt eine Beschreibung der Beispielanwendung ‘Rechnungsschreibung’. Die dafür benötigten Informationen stammen aus Gesprächen mit Mitarbeiter 1 aus der Abteilung, die für die Rechnungsschreibung zuständig ist. Hierbei wird vor allem der technische Aspekt beleuchtet. Anschließend wird der aktuelle Bereitstellungsprozess für Laufzeitumgebungen, den dazugehörigen Datenbanksystem und einer Messaging Lösung dargestellt.

4.1 Rechnungsschreibung

Für diese Arbeit wurde die Rechnungsschreibung als Beispielanwendung herangezogen, weil diese folgende Anforderungen erfüllt. Es handelt sich zum einen um eine in sich abgeschlossene Anwendung, die nur zu Beginn des Prozesses von anderen Anwendungen abhängig ist. Zum anderen benötigt die Rechnungsschreibung ein CICS als Laufzeitumgebung, eine Db2-Datenbank und MQ als Messaginglösung. Somit kann ein umfangreicher Bereitstellungsmechanismus in dieser Arbeit untersucht werden.

4.1.1 Beschreibung

Die Erzeugung der Rechnungen lässt sich in mehrere Schritte unterteilen, gesammelt werden diese Schritte als Rechnungsschreibung bezeichnet.

Bei dem Ablauf handelt es sich um einen Batch¹-Ablauf, der auf einem Großrechner läuft. Nur die Preisermittlung wird in ein CICS ausgelagert. Zunächst wird nach jeder kostenpflichtigen Leistungserbringung durch die dazugehörige Anwendung ein Berechnungssatz erzeugt. Ein Berechnungssatz beinhaltet die Metainformationen der Berechnung unter anderem die Artikelnummer, Menge und den Ordnungsbegriff. Der Preis und der Rechnungsempfänger wird zu einem späteren Zeitpunkt innerhalb der Rechnungsschreibung ermittelt.

Für das Einpflegen der Berechnungssätze in den Rechnungsschreibungsablauf stehen den Anwendungen drei Möglichkeiten zur Verfügung.

¹Stapelverarbeitung

Bei der Ersten Möglichkeit handelt es sich um die Verwendung des DMVIN²-Moduls und der dazugehörigen Schnittstelle. Dieses Modul ist in der Programmiersprache Assembler entwickelt worden. Das Ergebnis dieses Moduls ist eine sequentielle Datei am Großrechner, dieses Format lässt sich mit einer .txt Datei unter Windows vergleichen. Diese Datei, auch Berechnungsdatei genannt, hat folgenden Aufbau. Der erste Satz enthält Steuerinformationen wie zum Beispiel Datum/Uhrzeit, Produkt usw. Danach kommen die eigentlichen Berechnungssätze. Schließlich folgt noch die Anzahl der Sätze und die Summe der einzelnen Artikel in einem Satz mit Kontrollinformationen. Diese Kontrollinformationen werden im weiteren Verlauf mit den eingelesenen Werten abgeglichen, dadurch wird Datenverlust und unkontrollierte Eingriffsmöglichkeiten von außen ausgeschlossen. Aus dem Aufbau einer solchen Datei lässt sich schließen, dass verschiedene Schritte für die Erzeugung innerhalb der Anwendung notwendig sind. Für die nachfolgenden Schritte stellt das DMVIN-Modul jeweils Schnittstellen zur Verfügung. Zuerst wird beim sogenannten Open die Datei erstellt und der Steuersatz geschrieben. Danach folgt das eigentliche Schreiben der Berechnungssätze, dabei dürfen nur bestimmte Felder (Ordnungsbegriffe, Länderschlüssel und Mengen) verändert werden. Um unzulässige Veränderungen zu verhindern, haben diese einen Abbruch der Verarbeitung zur Folge. Schließlich folgt noch der 'Close' bei dem die Kontrollinformationen geschrieben werden. Hinzuzufügen ist, dass die variablen Informationen einer Formalprüfung unterzogen werden. So entstehen je nach fachlicher Logik und Laufhäufigkeit der Anwendung mehrere Berechnungsdateien.

Eine weitere Möglichkeit die Berechnungsinformationen in den Ablauf einzupflegen ist die Übergabe über einen mit der Programmiersprache Java realisierte Webservice. Hier werden die Berechnungsinformationen im XML-Format bereitgestellt. Das Ergebnis der entsprechenden Plausibilitätsprüfungen, die in einem Onlineverfahren durchgeführt werden, wird direkt an die aufrufende Anwendung zurückgegeben. Sind die Daten korrekt werden diese vorerst in einer Datenbank gespeichert. Vor dem nächsten Schritt wird diese Datenbank ausgelesen und mit der ersten Möglichkeit in den Kernablauf eingespeist.

