Performance Challenges in a Mainframe System

Dr. Stefan Fütterling
Michael Großmann

27.11.2014
Agenda

- Business Context and Objectives
- The Solution’s Big Picture
- Sources of Performance Data
- Data Aggregation and Mapping of Initiator Classes
- Reports
Agenda

- Business Context and Objectives
  - The Solution’s Big Picture
  - Sources of Performance Data
  - Data Aggregation and Mapping of Initiator Classes
  - Reports
Business Context

• The ordering system of a car manufacturer processes several hundred thousand car orders each year.

• It is used by thousands of dialog-users at car-dealerships, market systems, production plants and several other third party systems

• The system is running on an IBM mainframe and uses several technology stacks such as
  - CICS/Cobol
  - Websphere Application Server/Java
  - Messaging (Event Based System)
  - IBM DB2

• Many other applications are running concurrently on the same mainframe

• The ordering system is subject to a constant change and growth due to
  - enhanced/new functionalities
  - the addition of new markets and market systems
  - rise in car sales and/or the increasing complexity of the cars themselves.

Challenge: Performance Management
Business Context – Application Structure

Process Manager (Message driven)

- Change Order
- Primary Business Transaction
- Secondary Business Transactions
- GV Print Documents
- Recalculate Distributionway
- Update Multiple Order Management
- GV Plant Data supply
- GV Update Admission Data
- GV Sharing Distribution-Data with Neighbor Systems
- GV Sharing Order-Data with Neighbor Systems
- Plant
- Dialog
- Queue

Copyright © Capgemini 2014. All Rights Reserved
Capgemini-Performance Challenges in a Mainframe System.pptx
**Overall Process**

(identified via initial business transaction identifier)

**Business Transaction**

(identified via business transaction identifier)

**Platform-specific part of business transaction**

Initiator
(users, external systems, production plants)

Primary Business Transaction

Secondary Business Transactions

GV Zulassungsdoxumente Erstellen
JSS
Synchron abgearbeiteter Geschäftsvorfall

GV Zulassungsdoxumente Drucken
JSS
Synchron abgearbeiteter Geschäftsvorfall

UGV Zulassungsdaten lesen
Cobol

UGV Zulassungsdoxumente
JSS
Synchron abgearbeiteter Geschäftsvorfall

UGV Ermittle Empfänger Zulassungsdoxumente
Cobol

GV Auftrag Ändern
Cobol

UGV Meldungen Umsetzen
JSS

GV Transportweg Aktualisieren
Cobol

Synchron abgearbeiteter Geschäftsvorfall
Business Objectives

• The general questions to be answered are:
  ▪ Who is calling which business transaction and how much cpu-time is consumed?
  ▪ How expensive is a business transaction?
  ▪ How can the cpu-consumption of secondary transactions be evaluated in an event based system and how can it be attributed to user-classes and/or business transactions?
  ▪ Which technology (i.e. CICS/Cobol or WAS/Java) causes the growth? Where are the cost drivers?
  ▪ How large is the future cpu-consumption in the context of a forecast for the predicted growth in car sales and additional functionality due to new releases.

• Our Solutions:
  ▪ Classification of users, external systems and production plants into "Initiator classes" for cost allocation
  ▪ Classification of business transactions into "Process classes" for consumption analysis
  ▪ Introduction of a "process ID" to attribute Secondary cost to the initial business transaction (Primary cost) and to "follow" the overall process cost in a message based system
Agenda

- Business Context and Objectives
- **The Solution’s Big Picture**
  - Sources of Performance Data
  - Data Aggregation and Mapping of Initiator Classes
  - Reports
The Solution’s Big Picture

