Kapitel 8
Internetanbindung und SAP Fiori

In diesem Kapitel widmen wir uns den Webtechnologien der SAP, zu denen auch Fiori-UIs und deren Performanceoptimierung gehören.

Für die Arbeit mit einer SAP-Lösung gibt es zwei Zugriffsmöglichkeiten: Benutzer melden sich entweder über das klassische SAP GUI für Windows oder Java Environment oder über einen Webbrowser an SAP-Systeme an. Letzteres hat den Vorteil, dass auf den Desktop-Computern keine speziellen GUI-Programme installiert werden müssen.

In diesem Kapitel widmen wir uns der Performanceoptimierung von webbasierten Benutzeroberflächen und Webservices. Die Kommunikation zwischen dem Webbrowser bzw. dem Webclient und der SAP-Applikationsebene wird über eine der folgenden Alternativen vermittelt:

- Business Server Pages (BSP), Web Dynpro ABAP und integrierter Internet Transaction Server (ITS) auf Basis des SAP NetWeaver Application Servers (AS) ABAP
- Java Server Pages (JSP), Java Servlets und Web-Dynpro-for-Java-Anwendungen auf Basis des SAP NetWeaver AS Java
- SAPUI5 und OData auf Basis von SAP NetWeaver AS ABAP, SAP HANA Extended Services (XS) Engine oder SAP Cloud Platform. Zu dieser Klasse gehören auch Fiori-Anwendungen.

In diesem Kapitel lernen Sie, welche Herausforderungen Sie meistern müssen, um eine gute Performance von SAP-Webanwendungen zu erreichen.

Wann sollten Sie dieses Kapitel lesen?

Dieses Kapitel führt Sie als Administrator oder Entwickler in die Konfiguration und Überwachung von webbasierten Benutzeroberflächen und der Webschnittstellen auf Basis des SAP NetWeaver AS ABAP ein. Vor der Lektüre dieses Kapitels sollten Sie Kapitel 5, »Optimierung von ABAP-Programmen«, und Abschnitt 7.3, »Remote Function Calls (RFCs)«, gelesen haben.

8.1 SAP-Webanwendungen

Jedes Kind – und »Kind« kann hier in der Tat wörtlich genommen werden – kennt heutzutage die Protokollsprache des Internets. So erzeugt z. B. das folgende Kommando *http://wdrmaus.de*, in das Adressfeld eines Webbrowsers eingegeben, eine für einen Benutzer lesbare Bildschirmseite.

Als Transportprotokoll wird dabei das *Hypertext Transfer Protocol* (HTTP) verwendet. Den Ausdruck *http://wdrmaus.de* bezeichnet man als *Unified Resource Locator* (URL), da er eine bestimmte Seite im Internet eindeutig identifiziert. Das Ergebnis der Anfrage bezeichnen wir in diesem Buch als Webdokument. Dies können beispielsweise sein:

- ein *Hypertext-Markup-Language*-(*HTML*-)Dokument
- Dateien mit Formatvorlagen (CSS-Dateien), Schriftarten, Bild- und sonstige Mediendateien
- Datendokumente in der *eXtended Markup Language* (XML) oder der *JavaScript Object Notation* (JSON)
- Dateien mit JavaScript-Coding

Webanwendungen

Dem SAP NetWeaver AS (technisch SAP-Basis 7.00 und höher) liegt u. a. SAP ERP, SAP CRM, SAP Business Warehouse (BW) oder SAP S/4HANA (ab NetWeaver 7.50) zugrunde. Mit dem SAP NetWeaver AS und seinem Vorläufer stehen Ihnen drei Techniken zur Realisierung von Webanwendungen zur Verfügung:

- **Webanwendungen auf Basis des SAP Internet Transaction Servers (ITS)**
  Der ITS war die erste Internettechnik der SAP. Als eigenständige Installation stand er mit SAP R/3 3.1 zur Verfügung, seit Version SAP R/3 4.6C werden aber in dieser Technik keine neuen Anwendungen mehr realisiert. Um die Investitionen in ITS-Anwendungen zu schützen, wird der ITS von SAP weiterhin gewartet und in den SAP NetWeaver AS ABAP integriert (*integrierter ITS*). Zu dieser Klasse von Anwendungen gehören auch das SAP GUI for HTML, oder auch Web GUI, eine generische Übersetzung klassischer SAP-GUI-Anwendungen in das HTML-Format. Diese Technologie bleibt auch in SAP S/4HANA noch verbreitet, diese Anwendungen sind auch im Fiori-Portfolio vertreten.
- **Webanwendungen auf der Basis von Business Server Pages (BSP) und Web Dynpro ABAP**
  Seit dem *SAP Web Application Server ABAP* 6.10 können Webanwendungen direkt in der ABAP-Entwicklungsumgebung entwickelt und in der

ABAP-Laufzeitumgebung ausgeführt werden. Diese Anwendungen werden als BSP bezeichnet. Um diese Webanwendungen auszuführen, ist keine Installation zusätzlicher Softwarekomponenten (auch nicht eines separaten Webservers) nötig. Die Fortentwicklung ist die Technologie *Web Dynpro ABAP*. BSP und WebDynpro ABAP werden in Produkten wie SAP CRM und SAP ERP weiterhin genutzt.

- **Webanwendungen auf der Basis von Java Server Pages (JSP), Java Servlets und Web Dynpro Java**
  Mit dem *SAP NetWeaver AS Java* steht eine komplette Entwicklungs- und Laufzeitumgebung für Java-Webanwendungen (*JSP* und *Java Servlets*) sowie für Java-»Backend«-Anwendungen (*Enterprise JavaBeans*) zur Verfügung. Eine besonders komfortable Entwicklungsumgebung für User Interfaces (UIs) bietet die Technologie *Web Dynpro Java*. Diese Technologie wird mittlerweile nur noch sehr selten eingesetzt und wird daher in diesem Buch auch nicht weiter behandelt.
- **Webanwendungen auf der Basis von SAPUI5 und SAP Gateway (u. a. SAP-Fiori-Anwendungen)**
  SAPUI5 als UI-Technologie und *SAP Gateway* als Technologie für die Bereitstellung von Webservices (*OData-Services*) bilden die Grundlage für die modernsten Webanwendungen der SAP, zu der auch die *Fiori-Anwendungen* gehören. SAPUI5 als UI-Technologie und OData als Service-Technologie sind plattformübergreifend. Anwendungen auf Basis von SAPUI5 und OData laufen nicht nur auf dem SAP NetWeaver AS, sondern auch auf der SAP HANA XS Engine oder der SAP Cloud Platform.

Webservices

Webtechnologie wird nicht nur für die Erstellung von Benutzerschnittstellen verwendet, sondern in zunehmendem Maße auch zur Standardisierung von Schnittstellen zwischen Systemen; wir unterscheiden zwischen Webservices für die Kommunikation zwischen Anwendungen innerhalb eines Unternehmens (*Application-to-Application*, A2A) oder zwischen Unternehmen (*Business-to-Business*, B2B). Beide Arten von Webservices verwenden als Protokollsprache HTTP, als Datenübertragungssprache wird bei Webservices für Systemschnittstellen XML oder JSON verwendet.

Einsatz von Web UI und SAP GUI planen

Alle Anwendungen der SAP sind webfähig, von wenigen Anwendungen abgesehen. Das bedeutet, es besteht die Möglichkeit, sich in einem Unternehmen strategisch auf ein Web UI als einzige UI-Lösung festzulegen. Die Alternative ist, zweigleisig zu fahren und für einige Benutzer/Anwendungen ein Web UI als Oberfläche einzuführen und für andere das klassische SAP GUI. Wir wollen in diesem Abschnitt beleuchten, von welchen Argumenten Sie sich dabei leiten lassen sollten.

Dabei verwenden wir den Begriff *SAP GUI* für das SAP GUI for Windows und das SAP GUI for Java Environment, d. h. für die Technologien, die eine Installation eines SAP-Frontend-Programms erfordern. Die Bezeichnung *Web GUI* wird häufig als Abkürzung für das SAP GUI for HTML verwendet. Als Web UI bezeichnen wir in diesem Buch generell alle Webanwendungen.

Reine Webanwendungen

Zunächst gibt es einige SAP-Anwendungen, die grundsätzlich nur mit dem Web UI laufen; dazu gehören alle Lösungen für das Internet oder Intranet, z. B. SAP Employee Self-Services (SAP ESS), SAP CRM und Enterprise Buyer Professional (E-Procurement) sowie Fiori-Anwendungen oder sonstige Anwendungen auf der Basis von SAPUI5. Für diese Lösungen ist die Entscheidung bereits getroffen. Diese Webanwendungen sind für die Anforderungen des Internets optimiert.

Anders sieht die Situation für viele andere Transaktionen in SAP ERP oder SAP Advanced Planning and Optimization (APO) aus. Diese laufen sowohl im Web UI als auch im SAP GUI. Um hier eine Entscheidung zu treffen, sollten Sie Performanceaspekte mit einbeziehen.

Performance Web UI und SAP GUI

Grundsätzlich ist es so, dass die Verwendung des Web UI einen zusätzlichen Aufwand (CPU-Zeit und Datentransfer) bedeutet, der zu einem Laufzeitnachteil im Vergleich zum SAP GUI führt. Im Einzelnen sind zu berücksichtigen:

- die CPU-Zeit für die Umwandlung der SAP-Bildschirme in HTML-Seiten durch den Server
- der höhere Datentransfer zwischen Server und Browser (im Vergleich zum Datentransfer zwischen Applikationsebene und SAP GUI)
- die Generierungszeit im Webbrowser; sie ist höher als die Bearbeitungszeit im SAP GUI.

Ultra-thin Client

Um wie viel höher die Antwortzeit für einen Benutzer mit dem Web UI im Vergleich zum SAP GUI schließlich ist, hängt von den verwendeten Funktionen und der Hardware ab. Für die CPU-Zeit auf dem Server und dem Frontend sollte man mit einem Zusatzaufwand in der Größenordnung von einigen 10 bis 100 ms rechnen. Zum höheren Datenverkehr können Sie von folgenden Überlegungen ausgehen: Zu Recht bezeichnet SAP das SAP GUI als *Ultra-thin Client*, da der Datenverkehr zwischen Applikationsebene und GUI sehr gering ist, ca. 3–5 kB pro Bildwechsel bei Transaktionen mit Controls. Bei Webtransaktionen des SAP NetWeaver AS werden zwischen 10 und 40 kB übertragen. Damit liegen die Produkte der SAP mit ihrem Datenverkehr eher am unteren Ende des Internetstandards, bei typischen Webseiten im Internet werden mehrere 100 kB übertragen. In einem LAN mit

hoher Bandbreite macht sich dieser erhöhte Datenverkehr in der Laufzeit praktisch nicht bemerkbar.

Latenzzeiten im WAN

In einem WAN mit niedriger Bandbreite und hoher Latenzzeit kann der Unterschied zwischen dem SAP GUI und dem Web UI zu deutlichen Performancenachteilen für das Web UI führen. Als *Latenzzeit* bezeichnet man die Zeit zwischen dem Verlassen eines Datenpakets auf der Senderseite und dem Empfang auf der Empfängerseite sowie den Rückweg vom Empfänger zum Sender. Die Latenz setzt sich aus zwei Teilen zusammen:

- der sogenannten *letzten Meile* – vom Endpunkt des Internet- oder Mobilfunkproviders bis zum Endgerät über kabelgebundes Netzwerk, kabelloses Netzwerk (WLAN) oder Mobilfunk
- den *mittleren Meilen* – vom Webserver der SAP-Anwendung bis zum Endpunkt des Internet- oder Mobilfunkproviders

Latenzzeiten der letzte Meile

Typische Latenzzeiten für die letzte Meile sind:

- < 10 ms im LAN
- > 20 ms für ein kabelgebundes Netzwerk oder kabelloses Netzwerk (WLAN)
- > 50 ms für ein 4G-Mobilfunknetz (z. B. LTE)
- > 100 ms für ein 3G-Mobilfunknetz (z. B. HSPA)

In Expertenforen wird immer wieder von stark schwankenden Latenzen in Mobilfunknetzen berichtet. Insbesondere treten hohe Latenzen auf (im Bereich von Sekunden), wenn die Verbindung zum Netzwerk neu aufgebaut werden muss – dies macht sich für den Benutzer bei wenig frequenter Nutzung einer Anwendung bemerkbar.

Latenzzeiten der mittleren Meilen

Typische WAN-Latenzzeiten für die mittleren Meilen sind:

- < 100 ms innerhalb eines Kontinents
- > 200 ms für Interkontinentalverbindungen, z. B. zwischen Europa und Amerika

Latenzzeiten für die mittleren Meilen ergeben sich aus der Lichtgeschwindigkeit (ca. 200.000 km/Sekunde in optischem Fieberglas) und den Verzögerungen innerhalb von Schaltstellen im Internet und bei der Verbindung zwischen Ihrem Firmennetzwerk und dem Internet.

Für ein SAP GUI müssen Sie minimal mit folgenden Netzwerkzeiten durch die Latenzzeit rechnen:

- Für UIs ohne SAP GUI Controls: Latenzzeit (einmal die Laufzeit eines Datenpakets hin und zurück)

- Für UIs mit SAP GUI Controls; 2 × Latenzzeit (zweimal Laufzeit eines Datenpakets hin und zurück aufgrund der Datenrückgabe zum Control)

Für ein Web UI müssen Sie mit deutlich höheren Netzwerkzeiten durch die Latenzzeit rechnen, im Detail gehen wir darauf in den folgenden Abschnitten ein.

Bei langsamen WAN-Verbindungen kann also die Latenzzeit innerhalb der Antwortzeit einer Transaktion leicht dominieren. Bei Transaktionen, die große Datenmengen verarbeiten, kommt die Bandbreite, d. h. der Durchsatz des Netzwerkes als Faktor, der die Performance einschränken kann, hinzu. Wir empfehlen daher Evaluierungsmessungen, wie in Abschnitt 8.2, »Analysen auf dem Präsentationsserver«, beschrieben.

SAP-BW-Frontend-Technologien

Beim Einsatz von SAP Business Warehouse (BW) gelten ähnliche Überlegungen. Hier müssen Sie sich zwischen dem BEx Analyzer, einem Microsoft-Excel-basierten Frontend mit SAP-Add-in zur Kommunikation mit dem BW-Server, und dem webbasierten Reporting entscheiden.

8.2 Analysen auf dem Präsentationsserver

Es kommt vor, dass alle Serverkomponenten technisch einwandfrei laufen, die Benutzer sich aber dennoch über schlechte Performance oder Fehler beschweren. In diesem Fall können Sie die in diesem Abschnitt beschriebenen Methoden zur Performanceanalyse einsetzen. Sie sind im Wesentlichen unabhängig vom verwendeten Server, d. h., sie können sowohl für einen SAP NetWeaver als auch für jede andere Servertechnologie verwendet werden. Sie können z. B. auch den HTTP-Strom der Webanwendungen Ihrer Onlinebank mit dieser Methode analysieren.

Parallele Anfragen

Abbildung 8.1 zeigt den Ablauf eines Transaktionsschrittes einer Webanwendung aus Sicht des Präsentationsservers. Während im zuvor betrachteten SAP-GUI-Szenario die Interaktionen zwischen Präsentationsserver und Anwendungsserver sequenziell ablaufen, kann der Webbrowser mehrere Anfragen an den Webserver parallel starten und auch schon parallel zu den laufenden Anfragen die Daten aufbauen. Abbildung 8.1 ist wie folgt zu interpretieren:

❶ Ein Benutzer tritt in Interaktion mit dem Browser.

❷ Der Browser stellt fest, dass eine Anfrage an den Webserver gesendet werden muss.

