

Für die Performanceanalyse von ABAP-Programmen gibt es eine Vielzahl an Werkzeugen. In diesem Kapitel stellen wir Ihnen die Werkzeuge vor und zeigen Ihnen, wie Sie diese effizient einsetzen können. Dazu gehen wir auf die wichtigsten Messergebnisse ein und erklären diese im Detail.

8 Werkzeuge für die Performanceanalyse

Das SAP-System bietet statische und dynamische Werkzeuge zur Performanceanalyse von ABAP-Programmen, wie z. B. den ABAB Code Inspector, Performance Trace oder Single Activity Trace. Zusätzlich gibt es ein Werkzeug, das statische mit dynamischen Prüfungen verknüpft, die SQL Performance Tuning Worklist. Wir stellen diese Werkzeuge in den folgenden Abschnitten vor:


- ▶ Abschnitt 8.1, »Statische Prüfungen«
- ▶ Abschnitt 8.2, »Dynamische Prüfungen und Traces«
- ▶ Abschnitt 8.3, »Kombinierte Auswertung statischer und dynamischer Prüfungen (SWLT)«

In jedem Abschnitt zeigen wir Ihnen die Bedienung der Werkzeuge und gehen auf die Auswertung im Detail ein.

8.1 Statische Prüfungen

Bei statischen Prüfungen wird das Programm analysiert, ohne dass es ausgeführt werden muss. Sie benötigen also zur Durchführung der Prüfung keine (produktiven) Daten. Daher können Sie mit statischen Performanceprüfungen beispielsweise schon während des Entwicklungsprozesses potenzielle Performanceprobleme erkennen und vermeiden. Andererseits wird bei solchen Analysen nicht festgestellt, ob das Coding überhaupt ausgeführt wird. Im Folgenden beschreiben wir die Performanceprüfungen des Code Inspectors.

8.1.1 Code Inspector (SCI)

In Kapitel 5, »Werkzeuge für die Qualitätssicherung«, haben wir Ihnen das Konzept und die Bedienung des Code Inspectors als Werkzeug zur Qualitätssicherung beschrieben. Darüber hinaus enthält insbesondere der Code Inspector eine Reihe von statischen Performanceprüfungen, mit deren Hilfe schon während der Entwicklungsphase viele typische Performanceprobleme (siehe Kapitel 7, »Typische Performanceprobleme und Lösungen«) erkannt werden können. Die meisten Prüfungen sind konfigurierbar und können flexibel angepasst werden. Zu jeder Prüfung erreichen Sie über den Button INFO  eine ausführliche Beschreibung und Tipps zur Vermeidung des Problems.

Performanceprüfungen im Code Inspector

In der Transaktion SCI können Sie eine Prüfvariante für Performanceprüfungen definieren und die folgenden Prüfungen aktivieren.

Analyse der WHERE-Bedingung für SELECT

Zur Auswahl stehen die folgenden Einzelprüfungen:

- ▶ keine WHERE-Bedingung
- ▶ kein Indexfeld in WHERE-Bedingung
- ▶ kein 1. Indexfeld in WHERE-Bedingung
- ▶ CLIENT SPECIFIED ohne Mandant

Analyse der WHERE-Bedingung für UPDATE und DELETE

Es stehen dieselben Einzelprüfungen wie beim SELECT zur Verfügung.

- ▶ **SELECT-Anweisungen, die am Tabellenpuffer vorbei lesen**
Bestimmte Zugriffe auf gepufferte Tabellen gehen am Puffer vorbei direkt auf die Tabelle und sollten aus Performancegründen vermieden werden. Diese haben wir in Abschnitt 7.1.3, »Identische Datenbankzugriffe«, beschrieben.
- ▶ **Problematische SELECT *-Anweisungen suchen**
SELECT *-Anweisungen werden z. B. erkannt, wenn nur wenige Felder einer breiten Tabelle tatsächlich genutzt werden oder die Größenkategorie einer Tabelle einen Mindestwert überschreitet. Die Schwellenwerte sind einstellbar. Zugriffe auf gepufferte Tabellen können ausgeschlossen werden.

- ▶ **Nach zu transformierenden SELECT .. FOR ALL ENTRIES-Klauseln suchen**

Findet SELECT ... FOR ALL ENTRIES-Anweisungen, die alternativ durch einen JOIN ausgeführt werden können.

- ▶ **Nach SELECT-Anweisung mit DELETE-Anweisung suchen**

Erkennt SELECT-Anweisungen, deren Ergebnismenge durch ein anschließendes DELETE FROM <itab> gelöscht wird.

- ▶ **DB-Operationen in Schleifen über Modularisierungseinheiten suchen**

Erkennt Tabellenzugriffe in Schleifen, auch über Modularisierungseinheiten hinweg. Unter anderem kann die Schachteltiefe definiert werden.

- ▶ **Ändernde Datenbankzugriffe in Schleifen**

Findet INSERT-, UPDATE-, MODIFY- oder DELETE-Anweisungen in Schleifen.

- ▶ **Geschachtelte Schleifen**

Meldet, wenn zwei Schleifen ineinander geschachtelt sind, z. B. LOOP...ENDLOOP, SELECT...ENDSELECT usw.

- ▶ **EXIT oder keine Anweisung in SELECT...ENDSELECT-Schleife**

SELECT...ENDSELECT-Schleifen, die leer sind oder nur die Anweisung EXIT enthalten (z. B. zur Existenzprüfung eines Datensatzes), werden hier erkannt. Wahlweise können Einzelsatz- oder gepufferte Zugriffe weggelassen werden.

- ▶ **SELECT-Anweisungen mit anschließendem CHECK**

Meldet CHECK-Anweisungen, die direkt auf ein SELECT folgen und die Ergebnismenge des SELECT einschränken.

- ▶ **Inperformante Operationen auf internen Tabellen**

Erkennt sequenzielle Suchoperationen auf internen Tabellen, z. B. bei (unvollständigem) Schlüsselzugriff. Dabei wird zwischen den verschiedenen Tabellenarten unterschieden, die wir in Abschnitt 7.2, »Zugriffe auf interne Tabellen«, beschrieben haben. Die Onlinenhilfe enthält einige Verbesserungsvorschläge zur Optimierung der Zugriffe.

- ▶ **SORT-Anweisung in Schleife**

Die SORT-Anweisung sollte möglichst nicht innerhalb einer

Schleife ausgeführt werden und wird mit dieser Prüfung gemeldet.

► **Kopieren der aktuellen Tabellenzeile bei LOOP AT ...**

Findet LOOP AT <itab> oder ...INTO <wa>-Anweisungen. Aktuelle Tabellenzeilen sollten nicht in einen Arbeitsbereich oder die Kopfzeile kopiert werden. Stattdessen sollten Feldsymbole oder Referenzvariablen genutzt werden, um Kopierkosten zu vermeiden.

► **Inperformante Parameterübergaben**

Parameterübergaben z. B. an Methoden oder Funktionsbausteine können als Referenz- oder Wertübergabe deklariert werden, wobei Wertübergaben deutlich teurer sind und hier erkannt werden. Es kann nach dem Typ der übergebenen Werte unterschieden werden, z. B. geschachtelten internen Tabellen, Strings oder langen Feldern.

► **Kopieren großer Datenobjekte**

Findet Zuweisungen mit = oder MOVE, bei denen große Datenmengen kopiert werden müssen, wie z. B. bei geschachtelten internen Tabellen oder breiten Strukturen (>1000 Byte).

► **Prüfung der Tabelleneigenschaften**

Es werden die technischen Einstellungen von Datenbanktabellen und Indizes bezüglich der Performance untersucht. Gemeldet werden Einstellungen, die die Performance verschlechtern können.

► **Instanzerzeugung von BAdIs**

Es wird geprüft, ob die neue (GET BADI...) oder alte (CALL METHOD cl_exithandler=>get_instance) BAdI-Implementierung verwendet wird. Wahlweise kann bei Verwendung der alten Implementierung geprüft werden, ob die optionalen Parameter genutzt werden, die ebenfalls die Performance der Instanziierung verbessern (siehe SAP-Hinweis 537844).