**Backend**

- Performance DB
  - Aggregated + Historized Data

**ETL-1:**
- Extraction Transformation (incl. Mapping + Aggregation)
- Load
  - User-id
  - Initiator class

**Appl. Logs Cobol/Java**

**Mainframe SMF Records**

**Business Key Figures**

**ETL-2:**
- Qlikview LOAD Scripts
- Generates QVD Files

**Frontend**

- QVD Data Layer
  - QVD
  - QVD
  - QVD

- Report Definition
  - QVW

- Performance Manager (Web Browser)
Agenda

- Business Context and Objectives
- The Solution’s Big Picture

**Sources of Performance Data**

- Data Aggregation and Mapping of Initiator Classes
- Reports
Sources for Performance Data

- **SMF Records**
  - CPU-Consumption

- **Application Logs**
  - Business Transactions with runtimes
  - Business Transaction call count

- **Business Key Figures**
  - Car sales…

**Challenge:**
How to link Application runtimes to SMF CPU-Consumption?
Mainframe System Management Facility (SMF) Records

IBM System Management Facility (SMF) - Definition¹:
SMF is a component of IBM’s z/OS for mainframe computers, providing a standardised method for writing out records of activity running on that IBM mainframe operating system to a file (including CPU-costs, I/O, network activity...)

- SMF provides us information on
  - CPU-Consumption of different CICS and EJB transactions
  - Which type of CPU was used (CP, zIIP, …)

- There are different Types of SMF Records
  - SMF 30 - General Job Information
  - SMF 110 - CICS
  - SMF 120.9 - Java
  - …

- SMF Records can be "enriched" with application specific data (i.e. user-ids, current business transaction identifier)

- Example SMF Record

<table>
<thead>
<tr>
<th>CICS</th>
<th>CICS-Transaction</th>
<th>Timestamp</th>
<th>User</th>
<th>Business Transaction ID</th>
<th>nCPUmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS1</td>
<td>SABCD</td>
<td>15.09.2014 13:07:34</td>
<td>ZUSR1</td>
<td>GV1</td>
<td>3</td>
</tr>
</tbody>
</table>

---

Agenda

- Business Context and Objectives
- The Solution’s Big Picture
- Sources of Performance Data
- **Data Aggregation and Mapping of Initiator Classes**
- Reports
Initiator Classes

Design goals and challenges for the taxonomy of Initiator Classes:

• Protection of data privacy
• Adequate grouping of users in different classes
• Identification of external systems
  • via their technical users
  • via queue names
• Achieve a best fit between the granularity of data vs the cost of storage
• Initiator classes have to serve as a basis to charge-back cpu-cost on a user-class basis
• Taxonomy of Initiator Classes enables drill-functionality in the BI-Reports
Initiator Classes Taxonomy

- User ID / Queue:
  - ZUSRGBP
  - ZUSR009
  - ZUSR239
  - QUEUE.25
  - QUEUE.44
  - QUEUE.26
  - QUEUE.50
  - QUEUE.85

- Initiator group:
  - UK Ext-1
  - UK Ext-2
  - UK Dialogusers

- Initiator subclass:
  - United Kingdom
  - Plant-1
  - Plant-2
  - Sys-1
  - Sys-2

- Initiator class:
  - Headquarters
  - Sales
  - Plants
  - Central systems
  - Batches
  - Unknown

- Total
Aggregation and mapping of initiator classes for application log records

Log entry in application log

<table>
<thead>
<tr>
<th>CICS Name</th>
<th>CICS Transaction</th>
<th>Initial User</th>
<th>Initial Bus. Transaction</th>
<th>Zeit-stempel</th>
<th>User</th>
<th>Business Transaction</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS3</td>
<td>S07ABC</td>
<td>ZUSRGB2</td>
<td>XVABAACV</td>
<td>05.08.2014 12:03:01</td>
<td>ZUSRGB2</td>
<td>XVABAACV</td>
<td>60 ms</td>
</tr>
</tbody>
</table>

