




Leseprobe

Machen Sie Ihre SAP-Systeme schneller! In der neuen Auflage dieses Bestsellers finden Sie geballt das neueste Performance-Wissen. Diese Leseprobe zeigt Ihnen, wie Sie die Leistung von Webtechnologien der SAP optimieren, also Fiori-UIs und andere Web-UIs oder Webservices.

 **»Internetanbindung und SAP Fiori«**
»Lösungen zu den Fragen«
»Vorwort«
»Einleitung«

 **Inhaltsverzeichnis**

 **Index**

 **Der Autor**

 **Leseprobe weiterempfehlen**

Thomas Schneider

SAP-Performanceoptimierung

Analyse und Tuning von SAP-Systemen

867 Seiten, gebunden, 8. Auflage, Juli 2017

79,90 Euro, ISBN 978-3-8362-4479-4

 www.sap-press.de/4330

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.

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.

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.

Webservices

Einsatz von Web UI und SAP GUI planen

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.

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 kabelgebundenes Netzwerk, kabelloses Netzwerk (WLAN) oder Mobilfunk
- den *mittleren Meilen* – vom Webserver der SAP-Anwendung bis zum Endpunkt des Internet- oder Mobilfunkproviders

Typische Latenzzeiten für die letzte Meile sind:

- < 10 ms im LAN
- > 20 ms für ein kabelgebundenes 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.

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)

Latenzzeiten
im WAN

Latenzzeiten der
letzte Meile

Latenzzeiten der
mittleren Meilen

- Für UIs mit SAP GUI Controls; $2 \times$ 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.

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.

SAP-BW-Frontend-Technologien

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 Präsentationsserver. 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:

- 1 Ein Benutzer tritt in Interaktion mit dem Browser.
- 2 Der Browser stellt fest, dass eine Anfrage an den Webserver gesendet werden muss.

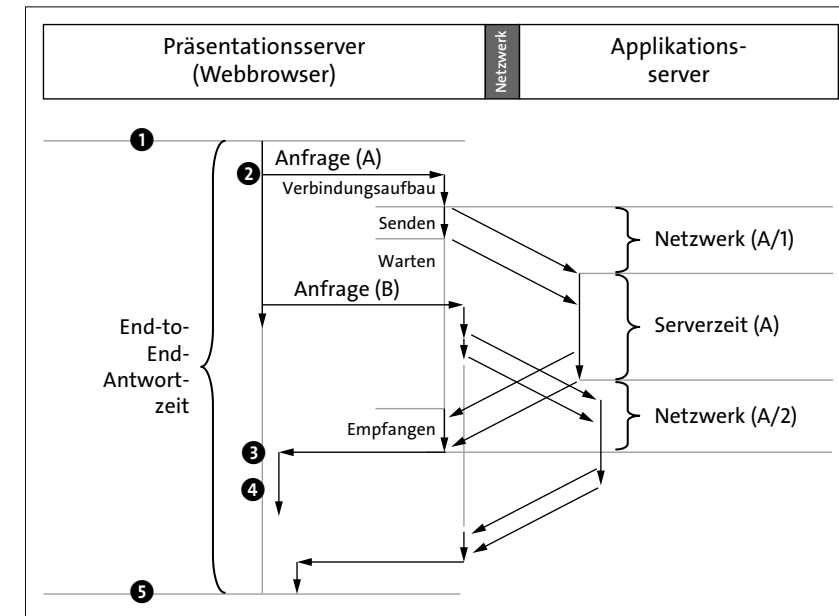


Abbildung 8.1 Transaktionsschritt einer Webanwendung zwischen Präsentationsserver (Webbrowser) und Anwendungsserver (Webserver)

Die Anfrage untergliedert sich in die folgenden Zeitanteile:

- *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.

Zeitanteile einer Webanfrage

– *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).

- ③ 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.
- ④ Schließlich ist die Aufbereitung der Anfrage (*Rendering*) abgeschlossen.
- ⑤ 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-Webrowsers 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.

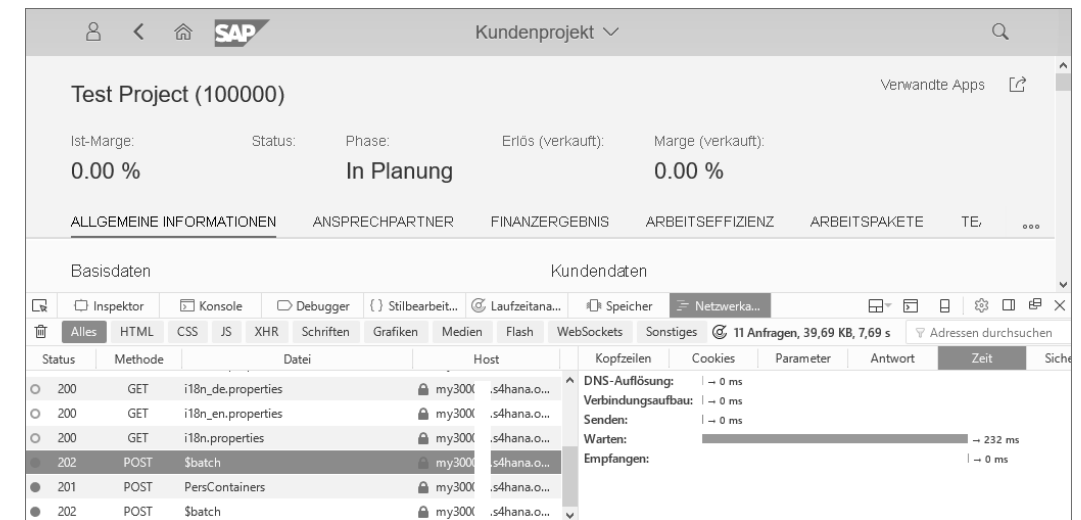


Abbildung 8.2 Entwicklerwerkzeuge im Firefox-Browser

Die Netzwerksicht im Browser zeigt den Zeitverlauf der Serveranfragen. In der rechten oberen Ecke finden Sie neben dem Uhrensymbol 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

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

Browserpuffer bei Performancemessungen aktivieren

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] + [U] + [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**.

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

SAP-Statistiksatz

- 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:
gwttotal=150,gwrfcoh=6,gwapp=65,gwhub=14,gwbe=62,icmtotal=184,
icmreqrcv=1,icmext=179,icmssl=5,wdttotal=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:

- wdttotal, 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. wdttotal=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.

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.)

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.

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.

End User Experience
Monitoring

Reaktive
Überwachung

Werkzeuge von
Drittanbietern

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, Java-Script-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 setzen, 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	<code>cache-control : no-cache</code>	Der Server darf die Antwort nicht aus dem Puffer beantworten.
	<code>cache-control : max-age = <Wert>, Wert in Sekunden</code>	Der Server muss ein Dokument schicken, das aktueller ist als der spezifizierte Wert. <code>max-age=0</code> bedeutet also, dass der Server immer ein aktuelles Dokument schicken muss.
	<code>if-modified-Since : <Datum></code>	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 (<i>Conditional Request</i>).
Antwort	<code>cache-control : no-cache</code>	Der Browser darf die Anfrage nicht puffern.
	<code>cache-control : max-age = <Wert>, Wert in Sekunden</code>	Der Browser darf das Dokument aus dem Puffer laden, bis das Verfallsalter <code>max-age</code> erreicht ist. Der Eintrag <code>max-age = 315360000</code> bedeutet, dass ein Dokument für ein Jahr gepuffert werden darf. <code>max-age = 0</code> bedeutet, dass das Dokument gepuffert werden darf, dass der Browser aber beim Server nachfragen muss, ob das Dokument noch aktuell ist (<i>Conditional Request</i>).
	<code>last-modified : <Datum></code>	Angabe, wann das Dokument auf dem Server zuletzt geändert wurde

Tabelle 8.1 HTTP-Parameter zur Steuerung des Browserpuffers

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

Puffer löschen

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 vom 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.

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.

Pufferinvalidierung im ICF

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, gepufferte 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

gepufferte 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.

[zB]

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 Anwendungstypen 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
Performance-
analyse

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 STO3) 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.

Performance-
statistiken
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 STO3) 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.

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:

Detailanalyse

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.

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

Spezielle
Performance-
analyse

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 <i>Execution Time</i>
Execution Time	Ausführungszeit der HTTP-Anfrage im ABAP
Sent Data	gesendete Datenmenge
Received Data	empfangene Datenmenge

Tabelle 8.3 Felder des HTTP-Profiles 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

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.

Aufzeichnung in Einzelsatzstatistik

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 **1** und in der ABAP-Coding **2** 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.

Darüber hinaus umfasst eine BSP-Anwendung sogenannte *Events*. Dies sind in ABAP geschriebene Programmteile, die zu bestimmten Zeitpunk-

ten, 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.

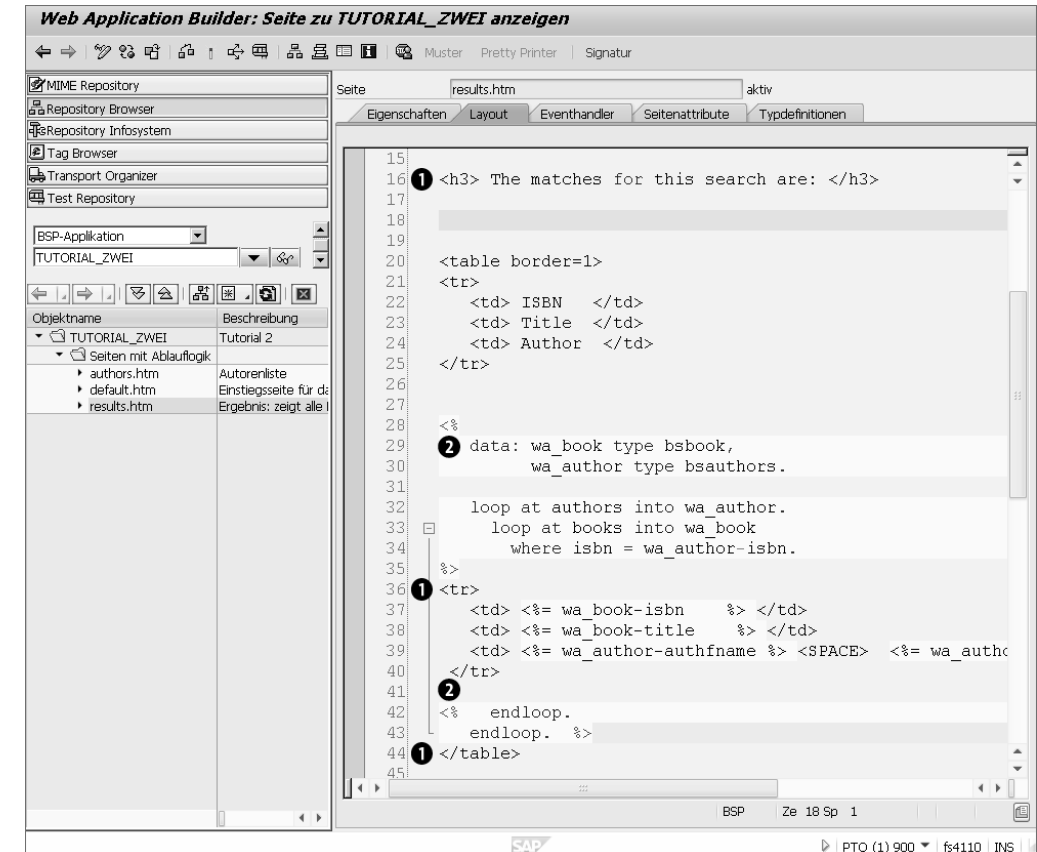


Abbildung 8.4 Zur HTML-Seite »Your Book Search Results« gehöriger Programmtext in HTML und ABAP

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:

BSP-Entwicklungsumgebung aufrufen

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**.

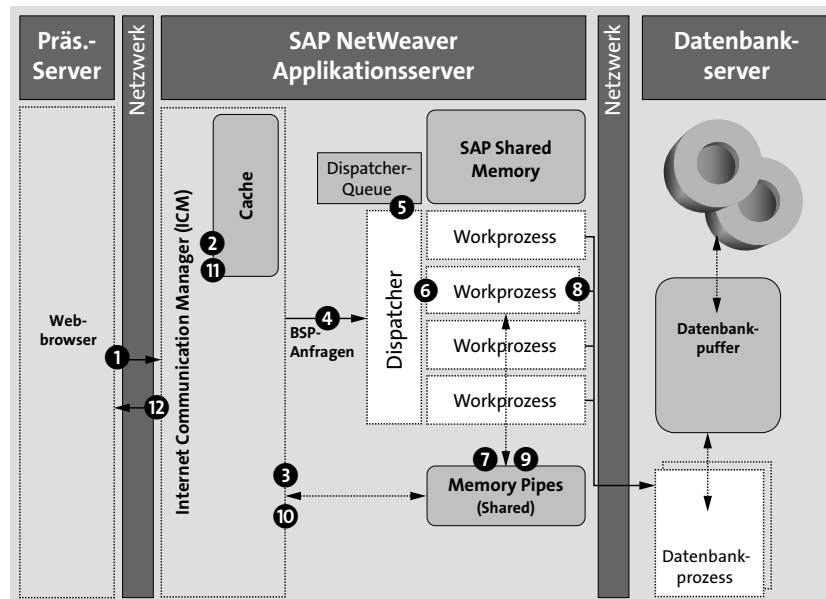


Abbildung 8.5 Ablauf eines Transaktionsschrittes im SAP NetWeaver AS beim Aufruf einer BSP-Seite

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 ①. Dieser prüft, ob die Browseranfrage mithilfe der in seinem Cache gespeicherten Informationen beantwortet werden kann ②. Ist dies nicht der Fall, übergibt er die Anfrage an den SAP-Dispatcher ④, nachdem er die Daten der Anfrage in sogenannte *Memory Pipes* gespeichert hat ③. 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 ⑤, übergibt der Dispatcher die Anfrage einem Workprozess ⑥. Dieser liest die benötigten Daten aus den Memory Pipes ⑦ und bearbeitet die Anfrage ⑧. Nachdem der Workprozess die BSP bearbeitet und die Webseite erstellt hat, stellt er die Daten in die Memory Pipe ⑨ und übergibt die Kontrolle zurück an den ICM ⑩. Dieser sendet die fertige HTML-Seite an den Webbrowser ⑪–⑫.

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

Performanceanalyse

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/!](https://<sapserver>:<port>/sap/bc/gui/sap/its/bwsp/)

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

Architekturänderungen

- 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.

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-

Hauptspeicher-verbrauch

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.

Performance-analyse 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 Webserver 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.

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.

Open Data (OData) Protocol

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

SAP Fiori

- 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 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.

SAP Gateway

Komponenten der Fiori-Systemlandschaft

8.6.2 Aufbau einer 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 Plattform betrieben. Für die SAP-Gateway-Serverkomponente gibt es in diesem Szenario zwei Optionen. Diese kann
 - ebenfalls auf der SAP Cloud Plattform 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).

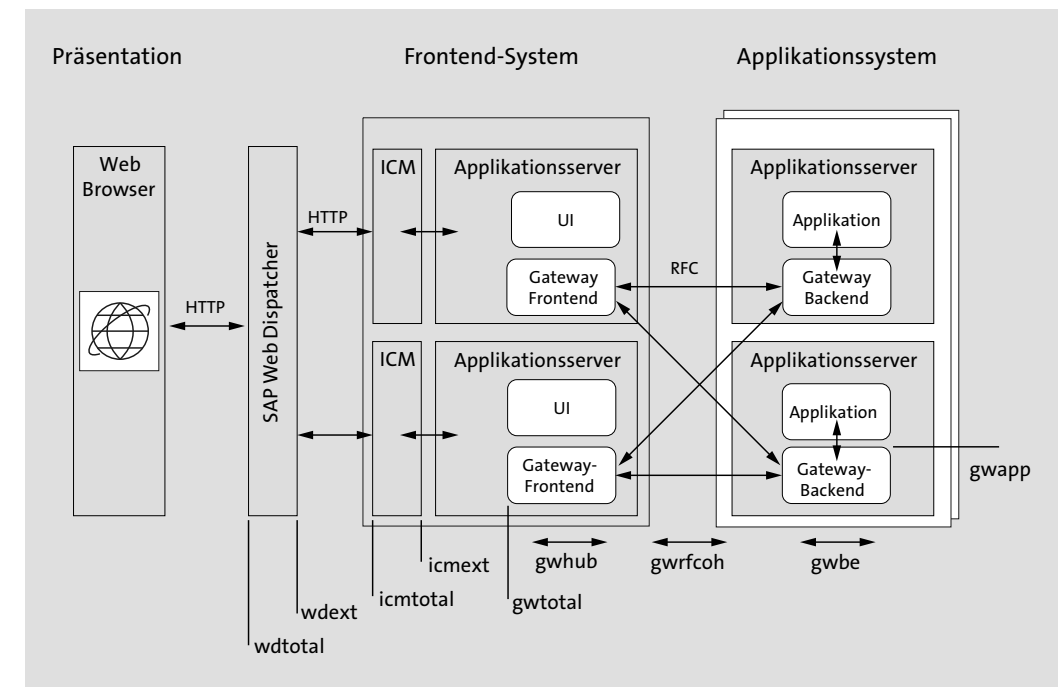


Abbildung 8.6 SAP Fiori-Systemlandschaft (Hub-Szenario)

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 abonnieren 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

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

Anfragen von Fiori-Anwendungen

1. SAPUI5-Basiskomponenten: Dazu gehören statische Ressourcen wie Bibliotheken mit den UI Controls, das Theming, wiederverwendbare Bild-dateien 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.