► **Aufrufe von Interface-Methoden innerhalb von Schleifen finden**

Erkennt Aufrufe bestimmter Methoden innerhalb von Schleifen. In der Voreinstellung werden massenfähige Methoden aus dem BOPF-Framework (siehe Abschnitt 2.2.1, »Architektur-Blau-pause«) erkannt, die in Schleifen pro Einzelsatz gerufen werden.

Wahlweise können auch andere Methoden erkannt werden. Details dazu finden Sie ebenfalls in der Hilfefunktion **F1** der Prüfung.

Um die typischen Performanceprobleme frühzeitig zu erkennen, sollten Sie den Code Inspector bereits während der Entwicklungsphase einsetzen. Dazu finden Sie im System einige vorkonfigurierte globale Prüfvarianten (siehe Abschnitt 5.1.1, »Einsatz des Code Inspectors«). Dazu gehören z. B. die Prüfvarianten PERFORMANCE_CHECKLIST_XL oder PERFORMANCE_DB. Diese können Sie in eigene lokale Prüfvarianten kopieren und Ihren Bedürfnissen anpassen.

8.2 Dynamische Prüfungen und Traces

Im Gegensatz zu statischen Prüfungen muss das Programm bei dynamischen Prüfungen und Traces ausgeführt werden. Die Messungen sollten in Ihrer Produktiv- oder vorgelagerten Qualitätssicherungsumgebung mit realistischen Datenmengen ausgeführt werden, um aussagekräftige Ergebnisse zu erhalten. In diesem Abschnitt stellen wir die folgenden Werkzeuge für die Systemkomponenten *Datenbank* und *ABAP-Schicht* vor:

- Laufzeitstatistiken (Transaktion STAD)
- Performance Trace (Transaktion ST05)
- ABAP Trace (Transaktion SAT)
- Single Activity Trace (Transaktion ST12)
- SQL Monitor (Transaktion SQLM)
- Runtime Check Monitor (Transaktion SRTCM)

8.2.1 Laufzeitstatistiken (STAD)

Die Transaktion STAD beinhaltet die Laufzeitstatistiken für alle Anfragen an das SAP-System. Typische Anfragen sind z. B.:

- Dialogschritte (z. B. ein Klick auf einen Button im SAP GUI)
- HTTP-Anfrage (z. B. ein Klick auf einen Button in einer Webapplikation)

- ▶ Hintergrundjob (ein ABAP-Programm, das im Hintergrund ausgeführt wurde)
- ▶ RFC-Aufruf (ein Funktionsbaustein, der remote ausgeführt wird)

Im Allgemeinen sprechen wir von einzelnen Statistiksätzen, die die Laufzeitstatistiken bilden. Statistiksätze enthalten wichtige technische Informationen, die während der Laufzeit des Programms gesammelt wurden. Dazu gehören unter anderem die verbrauchte CPU-Zeit oder Datenbankzeit, aber auch die Anzahl der verarbeiteten Datensätze. Die Statistiksätze werden vom jeweiligen Applikationsserver, auf dem die Anfrage ausgeführt wurde, gesammelt und in das lokale Dateisystem geschrieben. Dort stehen die Dateien standardmäßig für 48 Stunden zur Verfügung, bevor sie überschrieben werden (diese Einstellung kann geändert werden, siehe Kasten »Tipps zur Konfiguration der Transaktion STAD«, am Ende des Abschnitts).

Sehen wir uns nun an, wie Sie die Statistiksätze selektieren und auswerten können. Abgerundet wird der Abschnitt durch die wichtigsten Einstellmöglichkeiten und Parameter für diese Transaktion.

Selektion der Statistiksätze

Nach dem Ende der Anfrage werden die während der Laufzeit gesammelten und aggregierten Daten zunächst in einen Puffer und dann in das Dateisystem des Applikationsservers geschrieben. In der Transaktion STAD können Sie diese Daten dann selektieren (siehe Abbildung 8.1).

Im Feld für das Zeitintervall wählen Sie die Zeit so, dass das Ende der Anfrage im Selektionszeitraum liegt. Dies ist z. B. bei lange laufenden Anfragen wie Hintergrundprozessen von Bedeutung. Beachten Sie auch, dass Sie den Zeitraum nicht zu groß wählen, da sich dies negativ auf die Geschwindigkeit der Abfrage und den Speicherverbrauch im Applikationsserver auswirkt. Je mehr Anfragen im SAP-System angefallen sind, desto kürzer sollte der ausgewählte Zeitraum sein. In stark belasteten Systemen sollten hier nur wenige Minuten eingetragen werden oder die Selektion auf einen Applikationsserver (über den Button SERVER SELECTION) beschränkt werden.

Abbildung 8.1 Selektionsmaske in STAD

Darüber hinaus können Sie noch weitere Filter wie den Benutzernamen, das Programm oder die Transaktion, den Tasktyp (DIALOG, RFC, BACKGROUND usw.) und verschiedene Schwellenwerte, z. B. Mindestantwortzeit oder Mindestdatenbankzeit, angeben.

Auswertung der Statistiksätze

Nachdem Sie die Selektion ausgeführt haben, erhalten Sie eine Liste aller Statistiksätze, die im selektierten Zeitraum angefallen sind (Abbildung 8.2). In dieser Übersicht werden schon wichtige KPIs wie RESPONSE TIME, CPU TIME und DB REQUEST TIME angezeigt. Über den Button SEL. FIELDS können Sie auch weitere Detailinformationen in dieser Liste einblenden.

SAP Workload: Single Statistical Records - Overview												
Download Disp. mode Sel. fields Server ID												
System: A4H Number of RFCs which responded (without errors): 1 (1)												
Analyzed time: 28.06.2013 / 00:00:00 - 28.06.2013 / 23:50:00												
Display mode: All statistic records, sorted by time												
Started	Server	Transaction	Program	T Scr.	Up	User	Response time (ms)	Time in WFs (ms)	Wait time (ms)	CPU time (ms)	DB req. time (ms)	
	*	*	*	*	*	*	1.000			0	0	
19:00:34	a4hhost_A4H_00	SE38	ZR_A4H_CHAPTER8_TOP_CUST_2	D	0120	7	SCHMEIDERTHO	23.030	23.030	0	10	2
19:31:17	a4hhost_A4H_00	B	RSAL_BATCH_TOOL_DISPATCHING	B	11	DDIC	30.293	30.293	0	230	50	
19:35:17	a4hhost_A4H_00	R	RFC	R	3004	8	DDIC	1.192	1.192	0	890	344
19:35:17	a4hhost_A4H_00	B	SWNC_TCOLL_STARTER	B	12	DDIC	2.060	2.060	0	20	10	
19:37:13	a4hhost_A4H_00	SE38	ZR_A4H_CHAPTER8_TOP_CUST_2	D	0120	2	SCHMEIDERTHO	23.468	23.468	0	0	1
19:39:19	a4hhost_A4H_00	SE38	ZR_A4H_CHAPTER8_TOP_CUST_1	D	0120	7	SCHMEIDERTHO	76.257	76.257	0	55.040	25.353
20:00:25	a4hhost_A4H_00	SE38	ZR_A4H_CHAPTER8_TOP_CUST_2	D	0120	9	SCHMEIDERTHO	22.930	22.930	0	0	2
20:24:01	a4hhost_A4H_00	SE38	ZR_A4H_CHAPTER8_TOP_CUST_2	D	0120	0	SCHMEIDERTHO	23.062	23.062	0	20	2
20:31:17	a4hhost_A4H_00	B	RSAL_BATCH_TOOL_DISPATCHING	B	12	DDIC	30.314	30.314	0	230	55	
20:32:23	a4hhost_A4H_00	SE16	/LBCDWB/DBSPFLI	D	1000	1	SCHMEIDERTHO	1.163	33	0	30	8
20:35:17	a4hhost_A4H_00	B	SWNC_TCOLL_STARTER	B	12	DDIC	2.065	2.064	1	20	10	
21:31:17	a4hhost_A4H_00	B	RSAL_BATCH_TOOL_DISPATCHING	B	11	DDIC	30.315	30.314	1	230	57	
21:35:17	a4hhost_A4H_00	B	SWNC_TCOLL_STARTER	B	12	DDIC	2.058	2.057	1	20	11	
21:38:21	a4hhost_A4H_00	SE24	SEO_STARTUP	D	0200	1	SCHMEIDERTHO	1.387	70	1	50	21
21:54:47	a4hhost_A4H_00	SE16	/LBCDWB/DBSBOOK	D	1000	3	GAHM	637.030	637.030	0	401.800	40.290
22:04:43	a4hhost_A4H_00	STAD	RSSTAT26	D	0120	4	GAHM	2.105	2.105	0	2.090	3
22:05:14	a4hhost_A4H_00	STAD	RSSTAT26	D	0120	9	GAHM	2.187	2.187	0	2.190	0
22:05:33	a4hhost_A4H_00	STAD	RSSTAT26	D	0120	4	GAHM	2.088	2.088	0	2.080	0

Abbildung 8.2 Liste aller Statistiksätze

Komponenten der Antwortzeit

Beim Doppelklick auf einen Eintrag wird der Statistiksatz im Detail angezeigt. Im oberen Teil sehen Sie einen Überblick über die Zeitverteilung (siehe Abbildung 8.3).