Hourly Aggregation and Mapping of Initiator classes for the Performance Database

Aggregated record

<table>
<thead>
<tr>
<th>CICS Name</th>
<th>CICS Transaction</th>
<th>Initiator Class</th>
<th>Initial Bus. Transaction</th>
<th>Hour</th>
<th>Business Transaction</th>
<th>Call count</th>
<th>Overall Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS3</td>
<td>S07ABC</td>
<td>UK Ext-2</td>
<td>XVABAACV</td>
<td>05.08.2014 12:00</td>
<td>XVABAACV</td>
<td>1120</td>
<td>50000 ms</td>
</tr>
</tbody>
</table>
Splitting of large SMF Records of secondary transactions

- Secondary business transactions are processed asynchronously by workers sitting on queues
- Worker takes message out of queue and starts business transaction in CICS transaction
- CICS writes only a few large SMF records per CICS transaction (for many different business transactions)
- The large SMF records for each CICS Transaction are aggregated daily

<table>
<thead>
<tr>
<th>CICS Name</th>
<th>CICS TRANSACTION</th>
<th>Date</th>
<th>Calls</th>
<th>nCPUmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS1</td>
<td>ABC01</td>
<td>05.08.2014</td>
<td>47</td>
<td>2.410</td>
</tr>
</tbody>
</table>

- Runtimes for each secondary business transactions are aggregated from application logs
- Large SMF records are split up proportionally using runtimes for each secondary business transaction
- This approach does not provide exact CPU usage for secondary business transactions, but a first approximation

Can contain several secondary business transactions, each of them called many times
Agenda

- Business Context and Objectives
- The Solution’s Big Picture
- Sources of Performance Data
- Data Aggregation and Mapping of Initiator Classes
- Reports
Example Report Mockup - CPU-Consumption

Example Report-Mockup showing Drill-functionality
- Drill by Initiator-Class and -Subclass
- Drill by Access

<table>
<thead>
<tr>
<th></th>
<th>Jan 2014</th>
<th>Feb 2014</th>
<th>...</th>
<th>Dez 2014</th>
<th>Total</th>
<th>Average</th>
<th>StDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Headquarter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Sales</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Germany</td>
<td>426.324</td>
<td>403.327</td>
<td>-5.39%</td>
<td>413.900</td>
<td>354.588</td>
<td>-14.33%</td>
<td></td>
</tr>
<tr>
<td>+ China</td>
<td>634.234</td>
<td>893.455</td>
<td>40.87%</td>
<td>983.432</td>
<td>1.023.210</td>
<td>4.04%</td>
<td></td>
</tr>
<tr>
<td>- UK</td>
<td>235.433</td>
<td>345.287</td>
<td>46.66%</td>
<td>432.678</td>
<td>659.123</td>
<td>52.34%</td>
<td></td>
</tr>
<tr>
<td>Access System 1</td>
<td>426.324</td>
<td>403.327</td>
<td>-5.39%</td>
<td>413.900</td>
<td>354.588</td>
<td>-14.33%</td>
<td></td>
</tr>
<tr>
<td>Access System 2</td>
<td>634.234</td>
<td>893.455</td>
<td>40.87%</td>
<td>983.432</td>
<td>1.023.210</td>
<td>4.04%</td>
<td></td>
</tr>
<tr>
<td>Dialog-Users</td>
<td>235.433</td>
<td>345.287</td>
<td>46.66%</td>
<td>432.678</td>
<td>659.123</td>
<td>52.34%</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1.295.991</td>
<td>1.642.069</td>
<td>26.70%</td>
<td>1.830.010</td>
<td>2.036.921</td>
<td>11.31%</td>
<td></td>
</tr>
<tr>
<td>+ Plants</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Randomized Data
Drill-Functionality

- By Initiator-Class/-subclass
- By Process-Class/-subclass
Contact information

Dr. Stefan Fütterling

Email: Stefan.Fuetterling@capgemini.com
Phone: +49 711 50505 361

Michael Großmann

Email: Michael.Grossmann@capgemini.com
Phone: +49 711 50505 352
About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion. Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini