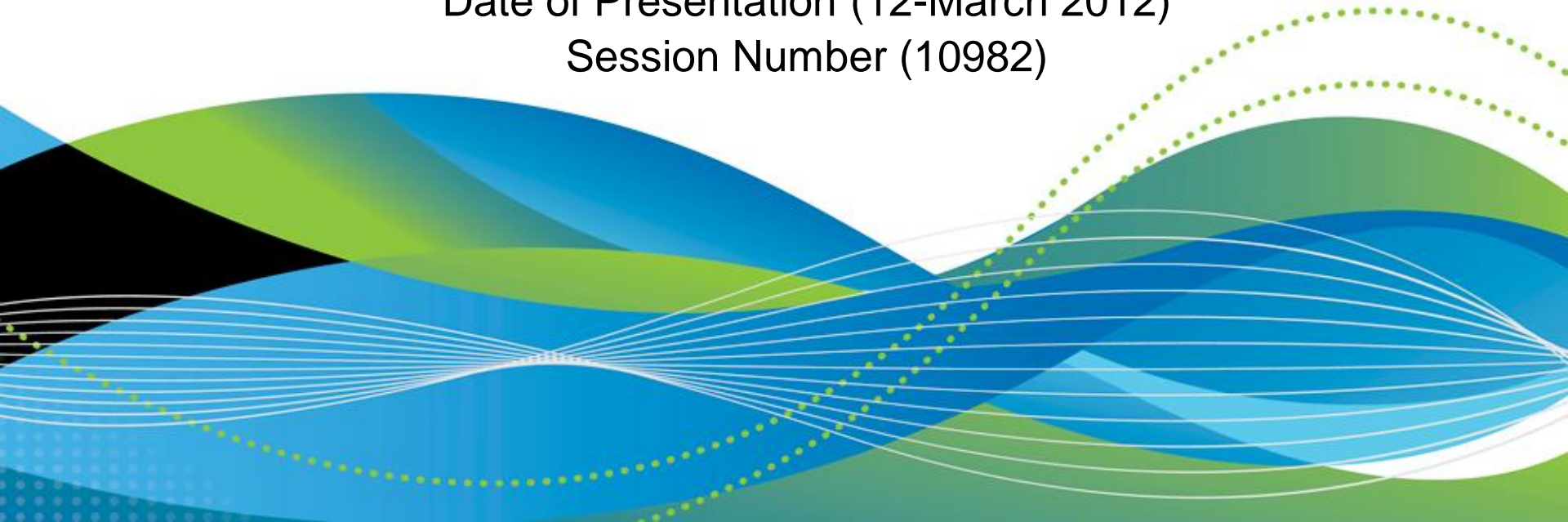


COBOL – VSAM Reporting using Java on Tomcat and ITEXT

Ramanathan Perinkolam
Tata Consultancy Services

Date of Presentation (12-March 2012)
Session Number (10982)



Agenda

- Objective and Scope
- Technical Environment – JZ/OS, ITEXT, TOMCAT
- Traditional Architecture
- Re-engineered Architecture
- Data Slicing in Re-engineering approach
- Enriched Graphical Representation of Report
- Traditional vs. Re-engineered – MIPS Usage
- Inference
- ROI
- References
- IBM Centre of Excellence – An Overview

Objective and Scope

Objective:

To Modernize a COBOL-VSAM Reporting Solutions using Java running on Tomcat/zOS and iText.

Scope:

- Install Tomcat on zOS
- Re-engineer COBOL Reporting Components to Java with access to VSAM enabled using JZOS on Tomcat
- Build Data Slicing functionality in Java as per the needs of the customer
- Integrate Java components with iText API with enriched Graphical Representation of reports
- Comparison in MIPS Consumption between Traditional Approach and Re-engineered approach

Technical Environment

- Enterprise Cobol for Z/OS 4.1
- IBM Java 1.5 on Z/OS
- Apache Tomcat V6 to be installed on Z/OS
- Apache Tomcat V6 to be installed on Distributed System to communicate with the Apache Tomcat running on Mainframe.
- iText V5.0 API to be added in the Java Program which is running in the distributed System

Introduction – JZOS, ITEXT

- **JZOS:**

JZOS is a part of IBM Java sdk's for z/OS. This Java programs enables access methods for accessing MVS assets like sequential files, partitioned datasets, VSAM files and DB2 (through JDBC).

- **iText :**

- iText is a free and open source library for creating and manipulating PDF files in Java.
- Versions of iText up to 2.1.7 were distributed under the Mozilla Public License or the LGPL and supported generating reports in RTF and HTML formats.
- Latest version 5.0.0 (released Dec 7, 2009) is distributed under the Affero General Public License version 3.
- iText is predominantly used for
 - Generate dynamic documents from XML file or databases
 - Use PDF's many interactive features
 - Add bookmarks, page numbers, watermarks, barcodes, etc.
 - Automate filling out PDF forms
 - Add digital signatures to a PDF file and many more

Introduction – Apache TOMCAT



Apache tomcat for z/OS:

Features:

- Latest version of Tomcat on Z/OS is 6.0.18.
- Additional Dovetailed Technologies developed JARs are included to enhance Tomcat on z/OS
- A Tomcat Security Realm for authenticating users and roles using SAF(RACF
- The JCL provided allows Tomcat to run as a batch job or started task using the IBM Java SD
- The configuration includes support for DB2 JDBC type-2 and type-4 Datasource connections.

System Requirements:

- One of the following Java SDKs:
 - IBM 31 or 64-bit SDK for z/OS, Java Technology Edition, V5, SR3 or later
 - IBM 31 or 64-bit SDK for z/OS, Java Technology Edition, V6
- 10 MB of HFS or zFS file system space, plus any space required for your web applications.

Licensing:

- Apache Tomcat is a pure-Java implementation of the Java Servlet and JavaServer Pages technologies and is licensed under the Apache Version 2.0 Open source license.

Introduction – Apache TOMCAT



Apache TOMCAT Security Features:

The Configuration File in Apache Tomcat Provides following Security Features

TOMCAT – users.xml

It enables the user to set the User id and Password for Security roles

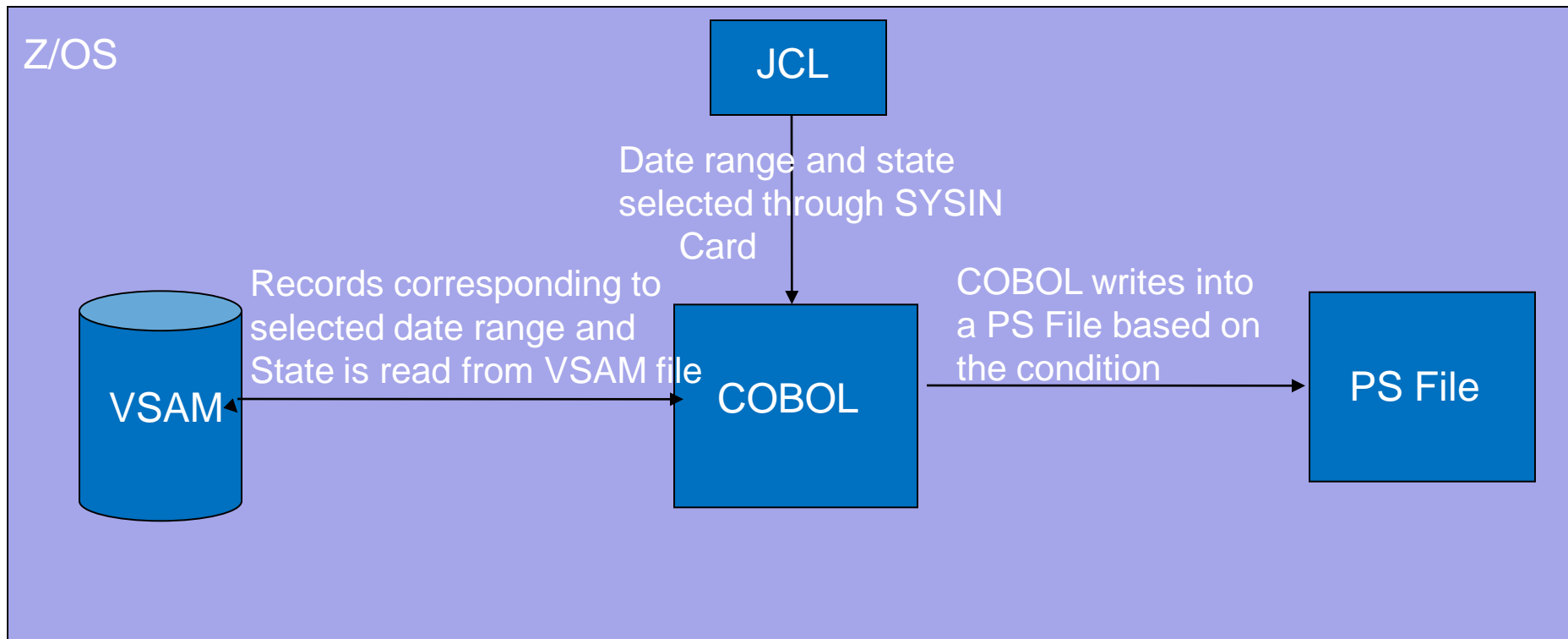
Web.xml

It contains Session Tracking using a JSESSIONID that is generated in random with a sequences of ID's which are not reproducible. This prevents a remote user from hijacking a client's session.

Server.xml

It enables to configure to Hyper Text Transfer Protocol – Secure Connection.

COBOL – VSAM Reporting – Traditional Architecture



Scenario :

- COBOL Program Reports CardNo Transaction Data based on Inputs (Data Range and / or State) passed via the JCL to the program
- The COBOL Program reads data from the Transaction master File(VSAM)
- The Program scans through the complete file and extracts records based on the selected date range and the state
- The Selected records are formatted and written as report into the PS file which is transferred to the distributed environment for analysis

COBOL – VSAM Reporting – Traditional Approach

Scenario

JCL in which date range and state are specified through SYSIN

```
EDIT          COEGRP.VSAM.JCL (RUNPROG) - 01.36          Columns
Command ==> _____ Scro
***** ***** Top of Data *****
000100 //VSAMRUN  JOB (TRG,TRG,TRGB99,D2,DT99X), 'RUNJCL',PRTY=13,
000200 //          MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000300 //STEP1    EXEC PGM=PROG2
000400 //STEPLIB  DD DSN=COEGRP.VSAM.LOADLIB,DISP=SHR
000500 //SYSOUT    DD SYSOUT=*
000600 //DD1        DD DSN=COEGRP.JAVA.VSAM7.CLUSTER,DISP=SHR
000700 //DD2        DD DSN=COEGRP.VSAM.PSFILE1,DISP=MOD
000800 //DD3        DD DSN=COEGRP.VSAM.PSFILE1,DISP=SHR
001000 //SYSIN      DD *
001100 Alabama
001200 20110512
001210 20020112
001300 /*
```