Die wichtigsten Komponenten der Antwortzeit sind in Tabelle 8.1 aufgeführt. Weitere Details finden Sie auch in SAP-Hinweis 8963.

Zeit	Bedeutung
WAIT FOR WORK PROCESS	Zeit, auf die auf einen freien Workprozess gewartet wurde (in der Dispatcherqueue)
PROCESSING TIME	Berechnet aus <i>Response time</i> – alle anderen hier genannten Zeiten. Beinhaltet in der Regel Zeit für ABAP Processing und CPU-Verbrauch, aber auch Wartezeiten (z. B. RFC, Verbu-chung, WAIT, ...), wenn der Workprozess nicht herausgerollt ist.
RFC + CPIC TIME	Zeit für die RFC-Kommunikation
LOAD und GENERATING TIME	Zeit für das Laden und Generieren von Programmen
ROLL (IN+WAIT) TIME	Zeiten, in denen der Workprozess herausgerollt war und die Zeit für das rein- und rausrollen des Benutzerkontexts.

Tabelle 8.1 Zeitkomponenten

Zeit	Bedeutung
DATABASE REQUEST TIME	Zeit, die für Datenbankzugriffe (Open SQL und natives SQL) verbraucht wurde
ENQUEUE TIME	Zeit für Sperranfragen an den SAP Enqueue Service

Tabelle 8.1 Zeitkomponenten (Forts.)

SAP Workload: Single Statistical Records - Details						
▲ Record ▼ Record Time DB DB Procedures Task / Memory Table Bytes Client Info						
System: A4H Instance: a4hhost_A4H_00						
Analyzed time: 29.06.2013 / 00:00:00 - 29.06.2013 / 20:10:00						
Record: 11:59:49 - 12:00:16 SE38 ZR_A4H_CHAPTER8_TOP_CUST_2 D GAHM						
Analysis of time in work process						
CPU time	1.550 ms	Number	Roll ins	1		
RFC+CPIC time	0 ms		Roll outs	1		
			Enqueues	0		
Total time in workprocs	26.922 ms	Load time	Program	0 ms		
Response time	26.922 ms		Screen	0 ms		
			CUA interf.	0 ms		
Wait for work process	0 ms	Roll time	Out	6 ms		
Processing time	437 ms		In	0 ms		
Load time	0 ms		Wait	0 ms		
Generating time	0 ms	Frontend	No.roundtrips	1		
Roll (in+wait) time	0 ms		GUI time	0 ms		
Database request time	3.292 ms		Net time	0 ms		
Enqueue time	0 ms					
DB procedure call time	23.138 ms		No. of DB procedure calls	2		
Analysis of ABAP/4 database requests (only explicitly by application)						
Connection	DEFAULT	Request time	3.292 ms			
Database requests total	17	Commit time	0 ms			
DB Proc. Calls	0	DB Proc. Time	0 ms			
Type of ABAP request	Database rows	Requests	Buffer rows	Database calls	Request time (ms)	Avg.time / row (ms)
Total	429.079	17	7	112	3.292	0,0
Direct read	0	10	7		0	0,0
Sequential read	429.079	7	0	112	3.292	0,0
Update	0	0		0	0	0,0
Delete	0	0		0	0	0,0
Insert	0	0		0	0	0,0
Note: Tables were saved in the tablebuffer.						

Abbildung 8.3 Statistiksatz – Zeitverteilung und Datenbankdetails

Details zur Datenbank Im unteren Teil der Abbildung 8.3 werden Details zur Datenbankzeit aufgeführt, z. B. ob die Zeit mit lesenden oder schreibenden Zugriffen verbraucht wurde und wie viele Datensätze jeweils verarbeitet wurden.

Falls die Transaktion STAD entsprechend konfiguriert ist (siehe Kasten), finden Sie hier auch Details zu Datenbankprozeduren und Datenbanktabellen. Dazu werden die Anzahl der verarbeiteten Datensätze und der Zeitverbrauch für Tabellen (siehe Abbildung 8.4) dargestellt bzw. die Anzahl der Aufrufe und der Zeitverbrauch für Datenbankprozeduren (hier nicht dargestellt).

Table accesses (list might be incomplete!)					
Table name	Number of rows accessed				Time (ms)
	Total	Dir. reads	Seq. reads	Changes	
TOTAL	429.079	0	429.079	0	3.292
SBOOK	429.079	0	429.079	0	3.292
AAB_ID_ACT	0	0	0	0	0
VARID	0	0	0	0	0

Abbildung 8.4 Statistiksatz – Details zu Datenbanktabellen

Konfiguration Schließlich möchten wir Ihnen noch zeigen, wie Sie die Transaktion STAD konfigurieren können, um die hier vorgestellten Details zu sehen.

[+] Tipps zur Konfiguration der Transaktion STAD

Das anfangs erwähnte Zeitfenster von 48 Stunden für die Persistenz der Daten kann über den SAP-Profilparameter STAT/MAX_FILES auf bis zu 99 Stunden verlängert werden

Die hier vorgestellten Detaildaten auf Tabellen- und Datenbank-Prozedurebene werden nur angezeigt, wenn die beiden SAP-Profilparameter STAT/TABREC und STAT/DBPROCREC auf Werte gesetzt werden, die größer als null sind. Diese Parameter können Sie in der Transaktion ST03 über den folgenden Pfad dynamisch für einen bestimmten Zeitraum ändern. Dabei gehen Sie in der Transaktion ST03 über folgenden Pfad: KOLLEKTOR & PERF. DATENBANK • STATISTIKSÄTZE UND -DATEI • ONLINE PARAMETER • DIALOGSCHRITTSTATISTIK.

Im dann angezeigten Dialog können Sie die Parameter für den gewünschten Analysezeitraum ändern. Der Parameter STAT/RFCREC für die RFC-Subsätze ist standardmäßig bereits auf »5« gesetzt.

8.2.2 Performance Trace (ST05)

Der Performance Trace (Transaktion ST05) ist ein sehr mächtiges Werkzeug und neben dem ABAP Trace eines der wichtigsten Werkzeuge zur Performanceanalyse von ABAP-Programmen.

In der Transaktion ST05 können Sie die folgenden Traces erstellen:

- ▶ SQL Trace
- ▶ Puffer-Trace
- ▶ Enqueue Trace
- ▶ RFC Trace
- ▶ HTTP Trace

Von diesen ist der SQL Trace für die Performanceanalyse der wichtigste. Wir stellen Ihnen anhand des SQL Traces die Aufzeichnung eines Traces vor, die für die anderen Tracetypen in derselben Art und Weise durchgeführt wird. Die Auswertung des SQL Traces besprechen wir im Detail, während wir für die anderen Traces einen Überblick der wichtigsten Inhalte geben.

Aufzeichnung eines SQL Traces

Ein SQL Trace wird mit den Drucktasten TRACE EINSCHALTEN bzw. TRACE MIT FILTER EINSCHALTEN aktiviert. Im ersten Fall gilt die Aufzeichnung für den aktuellen Benutzer und den Server, auf dem die Transaktion ST05 aufgerufen wurde.

Bei der Aufzeichnung mit Filter stehen Ihnen verschiedene Filter, wie z. B. BENUTZERNAME, PROZESS-NUMMER (aus der Workprozessübersicht SM50) oder TABELLENNAMEN, zur Verfügung (siehe Abbildung 8.5).

In beiden Fällen können Sie unter STACK-TRACE wählen, ob für die SQL-Anweisungen die ABAP-Aufrufhierarchie mit aufgezeichnet werden soll oder nicht.

Bei der Aufzeichnung sollten Sie Folgendes beachten:

- ▶ Es kann immer nur ein SQL Trace pro Applikationsserver aktiviert werden, das heißt, es ist nicht möglich, mehrere Benutzer oder Prozesse parallel auf einem Server zu tracen.

Hinweise zur Aufzeichnung

- ▶ Der Trace wird in Dateien auf dem jeweiligen lokalen Applikationsserver geschrieben, daher muss die Auswertung auf demselben Applikationsserver erfolgen wie die Aufzeichnung.
- ▶ Es ist zwar möglich, den SQL Trace auf einem anderen Applikationsserver als dem lokalen Server zu aktivieren (siehe Filteroptionen in Abbildung 8.5), für die Auswertung müssen Sie sich dann aber auf dem Server anmelden, auf dem das Programm mit der Aufzeichnung ausgeführt wurde.
- ▶ Der Trace wird im sogenannten Round-Robin-Verfahren in die Tracedateien geschrieben. Wenn auch die letzte Datei beschrieben wurde, wird die erste Datei wieder überschrieben, das heißt, ein lange laufender Trace kann sich so selbst überschreiben und nur das Ende der Aufzeichnung beinhalten. Achten Sie daher darauf, den Trace nicht zu lange aufzuzeichnen, um ein Überschreiben der Daten zu verhindern. Für länger laufende Programme bietet sich an, eine erste Analyse mit dem SQL Monitor (siehe Abschnitt 8.2.5, »SQL Monitor (SQLM)«) durchzuführen. Die Ergebnisse des SQL Monitors (z. B. Namen von Tabellen) können Sie anschließend als Filter für eine Aufzeichnung mit der Transaktion ST05 verwenden.

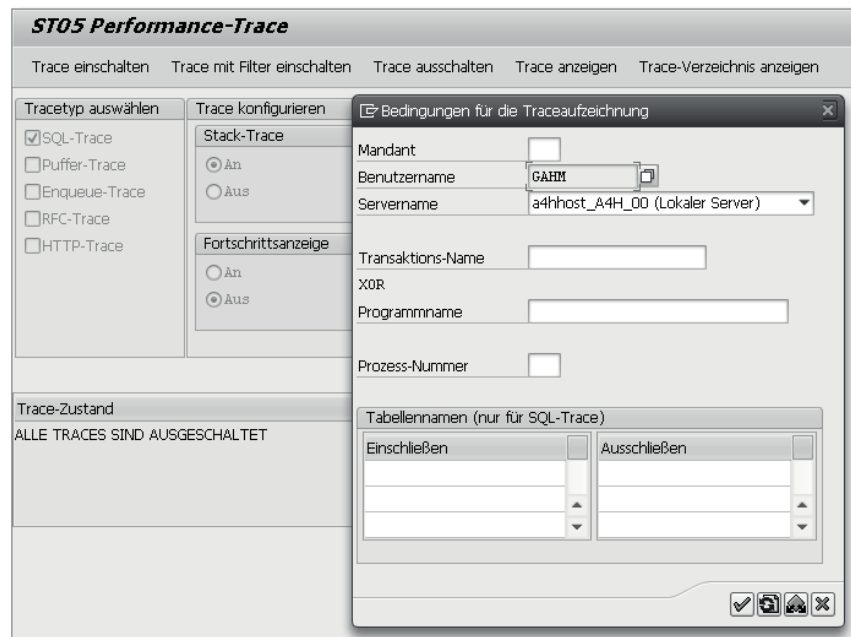



Abbildung 8.5 Transaktion ST05 – Aufzeichnung eines SQL Traces

Für die anderen Tracearten, wie z. B. den RFC Trace, erfolgt die Aufzeichnung analog.

Praxistipp: Tracedaten sichern

[+]

Sie können die aufgezeichneten Daten in der Transaktion ST05 sichern, um sie vor dem Überschreiben zu schützen. Gehen Sie dabei wie folgt vor: Auf dem Selektionsbildschirm (siehe Abbildung 8.6) wählen Sie das Symbol  TRACE-SÄTZE IN DB SCHREIBEN. Alle Daten, die den auf diesem Bildschirm eingegebenen Selektionskriterien entsprechen, werden dann in der Datenbank gespeichert. Über die Drucktaste TRACE-VERZEICHNIS ANZEIGEN (siehe Abbildung 8.5) können Sie dann auf die abgespeicherten Daten zugreifen und diese anzeigen.





Auswertung des SQL Traces

Um einen SQL Trace anzuzeigen, rufen Sie die Funktion TRACE ANZEIGEN auf. Im folgenden Dialog (siehe Abbildung 8.6) können Sie verschiedene Filter auswählen und die Anzeige beeinflussen. Im Bereich TRACE-TYPEN wählen Sie die Art des Traces, der angezeigt werden soll, in unserem Fall also den SQL-TRACE. Im Bereich TRACE-ZEITRAUM wird der Start- und Endzeitpunkt der Auswertung angegeben. Im Bereich TRACE-FILTER können Sie noch zusätzliche Filter wie den Benutzernamen, Tabellennamen usw. für die Anzeige eingeben.



Abbildung 8.6 Transaktion ST05 – Anzeigen eines SQL Traces

Über die Symbole am oberen Bildschirmrand können Sie folgende Sichten für den SQL Trace anzeigen:

- ▶ Mit dem Icon  AUSFÜHREN gelangen Sie auf die Liste der sogenannten Hauptsätze. Dort sehen Sie pro SQL-Anweisung eine Zeile, die alle Details zur SQL-Anweisung zusammenfasst.
- ▶ Mit dem Icon  TRACE-EINZELSÄTZE ANZEIGEN erhalten Sie für alle SQL-Anweisungen die Details wie OPEN oder FETCH.
- ▶ Mit dem Icon  STRUKTUR-IDENTISCHE TRACE-SÄTZE ANZEIGEN erhalten Sie eine aggregierte Liste, die alle ähnlichen SQL-Anweisungen zusammenfasst. Strukturidentisch meint hier, dass SQL-Anweisungen, die die gleichen WHERE-Bedingungen haben, in einer Zeile in dieser Sicht zusammengefasst und aggregiert werden. Diese Anweisungen können von unterschiedlichen Stellen im ABAP-Programm kommen.
- ▶ Mit dem Icon  TRACE-ÜBERSICHT erhalten Sie eine Übersicht über alle Anweisungen im SQL Trace. Dort sehen Sie eine Zusammenfassung und Auswertung der SQL-Anweisungen nach unterschiedlichen Kategorien.

Im Folgenden gehen wir auf die zwei wichtigsten Sichten ein:

- ▶ strukturidentische SQL-Anweisungen
- ▶ Liste der Hauptsätze

Strukturidentische SQL-Anweisungen

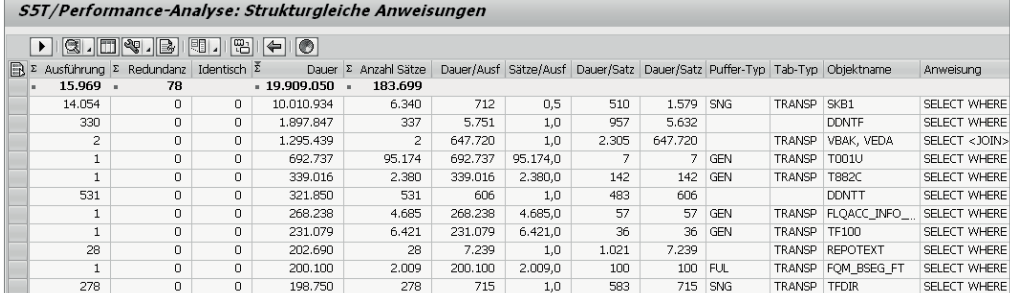
Sie sollten die Analyse möglichst immer mit der Sicht über die strukturidentischen SQL-Anweisungen beginnen. In dieser Sicht werden die teuersten SQL-Anweisungen mit den wichtigsten Details und Kennzahlen dargestellt. Die Liste ist so sortiert, dass die teuersten SQL-Anweisungen oben stehen, das heißt die SQL-Anweisungen, die (über alle Ausführungen summiert) am meisten Zeit verbraucht haben. Im Folgenden beschreiben wir die wichtigsten Werte, die Sie in dieser Ansicht auswerten können.

Wichtige Spalten

In der Spalte AUSFÜHRUNG sehen Sie die Summe der Ausführungen für die SQL-Anweisung. An dieser Spalte lässt sich das typische Problem vieler Ausführungen (siehe Abschnitt 7.1.2, »Viele Ausführungen«) gut erkennen.

In den Spalten REDUNDANZ und IDENTISCH sehen Sie, ob bzw. wie viele SQL-Anweisungen identisch ausgeführt wurden. Die Spalte REDUNDANZ gibt eine absolute Anzahl wieder und weist aus, wie viele SQL-Anweisungen exakt gleich (das heißt mit gleichen Parame-

tern in der WHERE-Bedingung) ausgeführt wurden. Die Spalte IDENTISCH gibt an, wie viele SQL-Anweisungen prozentual identisch waren. In diesen Spalten erkennen Sie das typische Problem der identischen Ausführungen (siehe Abschnitt 7.1.3, »Identische Datenbankzugriffe«) sehr gut.



Ausführung	Redundanz	Identisch	Dauer	Anzahl Sätze	Dauer/Ausf	Sätze/Ausf	Dauer/Satz	Dauer/Satz	Puffer-Typ	Tab-Typ	Objektname	Anweisung
15.969	78		19.909.050	183.699								
14.054	0	0	10.010.934	6.340	712	0,5	510	1.579	SNG	TRANSP	SKB1	SELECT WHERE
330	0	0	1.897.847	337	5.751	1,0	957	5.632			DONTF	SELECT WHERE
2	0	0	1.295.439	2	647.720	1,0	2.305	647.720			VBAK, VEDA	SELECT <JOIN>
1	0	0	692.737	95.174	692.737	95.174,0	7	7	GEN	TRANSP	T001U	SELECT WHERE
1	0	0	339.016	2.380	339.016	2.380,0	142	142	GEN	TRANSP	T882C	SELECT WHERE
531	0	0	321.850	531	606	1,0	483	606			DONTT	SELECT WHERE
1	0	0	268.238	4.685	268.238	4.685,0	57	57	GEN	TRANSP	FLQACC_INFO...	SELECT WHERE
1	0	0	231.079	6.421	231.079	6.421,0	36	36	GEN	TRANSP	TF100	SELECT WHERE
28	0	0	202.690	28	7.239	1,0	1.021	7.239			REPOTEXT	SELECT WHERE
1	0	0	200.100	2.009	200.100	2.009,0	100	100	FUL	TRANSP	FQM_BSEG_FT	SELECT WHERE
278	0	0	198.750	278	715	1,0	583	715	SNG	TRANSP	TFDIR	SELECT WHERE

Abbildung 8.7 Transaktion ST05 – strukturidentische SQL-Anweisungen

In der Spalte DAUER wird die Zeit angezeigt, die für alle Ausführungen der SQL-Anweisung im Trace benötigt wurde. Standardmäßig ist die Liste nach dieser Spalte sortiert.

In den Spalten ANZAHL SÄTZE und SÄTZE/AUSF sehen Sie, wie viele Datensätze von allen Ausführungen der SQL-Anweisung in Summe bzw. einer Ausführung im Durchschnitt verarbeitet wurden. Anhand der Werte in diesen Spalten lässt sich das typische Problem der Verarbeitung großer Datenmengen (siehe Abschnitt 7.1.1, »Verarbeitung großer Datenmengen«) gut erkennen.

Die Spalten DAUER/AUSF, MINIMALE DAUER/SATZ und MAXIMALE DAUER/SATZ zeigen an, wie lange die SQL-Anweisung pro Ausführung und Datensatz (minimal und maximal) gebraucht hat. Anhand dieser Werte können Sie das typische Problem ineffizienter Zugriffspfade (siehe Abschnitt 7.1.4) erkennen.

Die Spalten PUFFER-TYP und OBJEKTNAME zeigen, auf welche Objekte (Tabellen oder Views) die SQL-Anweisung zugegriffen hat und ob diese gepuffert sind. Gegebenenfalls können Sie damit Tabellen identifizieren, auf die unnötigerweise zugegriffen wurde, da nicht relevant für den Prozessschritt. Bei aktivierter Tabellenpufferung lassen sich auch Tabellen identifizieren, die am Tabellenpuffer vorbei

gelesen wurden. Wenn im SQL Trace eine gepufferte Tabelle auftaucht, bedeutet das, dass direkt auf die Datenbank zugegriffen wurde und nicht aus dem Puffer gelesen werden konnte. Mögliche Gründe dafür haben wir in Abschnitt 7.1.1 beschrieben.

In der Spalte ANWEISUNG sehen Sie den Text der SQL-Anweisung, bei dem die Feldliste und die FROM-Klausel ausgeblendet sind.

Details und Funktionen zu SQL-Anweisungen

Für die SQL-Anweisungen stehen Ihnen in der Symbolleiste eine Reihe von nützlichen Funktionen zur Verfügung (Tabelle 8.2).

Icon in Abbildung 8.7	Beschreibung
	Blendet die Standardfunktionen (z. B. Filtern, Sortieren, ...) für die Liste (Advanced List Viewer – ALV) ein.
	Zeigt die ganze SQL-Anweisung mit Platzhaltern (Bind-Variablen) oder in die WHERE-Bedingung eingesetzten Werten an. Hierfür wird die erste SQL-Anweisung der zugrunde liegenden SQL-Anweisungen verwendet.
	Zeigt die Data-Dictionary-Informationen (DDIC) zur Tabelle oder View an.
	Zeigt den Ausführungsplan für die SQL-Anweisung an.
	Zeigt die Aufrufstelle der SQL-Anweisung im ABAP-Programm an.
	Verzweigt auf andere Sichten, z. B. auf die Liste der Hauptsätze bzw. der wertgleichen SQL-Anweisungen (identical SELECTS).
	Ändert die Anzeige von absoluten Werten auf relative Werte (%).

Tabelle 8.2 Funktionen in der Symbolleiste des SQL Traces

Diese Symbolleiste mit Funktionen zu einer SQL-Anweisung finden Sie auf anderen Listen in der gleichen oder sehr ähnlichen Form wieder. Weiterhin hat die Liste noch einige nützliche Querverweise, die Sie bequem per Doppelklick aktivieren (Tabelle 8.3).

Doppelklick auf ...	Beschreibung
Spalte AUSFÜHRUNG	Springt auf die Liste der Hauptsätze, zeigt dort aber nur die ausgewählte SQL-Anweisung an.
Spalten REDUNDANZ/IDENTISCH	Springt auf die Liste der wertgleichen SQL-Anweisungen (identical SELECTS), zeigt dort aber nur die ausgewählte SQL-Anweisung an.
Spalten OBJEKTNAME/PUFFER-TYP/TAB-TYP	Springt auf die Informationen zum DDIC.
Spalte ANWEISUNG	Zeigt die SQL-Anweisung in voller Länge an.

Tabelle 8.3 Querverweise aus einzelnen Spalten des SQL Traces

Die Liste der Hauptsätze ist nach dem Zeitstempel sortiert. Dort sehen Sie pro ausgeführter SQL-Anweisung eine Zeile im Ergebnis. In dieser Ansicht finden Sie wichtige Informationen, wie z. B.:

- ▶ **Zeitstempel:** Zeitpunkt der Ausführung
- ▶ **Dauer:** Laufzeit der SQL-Anweisung
- ▶ **Sätze:** Ergebnismenge der SQL-Anweisung
- ▶ **Programm-Name:** Name des Programms, in dem die SQL-Anweisung ausgeführt wurde
- ▶ **Objektname:** Tabelle oder View

In der Symbolleiste stehen Ihnen, neben den in Tabelle 8.2 angezeigten Funktionen noch zusätzlich die in Tabelle 8.4 beschriebenen wichtigen Funktionen zur Verfügung.




Icons in Abbildung 8.7	Beschreibung
	Zeigt die Aufrufhierarchie der SQL-Anweisung, falls die Option bei der Aufzeichnung mit aktiviert wurde (siehe Abbildung 8.5).
	Verzweigt auf die Liste der Einzelsätze der SQL-Anweisungen.
	Verzweigt auf weitere Sichten und Listen.

Tabelle 8.4 Weitere Funktionen in der Symbolleiste des SQL Traces

Aufrufhierarchie einer SQL-Anweisung Eine sehr nützliche Funktion auf dieser Liste ist die Anzeige der Aufrufhierarchie. Wenn diese Option bei der Aufzeichnung aktiviert war, können Sie hier die Aufrufhierarchie sehen, über die eine SQL-Anweisung aufgerufen wurde (siehe Abbildung 8.8).

hh:mm:ss.ms	Dauer	Sätze	Programm-Name	Objektname	Anweisung
14:14:57.324	1.231	1	ZCO_UPLOAD_KBXX	TFDIR	SELECT WHERE "FUNCNAME" = 'K_DATE_TO_PERIOD_CONVERT' OR
14:14:57.327	697	1	ZCO_UPLOAD_KBXX	TFDIR	SELECT WHERE "FUNCNAME" = 'PERIOD_DAY_DETERMINE' ORDER B
14:14:57.331	1.459	1	SAPLSPO1	D347T	SELECT WHERE "PROGNAME" = 'SAPLSPO1' AND "SPRSL" = 'E' AND
14:14:57.412	977				
14:14:57.413	657				
14:14:59.348	1.012				
14:14:59.350	1.582				
14:14:59.354	2.338				
14:14:59.356	925				
14:14:59.357	967				
14:14:59.358	1.046				
14:14:59.360	1.071				
14:14:59.362	811				
14:14:59.363	2.091				

Stufe	ABAP Eventtyp	ABAP Eventname	Programm-Name	Include-Name	Zeile
6	MODULE (PBO)	INIT_500	SAPLSPO1	LSP01030	67
5	FUNCTION	POPUP_TO_CONFIRM	SAPLSPO1	LSP01006	248
4	FORM	CONFIRM_FORMAT_SETTING	ZCO_UPLOAD_KBXX	ZCO_UPLOAD_KBXX_FORMS	579
3	FORM	%_SEL_SCREEN_BLOCK_1000029	ZCO_UPLOAD_KBXX	ZCO_UPLOAD_KBXX	66
2	MODULE (PAI)	%_BLOCK_1000029	ZCO_UPLOAD_KBXX	ZCO_UPLOAD_KBXX	209
1	EVENT	SYSTEM-EXIT	ZCO_UPLOAD_KBXX	<SYSINI>	18

Abbildung 8.8 Transaktion ST05 – Hauptsätze mit Aufrufhierarchie

Die Analyse der Aufrufhierarchie ist bei den typischen Problemen der vielen Ausführungen bzw. identischer Ausführungen sehr nützlich. Sie können damit schnell die Verursacher identifizieren. Per Doppelklick können Sie von hier auf die jeweilige Aufrufstelle im ABAP-Quelltext springen.

Weitere Sichten im SQL Trace Für den SQL Trace gibt es noch ein paar speziellere Sichten, über die wir hier einen kurzen Überblick geben wollen:

- ▶ **Wertgleiche Anweisungen:** Hier werden die identischen SQL-Anweisungen angezeigt. Sie sehen, wie oft SQL-Anweisungen mit den gleichen Werten ausgeführt wurden.
- ▶ **Tabellenzugriffe:** Hierbei handelt es sich um eine Gruppierung auf Tabellenebene und SQL-Typ (SELECT, DELETE, INSERT, UPDATE ...). Sie sehen also, wie und wie oft auf Tabellen zugegriffen wurde.
- ▶ **Trace-Übersicht:** Zeigt den SQL Trace im Überblick und wertet verschiedene Aspekte aus, wie z. B. lesende, schreibende Verarbeitung, Ergebnismengen, Pufferung usw.

Weitere Traces in der Transaktion ST05

Neben dem SQL Trace gibt es in der Transaktion ST05 noch weitere Traces, die aber eher selten benötigt werden. Wir geben Ihnen hier

eine Übersicht, welche Informationen Sie in diesen Traces finden und wann Sie diese einsetzen können:

- ▶ **Puffer-Trace:** Der Puffer-Trace zeichnet Zugriffe auf Tabellen- und andere Puffer auf. Sie sehen, welche Zugriffe es gab, wie auf die Puffer zugegriffen wurde und wie lange diese gedauert haben. Dieses Werkzeug wird benötigt, wenn es beim Zugriff auf die Puffer zu Performanceproblemen kommt, was aber nur in Ausnahmefällen der Fall ist. Zugriffe, die am Tabellenpuffer vorbeigehen, kommen häufiger vor und werden mit dem SQL Trace analysiert.
- ▶ **Enqueue Trace:** Der Enqueue Trace zeichnet alle Zugriffe auf den Enqueue Service im SAP-System auf. Sie sehen, welche Zugriffe es gab, von welcher Art diese waren und wie lange diese gedauert haben. Dieser Trace wird benötigt, um Sperranfragen im SAP-System genauer zu analysieren.
- ▶ **RFC Trace:** Der RFC Trace zeichnet alle RFC-Anfragen (sowohl Client als auch Server) auf. Sie sehen, welche RFC-Zugriffe es gab, von welcher Art diese waren, welche Datenmengen übertragen wurden und wie lange diese gedauert haben. Dieser Trace wird benötigt, um RFC-Anfragen genauer zu analysieren.
- ▶ **HTTP Trace:** Der HTTP Trace zeichnet alle HTTP-Anfragen (sowohl Client als auch Server) auf. Sie sehen, welche HTTP-Zugriffe es gab, von welcher Art diese waren, welche Datenmengen übertragen wurden und wie lange diese gedauert haben. Dieser Trace wird benötigt, um HTTP-Anfragen genauer zu analysieren.

8.2.3 Laufzeitanalyse mit dem ABAP Trace/ABAP Profiler (SAT)

Mit der Transaktion SAT (Single Activity Trace) kann die Ausführung eines ABAP-Programms detailliert aufgezeichnet werden. Dabei werden die Ausführungszeiten von einzelnen ABAP- und SQL-Anweisungen und auch Modularisierungseinheiten (z. B. Methoden, Funktionsbausteine etc.) gemessen. Zusammen mit der optional aufgezeichneten Aufrufhierarchie und der Anzahl der Aufrufe kann damit die Performance eines Programms sehr genau untersucht werden.

Die ABAP-Laufzeitanalyse erfolgt in mehreren Schritten:

1. Einstellung der Messvariante
2. Messung
3. Auswertung des aufgezeichneten Traces

Im Folgenden beschreiben wir, wie Sie dieses Werkzeug effizient einsetzen können.

Einstellung der Messvariante

Die Messvariante definiert, welche Informationen im Detail aufgezeichnet werden sollen. Als Startpunkt können Sie die System-Messvariante `DEFAULT` in eine benutzerspezifische Variante kopieren und anpassen. Falls Sie diese ebenfalls `DEFAULT` nennen, wird sie als benutzerspezifische Standardeinstellung verwendet. Die Einstellungen der Messvariante sind in drei Register gegliedert:

- ▶ **DAUER UND ART** (Wie soll aufgezeichnet werden?)
- ▶ **ANWEISUNGEN** (Was soll aufgezeichnet werden?)
- ▶ **PROGRAMMTEILE** (weitere Filter)

Dauer und Art

In dem Register **DAUER UND ART** (Abbildung 8.9) können Sie die maximale Größe und Dauer der Aufzeichnung einstellen. Die Standardwerte (10.000 KB, 1.800 Sekunden) reichen für die meisten Traceszenarien aus.

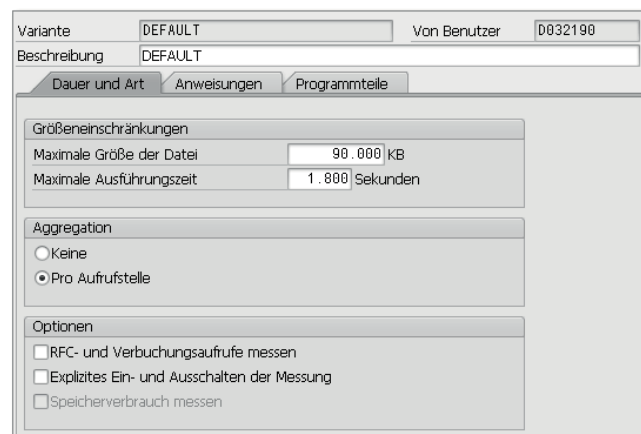


Abbildung 8.9 Messvariante – Dauer und Art der Aufzeichnung

Bei der Aggregation **PRO AUFRUFSTELLE** wird jede Aufrufstelle im Trace vermerkt, aggregiert über die Anzahl von Aufrufen. Falls Sie die Verdichtung des Traces deaktivieren (Aggregation **KEINE**), können Sie auch Aufrufhierarchien von Anweisungen darstellen und ebenfalls die Änderung des Speicherverbrauchs zur Laufzeit protokollieren. Der Nachteil einer Aufzeichnung ohne Aggregation ist aber, dass der Trace sehr groß werden kann, da jede Ausführung einzeln aufgezeichnet wird.

Aggregation


Mit der Option **EXPLIZITES EIN- UND AUSSCHALTEN DER MESSUNG** können Sie hier auch das explizite Ein- und Ausschalten der Aufzeichnung aktivieren, falls Sie z. B. nur das Speichern im letzten Dialogschritt einer Transaktion aufzeichnen wollen.

Optionen

Praxistipp: Explizites An- und Ausschalten der Messung

[+]

Sie haben verschiedene Möglichkeiten, die Traceaufzeichnung zu aktivieren:

- ▶ über OK-Codes `/ron` und `/roff`
- ▶ über das Menü: **SYSTEM • HILFSMITTEL • LAUFZEITANALYSE • EINSCHALTEN/AUSSCHALTEN**
- ▶ Sie können einen ABAP Trace für einen bestimmten Programmteil aktivieren, indem Sie zu Beginn und am Ende des Programmteils die ABAP-Anweisung `SET RUN TIME ANALYZER ON` bzw. `OFF` ausführen.
- ▶ Über eine Verknüpfung: Legen Sie mit dem Button  Verknüpfungen zu den Systembefehlen `/ron` und `/roff` z. B. auf dem Desktop an. Diese können Sie per Drag & Drop auf das SAP-GUI-Fenster ziehen und damit den Trace steuern.

Mit der Option **RFC- UND VERBUCHUNGSAUFRUFE** kann die Aufzeichnung an andere Workprozesse übergeben werden. Dazu muss folgender Profilparameter auf Quell- und Zielsever gesetzt werden:

```
rstr/accept_remote_trace = TRUE
```

Anweisungen

Sie können einstellen, welche Anweisungen in dem Trace separat aufgeführt werden sollen (siehe Abbildung 8.10). Hier sollten Sie auf jeden Fall noch die **LESE- und ÄNDERUNGSOPERATIONEN** für **INTERNE TABELLEN** aktivieren, um bei der Auswertung die Zeit, die zur Verarbeitung interner Tabellen benötigt wurde, angezeigt zu bekommen.

Interne Tabellen aktivieren

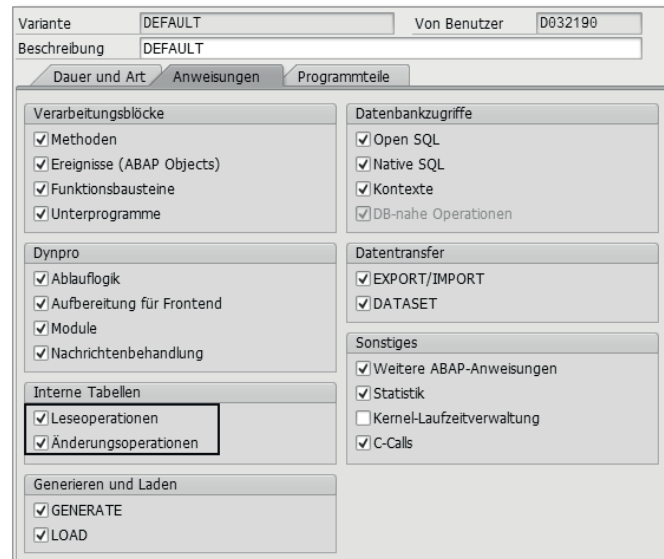


Abbildung 8.10 Messvariante – separat aufgeführte Anweisungen

Programmteile

Das Register PROGRAMMTEILE bietet die Möglichkeit, die Aufzeichnung auf einzelne Programmteile (Modularisierungseinheiten) einzugrenzen, z. B. wenn nur die Ausführung einer Methode oder eines Funktionsbausteins aufgezeichnet werden soll. In der Regel werden Sie bei der Performanceanalyse aber an den Laufzeitbeiträgen aller Programmteile interessiert sein und diese Option nicht nutzen.

Aufzeichnung des SAT Traces

Nachdem Sie die Messvariante angelegt haben, kann der Trace im Einstiegsbild der Transaktion SAT (Abbildung 8.11) aktiviert werden. Sie sollten dabei immer das Häkchen bei NAMEN FÜR INTERNE TABELLEN BESTIMMEN setzen, um bei der Traceauswertung interne Tabellen mit ihrem im ABAP-Programm verwendeten statt dem internen technischen Namen anzuzeigen.

Die Aufzeichnung kann auf mehrere Arten aktiviert werden, um unterschiedliche Ausführungs- und Messszenarien abzudecken:

- ▶ IM DIALOG
- ▶ IM PARALLELEN MODUS
- ▶ FÜR BENUTZER / SERVICE

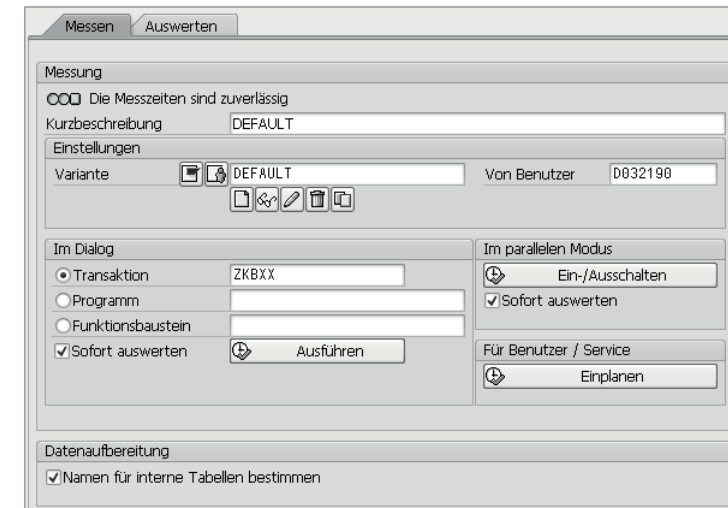


Abbildung 8.11 Einstiegsbild der Transaktion SAT

Bei der Aktivierung IM DIALOG wird die Anwendung direkt aus der Transaktion SAT heraus gestartet. Das bietet sich an, wenn Sie die Anwendung selbst ausführen können. Dabei werden sämtliche Aktivitäten in diesem Modus aufgezeichnet, bis Sie mit der ZURÜCK-Taste (oder **F3**) die Anwendung verlassen und zum Einstiegsbild der Transaktion zurückkehren. Falls Sie die Messung mit der Option EXPLIZITES EIN- UND AUSSCHALTEN DER MESSUNG durchführen wollen, muss die Aufzeichnung separat aktiviert werden, siehe Kasten »Praxistipp: Explizites An- und Ausschalten der Messung«.

Im Dialog

Zur Aufzeichnung bereits aktiver Prozesse können Sie die Messart IM PARALLELEN MODUS verwenden. Zunächst werden die aktiven Prozesse der aktuellen Instanz angezeigt. Prozesse anderer Instanzen können über den Schalter SERVERAUSWAHL ebenfalls angezeigt werden. Zur Aktivierung bzw. Deaktivierung des Traces markieren Sie den gewünschten Prozess und aktivieren den Trace über die Drucktaste SWITCH ON/OFF.

Im parallelen Modus

Es gibt auch die Möglichkeit, den Trace automatisch bei Eintreten bestimmter Bedingungen zu aktivieren. Dazu werden die Kriterien unter FÜR BENUTZER/SERVICE definiert und der Trace eingeplant. Tabelle 8.5 beschreibt die Einstellmöglichkeiten zur Einplanung einer Messung.