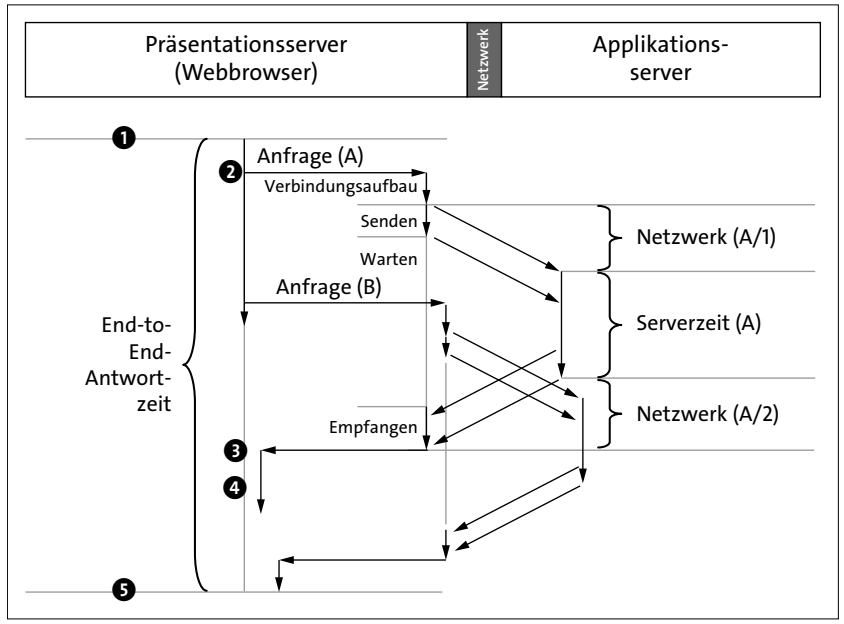

Abbildung 8.1 Transaktionsschritt einer Webanwendung zwischen Präsentationsserver (Webbrowser) und Anwendungsserver (Webserver)

Die Anfrage untergliedert sich in die folgenden Zeitanteile:

Zeitanteile einer Webanfrage

- *Verbindungsaufbau (Connecting Time)*: Bezeichnet die Zeit, die benötigt wird, um eine TCP-Verbindung zu öffnen. Ist diese Zeit erhöht, kann dies daran liegen, dass der Browser auf eine TCP-Verbindung wartet. Typischerweise öffnen Browser nur eine bestimmte Zahl von Verbindungen gleichzeitig (z. B. sechs), um den Server nicht mit Anfragen zu überlasten. Ist also schon eine bestimmte Zahl von Verbindungen offen, wartet der Browser, bis diese bearbeitet sind. Im SAP-Umfeld kann diese Situation eintreten, wenn das SAP Fiori Launchpad oder eine Übersichtsseite (*Dashboard*) geöffnet werden, die viele parallele Anfragen an den Server stellen.
- *Senden (Sending Time)*: Bezeichnet die Zeit, die es braucht, um die HTTP-Anfrage an den Server zu senden. Wenn dieser Zeitanteil hoch ist, sollten Sie überprüfen, ob eine große Datenmenge übertragen wird (Senden von Massendaten).
- *Warten (Waiting)*: Diese Zeit umfasst den Zeitraum zwischen dem vollständigen Absenden der Anfrage bis zur ersten Antwort, die den Browser erreicht. Diese Zeit wird auch *Time To First Byte (TTFB)* genannt.

– *Empfangen (Receiving Time)*: Bezeichnet die Zeit, die es dauert, um die komplette HTTP-Anfrage vom Server (oder Puffer) zu empfangen. Wenn dieser Zeitanteil hoch ist, sollten sie überprüfen, ob eine große Datenmenge übertragen wird (Empfangen von Massendaten).

❸ Der Webbrowser hat alle Informationen vom Webserver erhalten, und die Kommunikation wurde mit einem Rückgabewert, z. B. 200 bei einer erfolgreichen Übertragung, abgeschlossen. Zu diesem Zeitpunkt startet die Aufbereitungsphase.

❹ Schließlich ist die Aufbereitung der Anfrage (*Rendering*) abgeschlossen.

❺ In einem Browserfenster können mehrere Anfragen sequenziell oder parallel an den Server gesendet werden. Der Dialogschritt endet mit der Aufbereitung der letzten Anfrage. Im Gegensatz zu einer klassischen SAP-GUI-Transaktion sind Webanwendungen häufig so gestaltet, dass der Benutzer bereits mit der Arbeit fortfahren kann, während noch weitere Daten geladen werden. Das heißt, es ist nicht unbedingt eindeutig definiert, welche die letzte Anfrage eines Dialogschrittes ist.

Bei einer typischen Anfrage im Rahmen einer SAP-Transaktion ist zu erwarten, dass die Wartezeit dominiert. Die Summe aus Sende-, Warte- und Empfangszeit umfasst die Antwortzeit des Applikationsservers und die Netzwerkzeit. Oder anders ausgedrückt: Die Netzwerkzeit, die wir nicht explizit messen können, ergibt sich aus der Differenz von Sende-, Warte- und Empfangszeit und der Antwortzeit des Applikationsservers.

Richtlinien für Webanwendungen

Aufgrund der hohen Netzwerklatenz, die im WAN einen dominierenden Anteil an der Gesamtantwortzeit bilden kann, gelten die folgenden Richtlinien für Webanwendungen:

1. Pro Benutzerinteraktion sollten nur wenige nicht im Browser gepufferte Serveranfragen erfolgen, als Richtwert maximal zwei synchrone Anfragen, nachdem die Anwendung einmal geladen wurde.
2. Wenn möglich sollten Webdokumente im Browser gepuffert werden.
3. Die übertragene Datenmenge sollte möglichst klein sein.

Im folgenden Abschnitt lernen Sie nun die Werkzeuge kennen, mit denen Sie feststellen können, ob diese Richtlinien eingehalten werden.

8.2.1 Performancewerkzeuge der Internetbrowser

Entwicklerwerkzeuge im Browser

Um die Performance von Webanwendungen zu analysieren, verfügen moderne Webbrowser über Entwicklungswerkzeuge. Wir stellen diese anhand des Firefox-Webbrowsers dar, andere Browser bieten vergleichbare

Funktionen. In Firefox starten Sie die Entwicklungswerkzeuge über die Funktionstaste F12 oder über **Open Menü • Entwickler**. Da wir uns in diesem Abschnitt auf die Performance der Serveranfragen konzentrieren, ist der mit **Netzwerkanalyse** bezeichnete Abschnitt der Entwicklerwerkzeuge der für uns wichtige. Abbildung 8.2 zeigt den Firefox-Browser mit einem Ausschnitt einer Fiori-Anwendung im oberen Teil und die Firefox-Entwicklerwerkzeuge mit der geöffneten Netzwerksicht im unteren Teil.

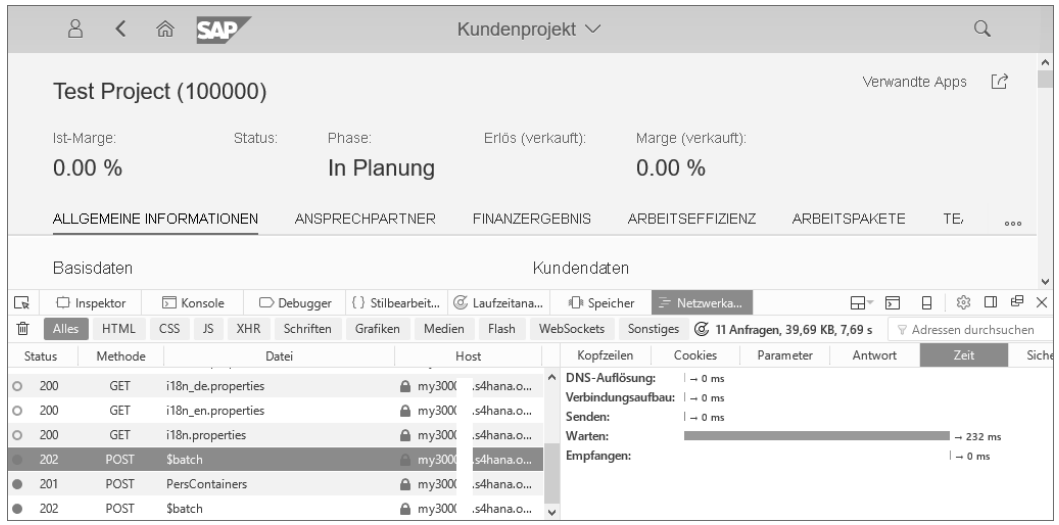

Abbildung 8.2 Entwicklerwerkzeuge im Firefox-Browser

Die Netzwerksicht im Browser zeigt den Zeitverlauf der Serveranfragen. In der rechten oberen Ecke finden Sie neben dem Uhrsymbol die gesamte Anzahl der Anfragen, die übertragene Datenmenge und die Gesamtantwortzeit aller Anfragen. Wenn Sie eine Anfrage auswählen, sehen Sie weitere Informationen, insbesondere den Inhalt von Anfrage und Antwort. Für die Performanceanalyse ist der Detailbereich **Zeit** besonders interessant. Hier wird die Antwortzeit der Anfrage weiter aufgeschlüsselt. Dieser Bereich ist in Abbildung 8.2 im rechten unteren Bereich zu sehen.

Hinweise zur Nutzung der Netzwerkanalyse

Browserpuffer bei Performancemessungen aktivieren

In den Einstellungen der Entwicklerwerkzeuge können Sie konfigurieren, ob der Browsercache genutzt werden soll, wenn die Entwicklerkonsole offen ist. Bei Firefox ist dies die Option **HTTP-Cache bei offenem Werkzeugkasten deaktivieren** (unter **Erweiterte Einstellungen**). Diese Einstellung ist nützlich, wenn Sie bei der Fehlersuche bzw. beim Entwicklungsprozess vermeiden wollen, dass veraltete Daten aus dem Cache verwendet werden. Bei

Performancemessungen sollten Sie allerdings darauf achten, dass diese Option *nicht* aktiviert ist.

Entwicklerwerkzeuge bei neuem Browserfenster

Die Entwicklerwerkzeuge sind an ein Browserfenster gebunden. Wenn ein neues Browserfenster geöffnet wird, sind diese in der Regel nicht geöffnet und schneiden daher die ersten Anfragen an den Server nicht mit. Einige Browser bieten die Einstellung an, beim Öffnen eines neuen Browserfensters direkt die Entwicklerwerkzeuge zu öffnen.

Plausibilitätscheck

Leider ist festzustellen, dass die Information, ob eine Anfrage aus dem Browserpuffer geladen wurde, in den Browser-Entwicklungswerkzeugen nicht zu 100 % zuverlässig ist. Gelegentlich wird angezeigt, dass eine Anfrage an den Server gesendet wurde, die in Wirklichkeit aus dem Puffer geladen wurde. Eine Antwortzeit von 1–3 ms ist ein Indiz dafür, dass die Anfrage aus dem Puffer beantwortet wurde. Wir empfehlen Ihnen daher, im Zweifelsfall unterschiedliche Browser zu vergleichen, im Serverlog (siehe Abschnitt 8.4.1, »HTTP-Trace im Internet Communication Manager«) zu verifizieren, ob wirklich eine Anfrage abgesetzt wurde, oder ein Analyse oder ein browserunabhängiges Werkzeug zur Netzwerkanalyse (wie z. B. Fiddler) zu verwenden.

8.2.2 SAP-Statistiken in der HTTP-Anfrage

Die Antwortzeit des Applikationsservers wird vom Server selbst gemessen. Die Werkzeuge, um diese Antwortzeit auszuwerten, sind also serverspezifisch, und Sie benötigen in der Regel Zugang zum Applikationsserver. Allerdings gibt es auch eine Möglichkeit, sich die Antwortzeit der SAP-Applikationsserver in der HTTP-Antwort mitschicken zu lassen, um diese direkt (also ohne Zugang zum Server) im Browser auswerten zu können.

SAP-Server-Statistikdaten einschalten

Das Senden der Server-Statistikdaten können Sie durch eine der folgenden Optionen einschalten:

- Fügen Sie der URL den Query-Parameter *sap-statistics=true* hinzu, z. B. *<server>/sap/opu/odata/sap/<odata service>/<entity>?sap-statistics=true*.
- Fügen Sie der HTTP-Anfrage das Kopf-Feld `sap-statistics:true` hinzu.
- Geben Sie in einem SAPUI5-basierten UI das Kommando Strg + Alt + ⇧ + P ein, und aktivieren Sie in dem Dialogfenster die Option **Enable SAP statistics for OData calls**.

Als Reaktion schickt der SAP-Server einen SAP-Statistiksatz im HTTP-Antwort-Kopf. Den HTTP-Antwort-Kopf finden Sie in den Browser-Entwicklerwerkzeugen wie folgt: Wählen Sie im Netzwerkbereich eine Anfrage aus, und öffnen Sie im Detailbereich zur Anfrage den Bereich **Kopfzeilen**.

SAP-Statistiksatz

Der SAP-Statistiksatz im HTTP-Antwort-Kopf ist wie folgt aufgebaut:

- Schlüssel: `sap-statistics`
- Wertebereich: Der Wertebereich enthält Serverantwortzeiten und ausgewählter Komponenten in der Form `<name1>=<zeit1>, <name2>=<zeit2> ...` Die Zeitangaben sind stets in Millisekunden.

Ein solcher SAP-Statistikdatensatz könnte also wie folgt aussehen:

```
sap-statistics:
gwtotal=150,gwrfcoh=6,gwapp=65,gwhub=14,gwbe=62,icmtotal=184,
icmreqrcv=1,icmext=179,icmssl=5,wdtotal=194,wdreqrcv=3,wdext=185,
wdssl=7
```

Das Senden des Statistiksatzes unterstützen zahlreiche Server und Komponenten der SAP. Die im Statistiksatz enthaltenen Informationen hängen von der konkreten Komponente ab. Die Namen beginnen mit folgenden Kürzeln:

- `wd`: SAP Web Dispatcher
- `icm`: SAP NetWeaver AS ABAP – Internet Communication Manager (ICM)
- `gw`: SAP Gateway
- `wdxs`: SAP Web Dispatcher für SAP HANA XS Engine

SAP Web Dispatcher oder ICM repräsentieren die Serverkomponente, die die HTTP-Anfrage entgegennimmt. Das heißt, die Antwortzeit des Applikationsservers wird repräsentiert durch:

- `wdtotal`, wenn ein SAP Web Dispatcher verwendet wird
- `icmtotal`, wenn der SAP NetWeaver ohne SAP Web Dispatcher verwendet wird
- `wdxstotal`, wenn eine SAP HANA XS Engine verwendet wird

In dem Beispiel oben können wir den Fluss der Anfrage durch die Komponenten wie folgt verfolgen (siehe auch Abbildung 8.6):

1. `wdtotal=194`: Brutto-Antwortzeit des SAP Web Dispatchers inklusive aller dahinterliegenden Komponenten
2. `wdext=185`: Antwortzeit der hinter dem SAP Web Dispatcher liegenden Komponenten
3. `icmtotal=184`: Brutto-Antwortzeit des ICMs inklusive aller dahinterliegenden Komponenten
4. `icmext=179`: Antwortzeit der hinter dem ICM liegenden Komponenten. Diese Zeit entspricht der **Calling Time** in der HTTP-Einzelsatzstatistik

(siehe Abschnitt 8.4.2, »Performanceanalyse von ABAP-Webanwendungen durchführen«).

5. `gwtotal=150`: Brutto-Antwortzeit des SAP Gateways, inklusive aller dahinterliegenden Komponenten
6. `gwapp=65`: Antwortzeit der hinter dem Gateway liegenden Anwendung

Mit den Informationen, die uns der Server zur Verfügung stellt, verfügen wir also nun über die folgenden Daten:

- am Browser gemessene Zeit für eine Anfrage
- Antwortzeit des Applikationsservers
- Die Differenz der beiden Zeiten ergibt die Netzwerkzeit.

Um die Netzwerkzeit zu optimieren, müssen Anwendungen die folgenden Maßnahmen ergreifen, auf die wir in den weiteren Abschnitten im Detail eingehen:

1. Die Optimierung der Anwendung selbst, d. h. die Reduktion der Anzahl der Anfragen und Datenmenge. Solche Optimierungen sind technologiespezifisch, wir gehen bei den konkreten SAP-UI-Technologien auf diese ein.
2. Die Pufferung von Dokumenten im *Browserpuffer* (*Browsercache*): jede Anfrage, die erst gar nicht über das Netzwerk geschickt wird, kann auch nicht durch hohe Latenz unangenehm auffallen.
3. Die Pufferung von Dokumenten in einem Servercache oder einem *Content Delivery Network*: Durch diese Pufferung kann die Antwortzeit von Anfragen reduziert werden.

Zur weiteren Analyse der eigentlichen Anwendung fahren Sie nun mit den entsprechenden Methoden der Performanceanalyse auf dem Server fort. Für den SAP NetWeaver AS ABAP ist dies im übernächsten Abschnitt beschrieben. Weitere Informationen zu den Gateway-Zeiten und zur Optimierung von OData-Anfragen finden Sie in Abschnitt 8.6.4, »SAP Fiori, SAPUI5 und OData auf dem SAP NetWeaver Application Server ABAP«.

8.2.3 Webanwendungen kontinuierlich überwachen

Die bisher vorgestellten Analysen im Browser eignen sich für die Analyse durch einen Entwickler oder Performanceexperten. Im kontinuierlichen Betrieb benötigen Sie aber Monitoring-Werkzeuge, die die Webanwendung kontinuierlich überwachen.

End User Experience Monitoring

Mit dem *End User Experience Monitoring* im SAP Solution Manager ist es möglich, Abfolgen von Webseiten durch die periodische Abfrage von Test-Requests zu überwachen. In einem Webshop könnten Sie mit diesem Werkzeug im Abstand von 10 Minuten testen, ob der Katalogzugriff auf ausgewählte Produkte, der Produktkonfigurator, die Preisfindung oder die Verfügbarkeitsprüfung noch möglich sind. Dazu bietet der SAP Solution Manager eine Umgebung an, in der Sie eigene Testskripts definieren und testen lassen können.

Das End User Experience Monitoring ist eng mit dem End-to-End-Trace verbunden. Mit diesem Monitoring erhalten Sie daher nicht nur Informationen über die gesamte Laufzeit einer Webseite, es werden auf dem Server (oder den Servern) auch Traces eingeschaltet und Informationen gesammelt, die Aufschluss über die Antwortzeitanteile der beteiligten Komponenten geben. Bei einem Alarm, also dem Ausfall einer Webanwendung oder einer Performanceverschlechterung, liefert es damit weiteren Einblick zur Analyse der Ursache des Fehlers.

Mithilfe dieses Monitorings können Sie zum einen die Verfügbarkeit »strategisch wichtiger« Webseiten zentral überwachen – egal, von welchem Server diese geliefert werden. Das Monitoring bietet dabei auch einen Content-Check an, d. h., Sie können prüfen, ob auch ein korrekter Inhalt auf der Webseite dargestellt wird. Zum anderen können Sie URL-Transaktionen, d. h. Folgen von HTML-Seiten, z. B. zum Anlegen eines Warenkorbs in einem Webshop, definieren, die periodisch abgespielt werden sollen. (Dabei müssen Sie natürlich darauf achten, dass keine echten Dokumente erzeugt werden.)

Reaktive Überwachung

Die Überwachung von Testanwendungen ergänzt in idealer Weise das »interne Monitoring« des SAP NetWeaver AS. Sie prüft nicht nur die technische Verfügbarkeit der Anwendung, sondern kann auch über das Ergebnis einer Anwendung eine Aussage machen (etwa ob ein »sinnvoller« Preis berechnet wird). Sie ist allerdings nicht proaktiv, d. h., sie meldet nur reaktiv Fehler, während z. B. die Überwachung der Workprozesse bereits einen Alarm auslösen könnte, wenn eine bestimmte Auslastung überschritten wird, der Endbenutzer aber noch keine Probleme erkennt.

Weitere Informationen zum End User Experience Monitoring finden Sie in Anhang E, »Informationsquellen«. Dieses Monitoring können Sie nicht nur für Webanwendungen, sondern auch für SAP-GUI-Anwendungen nutzen.

Werkzeuge von Drittanbietern

Darüber hinaus bieten zahlreiche Hersteller Werkzeuge zur Überwachung von URLs an. In Anhang E finden Sie den Verweis auf eine Internetseite, die Informationen zu solchen Monitoring-Produkten zusammenfasst.

8.3 Pufferung von Webdokumenten

Pufferbare und nicht pufferbare Webdokumente

Die Nutzung von Puffern (Caches) ist unverzichtbar für eine gute Performance von Webanwendungen, insbesondere im WAN. Das Internet verfügt dazu über eine Kaskade von Puffern, die wir in diesem Abschnitt vorstellen. Zunächst müssen wir aber zwischen pufferbaren und nicht pufferbaren Webdokumenten unterscheiden:

- Pufferbar sind Webdokumente, die sich über eine längere Zeit hinweg nicht ändern, beispielsweise statische Bilddateien, CSS-Dateien, JavaScript-Dateien sowie auch dynamische Metadatendokumente.
- In der Regel nicht pufferbar sind Dokumente mit betriebswirtschaftlichen Daten, die in Echtzeit vom Server berechnet werden. Auf Ausnahmen gehen wir in Abschnitt 13.5.4, »SAP HANA Cached Views«, ein.

8.3.1 Browserpuffer (Browsercache)

Browser verfügen über einen Puffer (Cache), in dem Webdokumente gespeichert werden können.

In den Browser-Entwicklungswerkzeugen wird in der Netzwerkansicht angezeigt, ob eine Anfrage aus dem Puffer gelesen wurde, z. B. durch die Information **(from cache)** neben dem Statuscode.

Definition der Puffereinstellungen

Der Server legt fest, welche Dokumente pufferbar sind, und steuert dies über HTTP-Kopfeinträge, die in der HTTP-Spezifikation RFC 7234 definiert sind. Tabelle 8.1 gibt eine Übersicht über die wichtigsten HTTP-Kopfeinträge, die das Verhalten des Puffers regeln. Die Spalte *Anfrage oder Antwort* gibt an, ob der Parameter in der Anfrage vom Browser oder in der Antwort vom Server gesetzt werden kann.

Pufferung mit Verfallsdatum

Um die Anzahl der Anfragen zu reduzieren, muss also der Server den Wert `max-age` auf einen Wert größer als null setzten, d. h., man legt über `max-age` ein Verfallsdatum fest.

Ein Problem mit am Browser gepufferten Dokumenten entsteht dann, wenn sich ein Dokument ändert. Da es keinen Kommunikationskanal vom Server zum Browser gibt, kann der Server den Browser über die Änderung nicht informieren. Als Konsequenz arbeiten Benutzer dann weiterhin mit gepufferten und daher veralteten Dokumenten. Im schlimmsten Fall kann dies zu Fehlern und Abbrüchen in der Anwendung führen, wenn die veralteten, gepufferten Dokumente nicht mehr zu den Dokumenten der Anwendung passen, die aktuell vom Server gelesen werden.

Anfrage oder Antwort	Parameter	Bedeutung
Anfrage	`cache-control : no-cache`	Der Server darf die Antwort nicht aus dem Puffer beantworten.
	`cache-control : max-age = <Wert>`, Wert in Sekunden	Der Server muss ein Dokument schicken, das aktueller ist als der spezifizierte Wert. `max-age=0` bedeutet also, dass der Server immer ein aktuelles Dokument schicken muss.
	`if-modified-Since : <Datum>`	Dieser Wert wird gesetzt, wenn sich ein Dokument im Browserpuffer befindet, der Browser aber die Aktualität des Dokuments am Server validieren möchte. Stellt der Server fest, dass das Dokument im Zeitraum seit diesem Datum nicht geändert wurde, sendet der Server einen Statuscode 304 und kein neues Dokument an den Browser. Dies bezeichnet man auch als bedingte Anfrage (*Conditional Request*).
Antwort	`cache-control : no-cache`	Der Browser darf die Anfrage nicht puffern.
	`cache-control : max-age = <Wert>`, Wert in Sekunden	Der Browser darf das Dokument aus dem Puffer laden, bis das Verfallsalter `max-age` erreicht ist. Der Eintrag `max-age = 315360000` bedeutet, dass ein Dokument für ein Jahr gepuffert werden darf. `max-age = 0` bedeutet, dass das Dokument gepuffert werden darf, dass der Browser aber beim Server nachfragen muss, ob das Dokument noch aktuell ist (*Conditional Request)*.
	`last-modified : <Datum>`	Angabe, wann das Dokument auf dem Server zuletzt geändert wurde

Tabelle 8.1 HTTP-Parameter zur Steuerung des Browserpuffers

Puffer löschen

Um das Problem kurzfristig zu lösen, kann der Benutzer die Anwendung neu starten, ohne den Puffer zu verwenden. In der Regel geschieht dies über

das Kommando Strg + F5. Alternativ kann auch in den Browsereinstellungen der Puffer gelöscht werden. Diese Empfehlung löst natürlich das Problem aber nicht befriedigend und nachhaltig.

Bedingte Validierung, Cache-Buster

Um das Problem für den Benutzer befriedigend zu lösen, stehen zwei Strategien zur Verfügung:

1. Pufferung mit bedingter Validierung
   Beim ersten Aufruf eines Webdokuments durch den Browser oder wenn der Puffer des Browsers geleert wurde, fordert der Browser das entsprechende Webdokument vom Server an. Der Server setzt in den Kopfinformationen der HTTP-Anfrage die Parameter `max-age = 0` und `last-modified = <Datum der letzten Änderung>` und den Statuscode 200. Damit schreibt der Browser das Webdokument in den Puffer.

   Wird die Anwendung neu gestartet, z. B. bei der nächsten Anmeldung des Benutzers, fordert der Browser das Webdokument erneut von Server an, denn die maximale Gültigkeit ist durch `max-age = 0` auf null Sekunden gesetzt. Der Browser sendet eine bedingte Anfrage mit der HTTP-Kopfinformation `If-Modified-Since = <Datum der letzten Änderung>`. Stellt der Server fest, dass das Webdokument in diesem Zeitraum nicht geändert wurde, sendet der Server einen Statuscode 304 (`not modified`) und kein neues Dokument an den Browser.

   Mit dieser Strategie ist man sicher, dass die Dokumente im Puffer nicht veraltet sind. Zu einer Performanceverbesserung trägt diese Strategie aber nur bedingt bei: Man reduziert zwar die übertragene Datenmenge der Antwort durch die Pufferung, nicht aber die Anzahl der Anfragen insgesamt.
2. Pufferung und Invalidierung über das Konzept des sogenannten *Cache-Busters*
   - Der URL von Webdokumenten, die gepuffert werden sollen, wird eine Versionsnummer bzw. ein Zeitstempel hinzugefügt, der auch als *Cache Buster Token* bezeichnet wird. (Beachten Sie, dass dieser Cache Buster Token keine neue Versionsnummer der Software bedeuten muss.)
   - Wenn eine Webanwendung gestartet wird, sendet diese (idealerweise genau eine) Anfrage an den Server, der Server sendet die aktuelle Liste der Cache Buster Tokens an den Browser.
   - Hat sich ein Dokument am Server geändert, dann wird der Cache Buster Token geändert. Damit stimmt der neue Token nicht mehr mit dem Token der gepufferten Dokumente überein. Der Browser liest daraufhin die geänderten Dokumente nach und greift nicht mehr auf die veralteten, gepufferten Dokumente zu.

Weitere Details zur Implementierung des Cache-Buster-Konzepts, z. B. für Fiori-Anwendungen, finden Sie im Verlauf dieses Kapitels.

8.3.2 Puffer im ICM und im SAP Web Dispatcher

Auf der SAP-Serverseite verfügen der ICM und der SAP Web Dispatcher über einen als ICM-Servercache oder Internet-Servercache bezeichneten Puffer, in dem Webdokumente gespeichert werden, bevor sie zum Browser geschickt werden. Beim nächsten Zugriff kann der Inhalt direkt aus dem Puffer gelesen werden, ohne das Dokument erneut vom eigentlichen Anwendungsserver (SAP NetWeaver AS ABAP, AS Java, XS Engine usw.) zu lesen.

Die Pufferung auf dem ICM und dem SAP Web Dispatcher wird über den Profilparameter `icm/HTTP/server_cache_<xx>` (im Profil des ICMs bzw. des SAP Web Dispatchers) aktiviert.

Über die HTTP-Kopfparameter steuert die Anwendung (z. B. in der ABAP-Implementierung der HTTP-Handlerklasse), welche Webdokumente gepuffert werden sollen:

- Wird von der Anwendung der SAP-spezifische HTTP-Kopfparameter `sap-cache-control: max-age = <Wert>` gesetzt, wird das Dokument im Puffer des ICMs gespeichert.
- Wird von der Anwendung der HTTP-Kopfparameter `cache-control: max-age = <Wert>` gesetzt, wird das Dokument im Puffer des SAP Web Dispatchers gespeichert.

Pufferinvalidierung im ICF

Da der ICM integrativer Bestandteil des SAP NetWeavers ist, kann die Anwendung Puffereinträge aktiv invalidieren. Dies geschieht auf dem SAP NetWeaver AS ABAP über Methoden der HTTP-Handlerklasse `CL_HTTP_SERVER` bzw. der Schnittstelle `IF_HTTP_SERVER`. Über weitere Methoden dieser Klasse können Dokumente auch proaktiv in den Puffer geladen werden.

Im Gegensatz dazu können die im SAP Web Dispatcher gepufferten Dokumente nicht vom SAP NetWeaver AS ABAP invalidiert werden, da dieser keine Kenntnis von dem vorgeschalteten SAP Web Dispatcher besitzt. Der Web Dispatcher ist, wie andere Webcaches auch, auf passive Invalidierung mithilfe einer Verfallszeit (`max-age`) bzw. eines Cache Buster Tokens angewiesen.

Administration des Puffers

Gepufferte Dokumente können mit der Administrationsumgebung des ICMs bzw. des SAP Web Dispatchers überwacht und notfalls auch invalidiert werden:

1. Rufen Sie im Falle des ICMs in der Administrationsumgebung (Transaktionscode SMICM) im Menü die Funktion **Springen • HTTP Plugin • Server Cache • Anzeigen** auf.
2. Das System zeigt eine Liste der gepufferten Dokumente mit folgenden Attributen an:
   - URL-Pfad
   - Einlagerungszeitpunkt und Gültigkeitsdauer (Datum und Uhrzeit)
3. Um einen Puffereintrag zu invalidieren, wählen Sie den betreffenden Eintrag aus, und wählen Sie **Cache-Eintrag invalidieren**.

Die Pufferstatistik stellt Informationen bereit, beispielsweise zur Größe und Belegung des Puffers, zu den Pufferzugriffen und zur Trefferquote. Um die Statistik anzuzeigen, wählen Sie in der Administrationsumgebung (Transaktionscode SMICM) **Springen • HTTP Plugin • Server Cache • Statistik anzeigen**. Analoge Administrationsfunktionen finden Sie in der Administrationsoberfläche des SAP Web Dispatchers.

In der SAP-Hilfe finden Sie weitere Informationen zur Funktionsweise der Puffer, ihrer Administration und zur Implementierung der HTTP-Handlerklasse `CL_HTTP_SERVER` und zur Schnittstelle `IF_HTTP_SERVER`, mit der Sie den Puffer aus dem ABAP-Programm heraus steuern können.

8.3.3 Content Delivery Network

Ein *Content Delivery Network (CDN)* ist ein weltweit verteiltes Netzwerk von Servern, die Inhalte von Webseiten puffern und optimieren. Wenn Sie Webinhalte über ein Content Delivery Network bereitstellen wollen, dann vereinbaren Sie vertraglich, dass bestimmte Basis-URLs über das Content Delivery Network abgewickelt werden. Stellt ein Benutzer eine Anfrage an Ihre Webanwendung, so verweist der Domain-Service im Internet ihn auf den nächstgelegenen Server des Content Delivery Networks. Das heißt, pufferbare Anteile müssen nicht von Ihrem SAP-System, das möglicherweise auf einem anderen Kontinent steht, bereitgestellt werden, sondern werden von einem nahegelegenen Server des Content Delivery Networks geliefert.

Content Delivery Networks sind aber mehr als reine Puffer. Sie nehmen für sich in Anspruch, dass sie abhängig von dem Browsertyp, den ein Benutzer verwendet, und der Last-Mile-Verbindung (kabelgebunden oder Mobilfunk) auch die Inhalte selbst optimieren. Weiterhin nehmen sie für nicht

pufferbare Anfragen eine Routenoptimierung für die mittleren Meilen der Webanfrage vor, d. h., sie finden einen vermeintlich schnelleren Weg über ihre eigenen Server vom Browser zu Ihrem Server.

Auf die Optimierungen im Einzelnen einzugehen, würde den Rahmen dieses Buches sprengen, in Anhang E, »Informationsquellen«, verweisen wir auf einige Artikel, die die Optimierungen beschreiben und auch weitergehende Informationen zu Content Delivery Networks bereitstellen.

Die SAP arbeitet bei der Bereitstellung ihrer Cloud-Services mit dem Content Delivery Network *Akamai* zusammen. Als Beispiel sind die Inhalte von SAPUI5 zu nennen sowie Software-as-a-Service-Angebote wie SAP Cloud for Customer. Diese Zusammenarbeit steht allerdings erst am Anfang, es ist aber zu erwarten, dass sie sich ausbauen wird. Sie sollten sich also gelegentlich darüber informieren, welche Services SAP über Akamai anbietet.

8.3.4 Zusammenfassung: Nutzung von Puffern in Webanwendungen

Tabelle 8.2 fasst die Vorteile der Nutzung der unterschiedlichen Puffer im Web zusammen.

Eigenschaft	Browser-puffer	Content Delivery Network	SAP-Web-Dispatcher-Puffer	ICM-Puffer
Reduziert die Anzahl der Browseranfragen.	ja (bei Pufferung mit Verfallsdatum)	nein	nein	nein
Reduziert das Datenvolumen der Browseranfragen.	ja	nein	nein	nein
Reduziert die Latenzzeit für Browseranfragen.	ja	ja	nein	nein
Reduziert die Last und Antwortzeit auf dem Server.	ja	ja	ja	ja
Puffer kann aktiv invalidiert werden.	über einen Cache-Buster	über einen Cache-Buster	über einen Cache-Buster	direkt

Tabelle 8.2 Zusammenfassende Übersicht zur Nutzung von Puffern in Webanwendungen

8.4 Performanceanalyse von Webanwendungen auf dem SAP NetWeaver Application Server ABAP

Internet Communication Manager (ICM)

Als Manager für an den SAP NetWeaver AS gerichtete Web-Requests dient der Internet Communication Manager (ICM). Dieser nimmt die Anfragen der Webclients entgegen und verteilt sie an den ITS, an den Dispatcher des SAP NetWeaver AS ABAP bzw. an den SAP NetWeaver AS Java. Optional ist dem SAP NetWeaver AS ein SAP Web Dispatcher vorgeschaltet (siehe Abbildung 1.5 in Abschnitt 1.1.3, »Client-Server-Architektur«).

Service-Pflege (SICF)

ABAP-basierte Webanwendungen werden in der Servicepflege (Transaktionscode SICF) definiert. Dort wird der Service aktiviert und der URL-Pfad definiert. Wenn Sie die Servicedefinition öffnen und dann zur Tabelle **Handler-Liste** navigieren, finden Sie die Klasse, die die Webanwendung implementiert und die Schnittstelle zwischen der HTTP-Anfrage und der eigentlichen Anwendung bildet.

Info-Service

Ein denkbar einfacher Webservice ist der Service *http://<server>:<port>/sap/public/info*. Dieser gibt ein XML-Dokument mit einigen Systemparametern zurück. In der Servicepflege (Transaktionscode SICF) finden Sie diesen unter **DEFAULT_HOST • SAP • Public • Info**. Mit der Funktion **Service testen** aus dem Kontextmenü können Sie diesen Service starten. Wenn Sie die Servicedefinition öffnen und dann zur Tabelle **Handler-Liste** navigieren, finden Sie die implementierende (ABAP-)Klasse, in unserem Beispiel `CL_HTTP_EXT_INFO`. Durch einen Doppelklick können Sie in die Klasse hineinnavigieren und sehen, wie dort das Antwortdokument zusammengebaut wird.

Nach diesem Schema sind alle ABAP-Webanwendungen gebaut. Als HTTP-Handlerklasse verfügen diese entweder über eine spezifisch für den Service implementierte Klasse, oder sie verwenden eine Rahmenanwendung wie WebDynpro ABAP oder OData, welche die HTTP-Handlerklasse implementiert und Standardaufgaben der HTTP-Kommunikation übernimmt, sodass Sie sich als Entwickler auf die betriebswirtschaftlichen Inhalte konzentrieren können.

In diesem Abschnitt stellen wir die Performanceanalyse für Webanwendungen auf dem SAP NetWeaver AS ABAP vor. Dabei unterscheiden sich die grundlegenden Methoden nicht, egal, ob es sich um BSP, WebDynpro ABAP,

den internen ITS oder auf dem SAP NetWeaver AS ABAP implementierte Fiori-Anwendungen und OData-Services handelt. Folglich behandeln wir in den nächsten Abschnitten die gemeinsamen Methoden, bevor wir uns den Spezifika der einzelnen Anwendungsstypen widmen.

8.4.1 HTTP-Trace im Internet Communication Manager

Wie zu Beginn dieses Kapitels erwähnt, laufen alle Webanfragen an den SAP NetWeaver AS über den Internet Communication Manager (ICM). In Kapitel 2, »Analyse von Hardware, Datenbank und ABAP-Applikationsserver«, haben wir Ihnen die Engpassanalyse für diese Komponente vorgestellt.

Auf dem ICM können Sie ein Logging einschalten:

ICM-Monitor

1. Rufen Sie dazu den ICM-Monitor (Transaktionscode SMICM) auf, und navigieren Sie in diesem Monitor weiter zu **Springen • http Plugin • Server Logs**.
2. Unter dem Menüpunkt **Loghandler** finden Sie die Funktionen **Einträge Anzeigen**, **Aktivieren** und **Deaktivieren**.

In seiner Standardanzeige zeigt der HTTP-Trace u. a. die IP-Adresse des Webservers, den Zeitstempel, das HTTP-Kommando und den Rückgabewert, z. B. 200 für eine erfolgreiche Verarbeitung, die Größe der Anfrage in Byte und die Dauer der Anfrage an. Der Inhalt der Datei wird nach einiger Zeit automatisch überschrieben.

Welche Daten zusätzlich verfolgt werden, wird über den Parameter `icm/HTTP/logging_*` konfiguriert. * ist normalerweise 0, kann jedoch auch eine beliebige Nummer sein. Diesen Parameter können Sie in der Parameterpflege (Transaktionscode RZ11) oder auch direkt im ICM-Monitor unter dem Menüpunkt **Springen • Parameter • Ändern** pflegen. In der Parameterpflege finden Sie eine ausführliche Dokumentation, die auch die Standard-Logfile-Formate beschreibt.

Für den End-to-End-Trace mit dem SAP Solution Manager wird das HTTP-Logging zielgerichtet aktiviert, und dadurch werden die Serverzeiten gemessen. Der ICM kann so konfiguriert werden, dass ein HTTP-Log-Eintrag nur dann geschrieben wird, wenn das E2E-Trace-Plug-in verwendet wird. Der ICM prüft, ob in der Anfrage das Feld `X-CorrelationID` vorhanden ist, und schreibt dann einen Eintrag ins Log. Weitere Informationen hierzu finden Sie in SAP-Hinweis 1252944.

8.4.2 Performanceanalyse von ABAP-Webanwendungen durchführen

Zur Performanceanalyse von ABAP-Webanwendungen können Sie alle Verfahren und Werkzeuge verwenden, die Sie bereits zur Performanceanalyse von SAP-GUI-Anwendungen kennen. Einige Besonderheiten sind jedoch zu beachten, die wir hier besprechen.

Generelle Performanceanalyse

Wenn Benutzer über schlechte Performance klagen, starten Sie, wenn das Performanceproblem aktuell besteht, den Workprozess-Monitor (Transaktionscode SM50 bzw. SM66). Sind alle Workprozesse belegt, nehmen Sie eine Engpassanalyse gemäß Kapitel 2, »Analyse von Hardware, Datenbank und ABAP-Applikationsserver«, vor, um die Ursache einzugrenzen und z. B. Benutzer und Programme zu identifizieren, die das Problem verursachen.

Können Sie im Workprozess-Monitor keine langlaufenden Prozesse beobachten, stellen die Benutzer aber trotzdem ein Performanceproblem fest, starten Sie den ICM-Monitor (Transaktionscode SMICM). Sind hier alle ICM-Threads belegt, kann möglicherweise das Problem behoben werden, indem Sie die Anzahl der ICM-Threads erhöhen. Verwenden Sie den integrierten ITS und liegt ein Problem mit hoher Speicherbelegung im Extended Memory vor, beachten Sie die Hinweise zum integrierten ITS in Abschnitt 8.5, »Business Server Pages (BSP), Web Dynpro ABAP und integrierter ITS«.

Um die Performance rückblickend zu bewerten, rufen Sie den Workload-Monitor (Transaktionscode ST03) auf. Der Monitor weist neben den bekannten Task-Typen **Dialog**, **Background**, **Update** etc. die Task-Typen **HTTP**, **HTTPS** und **SMTP** aus (natürlich nur, sofern entsprechende Anfragen vom System bearbeitet wurden). Anhand dieser Task-Typen können Sie einfach die Aktivität und die Antwortzeiten der entsprechenden Anfragen für Webanwendungen bewerten.

Performancestatistiken aktivieren

Um die Performanceanalyse mit dem Workload-Monitor durchführen zu können, müssen Sie allerdings das Schreiben der Performancestatistiken aktivieren. Im Auslieferungszustand schreibt der SAP-Kernel für HTTP-, HTTPS- und SMTP-Anfragen keine Statistiksätze. Um die Statistiken zu aktivieren, setzen Sie den SAP-Profilparameter `rdisp/no_statistic` auf den Leerwert. (Das heißt, Sie fügen in die Profilparameterdatei die Zeile `rdisp/no_statistic =` ein. Der Kernel-Standardwert ist `PLUGIN`.) Diesen SAP-Profilparameter können Sie bei laufendem System ändern, indem Sie wie folgt vorgehen:

1. Wählen Sie zunächst im Systemlastmonitor (Transaktionscode ST03) die Rolle **Experte** (Schaltfläche in der linken oberen Ecke).

2. Folgen Sie dann im Baum den Punkten **Kollektor & Perf. Datenbank • Statistiksätze & -datei • Online Parameter • Dialogschrittstatistik**.
3. Die Tabelle **Laufzeitparameter der Statistiksammlung** wird angezeigt. Löschen Sie in der Spalte `rdisp/no_statistic` den Wert `PLUGIN`. Sofern sich in der Spalte `rdisp/no_statistic` der Wert `PLUGIN` befindet, werden keine Statistiksätze geschrieben. Ist die Spalte leer, ist das Schreiben der Statistik aktiviert.
4. Aktivieren Sie Ihre Änderung über die Schaltfläche **Werte aktivieren**.

Die Onlineaktivierung ist nur bis zum nächsten Neustart der entsprechenden SAP-Instanz aktiv. Um die Statistiken permanent einzuschalten, ändern Sie den Parameter in der Konfigurationsdatei.

Detailanalyse

Im Transaktionsprofil des Workload-Monitors können Sie eine Detailanalyse der aufgerufenen Webanwendungen vornehmen. Sie können festlegen, in welcher Granularität er die Performancestatistiken für Webseiten im Transaktionsprofil darstellen soll:

1. Starten Sie dazu das Programm `SWNC_CONFIG_URL`.
2. Wählen Sie dann eine der folgenden Optionen aus:
   - **Auflösung vollständig nach Anwendung und Seite**
     Die Performancestatistiken werden pro Webseite erstellt, d. h., im Transaktionsprofil finden Sie für jede prozessierte Webseite einen Eintrag.
   - **Auflösung nach Anwendung**
     Eine BSP- oder Web-Dynpro-ABAP-Anwendung kann aus mehreren Webseiten bestehen. Im Transaktionsprofil wird bei dieser Auflösung eine Performancestatistik pro BSP-Anwendung erstellt, die Seiten selbst werden nicht aufgelöst.
   - **Summierung unter Report SAPMHTTP**
     Es erfolgt keine Auflösung nach ABAP-Webservice, d. h., alle Web-Requests werden im Transaktionsprofil unter dem Eintrag **SAPMHTTP** gesammelt.

Das Einschalten der Statistiken bringt nur geringe Performanceeinbußen und ist daher praktisch immer zu empfehlen. Nur bei Systemen unter sehr starker Last sollten Sie testen, ob sich die Performance entscheidend verbessert, wenn Sie die Statistiken für Webanwendungen wieder deaktivieren.

Spezielle Performanceanalyse

Wollen Sie die Performance einer bestimmten ABAP-Webanwendung im Detail analysieren, verwenden Sie zunächst die Einzelsatzstatistik (Transaktionscode STATS). Dazu wählen Sie im Selektionsbildschirm der Einzelsatzstatistik im Feld **Task type** den Wert »H« für HTTP-Anfragen bzw. »T« für

HTTPS-Anfragen aus. Um die Einzelsatzstatistik verwenden zu können, muss das Schreiben der Statistik im SAP-Kernel aktiviert sein, wie wir bereits weiter oben in diesem Abschnitt beschrieben haben.

Tabelle 8.3 stellt die wichtigsten Performancekennzahlen für HTTP-Anfragen zusammen.

Feld	Bedeutung
Quantity	Anzahl der HTTP-Anfragen
Calling Time	Gesamtantwortzeit für die HTTP-Anfrage, d. h. die für den Logon plus Zeit im ICF plus *Execution Time*
Execution Time	Ausführungszeit der HTTP-Anfrage im ABAP
Sent Data	gesendete Datenmenge
Received Data	empfangene Datenmenge

Tabelle 8.3 Felder des HTTP-Profils im Workload-Monitor bzw. im HTTP-Einzelsatz in der Einzelsatzstatistik

Trace für ABAP-Webanwendungen

Da es sich bei ABAP-Webanwendungen im Wesentlichen um in ABAP programmierte Anwendungen handelt, stehen Ihnen als Werkzeuge für die Detailanalyse z. B. der Performance-Trace, die ABAP-Laufzeitanalyse und der Debugger zur Verfügung. Die Beschreibung der entsprechenden Werkzeuge finden Sie in Kapitel 5, »Optimierung von ABAP-Programmen«.

Die Laufzeitanalyse für ABAP-Webanwendungen aktivieren Sie, wie in Abschnitt 5.2, »Performanceanalyse mit dem ABAP-Trace (Laufzeitanalyse)«, beschrieben. Achten Sie bei der Einplanung der Messung darauf, dass Sie den richtigen Prozesstyp (**HTTP**) und Objekttyp (**URL**) auswählen.

Alternativ können Sie die Laufzeitanalyse auch in der Servicepflege (Transaktionscode SICF) einschalten. Markieren Sie im Navigationsbaum den zu untersuchenden Service, und aktivieren Sie die Laufzeitanalyse über:

Bearbeiten • Laufzeitanalyse • Aktivieren

8.4.3 Aufrufe von Webservices überwachen

Aufzeichnung in Einzelsatzstatistik

Werden von einem ABAP-Server Webservices aufgerufen, werden diese Aufrufe im Statistiksatz ebenfalls aufgezeichnet. Die Details ausgehender HTTP-Aufrufe finden Sie in der Einzelsatzstatistik im Abschnitt **HTTP Records • As Client**. Die Zeit für die Aufrufe finden Sie dort als **Calling Time**, außerdem Daten zu den Aufrufzielen und zur übertragenen Datenmenge.

Da der Ablauf eines HTTP-Aufrufes dem des synchronen RFCs sehr ähnelt, kann Ihnen Abschnitt 7.3, »Remote Function Calls (RFCs)«, helfen, das Verständnis zu vertiefen. Die Rolle der RFC-Zeit übernimmt im Fall des HTTP-Aufrufes die **Calling Time** im Abschnitt **HTTP Records**. Im Fall eines Transaktionsschrittes mit HTTP-Aufruf, aber ohne RFC-Aufruf und SAP GUI Control gilt also, dass die HTTP-Calling-Zeit größer sein muss als die Roll-Wartezeit. Beobachten Sie eine hohe Roll-Wartezeit im Hauptteil des Statistiksatzes und können Sie weder die RFC-Zeit noch die GUI-Zeit dafür verantwortlich machen, dann sollte sich Ihr Augenmerk auf die Frage richten, ob eventuell ein zeitintensiver HTTP-Aufruf als Ursache infrage kommt.

8.5 Business Server Pages (BSP), Web Dynpro ABAP und integrierter ITS

In diesem Abschnitt stellen wir Ihnen weitere Details zu *Business Server Pages* (BSP) und *Web Dynpro ABAP* sowie zum integrierten ITS vor.

8.5.1 Business Server Pages und Web Dynpro ABAP

BSP und Web-Dynpro-ABAP-Anwendungen sind eine Möglichkeit, SAP-Webanwendungen zu schreiben. BSP bestehen aus einzelnen Webseiten, die in HTML und ABAP als Skriptsprache geschrieben sind. Mit derselben technologischen Basis ist Web Dynpro ABAP die Fortentwicklung zu einer deklarativen, modellorientierten Entwicklungsumgebung. Das Programmiermodell von Web Dynpro for ABAP basiert auf dem Design Pattern *Model View Controller* (MVC). Ursprünglich im Umfeld von Smalltalk-80 entwickelt, ist es heute De-facto-Standard für die Entwicklung von Benutzeroberflächen. Es erlaubt die strikte Trennung von Datenmodell (*Model*), Darstellung der Daten an der Oberfläche (*View*) und Verarbeitungskontrolle (*Controller*). Für die Methoden der Performanceanalyse macht es grundsätzlich keinen Unterschied, ob Sie das einfache BSP-Programmiermodell oder Web Dynpro ABAP als Entwicklungsumgebung verwenden. Im Folgenden werden wir Ihnen die Performanceanalyse daher anhand des statischen BSP-Entwicklungsmodells vorstellen.

BSP und Web-Dynpro-ABAP-Anwendungen werden vollständig in der SAP Development Workbench entwickelt, d. h., es gibt nicht, wie in der Vergangenheit beim externen SAP ITS oder der SAP J2EE Engine, Bestandteile von Webanwendungen, die in Dateisystemen außerhalb der Datenbank des SAP-Systems abgelegt werden. Die HTML-Seiten werden zur Laufzeit von einem SAP-Workprozess generiert. Die SAP Development Workbench und

die SAP-Laufzeitumgebung verfügen dazu über das sogenannte *Internet Communication Framework* (ICF) und den *Internet Communication Manager* (ICM). Beide sind Bestandteil der Auslieferung und der Installation des SAP NetWeaver AS.

Anmelden an den Applikationsserver

Eine BSP-Anwendung in einem SAP-System rufen Sie auf, indem Sie folgende URL in Ihren Webbrowser eingeben:

https://<sapserver>:<port>/sap/bc/bsp/sap/<bsp_anwendung>/<seite>

Dabei steht *<sapserver>* für den Namen eines Anwendungsservers, auf dem das SAP-System läuft. Dieser muss immer voll qualifiziert angegeben werden, also z. B. *sapapp1.stadt.company.com* und nicht nur *sapapp1*. *<port>* ist der TCP/IP-Port, auf dem der Anwendungsserver hört. Dieser Wert steht aus Sicherheitsgründen zunächst auf 0 und muss vom Administrator auf den gewünschten Wert geändert werden. *<bsp_anwendung>* ist der Name der BSP-Anwendung und *<seite>* der Name der Seite innerhalb der Anwendung.

Entwicklungsumgebung für BSP

Abbildung 8.3 und Abbildung 8.4 zeigen die Entwicklung einer BSP-Anwendung. In Abbildung 8.3 erkennen Sie die HTML-Seite **Your Book Search Results**, wie sie in einem Browser dargestellt wird.

Your Book Search Results

The matches for this search are:

ISBN	Title	Author
978-3-8362-1888-7	Adobe Flash CS6	Nick Weschkalnies
978-3-8362-1888-7	Adobe Flash CS6	Rojahn Ahmadi
978-3-8362-1646-3	SAP NetWeaver AS ABAP - Systemadministration	Liane Will
978-3-8362-1646-3	SAP NetWeaver AS ABAP - Systemadministration	Frank Föse
978-3-8362-2177-1	SAP-Performanceoptimierung	Thomas Schneider
978-3-8362-1993-8	SAP NetWeaver BW - Performanceoptimierung	Thomas Schröder
978-3-8362-1890-0	Adobe Dreamweaver CS6	Hussein Morsy

Abbildung 8.3 HTML-Seite »Your Book Search Results«

In Abbildung 8.4 ist das zugehörige Coding dargestellt. Eine BSP-Anwendung besteht aus der eigentlichen Webseite, die in HTML geschrieben ist ❶ und in der ABAP-Coding ❷ zur Datenaufbereitung in sogenannten *Tags* eingebettet ist (z. B. ABAP-Loops zum Füllen von Tabellen). Das Programmiermodell ist also vergleichbar mit Java Server Pages (JSPs), bei denen Java-Anwendungen in den HTML-Text eingebunden sind.

Events

Darüber hinaus umfasst eine BSP-Anwendung sogenannte *Events*. Dies sind in ABAP geschriebene Programmteile, die zu bestimmten Zeitpunkten, z. B. der Initialisierung der Webseite oder nach der Dateneingabe, prozessiert werden. Diese Events erlauben die Bildsteuerung der Seite und die Datenbeschaffung (z. B. Zugriffe auf die Datenbank). Insgesamt kann eine komplette BSP-Anwendung aus mehreren solcher Webseiten bestehen.

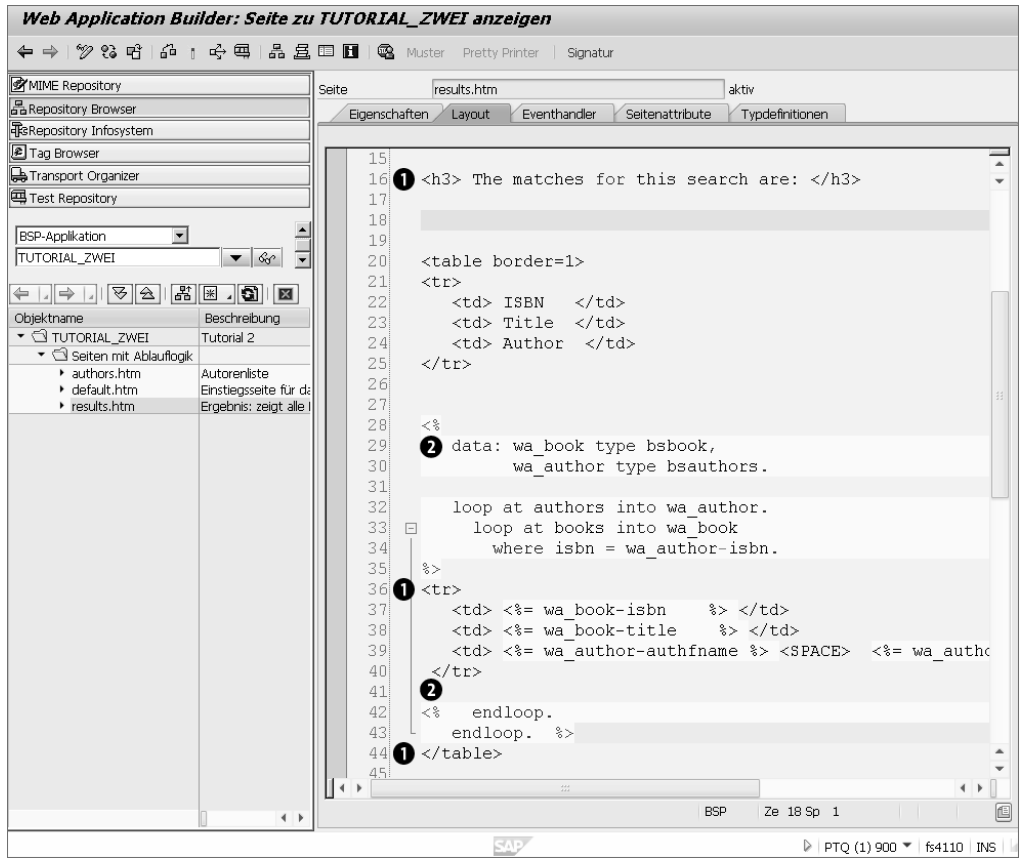

Abbildung 8.4 Zur HTML-Seite »Your Book Search Results« gehöriger Programmtext in HTML und ABAP

BSP-Entwicklungsumgebung aufrufen

Die SAP-Entwicklungsumgebung (ABAP Workbench, Transaktionscode SE80) wurde um die entsprechenden Funktionen zur Entwicklung von BSP erweitert. Um eine BSP-Seite in einem SAP-System anzuzeigen, gehen Sie in der ABAP Development Workbench wie folgt vor:

1. Wählen Sie die Schaltfläche **Repository Browser**.
2. In dem unter der Schaltfläche befindlichen Auswahlfeld wählen Sie die Kategorie **BSP-Applikation** aus. Geben Sie in dem darunterliegenden Eingabefeld den Namen einer BSP-Anwendung ein, und bestätigen Sie die Eingabe.

3. In dem nun erscheinenden Baum finden Sie im Teilbaum **Seiten** die HTML-Seiten, die zur entsprechenden BSP-Anwendung gehören. Wählen Sie eine Seite aus. Nun befinden Sie sich in der Entwicklungsumgebung für eine spezielle BSP-Seite.

Um sich eine Vorschau auf die entsprechende Webseite anzuschauen, wählen Sie die Registerkarte **Vorschau**. Um sich den Programmcode der Webseite anzeigen zu lassen, klicken Sie auf die Registerkarte **Layout**. Um die zu einer BSP-Seite gehörigen Events anzuzeigen, also den Programmcode, der z. B. beim Aufruf einer Seite oder bei einer Eingabe ausgeführt wird, wählen Sie die Registerkarte **Events**.

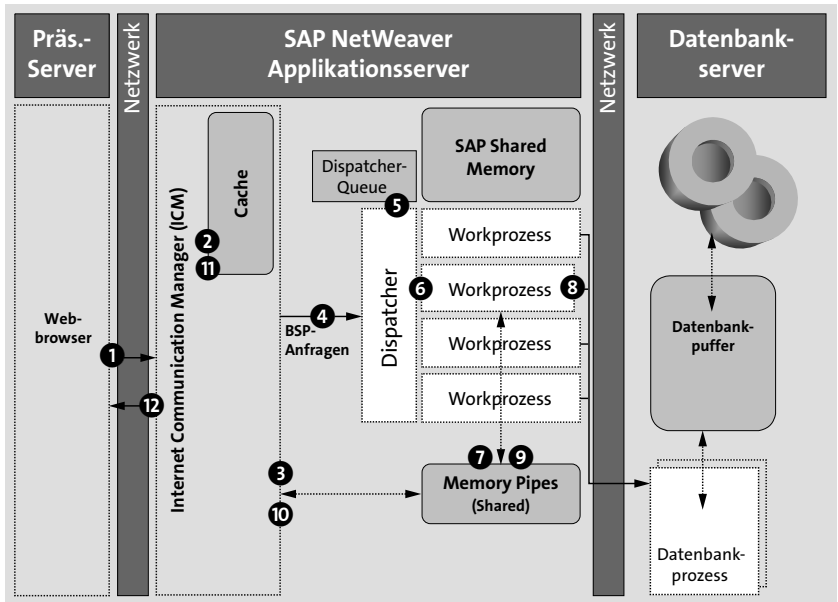

Abbildung 8.5 Ablauf eines Transaktionsschrittes im SAP NetWeaver AS beim Aufruf einer BSP-Seite

Generieren der HTML-Seite

Beim Aufruf einer BSP-Seite werden die in Abbildung 8.5 dargestellten Schritte durchlaufen, um eine HTML-Seite zu generieren. Der Webbrowser sendet seine Anfrage zunächst an den ICM ❶. Dieser prüft, ob die Browseranfrage mithilfe der in seinem Cache gespeicherten Informationen beantwortet werden kann ❷. Ist dies nicht der Fall, übergibt er die Anfrage an den SAP-Dispatcher ❹, nachdem er die Daten der Anfrage in sogenannte *Memory Pipes* gespeichert hat ❸. Memory Pipes gehören zum Shared Memory der SAP-Instanz und dienen der Kommunikation zwischen dem ICM auf der einen und den SAP-Workprozessen auf der anderen Seite. Sofern ein Dialog-Workprozess für die Bearbeitung zur Verfügung steht

und der Dispatcher die Anfrage nicht in der Queue zwischenparken muss ❺, übergibt der Dispatcher die Anfrage einem Workprozess ❻. Dieser liest die benötigten Daten aus den Memory Pipes ❼ und bearbeitet die Anfrage ❽. Nachdem der Workprozess die BSP bearbeitet und die Webseite erstellt hat, stellt er die Daten in die Memory Pipe ❾ und übergibt die Kontrolle zurück an den ICM ❿. Dieser sendet die fertige HTML-Seite an den Webbrowser ⓫–⓬.

Performanceanalyse

Zur Performanceanalyse von BSP und ABAP-WebDynpro-Anwendungen nutzen Sie die in Abschnitt 8.4.2, »Performanceanalyse von ABAP-Webanwendungen durchführen«, vorgestellten Methoden.

8.5.2 Integrierter ITS

Der integrierte ITS ist als HTTP-Request-Handler im Internet Communication Framework (ICF) implementiert. Die entsprechende ITS-Anfrage, d. h. die ABAP-Programmlogik und das Erzeugen der HTML-Seite, werden also in einen ABAP-Workprozess bearbeitet. ITS-Templates und `MIME`-Dateien werden direkt in der Datenbank gespeichert. Der integrierte ITS wird durch den Profilparameter `itsp/enable = 1` aktiviert. Als Programmiermodelle werden vom integrierten ITS SAP GUI for HTML und Easy Web Transactions (EWT) unterstützt.

Die Anmeldung an das SAP GUI for HTML erfolgt über die folgende URL: *https://<sapserver>.<company.com>:<port>/sap/bc/gui/sap/its/webgui?*

Eine typische Easy Web Transaction, den Business Workplace, starten Sie mit folgender URL: *https://<sapserver>:<port>/sap/bc/gui/sap/its/bwsp/!*

Architekturänderungen

Im Vergleich zum alten, externen ITS bringt die integrierte Version große TCO-Gewinne, die sich aus den folgenden Architekturänderungen ergeben:

- Der ICM übernimmt die Rolle des Webservers, sodass der integrierte ITS keinen separaten Webserver benötigt.
- Der integrierte ITS ist als HTTP-Request-Handler im Internet Communication Framework (ICF) implementiert, sodass keine separate Installation nötig ist.
- Softwarelogistik, Monitoring und Administration können genauso wie bei anderen ABAP-Programmen genutzt werden.

Hauptspeicherverbrauch

Der integrierte ITS nutzt den globalen Extended Memory (*SAP EG Memory*) zur Ablage der Laufzeitversion der HTML Business Templates (sogenannte *Pre-parsed Templates*). Dabei wird pro Browsertyp (Internet Explorer, Firefox etc.) und Sprache ein eigenes Laufzeit-Template gespeichert, sodass sich die Größe des Speicherbereichs aus dem Produkt aus Anzahl der unter-

schiedlichen Anwendungen, Anzahl der Browsertypen, Anzahl der Sprachen und einem mittleren Speicherbedarf pro Anwendung ergibt. Dabei können Sie bei einer SAP-GUI-for-HTML-Anwendung von einem mittleren Speicherbedarf von 10 MB pro Anwendung ausgehen. Hinzu kommen etwa 3 MB pro Session im Extended Memory.

Zu den im Shared Memory gespeicherten Objekten gehören nur die HTML Business Templates, Grafikdateien (MIME-Dateien) werden im lokalen Speicher des ICMs gespeichert.

Performanceanalyse und Überwachung

Zur Performanceanalyse von Web-GUI-Anwendungen nutzen Sie die in Abschnitt 8.4.2, »Performanceanalyse von ABAP-Webanwendungen durchführen«, vorgestellten Methoden. Der Überwachung des benötigten Speichers dient der ITS-Status-Monitor (Transaktionscode SITSPMON).

8.6 SAP Fiori, SAPUI5 und OData-Services

SAP Fiori, SAPUI5 und OData sind serverübergreifende Konzepte mit Implementierungen auf dem SAP NetWeaver AS ABAP, der SAP HANA XS Engine sowie auf der SAP Cloud Platform.

In diesem Abschnitt stellen wir zunächst die plattformübergreifenden Konzepte vor und gehen dann auf einige Spezifika der Implementierung auf dem SAP NetWeaver AS ABAP ein.

8.6.1 Grundlagen von Fiori, SAPUI5 und OData

Webanwendungen, die den gestiegenen Anforderungen von Benutzern entgegenkommen, verwenden interaktive und dynamische Webseiten mit asynchroner Kommunikation mit dem Webserver. Als Resultat wirken diese Anwendungen nicht eingefroren, während der Browser auf die Antwort des Webservers wartet. Über Webservices werden nur die geänderten Daten mit dem Webserver ausgetauscht.

Responsive Webdesign

Weiterhin erwarten Benutzer, dass sie Webanwendungen in unterschiedlichen Formfaktoren nutzen können, d. h. angepasst an ihr Smartphone, ihren Tablet- und ihren Desktop-Computer. Dies bezeichnet man auch als *Responsive Webdesign*.

SAPUI5

Diesen Anforderungen kommen mit SAPUI5-Technologie entwickelte Anwendungen nach. SAPUI5 implementiert ebenfalls das Design-Pattern Model View Controller (MVC), d. h. eine Trennung von Sicht (View), Controller und Modell, das wiederum mit dem Webserver kommuniziert.

Wenn wir in diesem Buch von SAPUI5 sprechen, so gelten die Aussagen analog auch für openUI5, die Open-Source-Variante von SAPUI5.

Open Data (OData) Protocol

Das *Open Data (OData) Protocol* ist ein anbieterneutraler OASIS-Standard für das Bauen und Verwenden von Webservices nach den Designprinzipien des *Representational State Transfer (REST) Protocol*. Es ist erweiterbar, d. h., es erlaubt Anbietern wie der SAP, eigene Datentypen und Annotationen zu verwenden.

SAP Fiori

SAP-Fiori-Anwendungen zeichnen sich durch die folgenden Merkmale aus:

- Sie folgen einem einheitlichen visuellen und Interaktionsdesign:
  - Sie sind für eine spezielle Nutzerrolle entworfen.
  - Sie folgen einem bestimmten grafischen Fiori-Design und verwenden ein einheitliches Theming.
  - Sie verwenden einfache und einheitliche Prinzipien der Benutzerführung und wiedererkennbare Interaktionsmuster.
- Fiori-Anwendungen werden über eine einheitliche Oberfläche, das *SAP Fiori Launchpad* (FLP), gestartet. Im SAP Fiori Launchpad richtet sich jeder Benutzer seine Sicht auf die Fiori-Anwendungen ein, die ihm sein Unternehmen aufgrund seiner Rolle zur Verfügung stellt.
- In ihrem technischen Design trennen Fiori-Anwendungen zwischen dem grafischen UI und der Datenbeschaffung. Als empfohlene Standardtechnologie verwenden sie SAPUI5 für den im Browser laufenden Anwendungsteil und OData für den Datenaustausch mit dem Server.

Nicht alle Fiori-Anwendungen erfüllen die Kriterien des technischen Designs in vollem Maße – auch »klassische« Web-GUI-Transaktionen mit Fiori-Visualisierung gehören zum Fiori-Portfolio. In diesem Abschnitt behandeln wir allerdings diejenigen Fiori-Anwendungen, die auf Basis von SAPUI5 und OData gebaut sind.

SAP Gateway

SAP Gateway als Bestandteil von SAP NetWeaver ist eine Komponente, die das OData-Protokoll implementiert und damit die Möglichkeit bietet, SAP-Systeme sicher, effizient und performant über das OData-Protokoll mit der Welt des Internets zu verbinden. SAP Gateway bietet die Services, um Anfragen zu analysieren und zu validieren und Antworten in den unterschiedlichen Formaten (z. B. XML/ATOM und JSON) zu erzeugen, Konvertierungen vorzunehmen und die Metadaten zum Service zu verwalten. Darüber hinaus verwaltet Gateway die Berechtigungen für OData-Services und bietet Werkzeuge zu Monitoring, Vermessung, Performance- und Fehleranalyse an. Das Konzept von SAP Gateway ist bisher auf dem SAP NetWeaver AS ABAP und der SAP Cloud Platform implementiert.

8.6.2 Aufbau einer Fiori-Systemlandschaft

Komponenten der Fiori-Systemlandschaft

Um SAP Fiori-Anwendungen für S/4HANA bzw. auf einem bestehenden SAP-System (SAP ERP, SAP CRM) zu betreiben, müssen Sie fünf Softwarekomponenten in Ihre Systemlandschaft integrieren:

1. die zentrale UI-Technologiekomponente mit SAPUI5 und SAP Fiori Launchpad
2. Fiori-Anwendungen (UIs), inklusive der SAP Fiori-Launchpad-Inhalte: Die Anwendungen und Inhalte sind spezifisch für ein SAP-Produkt, d. h., es gibt unterschiedliche Softwarepakete für SAP S/4HANA, SAP ERP, SAP CRM usw.
3. Die SAP-Gateway-Serverkomponente (und optional zentrale Gateway-Inhalte): Ab SAP NetWeaver 7.40 ist diese Teil von SAP NetWeaver, für ältere Versionen muss sie separat installiert werden.
4. Die SAP-Gateway-Backend-Komponente: Ab SAP NetWeaver 7.40 ist diese Teil des NetWeavers, für ältere Versionen muss sie separat installiert werden.
5. SAP Fiori-OData-Services (Backend-Integration): Die Services sind spezifisch für ein SAP-Produkt, d. h., es gibt unterschiedliche Softwarepakete für SAP S/4HANA, SAP ERP, SAP CRM usw. Bei SAP S/4HANA sind diese Bestandteil der Installation, für andere SAP-Produkte werden sie über Add-ons oder Supportpakete installiert.

Die SAP-Gateway-Backend-Komponente sowie die SAP Fiori-OData-Services werden immer auf dem S/4HANA-System bzw. auf dem bestehenden SAP-Anwendungssystem installiert, idealerweise ohne dieses selbst auf eine neue Version bringen zu müssen.

Für die verbleibenden drei Komponenten existieren unterschiedliche Szenarien:

- *Hub-Szenario*: separates SAP-Fiori-Frontend-System (auf Basis von SAP NetWeaver AS ABAP)
- *Embedded*: integriert in das SAP-Anwendungssystem (ebenfalls auf Basis von SAP NetWeaver AS ABAP)
- *SAP Fiori Cloud Edition*: Fiori-UI-Anwendungen und UI-Technologie werden in der SAP Cloud auf der SAP Cloud Platform betrieben. Für die SAP-Gateway-Serverkomponente gibt es in diesem Szenario zwei Optionen. Diese kann
  - ebenfalls auf der SAP Cloud Platform oder
  - auf einem lokalen Gateway-Server (auf Basis von SAP NetWeaver AS ABAP) betrieben werden.

Abbildung 8.6 stellt die erste Option, das *Hub-Szenario*, dar. Wird eine Fiori-Anwendung auf dem Browser gestartet, so erreicht diese zunächst den SAP Web Dispatcher, der die Anfrage an einen Anwendungsserver des SAP-Fiori-Frontend-Systems (FES) vermittelt. Von dort werden die Fiori-UI-Anwendungen und die SAPUI5-Bibliotheken geladen. OData-Anfragen werden über das RFC-Protokoll an das Anwendungssystem weitergeleitet.

In Kursivbuchstaben finden Sie in Abbildung 8.6 auch die Zeiten, die als SAP-Statistiken von den beteiligten Komponenten gemessen werden. Ein Wert am linken Rand einer Komponente repräsentiert die Brutto-Zeit, die am rechten Rand die Zeit, die eine Komponente den folgenden Komponenten zuweist. Weitere Details zu den SAP-Statistiken finden Sie in Abschnitt 8.2.2, »SAP-Statistiken in der HTTP-Anfrage«, und Abschnitt 8.6.4, »SAP Fiori, SAPUI5 und OData auf dem SAP NetWeaver Application Server ABAP«.

SAP-Fiori-Frontend-Server/-System (FES)

In der SAP-Dokumentation ist in der Regel von dem *SAP-Fiori-Frontend-Server* die Rede. In der Terminologie dieses Buches ist damit aber ein SAP-System gemeint, daher sprechen wir stattdessen auch von dem *SAP-Fiori-Frontend-System* (FES).

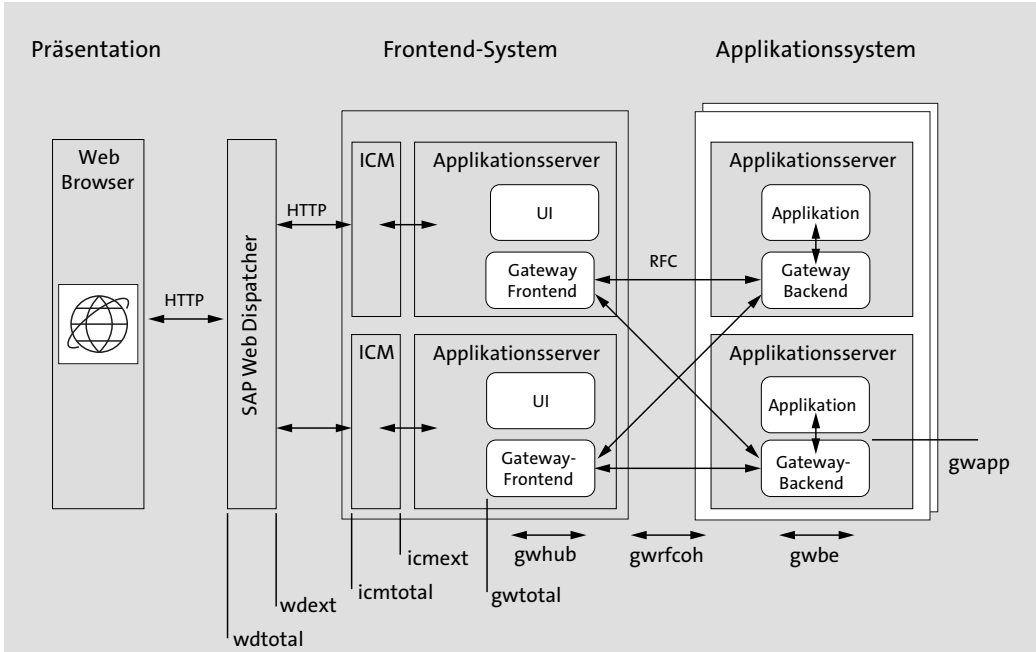

Abbildung 8.6 SAP Fiori-Systemlandschaft (Hub-Szenario)

Embedded-Szenario

Beim *Embedded-Szenario* wird kein separates SAP-Fiori-Frontend-System aufgebaut, d. h., alle UI- und Gateway-Komponenten werden auf dem Anwendungssystem konfiguriert. Der Fluss der Anfrage bleibt aber derselbe. Der Unterschied ist nur, dass Gateway-Serverkomponente und Backend-Komponente nicht über ein Netzwerk (RFC), sondern direkt miteinander kommunizieren.

Fiori Cloud Edition

Bei der Verwendung der *Fiori Cloud Edition* liegen die Fiori-UIs und die UI-Technologie in der SAP Cloud auf der SAP Cloud Platform. Bezüglich der Gateway-Serverkomponente können Sie wählen, ob Sie diese auch in der Cloud nutzen wollen oder als lokale Installation betreiben wollen. Als Verbindungskomponente zwischen Ihrem Anwendungssystem bzw. dem lokalen Frontend-System und der SAP Cloud Platform kommt der *SAP Cloud Connector* zum Einsatz, der im Netzwerk Ihres Anwendungssystems konfiguriert wird und eine sichere Verbindung zur SAP Cloud Platform aufbaut.

Das Fiori-Cloud-Edition-Szenario hat insbesondere den Vorteil, dass dort immer alle Versionen der Fiori-UIs vorliegen und Sie sich dort nur subskribieren müssen.

Bewertung der Optionen

Eine vollständige Darstellung der Vor- und Nachteile der Optionen darzustellen, geht über die Aufgabenstellung dieses Buches weit hinaus, dazu verweisen wir auf die SAP-Landschaftsempfehlungen, einen Verweis darauf finden Sie in Anhang E, »Informationsquellen«. An dieser Stelle schließen wir nur einige Performancebetrachtungen bezüglich Netzwerklatenz und Komponentenperformance an.

Netzwerklatenz

Bei der Entscheidung für eine der Landschaftsoptionen spielt die Netzwerklatenz eine wichtige Rolle, insbesondere die Entfernungen zwischen der Mehrheit der Benutzer, dem Fiori-Frontend-System und dem Anwendungssystem.

Aufgrund des *Same-Origin-Policy-(SOP-)*Prinzips laufen alle Anfragen, d. h. die Anfragen, die UI-Dokumente laden, und die OData-Anfragen, über das Fiori-Frontend-System. Die OData-Anfragen werden dann entweder über die Gateway-Serverkomponente in der Cloud oder über ihre lokale Gateway-Installation an das Anwendungssystem geleitet. Damit kommt der Netzwerklatenz zwischen der Mehrheit der Benutzer, dem Fiori-Frontend-System und dem Anwendungssystem eine entscheidende Bedeutung zu. Wenn diese groß ist, leidet darunter die Performance. Daraus lassen sich folgende Aussagen ableiten:

- Wenn Sie ein lokales Fiori-Frontend-System nutzen: Um die Netzwerklatenz zwischen diesem und dem Anwendungssystem einerseits sowie der Mehrheit der Benutzer andererseits gering zu halten, sollten Sie, wenn Ihre Benutzer und Anwendungssysteme global verteilt sind, mehrere Fiori-Frontend-Systeme einrichten, z. B. eines pro Kontinent.
- Wenn Sie die SAP Fiori Cloud Edition nutzen: Um die Netzwerklatenz zwischen dieser und dem Anwendungssystem einerseits sowie der Mehrheit der Benutzer andererseits gering zu halten, sollten Sie auf den Standort des Rechenzentrums achten, auf dem die SAP Fiori Cloud Edition läuft.

Neben der Netzwerklatenz spielen für die Performance der Durchsatz und die Antwortzeit der beteiligten SAP-Komponenten eine wichtige Rolle. Vergleichende Messungen zwischen einem lokalen SAP-Fiori-Frontend-System und der SAP Fiori Cloud Edition liegen hier allerdings nicht vor.

8.6.3 Allgemeine Performancegesichtspunkte von SAP Fiori, SAPUI5 und OData

Anfragen von Fiori-Anwendungen

Die Anfragen, die eine Fiori-Anwendung an den Webserver richtet, können unter Performancegesichtspunkten wie folgt klassifiziert werden:

1. SAPUI5-Basiskomponenten: Dazu gehören statische Ressourcen wie Bibliotheken mit den UI Controls, das Theming, wiederverwendbare Bilddateien usw.
2. Komponenten zur Anwendung: Diese bilden die eigentliche UI-Anwendung, die im Browser läuft.
3. Metadatenanfrage des OData-Service: Bevor die eigentlichen Daten des OData-Service gelesen werden, liest die Anwendung zunächst die Metadaten zum OData-Service.
4. Datenanfrage des OData-Service: In der eigentlichen Datenanfrage werden die Daten zur Anwendung geladen.

Optimierungen

In Abschnitt 8.1, »SAP-Webanwendungen«, haben wir festgehalten, dass pro Benutzerinteraktion nur wenige nicht im Browser gepufferte Serveranfragen erfolgen sollten und wenn möglich Webdokumente im Browser gepuffert werden sollten, um einem Benutzer auch bei hoher Netzwerklatenz noch gute Antwortzeiten zu bieten. Dazu verwenden Fiori-Anwendungen folgende Optimierungen:

- Reduktion der geladenen Dokumente einer UI-Anwendung: Beim Entwicklungsprozess werden für die verschiedenen Sichten der UI-Anwendung, die zugehörigen Controller, das Modell sowie die Textdateien mit den Texten in den unterstützten Sprachen einzelne Dateien angelegt.

Damit diese nicht in vielen einzelnen Anfragen in den Browser geladen werden müssen, werden sie im sogenannten *Build-Prozess* komprimiert (*minifiziert*) und zu einer einzigen Datei gebündelt. Diese finden Sie im Netzwerk-Trace des Browsers in der Regel unter dem Namen **component-preload.js**. Voraussetzung für diese Optimierung ist, dass Sie eine Komponente (Component) für Ihre Anwendung definieren, was für Fiori-Anwendungen Standard ist. Im Entwicklungssystem können Sie beobachten, dass diese Datei auf dem Webserver nicht gefunden wird und daher anschließend die Dateien der Anwendung einzeln geladen werden, was viele Netzwerkinteraktionen zur Folge hat. In einem Produktivsystem darf dies nicht der Fall sein, dort muss die Anwendung mit einer Preload-Datei geladen werden. Dies gilt analog auch für Bibliotheken, die im Entwicklungsprozess aus mehreren Dateien bestehen, in der Produktivnutzung aber über eine minifizierte Komponentendatei geladen werden.

- SAPUI5-Basiskomponenten, UI-Anwendungskomponenten sowie OData-Metadaten werden im Browserpuffer gespeichert.
- OData-Anfragen können in Massenanfragen (*Batch*) gebündelt werden. Sollten in einer Anwendung pro Nutzerinteraktion mehrere OData-Anfragen nacheinander gesendet werden, so sollte der Entwickler diese auf Massenanfragen umstellen. Bei ändernden Anfragen (HTTP-`POST`, `PUT` oder `MERGE`), die innerhalb einer Transaktion konsistent auf dem Server prozessiert werden sollen, *müssen* die Anfragen aus Integritätsgründen sogar in einer Massenanfrage in einem sogenannten *Change Set* gesendet werden.

Datenabfragen

Wenn Sie produktiv mit der Fiori-Anwendung arbeiten und diese einmal geladen wurde, werden nur noch die Datenabfragen mit dem Server ausgetauscht. Idealerweise wird pro Interaktionsschritt nur eine Datenanfrage mit dem Server ausgetauscht. Ausnahmen von dieser Regel sind:

- nachlaufende asynchrone Anfragen, d. h. Anfragen, die die eigentliche Benutzung der Anwendung nicht blockieren, sondern nur zusätzliche Informationen in den Browser laden
- Übersichtsanwendungen (z. B. sogenannte Dashboards) laden in der Regel viele unterschiedliche Informationen in eine Anwendung, die einzelnen Bestandteile der Anwendung sind aber lose gekoppelt. Eine Bündelung der Anfragen in einer Massenanfrage ist daher in der Regel nicht möglich bzw. gewünscht. In einer solchen Anwendung werden die

Anfragen parallel gesendet, und die Ergebnisse laufen nacheinander im Browserfenster ein.

Tabelle 8.4 stellt typische Performanceprobleme in SAPUI5-Anwendungen, die Sie im Netzwerk-Trace erkennen können, und deren Lösungen dar. In Anhang E, »Informationsquellen«, finden Sie Referenzen zu weiterführendem Material und zur SAPUI5-Dokumentation, die erklärt, wie die Optimierungen in die SAPUI5-Anwendung einzubauen sind.

Problem	Lösung
Beim wiederholten Aufruf einer Anwendung werden SAPUI5-Dateien (Bibliotheken etc.) geladen.	SAPUI5- Bibliotheken sollten gepuffert sein, unter Verwendung des Cache-Busters.
Beim ersten Aufruf (*Bootstrap*) einer Anwendung (oder nach dem Löschen des Browsercaches) werden SAPUI5-Bibliotheken synchron, d. h. eine nach der anderen, geladen.	Stellen Sie die Ladestrategie für SAPUI5-Bibliotheken auf asynchrones, paralleles Laden um (Attribut `data-sap-ui-preload` in der **index.html**-Datei). Wichtig: Beachten Sie dabei, dass die nachfolgenden Schritte der Anwendung erst dann prozessiert werden dürfen, wenn alle Bibliotheken vollständig geladen sind!
Beim ersten Aufruf (*Bootstrap*) beobachten Sie eine hohe Netzwerklatenz für das Laden der SAPUI5-Bibliotheken bei Standorten, die sehr weit von Ihrem Fiori-Frontend-System entfernt sind.	SAP stellt die SAPUI5-Bibliotheken auf dem *Akamai Content Delivery Network* zur Verfügung. Stellen Sie Ihre Anwendung so um, dass die SAPUI5-Bibliotheken von einem Akamai-Server (in der Nähe Ihrer Benutzer) geladen werden.
Beim ersten Aufruf (*Bootstrap*) werden die Dateien der Anwendung (z. B. View- und Controller-Dateien) einzeln geladen.	Führen Sie für Ihre Anwendung eine Komponente (*Component*) ein, integrieren Sie alle Dateien in diese Komponente, und führen Sie einen Build durch, mit dem alle Dateien in einer Preload-Datei verpackt werden. Hinweis: Für eine Fiori-Anwendung ist eine Komponente Pflicht.
Pro Nutzerinteraktion werden viele OData-Anfragen gesendet.	Bündeln Sie die OData-Anfragen in einer OData-Batch-Anfrage.

Tabelle 8.4 Performanceprobleme in SAPUI5-Anwendungen und ihre Lösungen

SAP-Fiori-, SAPUI5-, OData- und SAP-Gateway-Implementierungen auf unterschiedlichen Servern

Wie schon erwähnt, sind SAP Fiori, SAPUI5, OData und Gateway Konzepte, die von SAP auf unterschiedlichen Servern implementiert werden, insbesondere auf dem SAP NetWeaver AS ABAP, der SAP Cloud Platform sowie der SAP HANA XS Engine. Beispielhaft beschreiben wir die aktuellen Details zur Pufferung von SAPUI5- und OData-Metadatendokumenten sowie zu den Gateway-Statistiken im nächsten Abschnitt. In Anhang E, »Informationsquellen«, finden Sie die Verweise auf die Dokumentation der entsprechenden Implementierungen auf der SAP Cloud Platform.

8.6.4 SAP Fiori, SAPUI5 und OData auf dem SAP NetWeaver Application Server ABAP

Während sich die Konzepte von SAPUI5, OData, SAP Fiori und Gateway auf den unterschiedlichen Plattformen gleichen, unterscheiden sich die Implementierungen und damit die Möglichkeiten der Performanceanalyse und -optimierung. In diesem Abschnitt stellen wir die Optionen auf dem SAP NetWeaver AS ABAP vor.

SAPUI5

Wenn Sie eine SAPUI5-Anwendung auf dem SAP NetWeaver AS ABAP installieren, dann wird diese technisch in derselben Ablage gespeichert wie BSP- und ABAP-WebDynpro-Anwendungen (*BSP Repository*). In der ABAP-Entwicklungsumgebung finden Sie SAPUI5-Anwendungen konsequenterweise auch unter der Kategorie *BSP-Anwendungen*. Der zugehörige URL-Pfad lautet */sap/bc/ui5_ui5*.

Eine SAPUI5-Anwendung auf dem SAP NetWeaver AS ABAP starten Sie typischerweise wie folgt:

https://<sapserver>:<port>/sap/bc/ui5_ui5/<namespace>/<anwendung>/index.html

Dabei steht <sapserver> für den Namen eines Anwendungsservers, auf dem das SAP-System läuft, <port> ist der TCP/IP-Port, auf dem der Anwendungsserver hört, und <anwendung> steht für den Namen der SAPUI5-Anwendung.

Eine Anwendung, die über das Fiori Launchpad gestartet wird, verfügt über keine **index.html**-Datei, über die sie direkt gestartet werden könnte. Eine Fiori-Anwendung wird über die **Component.js**-Datei als Teil der SAP-Fiori-Launchpad-Anwendung gestartet. Das SAP Fiori Launchpad starten Sie über die folgende URL:

https://<sapserver>:<port>/sap/bc/ui5_ui5/ui2/ushell/shells/abap/FioriLaunchpad.html

OData

OData-Services auf dem SAP NetWeaver AS ABAP werden über das SAP Gateway implementiert und konfiguriert. Eine Übersicht über die konfigurierten OData-Services finden Sie in der SAP-Gateway-Servicepflege (Transaktionscode /n/IWFND/MAINT_SERVICE). Von dort können Sie auch in die ICF-Servicepflege (Transaktionscode SICF) und in die Testumgebung für OData-Services (*SAP Gateway Client*) navigieren.

OData-Services finden Sie unter der URL */sap/opu/odata*. Das Metadatendokument rufen Sie mit der URL */sap/opu/odata/<servicename>/$metadata*, der Datenabruf zu einer Entität hat die Struktur */sap/opu/odata/<servicename>/<entityname>/<weitere Parameter>*. Weitere Parameter können beispielsweise Filterbedingungen auf die Entitäten sein.

Änderungen in der Pufferimplementierung

In diesem Abschnitt stellen wir die Details der Pufferung von SAPUI5-Elementen und OData-Metadatendokumenten auf dem SAP NetWeaver AS ABAP vor. Die Implementierung von Puffern kann sich von Version zu Version stark ändern, d. h., wenn Sie dieses Buch in der Hand halten, kann sich die Pufferung im Vergleich zum hier Dargestellten schon wieder verbessert haben. Dennoch halten wir es für wichtig, den Status quo zur Drucklegung des Buches darzustellen, um Ihnen die Konzepte zu diesem Zeitpunkt nachzubringen. Auch wenn sich diese ändern werden, können Ihnen die hier dargestellten Konzepte einen Einstieg geben, um die später aktuellen Hinweise und Dokumentationen besser zu verstehen.

Pufferung von SAPUI5-Dokumenten

Bereits in Abschnitt 8.3, »Pufferung von Webdokumenten«, haben wir die Problematik des Pufferns von Webdokumenten im Browserpuffer vorgestellt: Wenn sich ein Dokument am Server ändert, hat dieser aufgrund der Architektur des Internets keine Möglichkeit, den Browser über diese Änderung zu informieren. Der Benutzer arbeitet also mit veralteten Daten weiter, bis er aktiv den Browserpuffer löscht oder das Gültigkeitsdatum überschritten wird. Als nachhaltige Lösung des Problems haben wir das Konzept des Cache-Busters eingeführt.

Im SAP-Fiori-Kontext können die folgenden Dokumente gepuffert und über den Cache-Buster verwaltet werden:

- SAPUI5-Bibliotheken
- SAP-Fiori-Apps und SAP-Fiori-Bibliotheken
- andere SAPUI5-Anwendungsdokumente wie klassische SAPUI5-Anwendungen, die nicht als SAPUI5-Komponenten oder Anwendungsbibliotheken implementiert sind

Cache-Buster für Fiori-Anwendungen aktivieren

Im Detail unterscheiden sich die Implementierungen des Cache-Buster-Konzepts für Fiori-Anwendungen, d. h. für Anwendungen, die über das SAP Fiori Launchpad laufen, und für klassische SAPUI5-Anwendungen. Um den Cache-Buster für Fiori-Anwendungen zu aktivieren, gehen Sie wie folgt vor:

1. Starten Sie die Servicepflege des ICF (Transaktionscode SICF).
2. Aktivieren Sie den folgenden Service: **/sap/bc/ui2/flp**.
3. Planen Sie das Programm **/UI5/APP_INDEX_CALCULATE** als regelmäßigen Hintergrundprozess ein. Dieser Report aktualisiert den SAPUI5-Anwendungsindex, der dem Cache-Buster zugrunde liegt.

Nach dem Aktivieren des Service kann das SAP Fiori Launchpad mit Cache-Buster über eine der folgenden URLs gestartet werden:

- *https://<server>:<port>/sap/bc/ui2/flp/*
- *https://<server>:<port>/sap/bc/ui2/flp/index.html*
- *https://<server>:<port>/sap/bc/ui2/flp/FioriLaunchpad.html*

Über die Launchpad-Konfiguration können Sie diese URL auch unter der URL (Alias) zur Verfügung stellen, die Sie bis dato in Ihrer Organisation verwendet haben.

Wenn Sie wie beschrieben vorgehen, sollte der Cache-Buster wartungsfrei funktionieren. Sollten Sie in einer Ausnahmesituation die UI-Puffer direkt invalidieren müssen, so können Sie dies über das Programm **/UI2/INVALIDATE_CLIENT_CACHES** tun. Mit diesem Programm können Sie die Puffer entweder für einen bestimmten Benutzer oder für alle Benutzer invalidieren.

Cache-Buster für SAPUI5-Anwendungen

Für SAPUI5-Anwendungen, die nicht über das SAP Fiori Launchpad gestartet werden, können Sie in Ihrer Anwendung (**index.html**-Datei) konfigurieren, dass Sie den Cache-Buster für die SAPUI5-Bibliotheken verwenden wollen. Dazu ändern Sie in der Anwendung den Pfad zu den SAPUI5-Ressourcen (`sap-ui-bootstrap`) von `resources/sap-ui-core.js` zu `resources/sap-ui-cachebuster/sap-ui-core.js`.

In Anhang E, »Informationsquellen«, finden Sie Verweise auf die Dokumentation und SAP-Hinweise mit Details zum Cache-Buster-Konzept, zu den genannten Programmen und zur Entwicklerdokumentation.

Pufferung von OData-Metadatendokumenten

Für die Pufferung von OData-Metadatendokumenten implementiert SAP Gateway ein dreistufiges Konzept:

1. Pufferung im Browser
2. Pufferung im Hub-Server
3. Pufferung im Applikationsserver

Die Pufferung im Hub-Server oder Applikationsserver verhindert die Neuberechnung des Metadatendokuments, die Pufferung im Browser verhindert die Neuberechnung und die notwendige Kommunikation zwischen Browser und Server.

Pufferung im Browser

Für die Pufferung im Browser wird die in Abschnitt 8.3.1, »Browserpuffer (Browsercache)«, dargestellte *Pufferung mit bedingter Validierung* verwendet, d. h., Metadatendokumente werden mit `max-age = 0` gepuffert und müssen daher bei Neustart der Anwendung validiert werden.

Pufferung im Hub-Server und im Applikationsserver

Puffer im Hub-Server und im Applikationsserver verhindern, dass das Metadatendokument, wenn es vom Browser angefordert wird, erneut aus dem OData-Modell berechnet wird. Gateway verwendet dazu nicht den ICM-Puffer, sondern implementiert einen eigenen Puffer.

Den Puffer im Gateway-Hub-Server aktivieren Sie wie folgt:

1. Starten Sie die Transaktion /n/IWFND/MED_ACTIVATE.
2. Wählen Sie **Aktivieren**. Seit NetWeaver Version 7.50 existiert auch die Option, für den Puffer den Shared Memory zu verwenden. Aktivieren Sie auch diese Option. Der Puffer im Gateway-Hub-Server ist nun aktiv.

Der Puffer im Gateway-Hub-Server sollte wartungsfrei funktionieren, d. h., bei Änderungen des OData-Modells wird dieser automatisch beim nächsten Zugriff invalidiert. Sollten Sie in einer Ausnahmesituation den Puffer direkt invalidieren müssen, so können Sie dies über die Transaktionen /n/IWFND/CACHE_CLEANUP (Löschen des Gateway-Puffers auf dem Hub) bzw. /n/IWBEP/CACHE_CLEANUP (Löschen des Gateway-Puffers auf dem Anwendungsserver) tun. Weitere Informationen zu diesen Transaktionen finden Sie in der SAP-Hilfe.

Gateway-Performanceverbesserungen mit SAP NetWeaver 7.50

Mit SAP NetWeaver 7.50 SP04 bietet SAP Gateway folgende Performanceverbesserungen:

- Der Shared Memory kann für den Metadatenpuffer genutzt werden (neue Option in Transaktion /n/IWFND/MED_ACTIVATE).
- Im Hub-Szenario können Sie den sogenannten Micro-Hub-Modus einstellen. Dies hat zur Konsequenz, dass der Hub Anfragen nach der Validierung und Berechtigungsprüfung direkt an das Anwendungssystem weitergibt und dass die gesamte Bearbeitung dort durchgeführt wird. Die Funktionsweise des Hubs als einheitlicher Eingangskanal und für das Verteilen der Anfragen ist von dieser Umstellung nicht betroffen. Durch die komplette Bearbeitung im Anwendungssystem können einige Optimierungen durchgeführt werden, die bei der verteilten Bearbeitung nicht möglich sind. In diesem Betriebsmodus kann beispielsweise die Serialisierung der Antwort in den meisten Fällen über eine *ABAP Simple Transformation* im ABAP-Kernel ausgeführt werden. Diese Berechnung im Kernel ist deutlich schneller als die Serialisierung im ABAP-Programmcode. Die dazu notwendige ABAP Simple Transformation wird automatisch beim ersten Aufruf erzeugt. Einen Verweis auf die Dokumentation zu diesem Szenario finden Sie in Anhang E, »Informationsquellen«.

SAP-Gateway-Statistiken

Dieser Abschnitt ist der Monitoring-Transaktion auf dem SAP Gateway im SAP NetWeaver AS ABAP gewidmet.

Gateway-Statistiken

SAP Gateway speichert für jede erfolgreiche OData-Anfrage statistische Daten in der Datenbank. Diese Statistiken können über die Transaktion /n/IWFND/STATS ausgewertet werden. Die dort angezeigten Messwerte sind identisch mit den Werten, die als `sap-statistics` auf Anfrage an den Browser zurückgegeben werden (siehe Abschnitt 8.2.2, »SAP-Statistiken in der HTTP-Anfrage«).

Um die Performance von OData-Anfragen zu analysieren, rufen Sie die Transaktion /n/IWFND/STATS auf. Der Bildschirm zeigt die Liste der OData-Anfragen der letzten Stunde mit Messwerten zur Performance. Abbildung 8.7 zeigt den Bildschirmabzug der SAP-Gateway-Statistik.

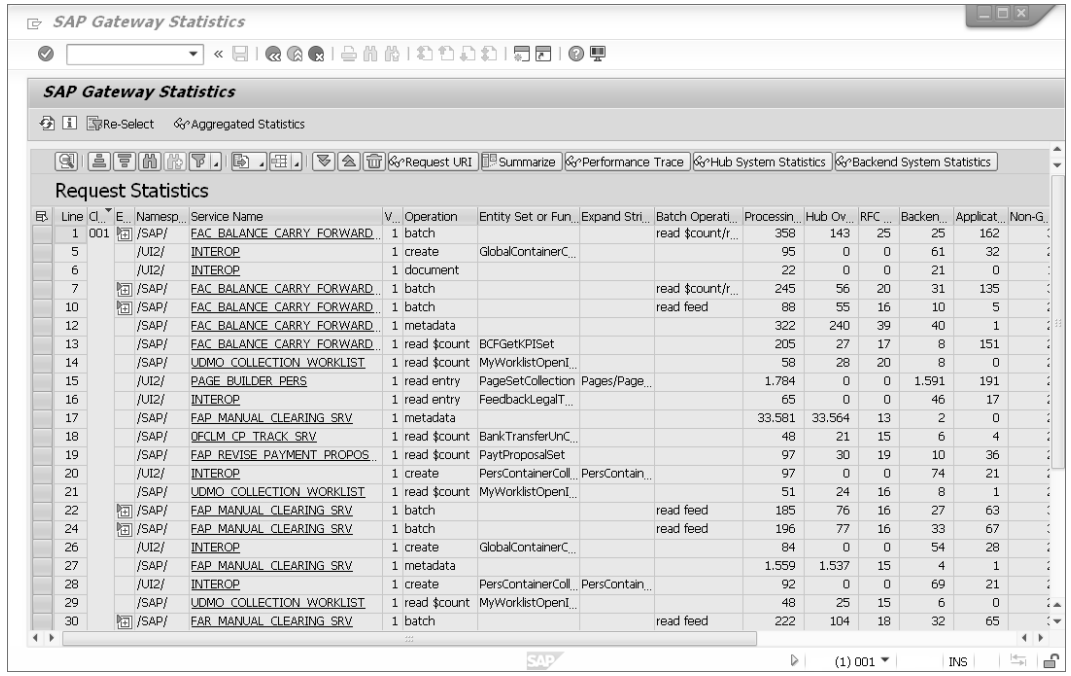

Abbildung 8.7 SAP-Gateway-Statistik

Tabelle 8.5 beschreibt die wichtigsten Felder. In Klammern finden Sie den Namen, unter dem der Messwert in der SAP-Statistik im HTTP-Kopf zu finden ist (siehe Abschnitt 8.2.2, »SAP-Statistiken in der HTTP-Anfrage«).

Feld	Bedeutung
Namespace	Namensraum des Service
Service Name	Name des Service
Version	Version des Service
Operation	ausgeführte Aktion, z. B. *Read, Create, Metadata* (Lesen der Metadaten), *Batch* (im Falle einer Batch-Anfrage finden Sie Details zu den einzelnen Anfragen im Batch im Feld **Batch Operation**)
Entity Set or Function	ausgeführte Entity oder Funktion
Processing Time in ms (`gwtotal`)	gesamte Antwortzeit im SAP Gateway inklusive der Anwendung

Tabelle 8.5 Felder der Gateway-OData-Statistik

Feld	Bedeutung
Hub Overhead in ms (gwhub)	Antwortzeit im SAP-Gateway-Hub-System
RFC Overhead in ms (gwrfcoh)	RFC- und Netzwerk-Anteil für die Kommunikation zwischen Hub und Anwendungssystem
Backend Overhead in ms (gwbe)	Zeit in der SAP-Gateway-Systemkomponente im Anwendungssystem (ohne Zeitanteil der Anwendung)
Application Time in ms (gwapp)	Zeit, die in der Anwendung benötigt wurde
Request Size in Bytes	Größe der Anfrage in Byte
Response Size in Bytes	Größe der Antwort in Byte

Tabelle 8.5 Felder der Gateway-OData-Statistik (Forts.)

Kommen wir an dieser Stelle noch mal auf das Beispiel des SAP-Statistikdatensatzes aus Abschnitt 8.2.2 zurück:

```
sap-statistics:
gwtotal=150,gwrfcoh=6,gwapp=65,gwhub=14,gwbe=62,icmtotal=
184,icmreqrcv=1,icmext=179,icmssl=5,wdtotal=194,wdreqrcv=3,wdext=
185,wdssl=7
```

Die Antwortzeit des Applikationsservers lässt sich in zwei Teile zerlegen:

- Den technischen Überbau: Der Zeitanteil für die Bearbeitung der HTTP-Anfrage bis zum Aufruf des betriebswirtschaftlichen Codings und vom Abschluss des betriebswirtschaftlichen Codings bis zur HTTP-Antwort (inklusive aller Services für die Validierung, De-Serialisierung/Serialisierung des Webdokuments, initiale Berechtigungsprüfung usw.) ergibt sich aus der Differenz zwischen wdtotal und gwapp. In dem oben dargestellten Beispiel sind dies *194 – 65 = 129 ms*.
- Den Zeitanteil in der eigentlichen Anwendung: gwapp. In unserem Beispiel sind dies 65 ms.

Weitere Funktionen

Die SAP-Gateway-Statistiken verfügen über folgende weitere Funktionen:

- Über die Schaltfläche **Re-Select** ändern Sie den Filter für die ausgewählten Statistiken, d. h. den Zeitraum, Nutzer, Servicenamen usw.
- Mit der Funktion **Summarize** können Sie die Statistiken der ausgewählten OData-Anfragen aggregieren. Der Bildschirm zeigt Ihnen dann die Mittelwerte bzw. Mediane der oben angegebenen Messwerte an.

- Über die Schaltfläche **Performance Trace** können Sie in den Gateway-Trace (Transaktionscode /n/IWFND/TRACES) springen, sofern zu der entsprechenden Anfrage ein Trace aufgezeichnet wurde.
- Über die Schaltflächen **Hub System Statistics** und **Backend System Statistics** erreichen Sie die statistischen Sätze (Transaktion STAD) im Hub- bzw. im Anwendungssystem.

Fiori-Anwendung tracen

Wenn Sie einen Trace für eine Fiori-Anwendung einschalten, dann werden unter Umständen mehrere Anfragen parallel an den Server geschickt (siehe Abbildung 8.1), und der aufgezeichnete Trace wird unübersichtlich und schwer zu analysieren, weil der Fluss verloren geht. Um dies zu vermeiden, gehen Sie wie folgt vor:

1. Schalten Sie beim Ausführen der Fiori-Anwendung zunächst nur den Gateway-Payload-Trace ein (Transaktionscode /n/IWFND/TRACES). Dieser Trace zeichnet die OData-Anfragen mit den Anfrage-Parametern (Payload) auf. Mit der Replay-Funktion können Sie die Anfragen anschließend erneut starten.
2. Schalten Sie nach dem Ausführen der Fiori-Anwendung die gewünschten Traces ein (Performance-Trace, ABAP-Laufzeitanalyse oder für SAP HANA den Expensive Statement Trace oder den Plan Trace).
3. Selektieren Sie im Gateway-Payload-Trace die Anfragen mit einer hohen Laufzeit, und starten Sie die kritischen Anfragen erneut mit der Funktion **Wiedergeben**. Indem Sie die kritischen Anfragen hintereinander ausführen, können Sie die Traces einzeln aufzeichnen. Achten Sie dabei darauf, dass Sie keine Inkonsistenzen erzeugen, wenn es sich um eine ändernde Anfrage handelt (POST, PUT etc.)

8.7 Zusammenfassung

Es gibt grundsätzlich zwei »todsichere« Methoden, ein Performanceproblem im Bereich der Frontend-Kommunikation zu erzeugen. Die erste ist, große Datenmengen an den Frontend-Client zu übertragen. Die zweite Methode ist, eine große Anzahl von Roundtrips (Kommunikationsschritten zwischen Frontend-Client und Server) zu programmieren.

SAP GUI vs. Web UI

Anwendungen auf Basis des SAP GUI reagieren bei einer nicht optimalen Programmierung oft noch halbwegs fehlertolerant, da das Protokoll zwischen SAP GUI und SAP NetWeaver AS das SAP-eigene DIAG-Protokoll ist, das auf eine minimale Datenübertragung getrimmt ist. Bei Webanwendun-

gen, die HTTP verwenden, werden dagegen in der Regel selbst bei optimaler Programmierung schon deutlich mehr Daten als bei SAP-GUI-Anwendungen übertragen. Kommt hier noch eine nicht optimale Programmierung hinzu, ist ein Performanceproblem unvermeidbar.

Eine nicht optimale Programmierung kann dazu führen, dass eine Anwendung zwar im LAN (Local Area Network), also etwa am Arbeitsplatz des Entwicklers, noch zufriedenstellend arbeitet, im WAN (Wide Area Network), etwa am Arbeitsplatz der Vertriebsmitarbeiterin in einer Außenstelle, aber zu katastrophalen Performanceproblemen führt.

Webanbindung

Eine Reihe von Punkten ist für die Performance der Webanbindung von Bedeutung. Zunächst ist die Auswahl des richtigen GUI zu nennen: Für manche Benutzergruppen ist das SAP GUI for HTML nicht die goldene Wahl. Es kann sinnvoll sein, auch weiterhin das SAP GUI for Windows (oder Java) zu verwenden.

Pufferung von Webdokumenten

Die Pufferung von Webdokumenten im Browserpuffer, auf dem SAP Web Dispatcher, im ICM sowie optional auch in einem Content Delivery Network ist eine wichtige Maßnahme, um eine gute Performance zu erreichen, wenn Ihre Benutzer weltweit über das Internet auf die SAP-Webanwendungen zugreifen.

Analysemethoden auf dem Präsentationsserver

Die in Abschnitt 8.2, »Analysen auf dem Präsentationsserver«, beschriebenen Methoden zur Performanceanalyse von Webanwendungen sind generisch und können nicht nur für Webanwendungen verwendet werden, die durch den SAP NetWeaver AS generiert werden, sondern auch für solche, die von anderen Servern generiert werden.

Analysemethoden auf dem SAP NetWeaver AS ABAP

Zur Analyse von Webanwendungen, die auf dem SAP NetWeaver AS ABAP betrieben werden – dazu gehören Anwendungen auf Basis des SAP GUI for HTML, Web Dynpro ABAP sowie Fiori-Anwendungen –, wurden die bekannten Werkzeuge wie der Workload-Monitor oder der ABAP- und der Performance-Trace entsprechend erweitert, sodass Sie die Performance lückenlos von der eingehenden HTTP-Anfrage bis in das betriebswirtschaftliche Coding analysieren und optimieren können. Hinzu kommt der Gateway-Performance-Monitor, mit dem Sie Performanceprobleme innerhalb von SAP Gateway aufspüren können.

Wichtige Begriffe aus diesem Kapitel

Mit den folgenden Begriffen sollten Sie nach der Lektüre dieses Kapitels vertraut sein:

- Auswahl des »richtigen« GUI: SAP GUI for Windows, SAP GUI for HTML, SAP GUI for Java Environment

- Internet Communication Manager (ICM) und SAP Web Dispatcher
- Performanceanalyse mit den Entwicklerwerkzeugen der Internetbrowser
- Pufferung von Webdokumenten
- SAP-Statistiken in der HTTP-Anfrage
- integrierter Internet Transaction Server (ITS), Business Server Pages (BSP) und Web Dynpro ABAP
- SAP Fiori, SAPUI5/openUI5, OData, SAP Gateway

8.8 Übungsfragen

Die Antworten finden Sie in Abschnitt C.7.

1. Welche der folgenden Aussagen zur Verwendung des Browserpuffers ist korrekt?
   a) Für Webanwendungen mit unternehmenskritischen Daten muss der Browserpuffer aus Sicherheitsgründen deaktiviert werden, nur für öffentliche Webseiten kann der Browserpuffer genutzt werden.
   b) Es muss zwischen pufferbaren und nicht pufferbaren Webdokumenten unterschieden werden. Benutzer müssen dafür im Browser konfigurieren, welche URL-Pfade gepuffert werden dürfen und welche nicht.
   c) Es muss zwischen pufferbaren und nicht pufferbaren Webdokumenten unterschieden werden. Die Serveranwendung legt über Parameter in der HTTP-Anfrage fest, welche Dokumente gepuffert werden dürfen und welche nicht.
   d) Grundsätzlich können alle Webdokumente gepuffert werden, d. h. prinzipiell auch Datenanfragen. Bei Änderungen der Daten informiert der Server über das sogenannte Cache-Buster-Konzept via Webchannels den Browser über die Änderung.
2. Eine Fiori-Anwendung, die unter Verwendung von SAPUI5 und OData-Technologie auf dem SAP NetWeaver AS ABAP realisiert wurde, läuft »zu langsam«. Welche Analysen nehmen Sie vor?
   a) Im Monitor des Internet Communication Managers (ICM) prüfen Sie, ob alle ICM-Threads belegt sind oder ob die CPU ständig belegt ist.
   b) In der Workprozess-Übersicht des angeschlossenen SAP-Systems prüfen Sie, ob alle Workprozesse belegt sind.
   c) Mit einem Performance-Trace und in der Einzelsatzstatistik auf dem angeschlossenen SAP-System analysieren Sie die Antwortzeit auf

dem SAP-System und vergleichen diese mit der vom Benutzer gemessenen Antwortzeit auf dem Präsentationsserver.

d) Mit der Netzwerksicht in den Entwicklerwerkzeugen Ihres Internetbrowsers überprüfen Sie die zum Browser übertragene Datenmenge und die Kompilierungszeit für die HTML-Seite im Browser und vergleichen die dafür benötigten Zeiten mit der Gesamtantwortzeit.

e) Mit dem SAP-Gateway-Monitor überprüfen Sie die Antwortzeiten von Gateway-Frontend-System, Gateway-Backend-System und RFC.

C.6 Fragen zu Kapitel 7, »Lastverteilung, Remote Function Calls und SAP GUI«

1. Wo sollen Hintergrund-Workprozesse konfiguriert werden?

   Antwort: c

2. Wie konfigurieren und überwachen Sie die dynamische Benutzerverteilung?

   Antwort: c

3. Was bedeutet eine hohe Roll-Wartezeit?

   Antwort: c

4. In einem Transaktionsschritt wird eine Transaktion prozessiert, die Controls verwendet, es wird aber weder ein externer RFC noch ein HTTP-Ziel gerufen. Welche Aussagen treffen zu?

   Antwort: a, d

5. In einem Transaktionsschritt wird eine Transaktion prozessiert, die keine Controls und keine synchronen RFCs verwendet, es werden aber asynchrone RFCs gerufen. Welche Aussagen treffen zu?

   Antwort: b, e

C.7 Fragen zu Kapitel 8, »Internetanbindung und SAP Fiori«

1. Welche der folgenden Aussagen zur Verwendung des Browserpuffers ist korrekt?

   Antwort: c

2. Eine Fiori-Anwendung, die unter Verwendung von SAPUI5 und OData-Technologie auf dem SAP NetWeaver AS ABAP realisiert wurde, läuft »zu langsam«. Welche Analysen nehmen Sie vor?

   Antwort: a, b, c, d, e

C.8 Fragen zu Kapitel 9, »Optimierung von Java-Programmen«

1. Sie beobachten häufig vollständige Garbage-Collection-Läufe auf Ihrem SAP NetWeaver AS Java – welche Schlüsse können Sie ziehen?

   Antwort: b

2. Sie haben Ihre JVM-Instanz mit einem Java-Heap von maximal 4 GB gestartet (`-Xmx4GB`). Nach einer Allocation Analysis stellen Sie in der