State and the data range which is given as input to COBOL

COBOL – VSAM Reporting – Traditional Approach

Scenario



COBOL Report generated in a PS File based

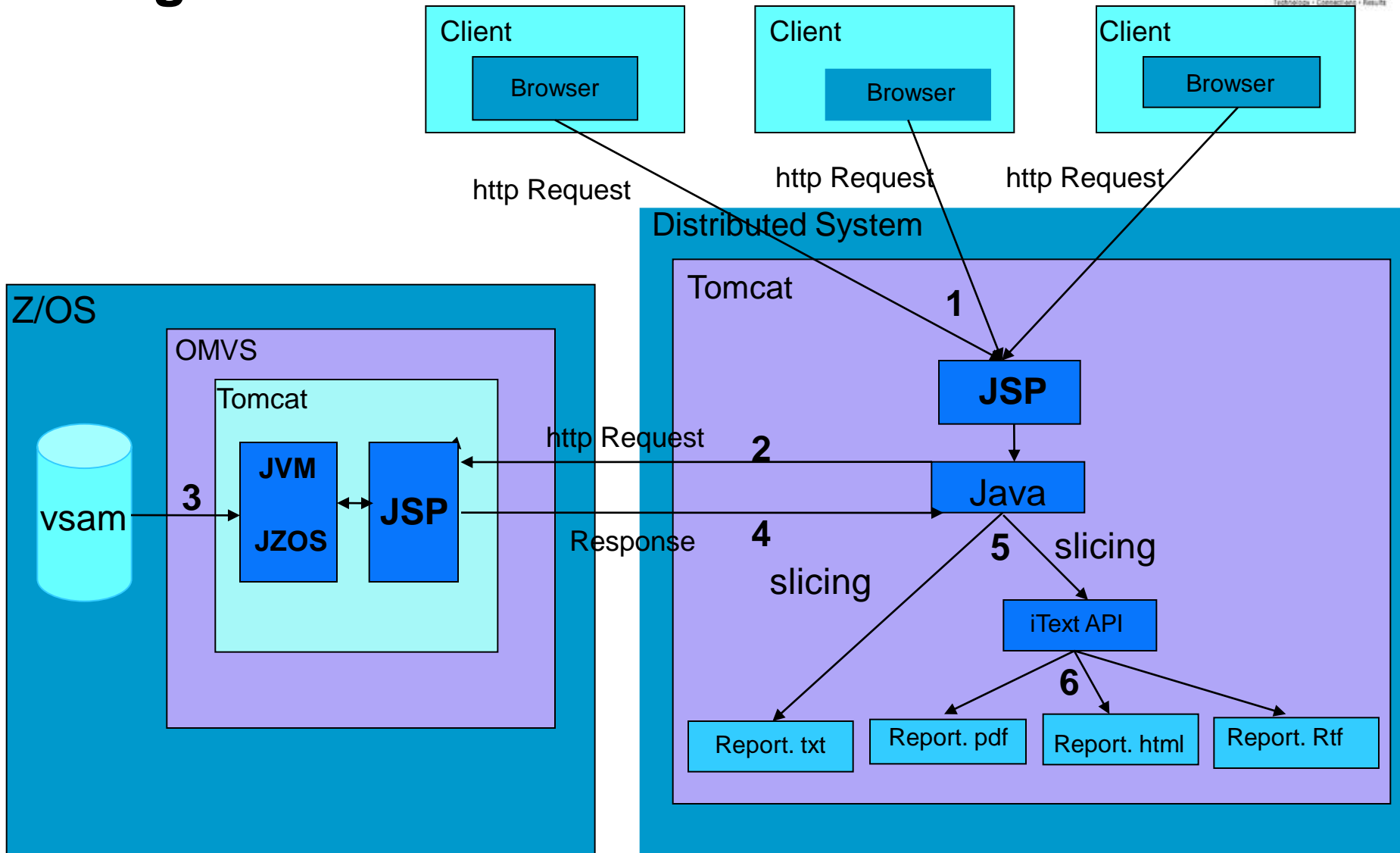
```

EDIT          COEGRP . VSAM . PSFILE1          Colu
Command ==> _____ S
*****
***** Top of Data *****

```

000001	CARD NUMBER	DATE	STATE	AMOUNT
000002	4552760798751003	20100730	Alabama	1842600
000003	4552760798751011	20060630	Alabama	1141900
000004	4552760798751011	20020720	Alabama	1906400
000005	4552760798751030	20080816	Alabama	1405900
000006	4552760798751042	20110111	Alabama	1558800
000007	4552760798751052	20030626	Alabama	1209000
000008	4552760798751060	20020915	Alabama	1088000
000009	4552760798751062	20090927	Alabama	1814000
000010	4552760798751069	20090723	Alabama	1657800
000011	4552760798751070	20070528	Alabama	1731500
000012	4552760798751072	20080312	Alabama	1061600
000013	4552760798751073	20020928	Alabama	1691900
000014	4552760798751078	20090717	Alabama	1312400
000015	4552760798751089	20040410	Alabama	1265100
000016	4552760798751096	20040920	Alabama	1464200
000017	4552760798751098	20060628	Alabama	1925100
000018	4552760798751108	20080727	Alabama	1841500

Java on Tomcat/zOS using iText – Reengineered Architecture



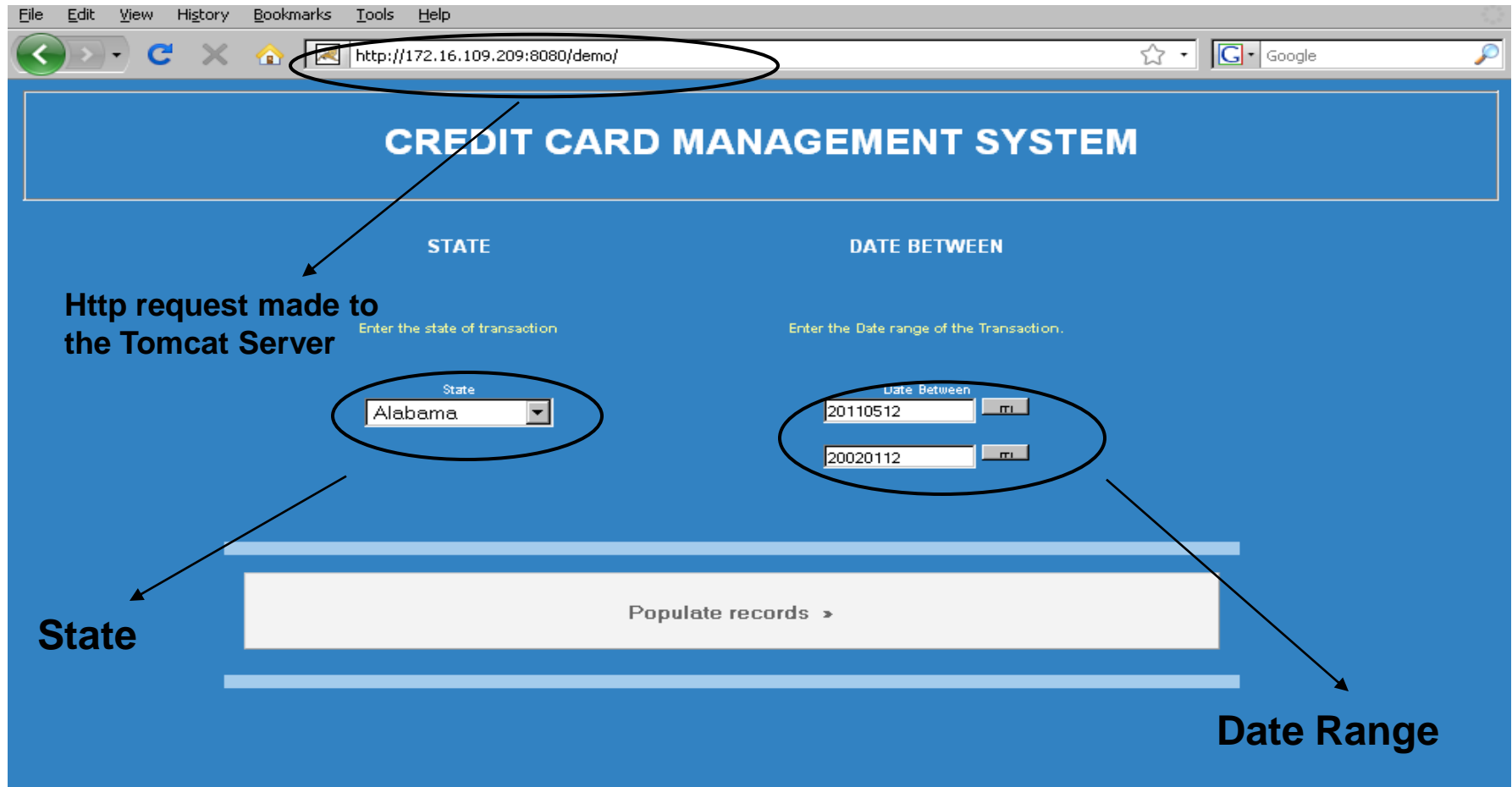
Java on Tomcat/zOS using iText – Reengineered Approach



1. The browser from the client side makes a http request to the Tomcat Server running on the distributed system .
2. JSP page invokes the Java component.
3. The Java Component in-turn makes a http request to the Tomcat Server running on Mainframe.
4. JZOS running in the Tomcat fetches the records from the VSAM.
5. The fetched records are then passed to the Java running in the distributed system in XML format.
6. The Java program performs the slicing operation on the fetched records .
7. The records are then generated in the form of PDF,HTML,RTF using iText API .

Java on Tomcat/zOS using iText – Reengineered Approach

The browser from the client side makes a http request to the Tomcat Server running on the distributed system .



File Edit View History Bookmarks Tools Help

http://172.16.109.209:8080/demo/

CREDIT CARD MANAGEMENT SYSTEM

STATE DATE BETWEEN

Enter the state of transaction Enter the Date range of the Transaction.

State Alabama

Date Between 20110512 20020112

Populate records >

Http request made to the Tomcat Server

State

Date Range

Java on Tomcat/zOS using iText – Reengineered Approach

The fetched records are then passed to the Java running in the distributed system .

```
File Edit Format View Help
<HTML>
<HEAD>
  <TITLE>VSAMData</TITLE>
</HEAD>
<BODY>
  <h1>VSAMData</h1>
<START>
455276079875100317974020100730AAlabama1842800
455276079875101116333520060630AAlabama1141900
455276079875101117135820020720AAlabama1906400
455276079875103014922120080816AAlabama1405900
455276079875104216068820110111AAlabama1558800
455276079875105215332420030626AAlabama1209000
455276079875106013340520020915AAlabama1088000
455276079875106214360520090927AAlabama1814000
455276079875106919424920090723AAlabama1657800
455276079875107015730820070528AAlabama1731500
455276079875107218383120080312AAlabama1061600
455276079875107314967820020928AAlabama1691900
455276079875107816569920090717AAlabama1312400
455276079875108910587220040410AAlabama1265100
455276079875109613071720040920AAlabama1464200
455276079875109818156420060628AAlabama1925100
455276079875110816270320080727AAlabama1841500
455276079875111514311020080118AAlabama1227700
<END>
</BODY>
</HTML>
```

→ **Fetches Records**

Data Slicing in Re-engineered Approach

File Edit View History Bookmarks Tools Help

http://172.16.109.209:8080/demo/package.jsp?data2=Alabama&data1=20110512&data3=20020112

CREDIT CARD MANAGEMENT SYSTEM

CARD NUMBER	DATE BETWEEN	STATE	AMOUNT BETWEEN
Enter the card number to be selected	Enter the Date range of the Transaction.	Enter the state of transaction	Enter the Amount range.
Card Number <input type="text"/>	Date Between <input type="text" value="20110512"/> <input type="text" value="20020112"/>	State <input type="text" value="Alabama"/>	Amount Between <input type="text" value="1200000"/> <input type="text" value="1550000"/>

First level filter disabled

Amount range is specified

[Edit / Slice up »](#) [Generate PDF Report »](#)

Click to slice based on the amount

Card number	Date	State	Amount
4552760798751003	20100730	Alabama	1842600
4552760798751011	20060630	Alabama	1141900
4552760798751011	20020720	Alabama	1906400
4552760798751030	20080816	Alabama	1405900
4552760798751042	20110111	Alabama	1558800
4552760798751052	20030626	Alabama	1209000
4552760798751060	20020915	Alabama	1088000
4552760798751062	20090927	Alabama	1814000
4552760798751069	20090723	Alabama	1657800
4552760798751070	20070528	Alabama	1731500

Data Slicing in Re-engineered Approach

Horizontal Forms symantically - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://172.16.109.209:8080/demo/EditData.jsp?sessionId=1306911548835&date=20110512&city=Alabama&date=20020112%2F&cardNo=&compVar1=

CREDIT CARD MANAGEMENT SYSTEM

CARD NUMBER DATE BETWEEN STATE AMOUNT BETWEEN

Enter the card number to be selected Enter the Date range of the Transaction. Enter the state of transaction Enter the Amount range.

Card Number Date Between State Amount Between

20110512 20020112/ Alabama 1200000

20020112/ 1550000

Both filters disabled

Edit / Slice up » Generate PDF Report »

Click to generate PDF Report

Card number	Date	State	Amount
4552760798751030	20080816	Alabama	1405900
4552760798751052	20030626	Alabama	1209000
4552760798751078	20090717	Alabama	1312400
4552760798751089	20040410	Alabama	1265100
4552760798751096	20040920	Alabama	1464200
4552760798751115	20080118	Alabama	1227700
4552760798751135	20070117	Alabama	1378400
4552760798751137	20040624	Alabama	1445500
4552760798751144	20100130	Alabama	1218900
4552760798751168	20090420	Alabama	1368500

Done

Enriched Graphical representation of Report in Reengineered approach




File Edit View History Bookmarks Tools Help

http://172.16.109.209:8080/pdfReports/1305625252911Alabama20110512.pdf

CREDIT CARD TRANSACTION DETAILS 1305625252911Alabama201105...

1 / 14 114% Find

CREDIT CARD TRANSACTION SUMMARY

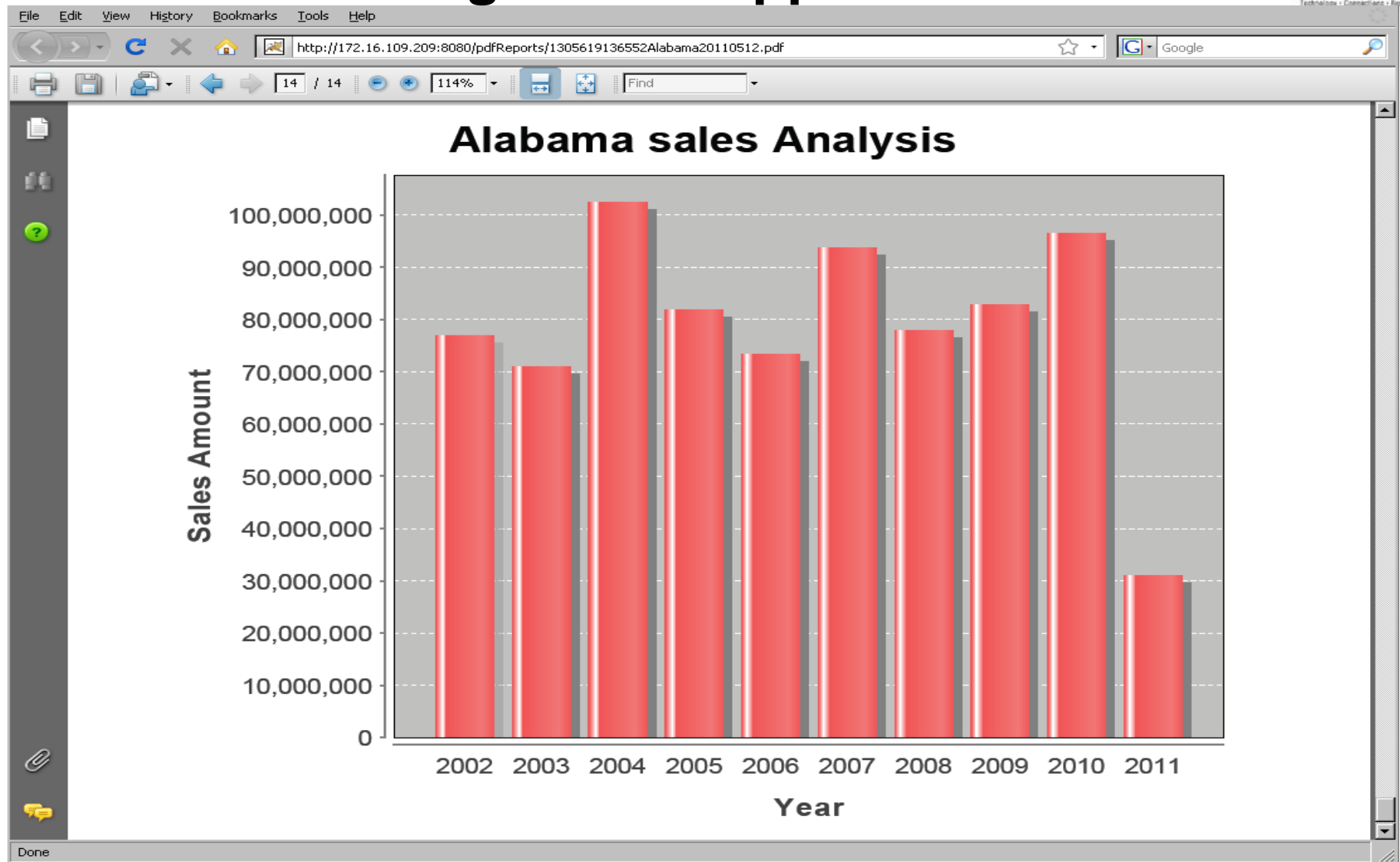


Alabama20110512

CREDIT CARD NUMBER	DATE	CITY	AMOUNT
4552760798751030	20080816	Alabama	1405900
4552760798751052	20030626	Alabama	1209000
4552760798751078	20090717	Alabama	1312400
4552760798751089	20040410	Alabama	1265100
4552760798751096	20040920	Alabama	1464200
4552760798751115	20080118	Alabama	1227700
4552760798751135	20070117	Alabama	1378400
4552760798751137	20040624	Alabama	1445500
4552760798751144	20100130	Alabama	1218900
4552760798751168	20090420	Alabama	1368500
4552760798751171	20100921	Alabama	1258500
4552760798751177	20070327	Alabama	1333300
4552760798751185	20090216	Alabama	1214500
4552760798751191	20060721	Alabama	1522500
4552760798751199	20050215	Alabama	1305800
4552760798751212	20050514	Alabama	1210100
4552760798751235	20090214	Alabama	1448800

Done

Enriched Graphical representation of Report in Reengineered approach



Traditional vs Re-engineered Approach - MIPS



Case 1 : Fixed Date Range and Variable State name

Case 2 : Variable Date Range and Fixed State name

Case 3 : Fixed Date Range and Fixed State name

	Reporting Run(s)	Measurement Window (min)	Java		COBOL
			CP (sec)	Zaap (sec)	CP (sec)
Case 1	600	15	16	142	144
Case 2	600	15	17	146	144
Case3	600	15	16	140	144
Average			16.33	142.66	144
Average per Run			0.027	0.237	0.24

Inference

Inference :

- Re-engineered UI which allows the user to perform multi-level slicing on the date range and state.
- Modernized Graphical Report Representation using iText for various Data-Slices
- Re-engineering the application layer and retaining only the Data Access Layer (VSAM) on the Mainframe is expected to significantly offload the MIPS to zAAP Speciality Engines and reduce the CP MIPS Footprint . Expected offload to Speciality Engines (89.72 %) and reduction of MIPS Footprint (88.65 %)
- The CP Time in COBOL Varies according to the number of Steps executed through JCL and accordingly there will be MIPS Reduction for Java