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:

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 (<i>Bootstrap</i>) 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 <code>data-sap-ui-preload</code> in der <code>index.html</code> -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 (<i>Bootstrap</i>) 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 <i>Akamai Content Delivery Network</i> 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 (<i>Bootstrap</i>) werden die Dateien der Anwendung (z. B. View- und Controller-Dateien) einzeln geladen.	Führen Sie für Ihre Anwendung eine Komponente (<i>Component</i>) 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/Fiori-Launchpad.html
```

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 Metadaten-dokument 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:

OData



- 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.

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 Browser

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.

Pufferung im Hub-Server und im Applikationsserver

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.

Line	Cl	E	Namesp...	Service Name	V	Operation	Entity Set or Fun...	Expand Stri...	Batch Operati...	Processin...	Hub Ov...	RFC	Backen...	Applicat...	Non-G...
1	001		/SAP/	FAC_BALANCE_CARRY_FORWARD	1	batch			read \$count/r...	358	143	25	25	162	
5			/UI2/	INTEROP	1	create	GlobalContainerC...			95	0	0	61	32	
6			/UI2/	INTEROP	1	document				22	0	0	21	0	
7			/SAP/	FAC_BALANCE_CARRY_FORWARD	1	batch			read \$count/r...	245	56	20	31	135	
10			/SAP/	FAC_BALANCE_CARRY_FORWARD	1	batch			read feed	88	55	16	10	5	
12			/SAP/	FAC_BALANCE_CARRY_FORWARD	1	metadata				322	240	39	40	1	
13			/SAP/	FAC_BALANCE_CARRY_FORWARD	1	read \$count	BCFGetKPISet			205	27	17	8	151	
14			/SAP/	UDMO_COLLECTION_WORKLIST	1	read \$count	MyWorklistOpen...			58	28	20	8	0	
15			/UI2/	PAGE_BUILDER_PERS	1	read entry	PageSetCollection	Pages/Page...		1.784	0	0	1.591	191	
16			/UI2/	INTEROP	1	read entry	FeedbackLegalIT...			65	0	0	46	17	
17			/SAP/	FAP_MANUAL_CLEARING_SRV	1	metadata				33.581	33.564	13	2	0	
18			/SAP/	DECLM_CP_TRACK_SRV	1	read \$count	BankTransferUnc...			48	21	15	6	4	
19			/SAP/	FAP_REVERSE_PAYMENT_PROPOS...	1	read \$count	PaytProposalSet			97	30	19	10	36	
20			/UI2/	INTEROP	1	create	PersContainerColl...	PersContain...		97	0	0	74	21	
21			/SAP/	UDMO_COLLECTION_WORKLIST	1	read \$count	MyWorklistOpen...			51	24	16	8	1	
22			/SAP/	FAP_MANUAL_CLEARING_SRV	1	batch			read feed	185	76	16	27	63	
24			/SAP/	FAP_MANUAL_CLEARING_SRV	1	batch			read feed	196	77	16	33	67	
26			/UI2/	INTEROP	1	create	GlobalContainerC...			84	0	0	54	28	
27			/SAP/	FAP_MANUAL_CLEARING_SRV	1	metadata				1.559	1.537	15	4	1	
28			/UI2/	INTEROP	1	create	PersContainerColl...	PersContain...		92	0	0	69	21	
29			/SAP/	UDMO_COLLECTION_WORKLIST	1	read \$count	MyWorklistOpen...			48	25	15	6	0	
30			/SAP/	FAP_MANUAL_CLEARING_SRV	1	batch			read feed	222	104	18	32	65	

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. <i>Read</i> , <i>Create</i> , <i>Meta-data</i> (Lesen der Metadaten), <i>Batch</i> (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 (gwttotal)	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 (gwrfcch)	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,gwrfcch=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.

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

Vorwort und Danksagung

Der große Erfolg der ersten Auflagen dieses Buches – man kann sicher sagen, dass es nicht nur bei den Kunden, sondern auch bei vielen Mitarbeitern der SAP ein Eckpfeiler in der Performanceausbildung geworden ist – bringt die Verpflichtung mit sich, eine aktualisierte und erweiterte Fassung vorzulegen. Dieser Verpflichtung komme ich gerne nach. Die Auswahl und die Darstellung der Themen sind im Wesentlichen geprägt von den Erfahrungen, die meine Kolleginnen, Kollegen und ich im konkreten Umgang mit vielen SAP-Systemen im produktiven Einsatz gemacht haben – sei es in den SAP-Services EarlyWatch und GoingLive Check, in Trainings zur Performanceanalyse und nicht zuletzt bei der Vor-Ort-Analyse von Systemen in Phasen mit kritischen Performanceproblemen. Aufgrund dieser Erfahrungen sind wir zuversichtlich, mit diesem Buch eine breite Palette wichtiger performancerelevanter Themen abzudecken.

In der nun vorliegenden achten Auflage haben wir aktuellen Entwicklungen Rechnung getragen und folgende Änderungen vorgenommen:

- Der Umfang zu den Themen rund um SAP HANA hat sich verdoppelt – dies trägt sowohl der strategischen Bedeutung dieses Themas als auch der gewachsenen Themenfülle Rechnung.
- Das Kapitel zu Web-UIs und Webservices haben wir komplett neu geschrieben, insbesondere werden jetzt die Themen SAP Fiori, SAPUI5, OData und SAP Gateway abgedeckt.
- Alle weiteren Kapitel sind gründlich durchgesehen und überarbeitet worden.

Dieses Buch konnte nur aufgrund der Mitwirkung vieler kompetenter Diskussionspartner entstehen. Es ist mir wichtig, zunächst unseren Kollegen und Mentor Augustinus Wohlfahrt hervorzuheben, dessen plötzlicher und unerwarteter Tod uns tief erschüttert hat. Er hatte als einer der Initiatoren dieser Buchreihe maßgeblichen Anteil daran, dass diese Veröffentlichung entstehen konnte. Ihm möchte ich dieses Buch widmen.

Mein ganz herzlicher Dank gilt Michael Wintergerst und Matthias Braun, die für die siebte Auflage wesentliche Teile des Java-Kapitels komplett neu geschrieben haben, sowie Mijta Sailer, der die Java-Abschnitte zu dieser achten Auflage kritisch durchgesehen hat.

Außerdem möchte ich mich namentlich bei folgenden Kolleginnen und Kollegen bedanken: Christian Bartels, Stefan Biedenstein, Achim Braemer, Ingo Bräuninger, Guido Derwand, Thomas Gauweiler, Heiko Gerwens,

Bernhard Glaser, Christiane Hienger, Manfred Hirsch, Karlheinz Kistner, Alexander Kosolapov, Tilman Model-Bosch, Jens Otto, Jörg Pfänder, Christopher Schmitz, Gerold Völker, Welf Walter und Torsten Ziegler – und darüber hinaus bei all den Kolleginnen und Kollegen, die mir über Jahre hinweg immer wieder Anregungen geliefert und Korrektur gelesen haben.

Kerstin Billen, Janina Karrasch und Florian Zimniak von SAP PRESS danke ich sehr für die gute und fruchtbare Zusammenarbeit.

Dr. Thomas Schneider

Einleitung

Warum ist die Performance Ihrer betriebswirtschaftlichen IT-Anwendung wichtig? Nur bei guten Antwortzeiten können Benutzer motiviert und effizient mit der Anwendung arbeiten. Ein langsames System führt zu Ausfallzeiten und Frustration. Eskaliert die Situation, wird im schlimmsten Fall der zur Bewältigung der Geschäftsprozesse erforderliche Durchsatz nicht mehr erreicht. Mehrarbeit, Produktionsverzögerungen und finanzielle Verluste sind die Folge. Umgekehrt steigert eine systematische, proaktive Performanceoptimierung maßgeblich den Nutzen Ihrer betriebswirtschaftlichen Anwendung.

Die Performance eines Datenverarbeitungssystems ist als die Fähigkeit definiert, gegebene Anforderungen an Antwortzeiten und Datendurchsatz zu erfüllen. Solche Anforderungen können z. B. sein, dass innerhalb einer Stunde ein Durchsatz von 10.000 gedruckten Rechnungen erreicht werden muss oder dass die Antwortzeit für das Erfassen eines Kundenauftrags unter einer Sekunde liegen soll. Eine gute Performance ist keine absolute Eigenschaft einer betriebswirtschaftlichen Anwendung, sondern immer relativ zu den Anforderungen an diese zu sehen.

Proaktives Performancemanagement

Unter *Performanceoptimierung* verstehen wir in diesem Buch einen Prozess, der immer fünf Phasen umfasst: Die ersten beiden Phasen sind, zu einem *Verständnis der Geschäftsprozesse* zu kommen sowie *Performanceziele festzulegen und zu quantifizieren*. Diese Schritte beziehen alle Beteiligten mit ein, d. h. Techniker und Anwendungsfachleute. Nur auf der Basis dieser Voraussetzungen kann eine Optimierung erfolgreich sein. Die Phasen drei bis fünf umfassen dann die *systematische Überwachung, Identifizierung und Analyse* von Problemen, die *Umsetzung von Optimierungsmaßnahmen* und die erneute Analyse zur *Verifizierung des Erfolgs* der eingeleiteten Maßnahmen (siehe Abbildung 1). Vor unreflektiertem »Herumschrauben« an Konfigurationsparametern und vergleichbaren »Tuningschnellschüssen« müssen wir warnen! Ziel dieses Buches ist es vielmehr, Ihnen die Möglichkeiten zu eröffnen, Performanceprobleme zu identifizieren und zu analysieren, um diese dann gezielt beheben zu können.

Performance

Performance-
optimierung

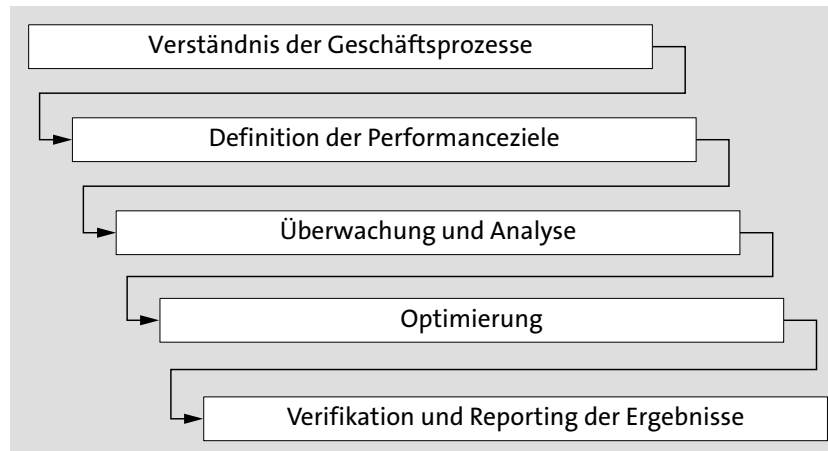


Abbildung 1 Performanceoptimierung in fünf Phasen

Technische Optimierung

Eine betriebswirtschaftliche IT-Anwendung besteht technisch gesehen aus zahlreichen Komponenten: aus logischen Komponenten, Prozessen wie Services, Threads oder Workprozessen und Speicherbereichen wie Puffern und Benutzerkontexten sowie physischen Komponenten wie Prozessoren (CPU), Hauptspeicher (RAM), Festplatten und Netzwerksegmenten. Diese Komponenten erlauben jeweils einen maximalen Durchsatz und eine optimale Antwortzeit. Ist das Zusammenspiel der Komponenten nicht ideal abgestimmt oder wird die Leistungsgrenze einzelner Komponenten erreicht, entstehen Wartesituationen, die sich negativ auf Durchsatz und Antwortzeit auswirken. Die Identifizierung, Analyse und Lösung solcher Probleme durch das Abstimmen der Komponenten aufeinander und die Verteilung der im System anfallenden Last sind die Aufgaben, die wir in diesem Buch als *technische Optimierung* bezeichnen.

Optimierung von Applikationen

Der zweite wichtige Bereich der Performanceoptimierung ist die Vermeidung unnötiger Last. Die Performance kann ebenso von ineffizienten Programmen oder dem ineffizienten Einsatz von Programmen negativ beeinflusst werden. Die Optimierung einzelner Programme bezeichnen wir als *Applikationsoptimierung*.

Ziel der Optimierung ist, zunächst die Systemeinstellung und die Applikationen derart zu verbessern, dass auf der Basis der bereits vorhandenen Hardwareressourcen die gewünschte Performance erreicht wird. Genügt dies nicht, müssen die Ressourcen entsprechend den gewonnenen Erkenntnissen erweitert werden.

Wie viel Tuning ist nötig?

Wie viel Aufwand ist für Performanceanalyse und Tuning einer SAP-Lösung nötig? Die Antwort auf diese Frage hängt stark von der Systemgröße ab. Für

eine kleine oder mittelgroße Installation ohne Modifikationen am SAP-Standard und ohne Eigenentwicklungen genügt in der Regel eine Performanceoptimierung kurz vor und kurz nach dem Produktivstart sowie nach größeren Veränderungen, z. B. durch Upgrades, Produktionsstarts neuer SAP-Lösungen, die Erhöhung der Benutzerzahl, größere Datenübernahmen oder Mandantentransporte. Darüber hinaus muss man natürlich nach Bedarf eingreifen, wenn akute Performanceprobleme vorliegen. Das Tuningpotenzial und damit auch der notwendige Analyse- und Optimierungsaufwand wachsen mit der Systemgröße an. Die Praxis zeigt, dass insbesondere kundeneigene Entwicklungen und Modifikationen am SAP-Standard zu Performanceengpässen führen. Ursache dafür sind meist ungenügende Tests, aber auch Zeitdruck und mangelnde Erfahrung der Entwickler. Den Extremfall bildet eine ständig weiterentwickelte Großinstallation mit vielen hundert Benutzern, komplizierten Prozessketten, einem Dutzend oder mehr Entwicklern (die oft verschiedenen Beratungsfirmen angehören und zu unterschiedlichen Zeiten und an unterschiedlichen Orten am System arbeiten) und einem ausgelagerten Systemmanagement. In einer solchen Systemumgebung ist es unumgänglich, dass eine kleine Gruppe von Administratoren und Entwicklern die Übersicht über das Gesamtsystem behält und die Performance nicht aus den Augen lässt.

Hilfe bei Performanceanalyse und Tuning bieten die Remote-Services der SAP. Namentlich sind dies der SAP GoingLive Check, der Ihnen den Produktivstart Ihres Systems erleichtert, und der SAP EarlyWatch Alert, der Ihr produktives System überwacht und Ihnen weitere Optimierungen vorschlägt.

Wie trägt ein *proaktives Performancemanagement* dazu bei, Ihr Ziel zu erreichen, eine betriebswirtschaftliche Anwendung erfolgreich zu betreiben? Wenn Sie dieses Ziel erreichen wollen, müssen Sie zwei Einflussfaktoren im Blick halten: die Zufriedenheit der Benutzer und die Kosten für den Betrieb der betriebswirtschaftlichen Anwendung. Die Kosten für den Betrieb ergeben sich zum einen aus den Kosten für Hardware (Infrastruktur, CPU, Hauptspeicher, Festplatten und Netzwerke) und Betriebspersonal (Administration, Wartung, Fehleranalyse). Nicht vernachlässigen dürfen Sie aber die Kosten, die entstehen, wenn eine Anwendung nicht zur Verfügung steht oder nicht die geforderte Performance erreicht – der Schaden übersteigt in diesem Fall innerhalb weniger Stunden oder Tage den Betrag, der im Mittel in einem Jahr für die proaktive Performanceoptimierung investiert wird. Diese Risikokosten müssen ebenfalls den Kosten für ein proaktives Performancemanagement gegenübergestellt werden. Tabelle 1 führt Ihnen anhand zweier konkreter Beispiele den Nutzen des proaktiven Performancemanagements vor Augen.

Proaktives Performancemanagement

Proaktive Maßnahme	Effekt im System	Sofortiger Nutzen durch höhere Benutzerzufriedenheit	Sofortiger Nutzen durch geringere Betriebskosten	Vermindertes Eskalationsrisiko
Optimierung von SQL-Anweisungen	Reduktion der Datenbanklast	schnellere Antwortzeiten bei bestimmten Transaktionen	Hardwareinvestitionen (Datenbankserver, Speichersystem) können gestreckt werden.	Überlastung des Datenbanksystems wird vermieden.
proaktives Datenmanagement (Datenvermeidung, Archivierung, Reorganisation)	<ul style="list-style-type: none"> Reduktion des Datenbankwachstums kürzere Zeiten für Wartungsarbeiten an der Datenbank (Backup/Recovery, Upgrade, Migration, Systemkopie) 	<ul style="list-style-type: none"> schnellere Antwortzeiten bei bestimmten Transaktionen kürzere Downtime bei Wartungsarbeiten 	<ul style="list-style-type: none"> Hardwareinvestitionen können gestreckt werden. geringerer Personalbedarf bei Wartungsarbeiten 	Datenbankgröße bleibt »beherrschbar«.

Tabelle 1 Beispiele für den Nutzen eines proaktiven Performancemanagements

Aktuelle Entwicklungen

Herausforderungen durch das Internet und mobile Geräte

Mit dem Aufstieg des Internets, der Smartphones und Tablet-Computer vollzieht sich ein Paradigmenwechsel in der Welt der betriebswirtschaftlichen Software: Nicht mehr auf den hoch spezialisierten Angestellten an seinem PC zielt die Software ab, sondern auf Benutzer des Internets (extern oder firmenintern als Intranet) oder den Benutzer eines mobilen Geräts. Der klassische Ansatz einer Prozessautomatisierung mit SAP R/3 basiert auf hoch spezialisierten Anwendern, die von festen Arbeitsplätzen über installierte SAP GUIs auf ihr ERP-System (Enterprise Resource Planning) zugreifen. Die Rolle dieses spezialisierten Vermittlers, der ausgebildet sein muss, um Software zu bedienen, wird allerdings an vielen Stellen zunehmend überflüssig. Stattdessen erhält der Endbenutzer durch Internet und Mobilgerät direkten Zugriff auf die ERP-Systeme der Unternehmen. Heute können z.B. in vielen Unternehmen die Mitarbeiter über das Intranet ihre Arbeits- und Fehlzeiten, Reisekostenabrechnungen etc. selbst in das System eingeben, wo dies früher über zentrale Benutzer geschah. Kunden

bestellen heute vermehrt ihre Produkte direkt über das Internet und nicht mehr über den Umweg eines Briefes, Faxes oder Telefonanrufs bei einem Vertriebscenter.

Die Erwartungen, die der Anwender einer Internet- oder mobilen Anwendung an die Bedienbarkeit und Performance hat, sind ungleich höher als die des klassischen Angestellten an sein ERP-System. Der Angestellte ist auf »sein« ERP-System angewiesen, und wenn es ihm normalerweise dabei hilft, seine tägliche Arbeit zu vereinfachen, wird er es akzeptieren und auch kleinere Fehler oder Performanceschwächen in Kauf nehmen. Ganz anders der Internetbenutzer: Funktionieren die Anwendungen, die ihm im Internet angeboten werden, nicht einfach und performant, kann er sofort zur Konkurrenzfirma wechseln und dort z. B. seine Einkäufe tätigen (»Die Konkurrenz ist nur einen Mausklick entfernt.«). Hinzu kommt: Das Internet macht nicht um 17 Uhr Feierabend – von einer E-Business-Lösung im Internet werden Verfügbarkeit und Performance an 365 Tagen 24 Stunden lang gefordert. Nutzer von mobilen Anwendungen legen die Maßstäbe in Bezug auf Bedienbarkeit und Performance an eine SAP-Anwendung an, die sie auch von anderen Mobile Apps gewohnt sind.

SAP HANA

Mit SAP HANA ist es SAP gelungen, ein Produkt auf den Markt zu bringen, das Analysten bereits heute als die wichtigste Innovation seit Jahren im Bereich von Business Software bezeichnen. Kern der Innovation ist eine *Hauptspeicherdatenbank*, um die sich weitere Services, z.B. ein als *XS-Engine* bezeichneter Applikationsserver, gruppieren. In diesem Buch gehen wir ausschließlich auf die SAP-HANA-Datenbankplattform ein, da zu den weiteren Services die Felderfahrungen noch fehlen.

Das wichtigste Argument für die Einführung von SAP HANA ist Performance! SAP HANA macht sich die Verfügbarkeit von riesigen Hauptspeichern und massiv parallelen Prozessorarchitekturen zunutze und setzt diese konsequent durch modernste Softwarearchitektur in Performance um. Vielleicht werden Sie sich fragen: Brauche ich überhaupt noch ein Buch über Performance, wenn ich SAP HANA habe? Oder löst SAP HANA alle Performanceprobleme? Die Antwort auf die zweite Frage ist – nach der festen Überzeugung des Autors – ein doppeltes Nein: Zum einen kann auch SAP HANA keine Wunder vollbringen, wenn Ihr Programm ganze Datenbanktabellen in den Applikationsserver liest und – schlimmer noch – die Daten weiter zum Webbrowser des Anwenders schickt. Der Löwenanteil der Laufzeit entfällt dann auf den Applikationsserver, das Netzwerk und den Browser. Die grundlegenden Regeln zur performanten Programmierung sind auch bei SAP HANA nicht außer Kraft gesetzt! Ein zweites Nein ergibt sich aus der Tatsache, dass jeder technischen Innovation »Begehrlichkeiten«

oder, anders formuliert, Herausforderungen entgegenstehen. Eine dieser Herausforderungen heißt *Big Data*.

Big Data Im Zentrum von Big Data steht die Erkenntnis, dass die Daten der Unternehmenskunden zum wichtigen Rohstoff für Unternehmen geworden sind. Um diesen Rohstoff zu fördern, braucht man eine neue Generation von Datenbankprogrammen, die schnell mit einer riesigen Menge strukturierter und unstrukturierter Daten umgehen können. Beispiele für solche Daten sind Beiträge in sozialen Netzwerken, Protokolle von Webzugriffen, Bewegungsdaten von Personen (z. B. über Mobilfunkortungen oder Posts mit Lokationsdaten in sozialen Netzwerken) und Produkten (die z. B. über RFID-Chips erfasst werden), Daten von Kameras, Mikrofonen und sonstigen Sensoren, Finanztransaktionen und Börsendaten sowie Verbrauchsdaten im Energiesektor. Unternehmen sind interessiert daran, diese Daten zu erfassen, zu verknüpfen, auszuwerten und so über ihre Kunden, Märkte und Produkte wertvolle Einsichten zu gewinnen. Als Beispiel sei ein Produktionsplaner in der Konsumgüterindustrie genannt. Um vorausschauender planen zu können, werden ihm in Zukunft nicht nur die Produktionszahlen der Vergangenheit und aktuelle Bestellungen als Grundlage dienen, sondern auch die aktuellen Trends der sozialen Netzwerke.



Soziale Netzwerke und das Internet der Dinge

In nur einer Minute liken Facebook-Nutzer mehr als vier Millionen Inhalte im sozialen Netzwerk, und Twitter-Nutzer verschicken mehr als 300.000 Tweets. Diese Zahlen stammen aus dem Jahr 2016, und es ist davon auszugehen, dass die Datenmenge weiter wächst. Von einer zeitnahen Auswertung dieser Daten können Handelsunternehmen und Hersteller von Markenartikeln profitieren, indem sie genauer und schneller als bisher erfahren, was die Endverbraucher wünschen, Banken können Finanztrends schneller analysieren und Gesundheitsbehörden den Ausbruch von Epidemien frühzeitig erkennen.

Im *Internet der Dinge*, einem 1999 von Kevin Ashton eingeführten Begriff, besteht das Internet nicht nur aus menschlichen Teilnehmern, sondern auch aus Dingen. Die automatische Identifikation von Dingen mittels Radio-Frequency Identification (RFID), Strich- oder 2D-Code, die Sensoren und Aktuatoren erfassen, ermöglicht es Unternehmen, über Ereignisdaten mit ihren Produkten fortlaufend Kontakt zu halten und die daraus gewonnenen Daten in ihre Unternehmensplanung mit einzubeziehen.

Eine dritte Kategorie neuer Datenquellen sind die Daten, die über Sensoren im Unternehmen direkt anfallen, z. B. Produktionsdaten oder Patientendaten in Kliniken. Diese können zu Analyse- und Diagnosezwecken für das Unternehmen nutzbar gemacht werden.

Neben den bisherigen Stamm- und Transaktionsdaten, die das Unternehmen direkt betreffen und die bisher im Fokus von betriebswirtschaftlicher Software standen, entsteht also eine neue Klasse von Daten, die wir als *Ereignisdaten* bezeichnen. Im Vergleich zu einer »klassischen« Transaktion, z. B. einem Vertriebsbeleg, Lieferschein oder einer Rechnung, ist ein Ereignisdatensatz klein. Die Anzahl der Datensätze dagegen ist gewaltig. Um diese auswerten zu können, werden neue Anwendungen benötigt, die sich z. B. auf Methoden des *Data Minings* stützen, im Fall der Auswertung von sozialen Netzwerken kommen auch noch Methoden der linguistischen Analyse hinzu. SAP HANA und SAP Vora sind die Antwort der SAP auf diese Herausforderungen.

Ereignisdaten

All diesen Datenquellen ist gemein, dass ihre Menge die traditioneller betriebswirtschaftlicher Daten (sogenannter *Stamm- und Bewegungsdaten*) um ein Vielfaches übersteigt. Übertragen auf unsere Performancefrage heißt das: Ein nicht performanceoptimiertes System oder Programm hat einen viel höheren Effekt als in einem »traditionellen« SAP-Business-Suite-System – als Konsequenz folgern wir: Auch in Zukunft wird Performance-Know-how ein wertvolles Gut sein.

Kleine »Fehler« – große Wirkung

Zu diesem Buch

Die in diesem Buch dargestellten Methoden der Performanceanalyse und -optimierung entsprechen den Verfahren, die ursprünglich von den Experten des SAP EarlyWatch Alert und des SAP GoingLive Checks verwendet und in den SAP-Basis-Trainings *Workload Analysis* und *Optimierung von ABAP-Programmen* vermittelt werden. Mit jeder neuen Auflage dieses Buches – mittlerweile der achten – wird die Chance genutzt, aktuelle Trends der Produktentwicklung bei SAP und – soweit relevant – Entwicklungen der IT-Welt allgemein adäquat zu behandeln.

Grundlagen des Buches

In dieser Auflage hat sich das Kapitel zu SAP HANA in seinem Umfang rund verdoppelt, was die Bedeutung von SAP HANA für die SAP-Produkte unterstreicht. Des Weiteren wird SAP HANA nun auch in den anderen Kapiteln als die führende Datenbank behandelt. So wurden Kapitel 4 um das Thema Sizing und Skalierung von SAP HANA und das Kapitel 10 um das Thema Multi Version Concurrency Control (MVCC) ergänzt. Kapitel 14 wurde um SAP-HANA-spezifische Empfehlungen für das SAP Business Warehouse (SAP BW) ergänzt.

SAP HANA

Nach wie vor bleibt der Fokus auf der SAP-HANA-Datenbankplattform, das Thema XS Engine bleibt aufgrund der aktuellen architektonischen Änderungen unbehandelt. Im Gegenzug ist das Kapitel zu TREX und SAP BW

SAP Fiori, SAPUI5, OData

Accelerator entfallen. Auch das Thema SAP Vora wird aufgrund der fehlenden Felderfahrung nicht behandelt.

Das Kapitel zu webbasierten User Interfaces (UIs) und Services ist im Lichte von SAP Fiori, der neuen UI-Strategie der SAP, praktisch komplett neu geschrieben worden und behandelt nun die neuen Weboberflächen und die Themen SAP Fiori, SAPUI5 und OData/Gateway. Auch dem Thema Pufferung von Webdokumenten (Webcaching) wurde breiter Raum eingeräumt.

Weiterhin sind folgende Themen neu aufgenommen worden:

- SQL-Monitor zur Identifizierung teurer SQL-Anweisungen (SQLM, Kapitel 2)
- Zero Administration Memory Management auf UNIX (Kapitel 5)
- neues ABAP-Lastverteilungskonzept und neue, schnelle Serialisierung für Remote Function Calls (RFCs, Kapitel 6)
- Standalone-Enqueue-Server (Kapitel 6 und 10)

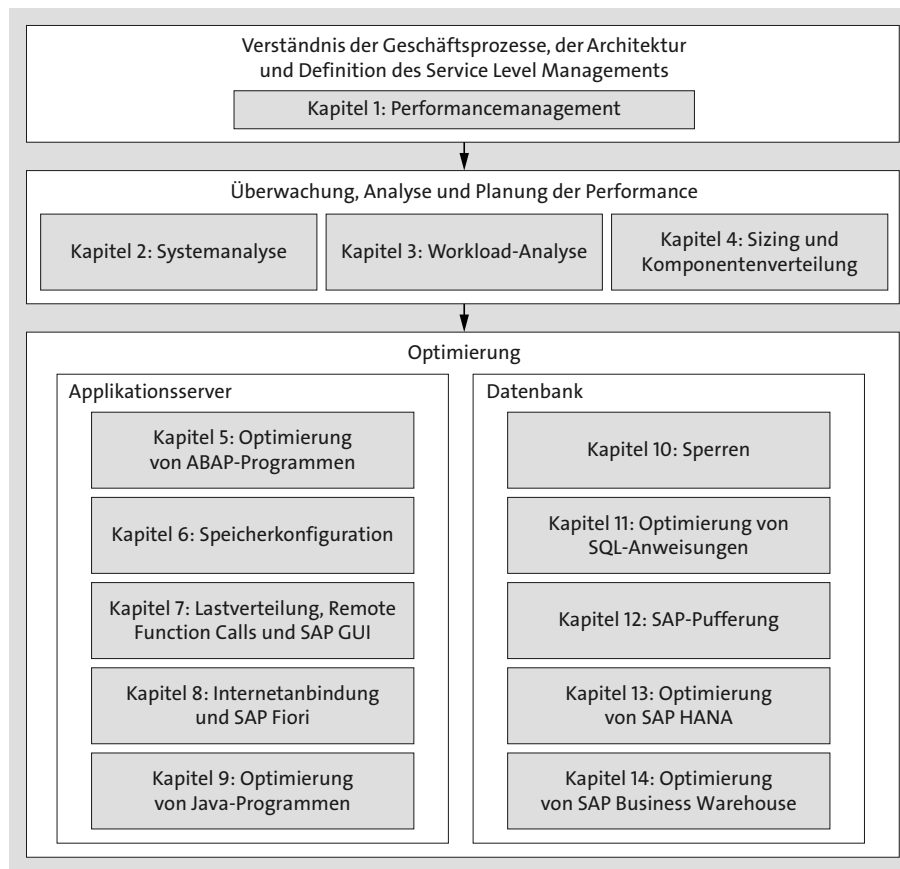


Abbildung 2 Die Kapitel dieses Buches

Aufbau

Abbildung 2 stellt die Kapitel dieses Buches, orientiert an den fünf Phasen der Performanceoptimierung, im Überblick dar. **Kapitel 1**, »Performance-management einer SAP-Lösung«, wendet sich sowohl an SAP-Administratoren und SAP-Berater als auch an Anwendungsentwickler und SAP-Projektleiter. Es behandelt auf einem nicht technischen Niveau die folgenden grundlegenden Fragen zur Performanceanalyse:

- Welche Vorkehrungen müssen getroffen werden, um eine optimale Performance einer SAP-Lösung zu gewährleisten?
- Welche Maßnahmen zum Performancetuning kommen in Betracht?
- Welche Personen sind in den Tuningprozess involviert?

Hinter der dem Benutzer angebotenen Leistung verbirgt sich in der Praxis oft ein Netz von Partnern, die jeweils Teilleistungen erbringen. Viele Teile werden von unterschiedlichen, manchmal externen Service Providern erbracht. Um diese Komplexität in den Griff zu bekommen, führen die meisten Leistungsanbieter und Kunden ein *Service Level Management (SLM)* ein. Als SLM bezeichnet man eine strukturierte proaktive Methode, die das Ziel hat, den Benutzern einer IT-Anwendung ein adäquates Servicenniveau zu garantieren – in Übereinstimmung mit den betriebswirtschaftlichen Zielen des Auftraggebers und bei optimalen Kosten. Wir zeigen in diesem Buch, mit welchen Werkzeugen und Methoden Sie ein SLM für eine SAP-Lösung einführen.

In Kapitel 2 und Kapitel 3 finden Sie die Darstellung der Performanceanalyse anhand des SAP NetWeaver Application Servers (AS) ABAP. Nach Lektüre dieser Kapitel sind Sie in der Lage, eine systematische Performanceanalyse für ein SAP-System auf Basis von AS ABAP, Datenbank und Betriebssystem durchzuführen.

In diesem Buch folgen wir zunächst der Strategie der Bottom-up-Analyse und beginnen in **Kapitel 2**, »Analyse von Hardware, Datenbank und ABAP-Applikationsserver«, mit den Untersuchungen der Teilsysteme Betriebssystem, Datenbank, SAP-Speicherkonfiguration und SAP-Workprozesse. Gleichzeitig werden Lösungsvorschläge angeboten, die den Administrator oder Berater befähigen sollen, die wichtigsten Performanceprobleme zu lösen. Für kleine und mittelgroße Installationen wird diese Stufe des Tunings in vielen Fällen bereits ausreichen.

Anschließend wird in **Kapitel 3**, »Workload-Analyse«, die komplexere Workload-Analyse als Top-down-Analyse diskutiert.

In **Kapitel 4**, »Hardware-Sizing und Komponentenverteilung«, stellen wir zum einen dar, was Sie beachten müssen, um eine optimal ausgelegte Hardware für Ihr SAP-System zu erreichen. Die Voraussetzung dafür ist,

dass auf der einen Seite keine Hardwareengpässe entstehen, dass aber auf der anderen Seite nicht zu hohe Kosten durch unnötig vorgehaltene Hardware anfallen. Zum anderen stellen wir in diesem Kapitel die Themen Serverkonsolidierung, Virtualisierung und Skalierung dar.

Die weiteren Kapitel 5 bis 14 vermitteln Wissen, das für eine umfassende Performanceanalyse notwendig ist. Sie wenden sich an SAP-Betreuer, die für das performante Funktionieren großer Systeme verantwortlich sind und die das Tuningpotenzial ihrer Systeme voll ausschöpfen wollen und müssen. Kapitel 5 bis 14 sind weitgehend eigenständige Einheiten und können mit Kenntnis der ersten vier Kapitel in beliebiger Reihenfolge gelesen werden. Eventuelle Abhängigkeiten sind am Anfang jedes Kapitels vermerkt.

In Kapitel 5 bis Kapitel 9 werden die Themen behandelt, die sich auf den Applikationsserver und den Präsentationsserver beziehen:

- **Kapitel 5, »Optimierung von ABAP-Programmen«:** Zeigt Methoden zur Analyse einzelner ABAP-Programme auf, u. a. mit den Hilfsmitteln Performancetrace und ABAP-Laufzeitanalyse, Debugger, Code Inspector, und gibt Hinweise zur Optimierung des ABAP-Codes.
- **Kapitel 6, »Speicherkonfiguration« (Memory Management):** Die Konfiguration der auf dem AS ABAP allokierten Speicherbereiche hat wesentlichen Einfluss auf die Performance.
- **Kapitel 7, »Lastverteilung, Remote Function Calls und SAP GUI«:** Eine optimale Lastverteilung von Webanfragen, Dialog-, Verbuchungs- und Hintergrundaufträgen hilft dabei, Hardware optimal zu nutzen und Engpässe aufgrund falscher Konfiguration zu vermeiden. Die Performance von Schnittstellen zwischen Softwarekomponenten trägt ebenfalls maßgeblich zur Performance der gesamten Lösung bei. Im Detail werden hier auch die Performanceaspekte von SAP GUI und SAP GUI Controls behandelt.
- **Kapitel 8, »Internetanbindung und SAP Fiori«:** Die Anbindung von SAP-Systemen über das Internet (HTTP-Protokoll) als Web-UIs oder Webservices kommt immer größere Bedeutung zu. Im Detail werden hier die Performanceaspekte von SAP GUI for HTML (Internet Transaction Server, ITS), von Business Server Pages und von Web Dynpro ABAP sowie von SAP Fiori, SAPUI5 und OData/Gateway behandelt.
- **Kapitel 9, »Optimierung von Java-Programmen«:** In diesem Kapitel finden Sie die Beschreibung der Werkzeuge, mit denen Sie die Performanceanalyse der Java Virtual Machine (SAP JVM) und von Java-Programmen durchführen können.

Kapitel 10 eröffnet die Reihe der Kapitel zu Datenbankthemen. Das zentrale Kapitel in diesem Block ist Kapitel 11, das Voraussetzung für die folgenden Kapitel 12 bis 14 ist:

- **Kapitel 10, »Sperren«:** Datenbank- und SAP-Sperren gewährleisten die Datenkonsistenz. Durch eine optimierte Verwaltung von Sperren (z. B. durch effiziente Programmierung oder auch durch die Pufferung von Nummernkreisen) werden Durchsatzengpässe vermieden.
- **Kapitel 11, »Optimierung von SQL-Anweisungen«:** Ineffektive SQL-Anweisungen beanspruchen die Datenbank übermäßig und werden daher zum Problem für die Performance der gesamten Anwendung. In diesem Kapitel stellen wir die Detailanalyse für »teure« SQL-Anweisungen sowie Optimierungsmöglichkeiten durch Datenbankindizes und durch Programmoptimierung (fünf goldene Regeln) dar.
- **Kapitel 12, »SAP-Pufferung«:** Die Pufferung von Tabellen auf den Applikationsservern beschleunigt den Zugriff auf oft gelesene Daten und dient der Entlastung der Datenbank.
- **Kapitel 13, »Optimierung von Datenbankabfragen mit SAP HANA«:** Dem neuen »Wunderkind« von SAP ist ein eigenes Kapitel gewidmet, das in die Grundlagen, Werkzeuge und Methoden der Performanceanalyse und -optimierung einführt.
- **Kapitel 14, »Optimierung von Anfragen an SAP Business Warehouse«:** Business-Warehouse-Anfragen sind spezielle SQL-Anweisungen, die in der Regel große Datenmengen bearbeiten – für diese Art von Anfragen existieren spezielle Optimierungsmöglichkeiten.

Kenntnisse in der Performanceoptimierung von SAP-Systemen und -Anwendungen sind für SAP-Administratoren, SAP-Anwendungsbetreuer, SAP-Entwickler und SAP-Projektleiter von größtem Nutzen – an diese Zielgruppen richtet sich dieses Buch. Jedem Kapitel ist am Ende der Einleitung ein kurzer Abschnitt mit dem Titel »Wann sollten Sie dieses Kapitel lesen« vorangestellt. Aus diesem ergibt sich die genaue Zielgruppe des jeweiligen Kapitels.

Dieses Buch setzt an den Stellen, an denen es um die konkrete Umsetzung von Empfehlungen geht, Kenntnisse in der Administration von SAP-Komponenten in Theorie und Praxis voraus. Insbesondere die Bedienung des *Computer Center Management Systems* (CCMS) sollte Ihnen vertraut sein. Als gute Vorbereitung kann z. B. der Band »SAP NetWeaver AS ABAP – Systemadministration« dienen (siehe Anhang E, »Informationsquellen«). Teile dieses Buches, z. B. die Kapitel 5 und 9, setzen Wissen über die Pro-

Zielgruppen

Voraussetzungen

grammiersprache ABAP bzw. Java, Kapitel 10 bis 14 Kenntnisse über die Funktionsweise von relationalen Datenbanken und SQL voraus.

Grenzen des Buches

In den folgenden Bereichen stößt das Buch an seine Grenzen:

■ **Hardware- und Netzwerktuning**

Ein Engpass bei CPU, Hauptspeicher, I/O oder Netzwerk lässt sich zwar aus dem SAP-System heraus mit großer Wahrscheinlichkeit identifizieren, für eine detaillierte Analyse werden jedoch die Werkzeuge der Hardware- bzw. Netzwerkanbieter benötigt.

■ **Datenbanken**

SAP bietet mit dem DBA-Cockpit ein Werkzeug an, das die Administration und Analyse der verschiedenen Datenbanksysteme so weit wie möglich vereinheitlicht. Wer jedoch tiefer in das Datenbanktuning einsteigt, kommt nicht daran vorbei, sich mit den unterschiedlichen Architekturen der Datenbanksysteme zu beschäftigen. Dabei ist es unmöglich, die Feinheiten aller mit SAP-Lösungen eingesetzten Datenbanksysteme in diesem Buch in der gebührenden Tiefe zu behandeln. Allerdings ist dies auch nicht notwendig, da zu allen Datenbanksystemen Literatur zum Tuning vorhanden ist. Dieses Buch kann und will diese Literatur nicht ersetzen. Sein Schwerpunkt liegt vielmehr im SAP-spezifischen Kontext des Datenbanktunings und in der Vermittlung datenbanksystemübergreifender Konzepte. Die konkreten Beispiele betreffen immer einzelne Datenbanksysteme. In Anhang A finden Sie eine Übersicht über die wichtigsten Monitore zur Datenbankanalyse für alle Datenbanksysteme.

■ **Applikationstuning**

Viele Performanceprobleme lassen sich nur mit detailliertem Anwendungswissen und Kenntnis der einzelnen SAP-Lösungen lösen. Oft ist es eine Änderung im Customizing, die die Lösung des Problems bringt. Know-how zum Tuning einzelner SAP-Lösungen ist nicht Teil dieses Buches. Allerdings vermittelt es Ihnen Analysestrategien, sodass Sie Performanceprobleme auf bestimmte Applikationen eingrenzen und somit an den richtigen Entwickler oder Berater adressieren können.

■ **SAP-Cloud-Anwendungen**

SAP bietet inzwischen fast alle ihre Produkte auch in der Cloud an. Hinzu kommen spezielle Software-as-a-Service-(SaaS-) und Platform-as-a-Service-(PaaS-)Anwendungen, die ausschließlich in der Cloud laufen. Auf die Spezifika dieser Anwendungen gehen wir in diesem Buch ebenfalls nicht ein.

Dieses Buch enthält zahlreiche zeitabhängige Informationen und Regeln. Nur so kann Ihnen dieses Buch als Nachschlagwerk für die tägliche Arbeit in der SAP-Administration dienen. Auf der anderen Seite ist klar: Eine neue Version, ein Patch (der SAP-Komponente, der Datenbank oder des Betriebssystems), eine neue Rechnergeneration – diese und andere Faktoren können mit einem Schlag alte Informationen wertlos machen; im schlimmsten Fall können sich veraltete Empfehlungen sogar kontraproduktiv auf die Performance auswirken. Dieses Buch ist also kein Gesetzbuch, und wer Performanceoptimierung nur als stures Befolgen von Regeln betreibt, handelt fahrlässig. Die direkte Auseinandersetzung mit der Lösung, der SAP-Onlinehilfe und aktuellen SAP-Hinweisen im SAP Support Portal kann dieses Buch nicht ersetzen, sondern nur fördern.

Alle Angaben zu Menüpfaden, Bezeichnungen in den Bildschirmen der Performancemonitore und Richtwerten für Performancekennzahlen beziehen sich – sofern nicht anders vermerkt – auf SAP NetWeaver 7.50 und SAP HANA Version 1, SPS12.

In diesem Buch finden Sie mehrere Orientierungshilfen, die Ihnen die Arbeit mit dem Buch erleichtern sollen.

In hervorgehobenen Informationskästen sind Inhalte zu finden, die wissenswert und hilfreich sind, aber etwas außerhalb der eigentlichen Erläuterung stehen. Damit Sie die Informationen in den Kästen sofort einordnen können, haben wir die Kästen mit Symbolen gekennzeichnet:

Die mit diesem Symbol gekennzeichneten *Tipps* geben Ihnen spezielle Empfehlungen, die Ihnen die Arbeit erleichtern können.

In Kästen, die mit diesem Symbol gekennzeichnet sind, finden Sie Informationen zu *weiterführenden Themen* oder wichtigen Inhalten, die Sie sich merken sollten.

Dieses Symbol weist Sie auf *Besonderheiten* hin, die Sie beachten sollten. Es *warnt Sie* außerdem vor häufig gemachten Fehlern oder Problemen, die auftreten können.

Beispiele, durch dieses Symbol kenntlich gemacht, weisen auf Szenarien aus der Praxis hin und veranschaulichen die dargestellten Funktionen.

Wie schon für die früheren Auflagen werden wir Aktualisierungen und gegebenenfalls Korrekturen zum Buch auf der Verlagswebsite (<https://www.sap-press.de/4330>) bereitstellen.

Release-abhängigkeit

SAP-NetWeaver- und SAP-HANA-Version

Hinweise zur Lektüre



Updates auf der Verlagswebsite

Auf einen Blick

1	Performancemanagement einer SAP-Lösung	33
2	Analyse von Hardware, Datenbank und ABAP-Applikationsserver	75
3	Workload-Analyse	131
4	Hardware-Sizing und Komponentenverteilung	181
5	Optimierung von ABAP-Programmen	223
6	Speicherkonfiguration	259
7	Lastverteilung, Remote Function Calls und SAP GUI	291
8	Internetanbindung und SAP Fiori	349
9	Optimierung von Java-Programmen	397
10	Sperrungen	443
11	Optimierung von SQL-Anweisungen	473
12	SAP-Pufferung	531
13	Optimierung von Datenbankabfragen mit SAP HANA	571
14	Optimierung von Abfragen an SAP Business Warehouse	681

Inhalt

Vorwort und Danksagung	17
Einleitung	19

1 Performancemanagement einer SAP-Lösung 33

1.1 Die Architektur von SAP-Lösungen	33
1.1.1 SAP-Lösungen und -Komponenten	34
1.1.2 SAP-HANA-Einsatzszenarien	37
1.1.3 Client-Server-Architektur	41
1.2 Das Überwachungs- und Optimierungskonzept für eine SAP-Lösung	49
1.2.1 Anforderungen an ein Überwachungs- und Optimierungskonzept	50
1.2.2 Service Level Management	53
1.2.3 Das Konzept für eine kontinuierliche Performanceoptimierung	60
1.2.4 Werkzeuge und Methoden für das Überwachungs- und Optimierungskonzept	66
1.2.5 SAP Solution Manager	69
1.3 Zusammenfassung	72

2 Analyse von Hardware, Datenbank und ABAP-Applikationsserver 75

2.1 Begriffsklärungen	76
2.2 Hardwareanalyse	77
2.2.1 Analyse eines Hardwareengpasses (CPU und Hauptspeicher)	79
2.2.2 Identifizierung von Schreib-/Lese-(I/O-)Problemen	85
2.2.3 Parameteränderungen und Netzwerkcheck	86
2.3 Datenbankanalyse	89
2.3.1 Der Performancemonitor im DBA-Cockpit	90

2.3.2	Analyse des Datenbank Hauptspeichers	91
2.3.3	Identifizierung teurer SQL-Anweisungen	93
2.3.4	Identifizierung von Schreib-/Lese-(I/O-)Problemen	100
2.3.5	Weitere Analysen auf Datenbankebene	101
2.4	Analyse der SAP-Speicherkonfiguration	110
2.4.1	Analyse der SAP-Puffer	111
2.4.2	Analyse des SAP Extended Memorys, des SAP Heap Memorys und des SAP Roll Memorys	113
2.4.3	Anzeige des allokierten Speichers	115
2.4.4	Weitere Monitore zur detaillierten Analyse	117
2.5	Analyse der SAP-Workprozesse	119
2.5.1	Felder der Workprozess-Übersicht	119
2.5.2	Analyse der Workprozesse	122
2.5.3	Überwachung der Dispatcher-Queue	126
2.6	Analyse des Internet Communication Managers (ICM)	128
2.7	Zusammenfassung	129
2.8	Übungsfragen	129
3	Workload-Analyse	131
3.1	Grundlagen der Workload-Analyse und der Laufzeitanalyse ...	132
3.2	Der Workload-Monitor	134
3.2.1	Mit dem Workload-Monitor arbeiten	135
3.2.2	Technische Einstellungen zum Workload-Monitor	138
3.3	Workload-Analyse	138
3.3.1	Ablauf eines Transaktionsschrittes	138
3.3.2	Weitere Zeitkomponenten	141
3.3.3	Interpretation der Antwortzeiten	143
3.3.4	Aktivität, Durchsatz und Last	145
3.4	Workload-Analyse durchführen	147
3.4.1	Allgemeines Performanceproblem analysieren	147
3.4.2	Spezielles Performanceproblem analysieren	154
3.5	Einzelstatistik	157
3.6	End-to-End-Workload-Analyse	162
3.6.1	Grundlagen der End-to-End-Workload-Analyse	162
3.6.2	Der zentrale Workload-Monitor	164

3.6.3	End-to-End-Workload-Monitor und End-to-End- Laufzeitanalyse im SAP Solution Manager	168
3.6.4	Zentrale Einzelstatistik	168
3.6.5	End-to-End-Laufzeitanalyse mit dem Solution Manager	172
3.7	Zusammenfassung	175
3.8	Übungsfragen	178
4	Hardware-Sizing und Komponentenverteilung	181
4.1	Initiales Hardware-Sizing	182
4.1.1	Übersicht über das Projekt zum initialen Sizing	183
4.1.2	Sizing-Methoden	188
4.1.3	Sizing-Projekt im Detail durchführen	193
4.1.4	SAP Standard Application Benchmarks	196
4.2	Sizing bei Laststeigerung, Versionswechsel oder Migration	202
4.2.1	Übersicht über ein Sizing-Projekt im Umfeld einer bereits produktiven Installation	202
4.2.2	Sizing im Umfeld produktiver Installationen im Detail durchführen	203
4.2.3	SAP-HANA-Migration	210
4.3	Systemlandschaft planen	212
4.3.1	Virtualisierung und Hardwarekonsolidierung	214
4.3.2	Skalierung der Datenbankebene, insbesondere von SAP HANA	216
4.3.3	Verteilung von SAP-Applikationsinstanzen	219
4.4	Zusammenfassung	220
4.5	Übungsfragen	221
5	Optimierung von ABAP-Programmen	223
5.1	Performance-Trace	223
5.1.1	Performance-Trace erstellen	224
5.1.2	SQL-Trace auswerten	225
5.1.3	Weitere Funktionen im SQL-Trace	230

5.1.4	Puffer-Trace auswerten	232
5.1.5	RFC-Trace auswerten	233
5.1.6	HTTP-Trace auswerten	234
5.1.7	Enqueue-Trace auswerten	235
5.2	Performanceanalyse mit dem ABAP-Trace (Laufzeitanalyse) ...	236
5.2.1	ABAP-Trace erstellen	237
5.2.2	ABAP-Trace auswerten	239
5.2.3	Varianten verwenden	241
5.2.4	Zeitachsensicht verwenden	243
5.3	Analyse des Speicherverbrauchs mit dem ABAP Debugger und im Memory Inspector	244
5.4	Code Inspector	247
5.5	Tipps und Tricks für performante ABAP-Programme	250
5.6	Zusammenfassung	256
5.7	Übungsfragen	257
6	Speicherkonfiguration	259
6.1	Grundlagen der Speicherkonfiguration	259
6.1.1	Begriffsklärungen	259
6.1.2	Speicherbereiche der SAP-Instanz	261
6.1.3	Zero Administration Memory Management	264
6.1.4	Technische Realisierung im Detail	264
6.1.5	Zusammenfassung	274
6.2	Speicherbereiche konfigurieren und überwachen	275
6.2.1	Auslagerungsspeicher überwachen	276
6.2.2	SAP-Speicherbereiche konfigurieren und überwachen	277
6.2.3	Hilfe zur Fehlerbehebung	281
6.3	Zusammenfassung	288
6.4	Übungsfragen	290
7	Lastverteilung, Remote Function Calls und SAP GUI	291
7.1	Services des SAP NetWeaver Application Servers	292

7.2	Lastverteilung innerhalb der ABAP-Instanzen	295
7.2.1	Message- und Enqueue- Service verteilen	296
7.2.2	Dialog-, Hintergrund- und Spool-Workprozesse verteilen	297
7.2.3	Benutzer und Workprozesse auf CPU-Ressourcen verteilen	298
7.2.4	Dynamische Benutzerverteilung: Anmeldegruppen konfigurieren	301
7.2.5	Lastverteilungskonzept für Dialog-Workprozesse	304
7.2.6	Ressourcen pro Benutzer beschränken	307
7.2.7	Betriebsarten	308
7.2.8	Dynamische Workprozesse konfigurieren	308
7.2.9	Hintergrundverarbeitung	309
7.2.10	Verbuchung	310
7.3	Remote Function Calls (RFCs)	317
7.3.1	Grundlagen und Begriffe	317
7.3.2	Ablauf eines RFCs	321
7.3.3	RFC-Verbindungen konfigurieren und testen	323
7.3.4	Serialisierung auswählen	327
7.3.5	Eingehende und ausgehende Last überwachen	328
7.3.6	Parallelisierung von Prozessen mit asynchronen RFCs konfigurieren	333
7.3.7	Datenübertragung mit transaktionalen RFCs überwachen	334
7.3.8	Hintergrund-RFCs	336
7.4	SAP GUI	336
7.4.1	Interaktionsmodell und Performancemessung	337
7.4.2	Performance der GUI-Kommunikation analysieren und optimieren	339
7.5	Zusammenfassung	344
7.6	Übungsfragen	347
8	Internetanbindung und SAP Fiori	349
8.1	SAP-Webanwendungen	350
8.2	Analysen auf dem Präsentationsserver	354
8.2.1	Performancewerkzeuge der Internetbrowser	356
8.2.2	SAP-Statistiken in der HTTP-Anfrage	358
8.2.3	Webanwendungen kontinuierlich überwachen	360

8.3	Pufferung von Webdokumenten	362
8.3.1	Browserpuffer (Browsercache)	362
8.3.2	Puffer im ICM und im SAP Web Dispatcher	365
8.3.3	Content Delivery Network	366
8.3.4	Zusammenfassung: Nutzung von Puffern in Webanwendungen	367
8.4	Performanceanalyse von Webanwendungen auf dem SAP NetWeaver Application Server ABAP	368
8.4.1	HTTP-Trace im Internet Communication Manager	369
8.4.2	Performanceanalyse von ABAP-Webanwendungen durchführen	370
8.4.3	Aufrufe von Webservices überwachen	372
8.5	Business Server Pages (BSP), Web Dynpro ABAP und integrierter ITS	373
8.5.1	Business Server Pages und Web Dynpro ABAP	373
8.5.2	Integrierter ITS	377
8.6	SAP Fiori, SAPUI5 und OData-Services	378
8.6.1	Grundlagen von Fiori, SAPUI5 und OData	378
8.6.2	Aufbau einer Fiori-Systemlandschaft	380
8.6.3	Allgemeine Performancegesichtspunkte von SAP Fiori, SAPUI5 und OData	383
8.6.4	SAP Fiori, SAPUI5 und OData auf dem SAP NetWeaver Application Server ABAP	386
8.7	Zusammenfassung	393
8.8	Übungsfragen	395
9	Optimierung von Java-Programmen	397
9.1	Garbage Collection	399
9.1.1	Aufbau des Java Heaps	399
9.1.2	Garbage-Collection-Algorithmen	401
9.1.3	Auswahl des Kollektors	402
9.1.4	Parametrisierung des Java Heaps und der Garbage Collection	403
9.1.5	Protokollierung von Garbage Collections	405
9.2	Just-in-time-Compiler	405
9.3	SAP Management Console	407

9.4	SAP Java Virtual Machine Profiler	409
9.4.1	Architektur	409
9.4.2	Verbindungsaufbau	410
9.4.3	Allocation Analysis	413
9.4.4	Performance Hotspot Analysis	419
9.4.5	Method Parameter Analysis	423
9.4.6	Synchronization Analysis	425
9.4.7	File- und Network-I/O-Analysis	428
9.4.8	Garbage Collection Analysis	430
9.4.9	Klassenstatistik und Heap Dump	432
9.5	Memory Analyzer	435
9.5.1	HPROF-Dateien erstellen	435
9.5.2	Analyse des Java Heaps mit dem Memory Analyzer	435
9.6	Weitere Werkzeuge zur Performanceanalyse auf dem SAP NetWeaver AS Java	438
9.7	Zusammenfassung	439
9.8	Übungsfragen	441
10	Sperren	443
10.1	Sperrkonzepte von Datenbanksystem und SAP-System	444
10.1.1	Datenbanksperren	444
10.1.2	SAP-Enqueues	445
10.1.3	Lesekonsistenz	447
10.2	Überwachung von Sperren	449
10.2.1	Datenbanksperren	449
10.2.2	SAP-Enqueues	454
10.2.3	Lesekonsistenz	457
10.3	Nummernkreispufferung	458
10.3.1	Grundlagen	458
10.3.2	Nummernkreispufferung einschalten	462
10.3.3	Nummernkreispufferung überwachen	463
10.4	Sperren mit Menge und ATP-Server	465
10.4.1	Grundlagen	466
10.4.2	ATP-Server konfigurieren	467
10.4.3	ATP-Server überwachen	470

10.5 Zusammenfassung	471
10.6 Übungsfragen	471
11 Optimierung von SQL-Anweisungen	473
11.1 Teure SQL-Anweisungen identifizieren und analysieren	475
11.1.1 Identifizierung und Voranalyse	475
11.1.2 Detaillierte Analyse	477
11.2 SQL-Anweisungen im Programm optimieren	481
11.2.1 Die fünf goldenen Regeln der performanten SQL-Programmierung	482
11.2.2 Beispiel für die Optimierung eines ABAP-Programms	488
11.2.3 Vorbelegung von Feldern in Reporttransaktionen	497
11.3 SQL-Anweisungen durch Sekundärindizes optimieren	502
11.3.1 Ein wenig Theorie	502
11.3.2 Indizes und Tabellenstatistiken administrieren	512
11.3.3 Regeln zum Anlegen oder Ändern von Sekundärindizes	518
11.4 Zusammenfassung	527
11.5 Übungsfragen	529
12 SAP-Pufferung	531
12.1 Grundlagen der Tabellenpufferung	533
12.1.1 Pufferungsarten	533
12.1.2 Pufferzugriffe	536
12.1.3 Puffersynchronisation	538
12.1.4 Pufferung einschalten	541
12.1.5 Welche Tabellen sollen gepuffert werden?	543
12.2 Tabellenpufferung auf dem ABAP-Server überwachen	545
12.2.1 Tabellenzugriffsstatistik	546
12.2.2 Analyse der gepufferten Tabellen	550
12.2.3 Analyse der bisher nicht gepufferten Tabellen	554
12.2.4 Detaillierte Tabellenanalyse	556
12.2.5 Monitor zur Puffersynchronisation (DDLOG-Einträge)	559
12.2.6 Teure SQL-Anweisungen durch falsche Pufferung	560

12.3 Objektorientierte Anwendungspuffer überwachen	562
12.4 Zusammenfassung	568
12.5 Übungsfragen	569
13 Optimierung von Datenbankabfragen mit SAP HANA	571
13.1 Grundlagen der Hauptspeicherdatenbank SAP HANA	573
13.1.1 Spaltenorientierte Datenspeicherung	574
13.1.2 Datenkomprimierung	576
13.1.3 Datenhaltung im Hauptspeicher	580
13.1.4 Indizierung	582
13.1.5 Textsuche und Kategorisierung	585
13.2 SAP-HANA-Architekturüberblick	586
13.3 Datenmodellierung aus Performancesicht	589
13.3.1 Tabellen- und Indexdesign	589
13.3.2 Design von Datenbanksichten und SELECT-Anweisungen	591
13.3.3 Design von schreibenden Anwendungen	601
13.4 SAP-HANA-Administrationswerkzeuge zur Performanceoptimierung	602
13.4.1 Hauptspeicher- und CPU-Ressourcen analysieren	606
13.4.2 Identifizierung teurer SQL-Anweisungen	613
13.4.3 Analyse teurer SQL-Anweisungen: Ausführungsplan und Planvisualisierung	622
13.5 Technische Optimierungsoptionen im Detail	637
13.5.1 Tabellengröße, Lade- und Entladevorgänge überwachen	640
13.5.2 Deltaindizes überwachen	646
13.5.3 Indizes anlegen	651
13.5.4 SAP HANA Cached Views	654
13.5.5 Partitionierung und Skalierung	661
13.5.6 Datenreduktion durch Löschen oder Auslagern	666
13.5.7 Replikation	675
13.6 Zusammenfassung	677
13.7 Übungsfragen	679

14 Optimierung von Anfragen an SAP Business Warehouse 681

14.1 Grundlagen des SAP Business Warehouse 682

14.1.1 Übersicht über die wichtigsten Begriffe in SAP BW 683

14.1.2 InfoCube und erweitertes Star-Schema 687

14.1.3 DataStore-Objekte 698

14.1.4 Datenmodellierung aus Performancesicht 701

14.1.5 Ablauf einer Query-Ausführung 704

14.2 BW-Administrationswerkzeuge zur Performanceoptimierung 706

14.2.1 Analyse teurer BW-Anfragen 707

14.2.2 BW-Statistiken im Workload-Monitor 711

14.3 Technische Optimierungsoptionen im Detail 714

14.3.1 Indizierung und Datenbankstatistiken administrieren 717

14.3.2 Einstellungen zur Datenselektion 723

14.3.3 Komprimieren der Faktentabelle 725

14.3.4 OLAP-Cache überwachen 726

14.3.5 Optimieren von Reports 731

14.3.6 Aggregate 732

14.4 Zusammenfassung 741

14.5 Übungsfragen 743

Anhang 745

A Datenbankmonitore 747

B Wichtige Transaktionscodes 807

C Fragen und Antworten 811

D Glossar 817

E Informationsquellen 829

F Der Autor 849

Index 851

Index

- 64 Bit 260
64-Bit-Architektur 275
- A**
-
- ABAP 477, 479, 817
ABAP Class Builder 565
ABAP Debugger 223, 244, 257
ABAP Dictionary 105, 490, 512, 531, 562, 817
 Anzeige 808
 Tabelle 808
ABAP Editor 286, 499, 809
ABAP Objects 565
ABAP Shared Objects 283
ABAP Workbench 248, 450, 501, 809
ABAP-Applikationsinstanz 76
ABAP-Instanz 345
ABAP-Klassenbibliothek 809
ABAP-Laufzeitanalyse 236, 239, 257, 482
ABAP-Laufzeitanalyse für Web-Dynpro-
 Anwendungen 238, 372
ABAP-Laufzeitfehler 810
ABAP-Programm 481, 488
 Abbruch 281
 Laufzeitanalyse 223
 Qualitätsanalyse 247
ABAP-Server 133, 140
ABAP-Trace 68, 223, 236
ABAP-Trace-Variante 241
ACID-Prinzipien 817
Activation Queue 700
Active Data 700
Active Session History 796
Adaptive Server Enterprise → ASE
Address Space 260
Adressraum 260, 817
Agent 163, 767, 769
Agent Private Memory 767
Aggregat 479, 571, 590, 732, 739
 anlegen 735
 Funktion 483
 Indizierung 739
 komprimieren 741
 Pflege 740
 Vorschlag 737
Aggregation 241
Ähnlichkeitsmaß 586
Aktionsprofil 165
Aktivität → Durchsatz
ALE 817
ALE-Administration 807
Alert-Monitor 817
Allocation Analysis 413
Allocation Trace 440
Allokationsrate 418
ALV Control 337
American National Standards
 Institute 817
Analyse
 linguistische 25, 585
 logische 63
 technische 63
analytische Anwendung 682
Änderung, logische 150
Änderungs- und Transportsystem 541
Änderungsauftrag 541
Änderungslauf 741
Anfrage, analytische 580
Anmeldegruppe 301, 326
Anmeldegruppe, Pflege 810
Anmeldegruppen-Monitor 296
ANSI 817
Antwortzeit 57, 131, 143, 298, 748
Antwortzeit, mittlere 155
Antwortzeitverteilung 165
Anwendungsanalyse 131, 810
Anwendungspuffer, objektorien-
 tierter 531, 562, 568
Anwendungsserver → Applikationsserver
Anzeigeattribut 690
Appliance 218
Application Link Enabling → ALE
Application Support Layer 767
Applikationsanalyse 131, 810
Applikationsebene 42
Applikationsfehler 284
Applikationsoptimierung 20
Applikationsserver 76, 140, 230, 817
Applikationstuning 30, 64
Archiver Stuck 106, 150
Array Fetch 228

- ASE 761
Ausführungsplan 765
Datenbankprozess 763
Datenpuffer 762
Engine 763
ATP-Logik 443, 465
ATP-Server 296, 465–467
ATP-Service 292
Attribut 683, 702
Anzeigattribut 690
Navigationsattribut 690
Vektor 577, 583
Ausführungsplan 504, 622, 749, 817
IBM DB2 für z/OS 788
Microsoft SQL Server 803
Oracle 797
Auslagerungsspeicher 80, 114, 259,
275–276, 288, 784, 817
Automatic Workload Repository 796
Auxiliary Storage Pool 784
- B**
- B*-Baum 719
Backup 51, 56
Bandbreite 581
BAPI 817
Batch-Input 191, 817
Benchmark 196, 200, 826
Benutzer
aktiver 146
gelegentlicher 146
Power-Benutzer 146
Sachbearbeiter 146
Benutzerkontext 140, 261, 305, 818
Benutzerliste 809
Benutzermodus
Experte 135
Service Engineer 136
Benutzeroberfläche, grafische 819
Benutzerprofil 146, 165
Benutzerübersicht 321
Bestand 684
Betriebsart 308, 809, 818
Betriebssystem
Auslagerungsspeicher 281
Grenze 285
Kommando 809
Konfigurationsparameter 281
Paging 261
Parameter 86
Betriebssystem (Operating System) 823
Betriebssystemdatei 807
Betriebssystemmonitor 67, 77, 85, 203,
225, 342, 775, 810
Betriebssystemmonitor, Prozess-
übersicht 121
Betriebssystemprozess 82
Betriebszeit 55
Bewegungsdaten 543
BEx Analyzer 354, 686
bgRFC-Monitor 336
Big Data 24
binäre Suche 583, 750
Blade-Server 218
Blatt 719
Block 92
Browser 818
BSP-Anwendungen, Laufzeit-
analyse 238, 372
BSP-Entwicklung 375
Buffer Pool 767
Buffer Trace 224
Business Application Programming Inter-
face → BAPI
Business Objects 818
Business Process 50
Business Server Pages (BSP) 42,
349–350, 373
BW Administrator Workbench 707,
720, 724, 738, 808
BW Query Monitor 808
BW Query → BW-Anfrage
BWA → SAP BW Accelerator
BW-Aggregatpflege 808
BW-Analyse 707
BW-Anfrage 682, 686
BW-Prüfreport 808
BW-Statistik 709
BW-Workload-Statistik 706
- C**
- CA Wily Introscope 68, 440
Cache → Puffer
Calc Engine 588, 600
CALL-Befehl 262
Catalog Cache 752, 767
CBO → Cost-Based Optimizer (CBO)
CCMS 29, 35, 66, 515, 818
System Component Repository
(SCR) 164

- CCMS (Forts.)
Überwachungsmonitor 327
Change 547, 555
Change and Transport Organizing 818
Change Run 741
Change-Log 700
charakterisierender Parameter 132
Checkpoint 93
Class Statistic 440
Classloader 399
CLEAR-Anweisung 251
Client-Destination-Statistiksatz 331
Client-Server-Architektur 41
Client-Server-Architektur, Skalier-
barkeit 49
Client-Statistiksatz 331
Cloud-Anwendungen 30
Cluster-Coding 579
Clustered Index 803
Code Completion 614
Code Inspector 223, 247, 257, 538
Code Push-down 571
Codecache 406
Column Store → spaltenorientierte
Datenspeicherung
Common Programming Interface
Communication 818
Computer Center Management System →
CCMS
COM-Routine → Database Procedure
Concurrent-Mark-Sweep-Kollektor .. 402
Container 89
Control 291
CO-PA Accelerator 38
Core 80, 763
Cost-Based Optimizer (CBO) 508
CPI-C 818
CPU 49, 80, 818
Auslastung 79, 82, 104, 204
Bedarf 200, 344
Belastung 151
Engpass 81, 84, 104, 150
Ressourcen 298
Trace 236
Wartezeit 300
Zeit 142, 144, 148, 153, 161, 299
CTO 818
Cursor 229
Cursor Cache 260
Cursor-ID 229
Customer Interaction Center 247
Customizing 818
Customizing-Daten 544
- D**
- Data Buffer 92
Data Cache 92, 752, 801
Data Control Language (Datenkontroll-
sprache) 819
Data Definition Language (Daten-
definitionssprache) 819
Data Manipulation Language (Daten-
manipulationssprache) 819
Data Mining 25, 683
Data Warehouse 683
Database
Analyzer 758
Global Memory 767
Heap 767
Procedure 141
Procedure Call 141
Procedure Subrecord 141
Procedure-Zeit 141
Database Lock → Datenbanksperre
DataStore-Objekt 684, 698, 702
indizieren 722
SAP-HANA-optimiertes 700
Dateioperation 428
Datenarchivierung 818
Datenbank 76, 818
Administration 108
Analyse 75, 89
Fehlerprotokolldatei 103, 772
Last 486
Stillstand 106
Tasks 753
Datenbankadministrator-Cockpit → DBA-
Cockpit
Datenbankanfrage, Parallelisierung 574
Datenbankindex, fehlender 105
Datenbankinstanz 76, 278, 818
Datenbankmonitor 67, 89, 116, 450, 747
Datenbankobjekt, fehlendes 807
Datenbankoperation 228
Datenbankoptimierer 105, 505, 507,
524, 622, 817–818
Datenbankoptimierer, kosten-
basierter (CBO) 105
Datenbankperformance, Problem 152
Datenbankperformancemonitor 90, 810
Datenbankprozess 83, 614

Datenbankprozessmonitor 94, 121, 225, 438, 440, 753, 763, 776, 788, 793, 803
 Datenbankprozessor 104
 Datenbankpuffer 91, 573
 Datenbankserver 47, 76, 818
 Datenbankservice 344
 Datenbanksperre 101, 438, 444, 446, 449, 771, 818
 exklusive 102
 IBM DB2 für z/OS 788
 SAP MaxDB 757
 SQL Server 803
 Datenbanksperremonitor 121, 450
 Datenbanksystem 444, 818
 Datenbanksystem, paralleles 76
 Datenbanktabelle 681
 Datenbanktuning 30
 Datenbank-View 496, 572, 750, 760, 766, 773, 790, 799, 804
 Datenbankzeit 140, 148, 152–153
 Datenbankzeit, lange 123
 Datenbankzugriff, voll qualifizierter 227
 Datenfile 89
 Datenlokalität 662
 Datenmenge, übertragene 338
 Datenmodellierung, SAP HANA 589, 591
 Datenpaket 741
 Datenpuffer 752, 767, 787, 792, 801
 Daten-Volume 757
 DB2 → IBM DB2
 DBA 819
 DBA-Cockpit 69, 90, 122, 450, 513–514, 602, 747, 751, 776
 DBA-Planungskalender 514
 DBA-Protokoll 807
 DCL 819
 DDL 819
 Deadlock 453, 819
 Debugger 245, 406
 DELETE-Anweisung 251
 Deltaindex 579, 646, 651
 demilitarisierte Zone (DMZ) 44, 294
 Denkzeit 298
 Deoptimierung 406
 Dequeue-Baustein 446
 Destination 319, 323
 Development Workbench 35
 DIAG-Protokoll 819
 Dialog Step → Transaktionsschritt

Dialogantwortzeit 316
 Dialogbenutzer 190, 308
 Dialoglast 149
 Dialogservice 292, 296–297
 Dialog-Workprozess 140, 304, 311, 819
 Dictionary 583
 Dictionary-Coding 576
 Differenz-Coding 579
 Dimensions-ID (DIMID) 687
 Dimensionstabelle 687
 Dimensionstabelle, Index 720
 Direct Read 228, 750, 759, 765, 789
 direktes Lesen → Direct Read
 disp+work 292
 Dispatcher 296, 819
 Dispatcher-Prozess 292
 Dispatcher-Queue 126, 139
 Dispatcher-Wartezeit 139, 147, 153, 299
 Dispatching 292
 Distributed Statistic Record → Statistik-satz, verteilter
 Distributed Statistics Records (DSR) 163
 DML 819
 Dokumentation 64
 Dominator Tree 436
 Drill-down 685
 Dump 281
 Durchsatz 145, 155, 443
 dynamische Benutzerverteilung 301
 Dynamischer Statement Cache 787
 Dynpro 819

E

E2E-Trace → End-to-End-Trace
 EarlyWatch Alert → SAP EarlyWatch Alert
 Easy Web Transaction 377, 819
 Eclipse 819
 Eden 399, 401
 EDI 819
 EDM DSC Cache 787
 EDM-Pool 787
 E-Faktentabelle 726
 Einführungsleitfaden 728
 Einheitendimension 692
 Einkernprozessor 80
 Einzelsatz, statistischer 132, 176, 223
 Einzelsatzpuffer 531, 533

Einzelsatzstatistik 67, 138, 322, 339, 707, 810
 Einzelsatzstatistik, zentrale 164, 168
 Electronic Data Interchange 819
 End User Experience Monitoring 57, 361
 End-to-End-Diagnostik 70
 End-to-End-Laufzeitanalyse 168, 172, 174, 177
 End-to-End-Trace 242, 369
 Engpassanalyse 131, 300
 Enqueue 444, 819
 Enqueue-Baustein 446
 Enqueue-Operation 223
 Enqueue-Service 292, 295–296
 Enqueue-Trace 224, 235
 Enterprise JavaBeans 43
 Entität 819
 Entwicklerprotokoll 285
 Entwickler-Trace 810
 Ereignisdaten 25
 erweiterter Speicherbereich (EM) 283
 Eskalationsverfahren 59
 ETL-Prozess 687
 Event 374
 EWT 819
 Exclusive Lockwait 101, 110, 127, 130, 449, 471, 788, 803
 Execution Plan → Ausführungsplan
 Executive Information System (EIS) 681
 Expertenmonitor 66
 Export-Import-Puffer 467, 532, 562
 Export-Import-SHM-Puffer 283, 532, 562
 Extended Memory 285, 819
 Extended-Storage-Server 587
 Extensible Markup Language 827
 Extrahieren 687

F

Failover-Lösung 297
 Faktentabelle 687, 741
 Index 720
 komprimieren 725
 SAP HANA 695
 FDDI 819
 FE Net Time 338
 Fehlerbehebung 57
 Fehlercode
 DBIF_RSQL_NO_MEMORY 287
 EXPORT_NO_SHARED_MEMORY 283

Fehlercode (Forts.)
 EXSORT_NOT_ENOUGH_MEMORY 287
 PXA_NO_SHARED_MEMORY 282
 SET_PARAMETER_MEMORY_OVERFLOW 283
 STORAGE_PARAMETERS_WRONG_SET 282, 285, 287
 SYSTEM_NO_MORE_PAGING 272
 SYSTEM_NO_ROLL 282
 TSV_TNEW_PAGE_ALLOC_FAILED 282
 TSV_TNEW_PG_CREATE_FAILED 272
 Festplatte 581
 Festplattenzugriff 757, 803
 FETCH-Operation 228, 492, 547
 F-Faktentabelle 726
 Fiber Distributed Data Interchange .. 819
 File System Cache 85
 File- und Network-I/O-Analysis 428
 Files-Statistik 428
 Firewall 819
 Flow 684
 FOR ALL ENTRIES 492, 495
 Fragmentierung 557, 567
 FREE-Anweisung 250
 Frontend, Trace 172
 Frontend-Zeit 338
 Full Table Scan 505, 511, 519, 718, 749
 Function Builder 315
 funktionaler Trace → Einzelsatzstatistik, zentrale
 Funktionsbaustein 313

G

Ganzzahlidentifikator 576
 Garbage Collection 398–399, 439, 819, 825
 Analysis 430, 440
 Compacting 401
 JVM 399
 Kollektor 402
 Mark and Copy 401
 Mark and Sweep 401
 partielle 400
 Protokoll 405
 vollständige 400
 Gateway-Monitor 321
 Gateway-Service 292

Generation 399
 Generierungszeit 140
 Geschäftsprozess 50
 Global Cache Hit Ratio 787
 Globally Unique Identifier (GUID) 820
 GoingLive Check → SAP GoingLive Check
 GoingLive Functional Upgrade
 Service 208
 GUI 819
 GUI Control 336
 GUID 820
 GUI-Kommunikation 339
 GUI-Zeit 141, 161, 338, 341

H

HANA → SAP HANA
 Hardware 49
 Analyse 75, 77, 79
 Engpass 82, 150
 Kapazität 152
 Konsolidierung 213
 Landschaft 296
 Sizing 181–182, 468
 Tuning 30
 Hardwarepartner 182, 184
 Hash-Tabelle 251
 Hauptspeicher 49, 571, 573, 581
 Auslastung 80
 Bedarf 204, 277
 Engpass 81–82, 85, 150
 Profil 153
 Pufferung 459
 Sizing 192, 210
 virtueller 288
 Hauptspeicherkonfigurations-
 monitor 204
 Heap 261
 Heap Dump 435
 Heap Dump Analysis 440
 Heap Memory 820
 Hierarchie 692, 703
 High Availability 296–297
 High Water Mark 113
 Hint 524
 Hintergrundjob, Scheduling-Moni-
 tor 808
 Hintergrundlast 149
 Hintergrundprogramm 190, 452
 Hintergrundservice 292, 297
 Hintergrundverarbeitung 809, 820

Hitratio 92
 Hochverfügbarkeit 296, 820
 Hotspot 100, 419, 663, 803
 Hotspot Java Virtual Machine 401
 HPROF-Datei 435
 HTML 820
 HTML Business Template 377
 HTTP 820
 HTTP-Aufruf 372
 HTTP-Trace 234, 256, 369
 HybridProvider 684
 Hypertext Markup Language 820
 Hypertext Transfer Protocol 820

I

I/O-Engpass 86, 100, 230
 I/O-Operation 440
 IBM DB2 768
 für IBM i 775
 für Linux, UNIX und
 Windows 767–768
 für z/OS 786
 IBM i 775–776, 784
 ICM-Monitor 67, 128, 369–370
 IDES 820
 IDoc 820
 Implementation Guide 821
 Index 571, 591, 651, 733
 administrieren 512
 anlegen 512
 B*-Baum-Index 722
 B*-Index 583
 Bitmap-Index 719, 722
 effizienter 228
 fehlender 105
 Fragmentierung 516
 invertierter 582–583
 konkatenierter 582, 585
 Primärindex 105
 Qualität 517
 reorganisieren 516
 verwalten 512
 Index Range Scan 506, 511
 Index Scan 718
 Index Unique Scan 505
 Indexserver 587
 Indizierung 717
 InfoCube 684, 687, 702, 734
 HANA-Migration 808
 Optimierung 715

InfoCube (Forts.)
 Tabellen 693
 transaktionaler 720
 Übersicht 807
 InfoObject 683
 InfoProvider 684, 702, 727
 InfoSet 684
 In-Memory-Anwendung 38
 In-Memory-Computing 571, 574
 In-Memory-Datenbank 581
 Installation
 verteilte 296
 zentrale 296
 Instanz → SAP-Instanz
 Integer ID 576
 Integrität 51
 Inter Process Communication 821
 Interaktionsmodell 337
 Internal Document 820
 International Demo- and Education
 System 820
 Internet Communication Frame-
 work (ICF) 374, 377
 Internet Communication Mana-
 ger (ICM) 128, 292, 368–369, 373–374
 Internet Communication Manager,
 Monitor 810
 Internet der Dinge 24
 Internet Pricing and Configura-
 tion (IPC) 821
 Internet Transaction Server (ITS) 303,
 350, 377, 821
 integrierter 271, 377, 810
 Performanceanalyse 370
 Internetanbindung 349
 Interpreter 405
 Intranet 821
 Introscope-Statistik 163
 Invalidierung 112, 541
 IPC 821
 iSeries 784
 ITS → Internet Transaction Server (ITS)
 ITS-Monitor 67
 ITS-Status-Monitor 378

J

J2EE Engine → SAP NetWeaver Applica-
 tion Server Java
 Java 821
 Java Development Kit (JDK) 397

Java Dictionary 531
 Java Heap 399, 404, 440
 Java-Heap-Analyse 435
 Java Runtime 410
 Java Server Page 43, 349, 351
 Java Servlet 43, 349
 Java Virtual Machine ... 397, 440, 821, 827
 Speicherbereich 399
 Workprozess 407
 Java-Applikationsinstanz 76
 Java-Bytecode 405
 JavaScript 588
 Java-Statistik 163
 Java-Trace 68
 Java-Workload-Monitor 168
 JIT-Compiler 397–398, 405, 420
 Jobanalyse 809
 Jobübersicht 740
 Join 703, 718
 Join Engine 588, 600
 Just-in-time-Compiler 397, 405

K

Karenz 539, 548
 Kennzahl 132, 683
 Kommandomonitor 754
 Kompilat 405
 Komprimierbarkeit 590
 Komprimierung 573, 576
 Cluster-Coding 579
 Präfix-Coding 578
 Run-length-Coding 579
 Sparse-Coding 578
 Konditionstabelle 544, 561
 Kontextwechsel 80, 261, 822
 Kopplung
 Application Link Enabling 318
 harte 318
 weiche 318

L

Laden 687
 Ladezeit 140
 LAN 48, 822
 LAN-Check 87
 Landschaftsreplikation 588
 Large IO Pool 762
 Last 145, 155
 ausgehende 328

- Last (Forts.)
eingehende 328
externe Systeme 165
 Lastprofil 133
 Lastverteilung 133, 151, 291
 ABAP 295
 falsche 82, 124, 152
 Lastverteilungskonzept, neues 304
 Latenzzeit 353, 429, 581
 Laufzeit 419
 Analyse 132, 223
 konstante 256
 logarithmisch ansteigende 256
 Laufzeitanalyse 439
 Leaf 719
 Lesen
 satzweises 511
 sequenzielles ... 228, 511, 749, 758, 765,
 772, 788, 790, 798, 803–804
 Lesezugriff 465
 logischer 92
 physischer 92
 Linearität 192
 Line-Item-Dimension 697, 703
 Linux 273
 Local Memory 260
 Lock Handler 444
 Lock List 767
 Lock-Eskalation 757
 Log-Bereich 106, 758
 Logical Unit of Work (LUW) 334, 822
 Logistic Information System
 (LIS) 314, 681
 Logon-Gruppe → Anmeldegruppe
 Loop, geschachtelter 252
 Low Speed Connection 344
 LRU 822
- M**
- Mandant 822
 Mark-and-Copy-Kollektor 402
 Maschinencode 405
 Master-Service 664
 Mehrkernprozessor 80
 Mehrsprachigkeit 703
 Memory Analyzer 434, 440
 Memory Inspector 244, 246
 Memory Leak 435, 437, 440
 Memory Management 822
 IBM i 775, 784
- Memory Management (Forts.)
 Linux 273
 Memory Pipe 128, 376
 Mengengerüst 189
 Merge 580, 646
 Auto-Merge 647
 Critical-Merge 647
 Hard-Merge 647
 Smart-Merge 647
 Merkmal 683, 690, 702
 Message-Server 291–292, 297, 345
 Metadaten 91
 Method Parameter Trace 440
 Methode 405, 440
 Methode, Speicherbedarf 415
 Microsoft SQL Server 503
 Microsoft Windows 272, 300
 Migration 210
 Missing Index 513
 Model View Controller (MVC) 373
 Modularisierungseinheit 236
 Modus 262, 822
 externer 262
 interner 262
 PRIV-Modus 267, 280
 Modusliste (Mode List) 279
 Monitoring 50, 58
 Monitoring, zentrales 53
 Monitoring-Agent 168
 Monitoring-System, zentrales 164
 mooresches Gesetz 822
 MultiProvider 684–685, 702
 Multi-Threaded-CPU 80
 MVC 373
- N**
- Name-Server 587
 Nametab-Puffer 532
 Navigationsattribut 690, 703
 Nested Loop Join 497, 718, 766
 Net Time 338
 Network-I/O- und File-I/O-Trace 440
 Netzplangrafik 808
 Netzwerk 87, 229, 299
 Problem 229
 Tuning 30
 Netzwerkoperation 428
 Nummernkreis 458
 Intervall 458
 Nummernkreisstand 463

- Nummernkreis (Forts.)
 Objekt 458
 Puffer 466
 Pufferung 458–459, 462, 809
 Pufferungsmodus 464
- O**
- Object Linking and Embedding (OLE) 823
 OLAP 201, 682, 823
 OLAP Engine 588, 600
 OLAP-Cache 565, 704
 Konfiguration 728, 808
 Monitor 729–730, 808
 überwachen 726
 OLAP-Prozessor 704
 Old Generation 399, 403
 Old Generation, Kollektoren 402
 OLE 823
 OLTP 201, 682, 823
 Open Object 802
 OPEN-Operation 228
 Optimierer → Datenbankoptimierer
 Optimierung, technische 20
 Optimierungskonzept 49, 53
 Optimizer → Datenbankoptimierer
 Oracle 792
 Oracle Wait Event 794
- P**
- Package Cache 767
 Page-in 817
 Page-out 817
 Pages 92
 Paging 823
 Paging-Datei → Auslagerungsspeicher
 Paging-Rate 81, 85, 775
 Paketdimension 692
 Parallelisierung 574, 661
 Parameter, charakterisierender 132
 Parameteränderung 86
 Parameterpflege 369
 Parsen 505
 Partitionierung 312, 661, 665
 bereichsweise 662
 umlaufende 663
 PASE-Laufzeitumgebung 784
 Performance 50, 823
 Forum 73
 Management 33
- Performance (Forts.)
 Messung 337
 Performance Hotspot Analysis 419
 Performance Hotspot Trace 440
 Performancemanagement, pro-
 aktives 21
 Performancemenü 810
 Performanceoptimierung, konti-
 nuierliche 60
 Performanceproblem
 allgemeines 147
 dauerhaftes 148
 spezielles 147, 154
 temporäres 148
 Performance-Trace 223–224, 340, 713
 Permanent Generation 399, 404
 Pflege-View DBDIFFVIEW 514
 Pflichteingabefeld 499
 physischer Hauptspeicher
 (RAM) 275, 288
 Pivotieren 685
 Planungsanwendung 682
 Pop-up 823
 Präfix-Coding 578
 Präprozessor 587
 Präsentationsebene 41
 Präsentationsserver 42
 Präsentationsserver-Analyse 354
 PREPARE 229
 Prepared Statement 787
 Pre-parsed Template 377
 Primärindex 503, 513, 750
 Prioritätsklasse 305
 Private Mode 267
 PRIV-Modus 123, 269
 Procedure Cache 801
 Processing-Zeit 142, 144, 148
 Profil bearbeiten 808
 Profilparameter 281
 abap/atrapath 239
 abap/atrasizequota 239
 abap/heap_area_dia 266, 270,
 284, 287
 abap/heap_area_nondia 266, 270,
 284, 287
 abap/heap_area_total ... 266, 277, 285
 abap/heaplimit 154, 267
 db/io_buf_size 228
 Einstellung 286
 em/address_space_MB 273
 em/blocksize_KB 266

Profilparameter (Forts.)
em/initial_size_MB 114, 123, 266, 279, 285
em/max_size_MB 272
Pflege 808
PHYS_MEMSIZE 115, 264, 278
rdisp/atp_server 468
rdisp/bufrefmode 540
rdisp/bufreftime 540
rdisp/enqname 297
rdisp/max_wprun_time 302
rdisp/mshost 297
rdisp/PG_MAXFS 272
rdisp/PG_SHM 272
rdisp/ROLL_MAXFS 265, 285
rdisp/ROLL_SHM 114, 265
rdisp/vb_dispatching 313
rdisp/vbstart 311
rsdb/max_blocking_faktor 494
rsdb/obj/buffersize 468
rsdb/obj/max_objects 468
rstr/file 225
rstr/max_diskspace 225
stat/dbprocrec 142
ztta/roll_area 265, 270
ztta/roll_extension 266, 270, 280, 284, 289
ztta/roll_extension_dia 266, 280
ztta/roll_extension_nondia ... 266, 280
ztta/roll_first 267–268, 270
Program Counter 80
Program Global Area (PGA) 792
Programm
 Abbruch 281
 Fehler 281
 Puffer 282, 532
 RSCOLLOO 138
 SAP_COLLECTOR_FOR_PERFORMANCE 138
 saposcol 79
Promotion 399
Prozess
 beendeter 124
 externer 84
 haltender 124
Prozess-ID 121
Prozesskette 687
Prozessor 80, 209, 822
Prozessor-Thread 346
Prüfgruppe 468
PTF-Paket 778
Puffer 46, 91, 531
 Arten 531
 Betriebssystempuffer 46
 Datenbankpuffer 46
 Datenpuffer 92
 Einstellung 112
 Einzelatzpuffer 536
 Katalogpuffer 767
 Kommunikationspuffer 767
 Management 536
 Metadatenpuffer 763
 *objektorientierter Anwendungs-
 puffer* 46
 Package-Puffer 767
 Prozesspuffer 762
 Pufferhierarchie 46
 Qualität 92, 108
 Speichersubsystem 46
 Status 548
 Synchronisation 538, 807
 Synchronisationsmonitor 559
 Tabellenpuffer 536
 TABL 536
 TABLP 536
 Trace 224, 232
 Zugriff 536
Pufferung → SAP-Pufferung

Q

quadratische Abhängigkeit 256
Qualitätsanalyse 247
Quick Sizer 188, 288
Quick-Sizer-Projekt 193

R

R/3 823
Radio-Frequency Identification (RFID) 24
RAID 823
Rangliste 586
RBO → Rule-Based Optimizer (RBO)
RDBMS 823
Read Random Hit Ratio 787
READ TABLE 251
READ TABLE .. WITH KEY ... 251
Reaktionszeit 57
Rechner 76, 121
Rechnerprofil 152
Recoverability 51
Recovery 56

Recursive Call 793
Redo-Log-Datei 150
Region, generische 534
Reisepass 162, 168, 826
Relational Database Management
 System 823
relationale Datenbank 36
Remote Function Call (RFC) 42, 191, 304, 823
 ABAP-Coding 319
 Ablauf 321
 asynchroner (arRFC) 303, 319, 333, 809
 Grundlagen 317
 Hintergrund-RFC (bgRFC) 319, 336
 queued (qrRFC) 319
 RFC-Zeit 323
 Statistiken 331
 synchroner 319
 transaktionaler (trRFC) 319, 334, 827
RemoteCube 727
REOPEN-Operation 228
Reorganisation 665
Replikation 588
Reporting, Vorberechnung 731
Repository Browser 809
Repository-Infosystem 809
Request 547, 555
Ressourcenbedarf 182, 192
Ressourcenmonitor 756
RFC → Remote Function Call (RFC)
RFC-Aufruf 223
RFC-Client-Profil 331
RFC-Trace 161, 224
RFC-Verbindung 323, 809
RFC-Verbindung überwachen 327
RFID 823
Roll Memory → SAP Roll Memory
Roll-in 139, 263, 822
Roll-out 139, 263, 822
Roll-Puffer 278
Roll-up 685
Roll-Wartezeit 141, 161, 322, 338
Roundtrip 338, 429
Row Cache 793
Row Store 588, 644
Row-ID 504
Rule-Based Optimizer (RBO) 507, 509
Run-length-Coding 579

S

SAP Application Performance Stan-
dard 198
SAP Business Suite
 Architektur 33
 Einführung 213
 SAP HANA 37–38
SAP Business Warehouse ... 133, 354, 565, 681, 823
 Administrationswerkzeuge 706
 Datenselektion 723
 Frontend 686
 hohe Datenbankzeiten 712
 Indizierung 717
 Lastprofile 712
 Optimierung 714
 Performanceoptimierung 701
 SAP HANA 37
SAP BusinessObjects 686
SAP BusinessObjects Analysis 686
SAP BusinessObjects Business Intel-
ligence 686
SAP BusinessObjects Dashboards 686
SAP BusinessObjects Explorer 687
SAP BusinessObjects Web Intel-
ligence 686
SAP BW 35
SAP BW Accelerator 37, 572, 706, 823
SAP BW Accelerator, Monitor 808
SAP BW Administrator Workbench ... 706
SAP Crystal Reports 686
SAP Cursor Cache 229
SAP Customer Relationship Manage-
ment (SAP CRM) 821
SAP EarlyWatch Alert 25, 59, 72, 202, 286
SAP Enterprise Portal 35
SAP Extended Global (EG) Me-
mory 271, 377
SAP Extended Memory 113, 266, 268, 271, 278–279, 288
SAP GoingLive Check 25, 183–184, 202, 286
SAP GoingLive Functional Upgrade
 Check 203
SAP GoingLive Migration Check 203
SAP GUI 352, 823
 Controls 291, 336
 for HTML 352
 for Java Environment 41, 352
 for Windows 41, 352

SAP HANA 488, 571, 823
Administration 697
Datenbankplattform 572
Hauptspeicherbereich 606
Indizierung 582
Komprimierung 576
Sizing 192
Skalierbarkeit 488
Skalierung 49
SAP HANA Studio 68, 602
SAP Heap Memory 113, 266, 285, 288
SAP Host Agent 776
SAP Internet Transaction Server (SAP ITS) 42, 821
SAP Java Virtual Machine (JVM) 398
SAP Java Virtual Machine Profiler 409, 435, 440
SAP List Viewer 241
SAP List Viewer Control 337
SAP liveCache 35, 133, 748, 822
SAP Logical Unit of Work 445
SAP Management Console 439
SAP MaxDB 503, 751
Ausführungsplan 758
Kommandomonitor 754
Ressourcenmonitor 756
SAP NetWeaver
Release 7.40 538
Release 7.50 31
SAP NetWeaver Administrator 66, 68
SAP NetWeaver Administrator,
SQL-Trace 438
SAP NetWeaver Application
Server 42, 350
SAP NetWeaver Application Server
ABAP 76
SAP NetWeaver Application Server
Java 35, 76, 397
Serverknoten 410
SQL-Trace 438
SAP NetWeaver Application Server,
Services 292
SAP NetWeaver Enterprise
Search 35, 824
SAP Paging Memory ... 270–271, 567, 823
SAP Process Integration 35
SAP R/3 22
SAP Roll Memory 113, 285, 288
SAP Service Marketplace, *Service* 71
SAP Solution Manager 134, 163, 172, 178, 242, 361, 748
Analyse 70
Performanceoptimierung 71
Service Level Management 59
Tracing 70
Überwachung 69
Workload-Analyse 70
SAP Standard Application Benchmark 181, 196
SAP Support Portal 288
SAP System Identifier 824
SAP Web Dispatcher 44, 291–293, 303
SAP-Anwendung, *Architektur* 33
SAP-Applikationsinstanz → SAP-Instanz
SAP-Basis 35
SAPCCMSR 168
SAP-Client-Plug-in 172, 177
SAP-Easy-Access-Menü 344
SAP-Einführungsleitfaden (IMG) 810
SAP-Enqueue 445, 454, 465
SAP-Enqueue-Monitor 470
SAP-GUI-Transaktion, *End-to-End-*
Laufzeitanalyse 174
SAP-Instanz 76, 136, 216, 219, 278, 821
Pflege 808
Übersicht 809
SAP-Kernel 285
SAP-Komponente 34
SAP-Lösung 34
SAP-Lösungslandschaft 42
SAP-Paging 261
SAP-Parameter, *Änderung* 117
SAP-Performance-Trace 223
SAP-Puffer 111, 141, 278, 288
Trace 225
Zugriff 223
SAP-Pufferung 486, 531, 534
Art 533
einschalten 541
generische 534
vollständige 534
SAP-Roll-Bereich
globaler 265
lokaler 265
SAP-Roll-Datei 265
SAP-Roll-Puffer 265
SAProuter 824
SAPS 198, 824
SAP-Service 291, 296
SAP-Speicherbereich 277

SAP-Speicherkonfiguration 110, 123
SAP-Speicherkonfigurations-
monitor 67, 110
SAP-Speicherverwaltung 153
SAP-System 42, 77
SAP-Systemservice 824
SAP-System-Trace 810
SAP-Transaktion 445
SAP-Workprozess 83, 102, 119, 278, 288, 752
SAP-Workprozess, *Übersicht* 119
Savepoint 93
Schaltfläche 824
Schattenprozess 794
Schnittstelle 291
Schreib/Lese-(I/O-)Problem 85, 100, 771
SD-Benchmark 196
Sekundärindex 502, 504, 512, 514
ändern 518
anlegen 518
SELECT
identischer 231
verschachteltes 491
SELECT*-Klausel 492
Selektionsbildschirm 498
Selektivität 508, 510, 519
Semaphor 122
Sensordaten 24
Sequential Read → Lesen, *sequenzielles*
sequenzielles Lesen → Lesen, *sequenzielles*
Lesen 122
Serialisierung 77, 824
Server 77, 824
Server, *Konsolidierung* 213
Server-Destination-Statistiksatz 331
Service 295
Service Level Agreement 54
Service Level Management 53, 72, 824
Service Level Reporting 54, 56, 59, 73
Servicepflege 238, 368, 372
Session Monitor 793
Shared Cursor Cache 793, 795
Shared Memory 260, 282, 376, 466, 565, 767, 824
Shared Object 562
Gebietskonfiguration 565
Gebietsmonitor 566
Pufferung 532, 565
Shared Pool 792–793
Shared SQL Area 795
Shared SQL Cache 795
Sicherheit 51
SID 690, 824
SID-Tabelle 691
Simulationsanwendung 682
Single Level Storage 784
Single Point of Failure (SPOF) 297
Sizing 181, 192
benutzerbasiertes 189
durchsatzbasiertes 189
initiales 183, 187
Sizing Plausibility Check 184
T-Shirt-Sizing 188
Sizing-Projekt durchführen 193
Sizing-Prozess 182
Skalierbarkeit 192, 665, 825
horizontale 49, 825
Programm 253, 255
vertikale 49, 825
Skeleton 787
Slave-Service 665
Slice & Dice 685
Socket-Statistik 428
Solution Monitoring 50
Sonde 438
soziales Netzwerk 24
spaltenorientierte *Datenspeicherung* 571, 573–574, 588, 640
Sparse-Coding 578
Speicher
allokierter 115, 607
Belegung 267
Bereich 275, 288
fest allokierter (HEAP) 284
genutzter 607
globaler 260
Konfiguration 110, 259
lokaler 260, 822
physischer 116, 607
residenter 607
verfügbarer 276
Verwaltung 825
virtuell benötigter 276
virtueller 259, 607, 827
Speicherabzug 246
Speicherkonfiguration 114
Speicherkonfigurationsmonitor 110, 279, 285, 470, 553, 564, 567, 728, 810
Speicherverwaltung 153
Speicherverwaltung, *integrierte* 399
Sperrung 425, 440, 443
Datenbanksperre 444

Sperre (Forts.)	Statistikserver	666
<i>SAP-Enqueue</i>	statistischer Einzelsatz	132, 176, 223, 546
<i>Shared-Sperre</i>	Stemming	585
<i>Sperren mit Menge</i>	Stock	684
<i>Sperrkonflikt</i>	Stop the World	401
<i>Sperrkonzept</i>	Stored Procedure	572, 826
<i>Sperrobjekt</i>	Structured Query Language	826
<i>Tabellensperre</i>	SUBMIT-Befehl	262
<i>Überwachung</i>	Subrecord	141
<i>Wartesituation</i>	Suite Accelerator	38
Spool-Service	Support Package	826
SQL	Survivor	401
<i>Native SQL</i>	Swap	112, 541, 545
<i>Open SQL</i>	Swap Space → Auslagerungsspeicher	
SQL Editor	Sybase	826
SQL Plan Cache	Synchronization Analysis	425
SQL Server	Synchronization Trace	440
SQL Server, Primärindex	System Global Area (SGA)	792
SQL-Anweisung	Systemlandschaft	212, 826
502, 519, 521, 622, 749, 754, 793–794	Systemlastanalyse, globale	164
<i>dynamische</i>	Systemlastmonitor → Workload-Monitor	
<i>optimieren</i>	Systemlog	809
<i>teure</i>	System-Monitoring	52
<i>verschachtelte</i>	Systemnachricht versenden	809
SQL-Anweisungspuffer	Systempuffer	532
SQL-Coding	Systemverfügbarkeit	297
SQL-Programmierung		
<i>goldene Regeln</i>		
<i>performante</i>		
SQLScript		
SQL-Statistik		
95, 438, 475, 477, 480,		
526, 560, 617, 796, 810		
<i>Analyse</i>		
<i>DB2</i>		
<i>IBM DB2 für z/OS</i>		
<i>SAP MaxDB</i>		
<i>SQL Server</i>		
SQL-Trace		
68, 159, 224–225, 230, 256,		
440, 475–476, 479, 482, 488, 561, 810		
Stack Pointer		
Stammdaten		
Stammdatentabelle indizieren		
Stammformerkennung		
Star-Join-Ausführungsplan		
Star-Schema		
Star-Schema, erweitertes		
Statement Cache		
Statement String		
Statistik, abgeleitete		
Statistiksatz		
157, 162, 341, 826		
Statistiksatz, verteilter		
162–163, 168		

T

Tabelle	
<i>Analyse</i>	556, 807
<i>ARFCSDATA</i>	334
<i>ARFCSSTATE</i>	334
<i>ATAB</i>	562
<i>DOIO*</i>	562
<i>DOIOS</i>	561
<i>DO20*</i>	562
<i>DBDIFF</i>	514
<i>DBSTATC</i>	515
<i>DDLOG</i>	539, 559, 565
<i>DDNTF</i>	555, 562
<i>DDNTT</i>	555, 562
<i>gepufferte</i>	550, 554
<i>Hash-Tabelle</i>	251
<i>interne</i>	244, 836
<i>KAPOL</i>	549, 562
<i>Konditionstabelle</i>	544
<i>kundeneigene</i>	552, 556
<i>NRIV</i>	449, 458, 464
<i>NRIV_LOKAL</i>	460
<i>reorganisieren</i>	516

Tabelle (Forts.)	Transaktion (Forts.)
<i>RESB</i>	<i>DB12</i>
64, 465, 517	91, 807
<i>SAP-eigene</i>	<i>DB13</i>
552, 556	91, 807
<i>sortierte</i>	<i>DB13C</i>
251	91
<i>SWNCMONI</i>	<i>DB20</i>
138	807
<i>TCURR</i>	<i>DB21</i>
557	515
<i>tRFC-Tabelle</i>	<i>DB24</i>
335	91
<i>VBBE</i>	<i>DBACOCKPIT</i>
465	67, 69, 90, 121, 450,
<i>VBDATA</i>	513–514, 602, 748, 751, 807, 819
312	<i>DWDM</i>
<i>VBHDR</i>	337
312, 555	<i>kritische</i>
<i>VBMOD</i>	56
312	<i>LISTCUBE</i>
<i>Verbuchungstabelle</i>	707, 807
312, 316	<i>ME57</i>
Tabellengröße	498
Tabellenoperation	<i>OSO6</i>
236	77
Tabellenpflege	<i>OSO6N</i>
514, 654	77
Tabellen-Pool	<i>OSO7</i>
<i>ATAB</i>	77
561	<i>OSO7N</i>
<i>KAPOL</i>	77
561	<i>OSSI</i>
561	808
<i>RSAI</i>	706–707, 709, 720, 738, 808
160, 531	<i>RSCACHE</i>
<i>generischer (TABL)</i>	808
531	<i>RSCUSTV14</i>
<i>partieller (TABLP)</i>	808
531	<i>RSDDBWAMON</i>
533	808
545	<i>RSDDV</i>
545	706, 808
105, 509, 514, 808	<i>RSMIGRHANADB</i>
546, 550,	697, 808
555, 810	<i>RSODSO_SETTINGS</i>
89	808
74	<i>RSRCACHE</i>
136	729
826	<i>RSRT</i>
784	706, 708, 713, 727, 808
579	<i>RSRV</i>
337	808
79–80, 440, 614, 752, 763	<i>RZO1</i>
80	808
80	<i>RZO2</i>
408	808
617	<i>RZO3</i>
420	808
826	<i>RZO4</i>
172	808
174	<i>RZIO</i>
830	808
445, 826	<i>RZ11</i>
807	369, 808
807	<i>RZ12</i>
223	808
807	<i>RZ20</i>
807	68, 327, 808
91, 102, 450, 471	<i>SAT</i>
91, 509, 807	236, 250
557, 807	<i>SBGRFCCONF</i>
	336
	<i>SBGRFCMON</i>
	336
	<i>SCII</i>
	248
	<i>SE11</i>
	490, 512–513, 808
	<i>SE12</i>
	235
	<i>SE14</i>
	512, 808
	<i>SE15</i>
	809
	<i>SE16</i>
	709, 809
	<i>SE24</i>
	248, 809
	<i>SE30</i>
	68
	<i>SE37</i>
	248, 315
	<i>SE38</i>
	248, 286, 450, 499, 809
	<i>SE80</i>
	375, 501, 809
	<i>SEU</i>
	809

- Transaktion (Forts.)
- SHMA 565
 - SHMM 566
 - SICF 238, 368, 372
 - SITSMON 809
 - SITSPMON 67, 378
 - SM01 809
 - SM02 809
 - SM04 321, 809
 - SM12 456, 470, 809
 - SM13 107, 123, 312, 809
 - SM21 107, 123, 267, 809
 - SM24 565
 - SM30 514
 - SM36 809
 - SM37 722, 740, 809
 - SM49 809
 - SM50 67, 119, 245, 285, 296, 315, 321, 370, 451, 753, 809
 - SM51 77, 119, 126, 809
 - SM56 463, 809
 - SM58 335, 809
 - SM59 323, 325, 327, 335, 809
 - SM63 809
 - SM65 809
 - SM66 119, 122, 296, 315, 321, 370, 753, 809
 - SM69 809
 - SMGW 321
 - SMICM 67, 128, 369–370, 810
 - SMLG 125, 296, 302, 810
 - SNRO 462
 - sperrn 809
 - SPRO 468, 810
 - SQLM 810
 - ST01 810
 - ST02 ... 67, 110, 204, 276, 279, 283, 285, 470, 553, 564, 810
 - ST03 ... 67, 133–134, 164, 296, 322, 328, 370, 706, 709, 711, 810
 - ST03G 67, 163–164
 - ST03N 134, 205
 - ST04 91, 116, 810
 - ST05 224, 340, 475, 810
 - ST06 67, 77, 82, 85, 100, 116, 121, 203, 276, 342, 775, 810
 - ST06N 77
 - ST07 810
 - ST10 810
 - ST11 810
 - ST14 810
- Transaktion (Forts.)
- ST22 281, 810
 - STAD 133, 157, 164, 322, 331, 339
 - STAD/STATS 810
 - STAT 546
 - STATTRACE 163–164
 - STMS 810
 - STUN 66, 810
 - SWLT 100, 810
 - SXMB_MONI 810
 - TREXADMIN 122, 810
 - TU02 810
 - VA01 155, 158
 - Variante 501
 - VLOIN 158
 - Transaktionscode 827
 - Transaktionsprofil 154
 - Transaktionsschritt 137–138, 261
 - Transformieren 687
 - Transmission Control Protocol/Internet Protocol 826
 - Transport 827
 - Transport Management System 810, 826
 - Tree 719
 - Tree Control 337
 - Trefferrate 92
 - TREX 35, 823, 827
 - Administration 810
 - Komprimierung 576
 - Suchfunktionen 585
 - T-Shirt-Sizing 188
 - Tune Summary 110
 - Tuning 528
 - Applikationstuning 60, 73
 - Programmoptimierung 61
 - technisches 60, 64
 - Tupel-Rekonstruktion 576
- U**
- Überwachungskonzept 49, 53
 - Unicode 207, 260, 278
 - Unicode-Konvertierung 208
 - Uniform Resource Locator 827
 - UNIX 300
 - UNIX-Standardimplementierung 273
 - Untersatz 133
 - Upgrade 207
 - URL 827
 - User Call 793

- User Kernel Thread (UKT) 752
 - User-Exit 188
- V**
- Value-Per-Unit 684
 - Variante 241
 - Verbuchung 107, 310, 345
 - asynchrone 311
 - Auftrag 311
 - deaktivierte 123
 - Dispatching 313
 - lokale 315
 - Prioritätensteuerung 314
 - Service 296
 - synchrone 315
 - Tabelle 310
 - V1-Verbuchung 313
 - V2-Verbuchung 313
 - V3-Verbuchung 313
 - Verbuchungsart 313
 - Verbuchungszeit 316
 - Workprozess 311–312
 - Verbuchungssatz 809
 - Verbuchungsservice 292
 - Verdrängung → Swap
 - Verfügbarkeit 50, 165, 296
 - Verfügbarkeitsprüfung 465, 564
 - Verifikation 64
 - verteilter Statistiksatz 163
 - Virtual Machine Container (VMC) 292, 827
 - virtueller Hauptspeicher 288
 - virtueller Provider 684
 - virtueller Speicher 259
 - VMC-Service 292
- W**
- WAN 827
 - Web Dynpro 827
 - Web Dynpro ABAP 42, 238, 349–350, 373
 - Web Dynpro Java 349, 351
 - Web GUI 352
 - Web Reporting 686
 - Webanwendung 352, 360
 - Webbrowser 818
 - Webservice 42, 191, 351
 - Webservice, Aufrufe überwachen 372
 - Webtransaktion 172
 - WHERE-Klausel 485
 - Wide Area Network (WAN) 48, 827
 - Windows 272, 300
 - Workgroup 301
 - Workload-Analyse 25, 61, 131–132, 138, 147, 439–440
 - Workload-Monitor 67, 133, 135, 138, 153, 205, 296, 322, 370, 707, 810
 - BW-Systemlast 711
 - Java 168
 - Rolle 135
 - zentraler 164
 - Workload-Übersicht 165
 - Workprozess 122, 125, 262, 292, 298, 344, 827
 - ABAP-Trace 237
 - Anzahl erhöhen 346
 - Java Virtual Machine 407
 - Nicht-Dialog-Workprozess 269
 - Workprozess-Monitor 315, 370
 - Workprozess-Typ 136
 - Workprozess-Übersicht 67, 94, 119, 225, 245, 285, 296, 321, 451, 753, 809
 - Workprozess-Übersicht, globale 809
 - WP → Workprozess
 - WWW 827
- X**
- XML 827
 - XS-Server 588
- Y**
- Young Generation 399, 403
- Z**
- zeilenorientierte Datenspeicherung 573
 - Zeitabhängigkeit 691, 703
 - Zeitdimension 692
 - Zeitprofil 165
 - zentrale Einzelsatzstatistik 168
 - zentraler Überwachungsmonitor 68, 808
 - zentrales Monitoring-System 164
 - Zero Administration Memory Management 114, 264, 275, 289
 - Zugriffsplan 770



Thomas Schneider

SAP-Performanceoptimierung

Analyse und Tuning von SAP-Systemen

867 Seiten, gebunden, 8. Auflage, Juli 2017

79,90 Euro, ISBN 978-3-8362-4479-4

 www.sap-press.de/4330

Dr. **Thomas Schneider** arbeitet seit 1996 bei SAP. Unter anderem leitete er im Bereich Service & Support das Center of Expertise für Performance und betreute als Support Alliance Manager Key Accounts. 2004 wechselte er als verantwortlicher Architekt für IT Service & Application Management in den Bereich Research & Breakthrough Innovation, seit 2009 verantwortet er in gleicher Rolle das Thema Partner Development Infrastructure in der SAP-Cloud-Organisation.

Wir hoffen sehr, dass Ihnen diese Leseprobe gefallen hat. Gerne dürfen Sie diese Leseprobe empfehlen und weitergeben, allerdings nur vollständig mit allen Seiten. Die vorliegende Leseprobe ist in all ihren Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und beim Verlag.

Teilen Sie Ihre Leseerfahrung mit uns!