Bei der letzten Möglichkeit handelt es sich um die Übergabe mittels einer CSV-Datei. Die Datei wird auf den Großrechner übertragen und dort mit dem DMVIN-Modul verarbeitet. Dieses Verfahren wird kaum von produktiven Anwendungen sondern hauptsächlich für Test- oder Qualitätssicherungszwecke genutzt.

Mittels dieser drei Möglichkeiten werden insgesamt monatlich circa 30 Millionen Datensätze bereitgestellt und weiterverarbeitet. Diese Datensätze stehen innerhalb der durch das DMVIN-Modul erzeugten Berechnungsdateien dem weiteren Verlauf als Input zur Verfügung. Um sicher zu stellen, dass all diese Dateien auch verarbeitet werden, wird bei Erstellung einer solchen ein Eintrag in eine Kontrolldatei vorgenommen. In dieser Kontrolldatei wird jedes Lesen und somit auch das Lesen im weiteren Verlauf gekennzeichnet. Eine monatliche Überprüfung führt die zuständige Abteilung durch.

²DatevMakroVerarbeitungsinformation

Der nächste Schritt des Rechnungsschreibungsprozesses ist die sogenannte Tägliche Bewertung. Dieser Ablauf läuft einmal täglich von Montag bis Freitag und ist für die Preis- und Rechnungsempfängerermittlung zuständig. Zur Realisierung wurden die Programmiersprachen Assembler und COBOL genutzt. Am Ende dieses Ablaufes steht die ARUBA³-Db2-Datenbank. Dort werden die Berechnungsdaten der letzten 36 Monate aufbewahrt. Dabei handelt es sich um insgesamt circa 3,8 Milliarden Datensätze von einer Gesamtgröße von circa 400 GB mit Indizes. Diese Datensätze beinhalten alle Informationen für die entgeltliche Erzeugung der Rechnungen.

Der erste Schritt der Täglichen Bewertung ist das Zusammenführen der Berechnungsdateien aus dem vorherigen Schritt und aus den bereits vorhandenen Daten des laufenden Monats aus der ARUBA-Db2-Datenbank. Zusätzlich werden während dieser Zusammenführung den Berechnungssätzen auf Basis der abgebenden Anwendung die entsprechenden Rechnungstellungsrythmen (täglich oder monatlich) zugewiesen. Anschließend wird mit Hilfe der Beraternummer die zugehörigen Betriebsstätte-, Rechnungsempfänger-, Hauptberater- und Mitglieds- bzw. Geschäftspartnernummer ermittelt. Die Beraternummer ist als oberster Ordnungsbegriff in den Berechnungssätzen enthalten. Außerdem wird die Debitorenkontonummer entweder durch die Mitglieds- oder durch die Geschäftspartnernummer zugeordnet. Für die Preisermittlung werden die Datensätze nach Geschäftspartner gruppiert. Im DATEV eG Umfeld ist ein Geschäftspartner entweder eine Kanzlei oder ein einzelner Mandant, dieser ist jedoch meist einer Kanzlei zugeordnet. Dann werden die gruppierten Daten aus Performancengründen an ein CICS übertragen. Die Architektur wird in 4.1.2 beschrieben. Dort findet die Preisermittlung mit den dazugehörigen kundenindividuellen Abhängigkeiten, wie zum Beispiel Rabatte, statt. Anschließend werden die Daten wieder zurück an den Batch-Ablauf übertragen. Hier werden die Rechnungsnettoeträge geprüft, ob diese über einem bestimmten Rechnungslimitbetrag liegen. Falls dies nicht der Fall ist, werden die Berechnungssätze als BUL⁴ gekennzeichnet und in die folgende Rechnungsperiode vorgetragen. Schließlich wird noch die Umsatzsteuer ermittelt. Abschließend werden die neu erzeugten Berechnungsdaten in die ARUBA-Db2-Datenbank eingepflegt und entsprechend gekennzeichnet.

Als letzter Schritt folgt die Rechnungsaufbereitung. Diese erfolgt am ersten Werktag im Monat. Mit Hilfe der ARUBA-Db2-Datenbank wird ermittelt, welchen Kunden eine Rechnung zugestellt werden muss. Außerdem wird dabei der Zustellungsweg, per Post oder E-Mail, bestimmt. Darauf folgt die Aufbereitung der Druckrohdaten und letztlich das Versenden der Rechnungen an die Berater. Zusätzlich werden die Rechnungen noch im PDF Format archiviert.

³Abrechnungs- und Umsatz-Basis

⁴Beratuer unter Limit

4.1.2 Architektur

In dieser Arbeit steht das automatisierte Provisionieren von Laufzeitumgebungen im Fokus. In diesem Fall handelt es sich um die Laufzeitumgebung CICS. Deshalb wird im Folgenden nur darauf eingegangen.

Das System muss an Lasttagen bis zu 180.000 Geschäftspartner verarbeiten können. Um all diese an das CICS zu übertragen stehen dem System mehrere IBM MQ Queues zur Verfügung. Darunter ist eine allgemeine Queue in der alle Aufträge, die für die Weiterverarbeitung zur Verfügung stehen, geschrieben werden. Pro Geschäftspartner wird ein Auftrag angelegt. In diesem Auftrag befinden sich die Namen vier weiterer Queues. Eine dieser Queues beinhaltet alle Informationen, die für die Preisermittlung des dazugehörigen Geschäftspartners notwendig sind. In den restlichen drei Queues sind die Ergebnisse der Preisermittlung gespeichert. Die Queuenamen werden nicht dynamisch generiert, da dies zu Performanceproblemen führt. Deshalb existieren für jede der vier Queues jeweils 100 vorgefertigte Namen. Somit können auch maximal nur 100 Aufträge gleichzeitig auf Weiterverarbeitung warten. Falls dieses Limit erreicht ist, wartet der Batch-Ablauf bis ein Auftrag fertig wird. Sobald ein Auftrag in die allgemeinen Auftragsqueue geschrieben wird, wird eine CICS-Transaktionen gestartet. Diese führt die Preisermittlung durch und schreibt das Ergebnis auf die dazugehörigen Queues. Ist dies geschehen stehen die Queues wieder für einen neuen Auftrag zur Verfügung. Es können maximal 30 Transaktionen zeitgleich arbeiten.

Für die Preisermittlung wird auch eine Db2-Datenbank, in der die Einzelpreise der Artikel gespeichert sind, verwendet. Wenn alle Transaktionen direkt auf diese Datenbank zugreifen würden, hätte dies über 60 Millionen Datenbankzugriffe zur Folge. Dies führt zu massiven Performancenproblemen. Deshalb werden bevor die eigentliche Preisermittlung stattfindet, alle benötigten Einzelpreise und Preisabhängigkeiten ermittelt. Diese Informationen werden dann in einen sogenannten 'SHARED GETMAIN'-Bereich gespeichert. Dabei handelt es sich im Prinzip um einen Hauptspeicherbereich des Großrechners. Die Adresse von diesem Bereich wird dem Ablauf zur Verfügung gestellt. Somit greifen die einzelnen Transaktionen nicht mehr direkt auf die Datenbank zu, sondern auf den schnelleren Hauptspeicher.

4.2 Aktueller Bereitstellungsprozess

Mit vielen anderen Abteilungen sprechen

Viel auf 'Zuruf' und Besprechungen

Genauere Infos noch von den CICSAdmins nachfragen

Kapitel 5

Realisierung

Umsetzung mit Tool

5.1 Testplex

5.2 Entwicklungsumgebung

5.3 Nutzwertanalyse

Kapitel 6

Ausblick

Bezug auf den Anfang Integrieren in eine Buildpipeline Möglichkeit (NUR MÖGLICHKEIT)
Bereitstellen der CICSe auch in Produktion

Kapitel 7

Zusammenfassung

Abbildungsverzeichnis

1.1 Annual amount of unplanned server downtime worldwide in 2019, by hardware platform	2
--	---

Tabellenverzeichnis

Quellcodeverzeichnis

Literaturverzeichnis

- [Also 93] S. Alsop. “IBM still has the brains to be a player in client/server platforms”. *InfoWorld*, Vol. 15, No. 10, p. 4, 1993.
- [Ceru 03] P. E. Ceruzzi. *A history of modern computing. History of computing*, MIT Press, Cambridge, Mass., 2. ed. Ed., 2003.
- [DATE 17] DATEV eG. “Geschichte der Datev”. 2017.
- [IBM 14] IBM. “Who uses mainframes and why do they do it?”. 2014.
- [IBM 19] IBM. “CICS Application Server Software for IBM Z”. 2019.
- [Kuhn 19] J. B. Kühnapfel. *Nutzwertanalysen in Marketing und Vertrieb. essentials*, Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2. auflage 2019 Ed., 2019.
- [Roge 11] P. Rogers. *ABCs of z/OS system programming: Volume 4. IBM redbooks*, IBM International Technical Support Organization, Poughkeepsie, N.Y.?, 2011.
- [Sull 16] D. Sullivan. “Google now handles at least 2 trillion searches per year - Search Engine Land”. 2016.
- [TIOB 19] TIOBE Software BV. “TIOBE Index | TIOBE - The Software Quality Company”. 25.11.2019.