ROI



ROI MODEL

	Year-1	Year-2	Year-3	Year-4	Year-5	Remarks/Assumptions
zAAP Engine Cost	125000	0	0	0	0	
Re-engineering Cost	92,656	0	0	0	0	Assuming an Application Size of 50 COBOL Components (~37500 LOC)
Support Cost	20,000	20,000	20,000	20,000	20,000	Maintenance of TOMCAT , iTEXT Environment
Cumulative Cost	237,656	257,656	277,656	297,656	317,656	
Annual Savings						
License Cost Savings - COBOL	28,554	28,554	28,554	28,554	28,554	Assumed 200 MIPS (25 MSU) reduction in COBOL MIPS
License Cost Savings - zOS	101,671	101,671	101,671	101,671	101,671	Assumed 75 % Reduction in zOS MIPS i.e from 200 MIPS (25 MSU) to 50 MIPS(7 MSU) on account of offload to zAAP Engine
Cumulative Savings	130,225	260,450	390,675	520,900	651,125	
ROI	0	2,794	113,019	223,244	333,469	Investments recovered from Year-2 and subsequent Year on Year Savings

Note: Reengineering investments recovered from Year-2 onwards



ROI Calculation

References

References :

Tomcat Download

<http://dovetail.com/downloads/tomcat/index.html>

Tomcat installation

<http://dovetail.com/docs/tomcat/install.html>

Tomcat installation doc

<http://dovetail.com/docs/tomcat/install.html>

Dovetail tomcat forum

<http://dovetail.com/forum/>

iText download

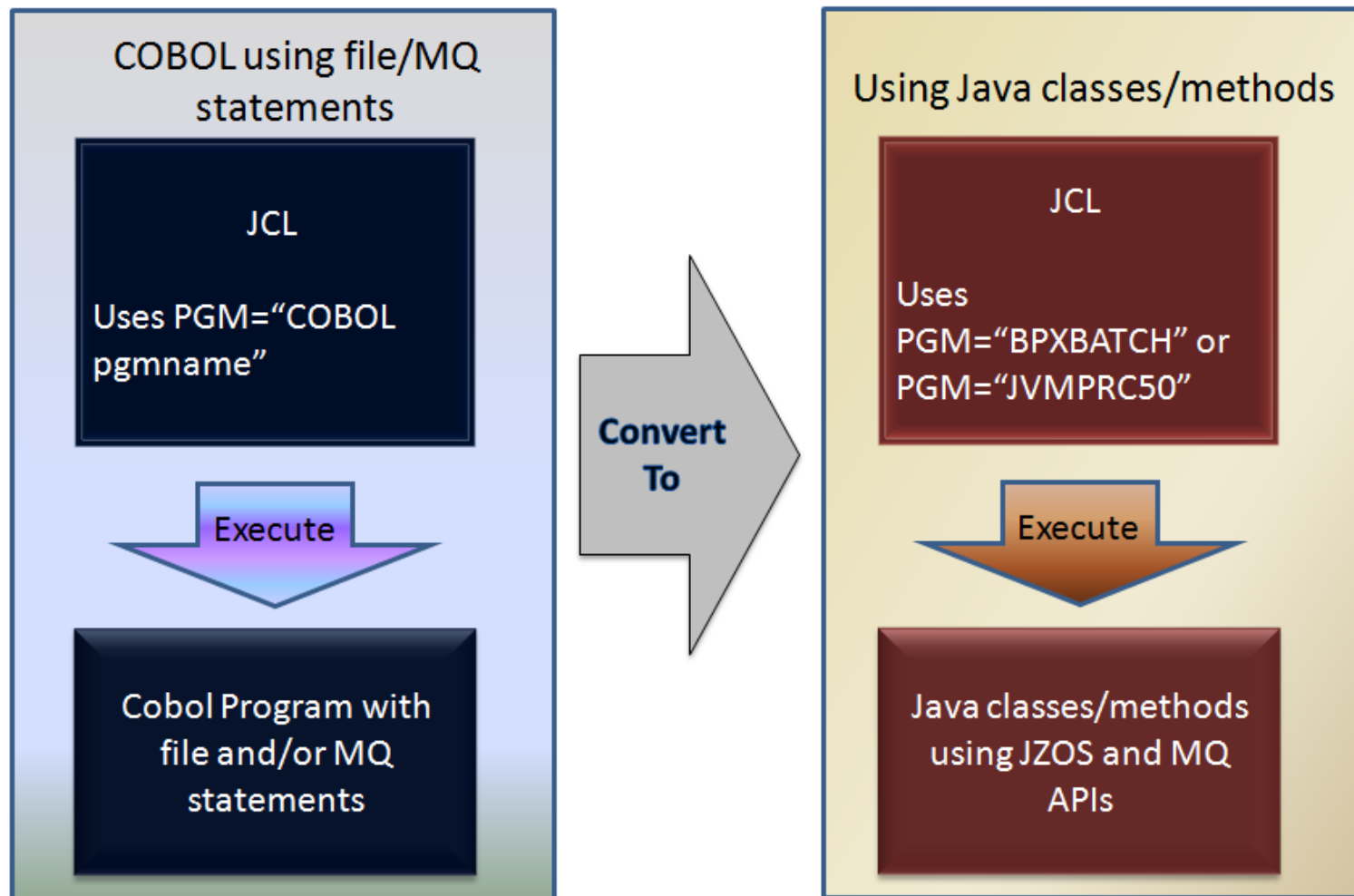
<http://itextpdf.com/download.php>

iText forum

<http://support.itextpdf.com/forum>

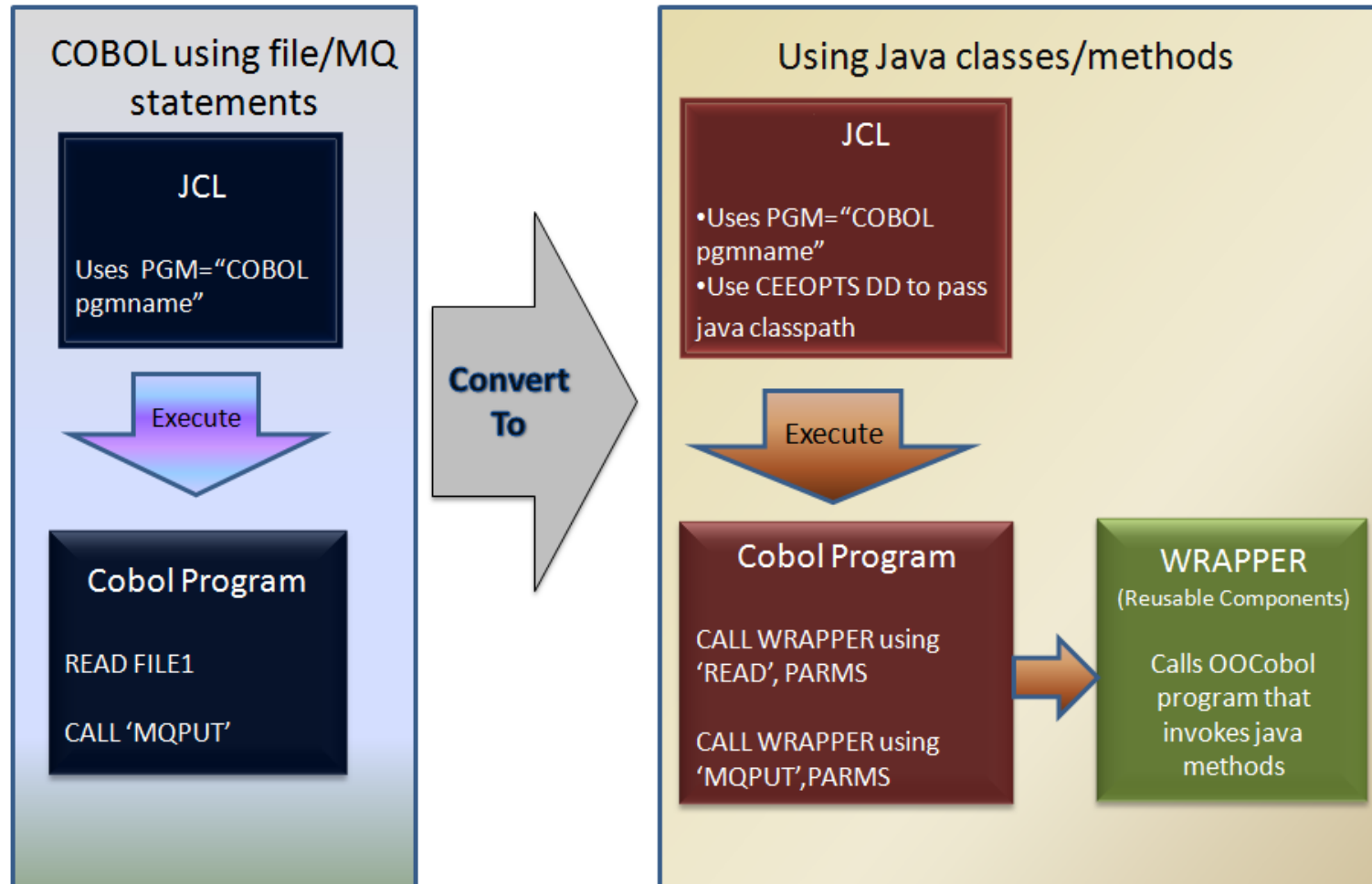
Alternate Approach for ZAAP utilization in Mainframe

Fig. 1 - Currently followed approach for ZAAP Utilization



Alternate Approach for ZAAP utilization in Mainframe

Fig. 2 - Suggested approach for ZAAP Utilization

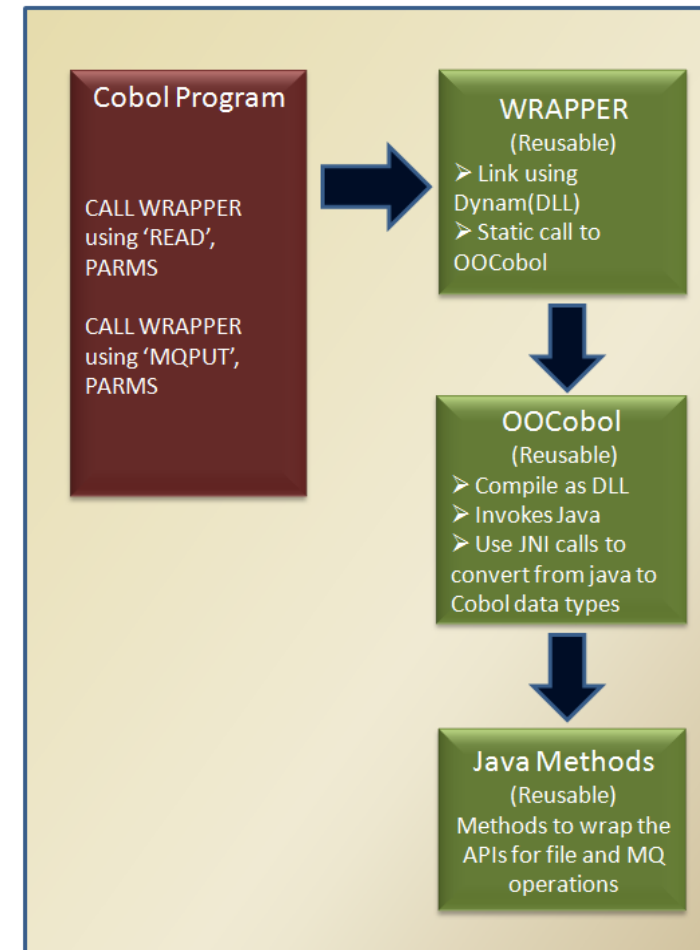


Alternate Approach for ZAAP utilization in Mainframe

Fig. 3 - Cobol Invoking Java methods

Steps followed to execute java from procedural Cobol:

1. Create java classes and methods to wrap the file and MQ operations functionality using java API (Reusable component).
2. Create Object Oriented Cobol (OOCobol) program that uses invoke statements to create java objects and to execute java methods. This OOCobol program should be compiled with DLL option (Reusable component).
3. Create a wrapper procedural Cobol program that calls the OOCobol program. Any program calling a program compiled with DLL option should be linked with DYNAM(DLL) option and can only use static call to invoke the DLL(Reusable component).
4. In the procedural Cobol program that has the file and MQ operations, those calls will be replaced by the calls to the wrapper procedural Cobol program passing parameters specific to the operation.



Alternate Approach for ZAAP utilization in Mainframe – Proof of Concept



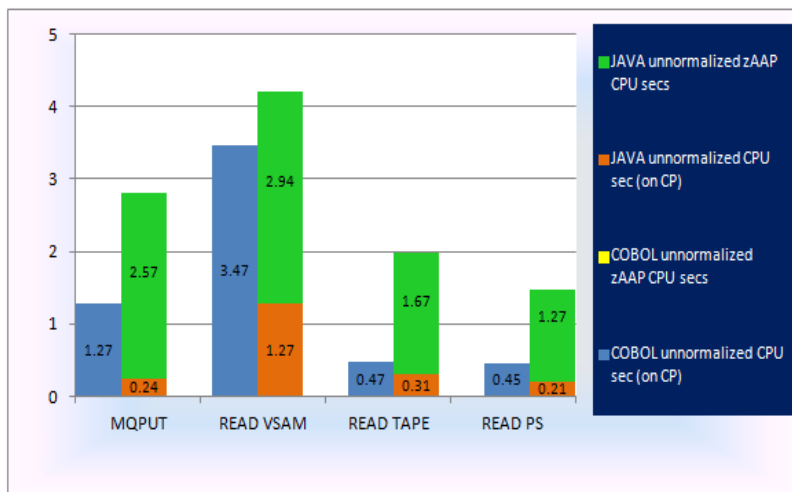
The Proof of Concept was executed for the below scenarios.

1. To compare the MQPUT Cobol call against the wrapper call executing MQPUT in java method
2. To compare the READ Cobol Statement for a VSAM file against the wrapper call executing READ in java method
3. To compare the READ Cobol Statement for a tape file against the wrapper call executing READ in java method
4. To compare the READ Cobol Statement for a sequential file (PS) against the wrapper call executing READ in java method

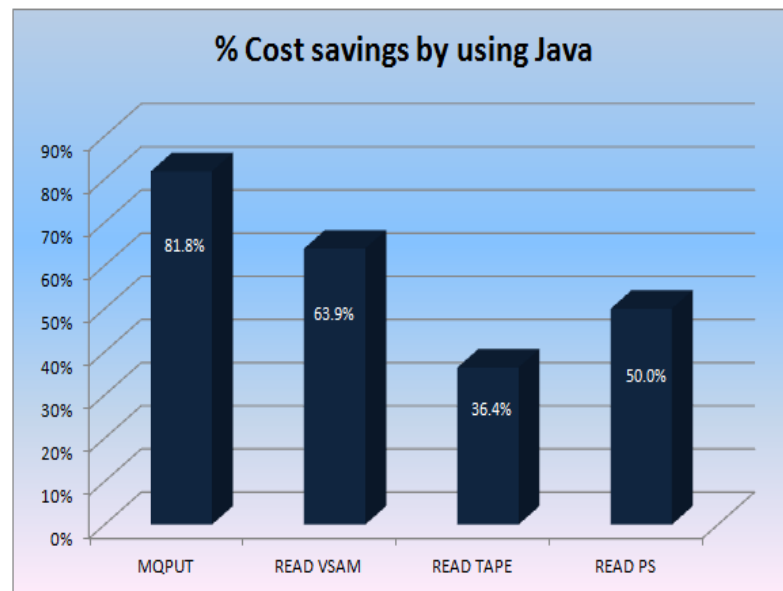
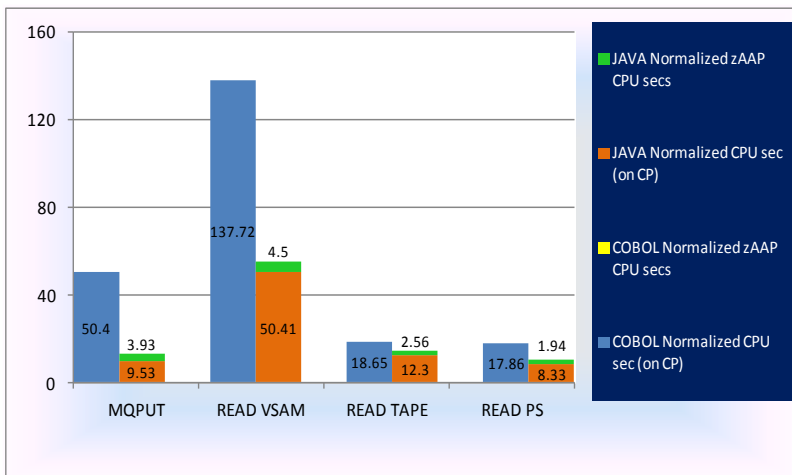
The charts below presents the results of the tests executed for the above four scenarios.

Alternate Approach for ZAAP utilization in Mainframe – Proof of Concept Results

Un-normalized CPU time for the Tests



Normalized CPU time for the Tests



Technology Trends – Way ahead for Customers

IT Optimization and Consolidation

- LPAR/ Vendor Consolidation
- SW Stack Consolidation
- Server Consolidation

- 54% Midrange – Legacy Integration
- 8% Port to LINUX

Technology Consumption Reduction

- Reduce License Cost, Automate IT Op
- Leverage Specialty Engines workloads.
- Investment on Application Re-engg, System tuning.

Competency

Effective use of Alliance for Training needs and getting abreast with latest in technology

High Availability and Real Time

- Parallel Sysplex Implementation
- Next Generation (VSAM-Db2)

Enterprise Modernization

- Modernize Candidate Applications : Code generators, Unsupported, Multiple run time Environments.

- 59% - W/Services
- 31% - Java in m/f

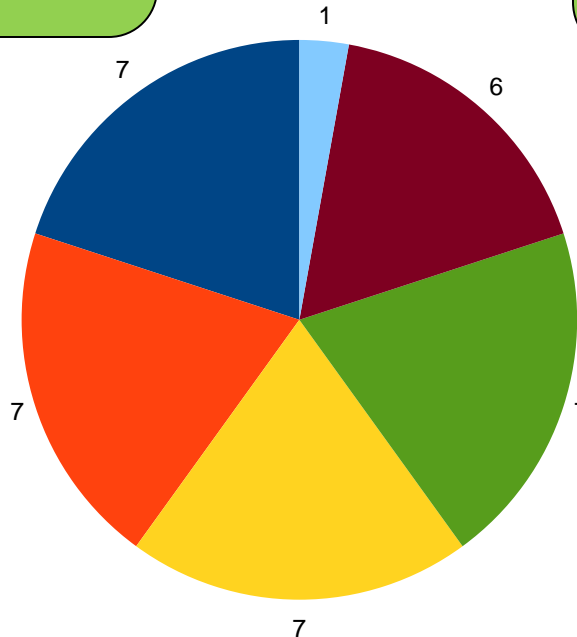
Others

- BI – Leverage Cognos
- Application Co-location

Virtualization

- Migrate non critical Batch
- Use z/Linux Private Cloud

■ CITI
■ AMEX
■ JPMC
■ BOA
■ LTSB
■ Others



Legacy Integration

- Usage of Web Services to merge disparate systems.

Thank You