Für Benutzer/Service

Kriterium	Mögliche Einstellungen
SERVER-NAME	<bestimmter> oder <alle> Server
EXTERNER MODUS	1, ..., 6 oder beliebig (Modus/Session kann in SM04 nachgesehen werden und ist abhängig von der Serverinstanz)
PROZESS-TYP	Dialog, Verbuchung, Hintergrund, RFC, HTTP, SMTP, ITS, SHO Gebietskonstruktor, beliebig
OBJEKT-TYP	Transaktion, Funktionsbaustein, Report, SHO Gebietskonstruktor, URL, beliebig
MAX. ANZAHL GEPLANTER MESSUNGEN	wie viele Traces nacheinander aktiviert werden dürfen
ABLAUFdatum/-zeit	Gültigkeit der eingestellten Kriterien

Tabelle 8.5 SAT-Kriterien für die Traceart Benutzer/Service

Auswertung des SAT Traces

Mit dem Ende der Aufzeichnung wird der SAT Trace angezeigt, falls die Option SOFORT AUSWERTEN gesetzt wurde. Alternativ können Sie die Messung über den Karteireiter AUSWERTEN auswählen.

Trace Overhead

Bei der Auswertung des Traces sollten Sie beachten, dass die Aufzeichnung selbst natürlich auch einen CPU-Aufwand verursacht. Bei der Aufzeichnung eines komplexen Programms kann es daher vorkommen, dass die während des Traces gemessene Laufzeit größer ist, als ohne Trace. Der ABAP Trace versucht zwar, den von sich selbst verursachten Overhead herauszurechnen, allerdings gelingt das nicht immer, und es kann zu Abweichungen der *ohne* und *mit* Trace beobachteten Laufzeiten kommen. Wir können allerdings davon ausgehen, dass der Overhead für alle Traceereignisse derselbe ist, das heißt, die Laufzeiten der einzelnen Anweisungen werden relativ zueinander korrekt wiedergegeben.

Die Transaktion SAT bietet verschiedene Sichten und Auswertemöglichkeiten für den ABAP Trace. Alle Sichten sind dabei mit der Sicht PROFIL verlinkt, das heißt, Sie können über das Kontextmenü eines Profils die entsprechenden Daten aus dem Trace in unterschiedlichen Sichten analysieren.

Im ABAP Trace werden sowohl Brutto- als auch Nettolaufzeiten für einzelne Anweisungen und Verarbeitungsblöcke angezeigt. Die Bedeutung dieser Zeiten wird im Kasten erklärt.

Brutto- und Nettolaufzeiten

Brutto- und Nettolaufzeiten im ABAP Trace [*]

- ▶ **Bruttolaufzeit:** Über alle Ausführungen summierte Laufzeit einer ABAP-Anweisung. Bei Verarbeitungsblöcken (z. B. Formroutine, Methode, Funktionsbaustein usw.) werden die Zeiten anderer Anweisungen innerhalb und unterhalb der aktuellen Aufrufhierarchie dazuaddiert.
- ▶ **Nettolaufzeit:** Über alle Ausführungen (Schleifen) summierte Laufzeit einer einzelnen ABAP-Anweisung. Bei Verarbeitungsblöcken werden hier die Zeit zum Aufruf des Verarbeitungsblocks und alle Zeiten angegeben, die innerhalb des Verarbeitungsblocks anfallen und die nicht separat ausgewiesen werden.

Welche Sichten im Einzelnen zur Auswahl stehen, hängt dabei von der Art der Aufzeichnung ab. Bei *aggregierter* Aufzeichnung pro Aufrufstelle stehen Ihnen die in Tabelle 8.6 genannten Sichten zur Verfügung.

Sicht	Beschreibung
PROFIL	Zeigt die Ereignisse nach verschiedenen Profilen an. Es stehen folgende Profile zur Auswahl: TRACE-EREIGNISSE, PAKETE, KOMPONENTEN, PROGRAMME.
HITLISTE	Zeigt alle Traceereignisse mit Anzahl der Aufrufe, Brutto- und Nettozeit. Die Bruttozeit entspricht in etwa der Total Time und die Nettozeit der Own Time des ABAP Profilers.
DB-TABELLEN	Zeigt alle Zugriffe auf Datenbanktabellen an, entspricht der Sicht im ABAP Profiler.
ZEITEN	Zeigt eine detaillierte Liste der Ereignisse mit einer weiteren Unterteilung der Zeit in Komponenten wie Datenbankzeit, Datenbankinterface-Zeit, Zeit für interne Tabellen usw.

Tabelle 8.6 Sichten bei Aggregation pro Aufrufstelle

Bei *nicht-aggregierter* Aufzeichnung stehen zusätzlich die in Tabelle 8.7 genannten Sichten zur Auswahl.

Sicht	Beschreibung
VERARBEITUNGS-BLÖCKE	Zeigt eine interaktive, hierarchische Darstellung der Ereignisse mit Brutto- und Nettozeit. Die Ebenen der Aufrufhierarchie können beliebig analysiert werden. An dieser Stelle gibt es auch eine automatische Analyse zur Darstellung kritischer Verarbeitungsblöcke. Es können so z. B. alle Modularisierungseinheiten, die einen Anteil von mehr als 5 % an der Nettozeit haben, farblich hervorgehoben werden.
AUFRUFHIERARCHIE	Zeigt eine Liste der Ereignisse mit Aufrufebene, Brutto- und Nettozeit.

Tabelle 8.7 Zusätzliche Sichten, wenn ohne Aggregation aufgezeichnet wird

Beim Einstieg in die Auswertung des Traces werden zunächst die Sichten PROFIL: TRACE-EREIGNISSE und HITLISTE dargestellt. In der HITLISTE wird der vollständige ABAP Trace angezeigt, sortiert nach den Nettozeiten der Anweisungen bzw. Ereignisse. Sie haben nun die Möglichkeit, den ABAP Trace mithilfe unterschiedlicher Sichten zu untersuchen und den Trace zu filtern.

Sicht »Profil«

Die Profilsicht bietet den zentralen Einstieg in die Traceanalyse und dient dazu, den Trace nach unterschiedlichen Profilen sortiert darzustellen. Folgende Profile stehen zur Auswahl:

- ▶ TRACE-EREIGNISSE: Anzeige ausgewählter Anweisungen
 - ▶ Verarbeitungsblöcke (Methoden, Funktionsbausteine etc.)
 - ▶ interne Tabellen (READ, LOOP AT, ...)
 - ▶ Datenzugriffe intern (Tabellenpuffer, native SQL)
 - ▶ Datenzugriffe extern (OPEN SQL)
- ▶ PAKETE: Anweisungen, nach Paketen gegliedert
- ▶ KOMPONENTEN: Gliederung nach SAP-Softwarekomponenten
- ▶ PROGRAMME: Anweisung gegliedert nach:
 - ▶ Programmen
 - ▶ Funktionsgruppen
 - ▶ Klassen

Das Profil TRACE-EREIGNISSE wird in Abbildung 8.12 dargestellt. Durch Auswahl einzelner Ereignisse kann der ABAP Trace entsprechend gefiltert und in der HITLISTE oder der Sicht ZEITEN dargestellt werden. Mit einem Doppelklick auf LAUFZEITMESSUNG wird der vollständige Trace gezeigt.

Profil	Auswahl	Anzahl	Netto [µsec]	Netto [%]
Laufzeitmessung	<input checked="" type="checkbox"/>	6.028.754	92.286.948	100,00
Verarbeitungsblöcke intern	<input checked="" type="checkbox"/>	2.715.593	21.288.040	23,07
Funktionsbausteine	<input type="checkbox"/>	599.782	7.104.826	7,70
Unterprogramme	<input type="checkbox"/>	2.052.925	14.039.398	15,21
Methoden / Ereignisse	<input type="checkbox"/>	62.749	141.646	0,15
Module	<input type="checkbox"/>	137	2.170	<0,01
Verarbeitungsblöcke extern	<input type="checkbox"/>	14	332.038	0,36
Remote Function Call	<input type="checkbox"/>	14	332.038	0,36
Interne Tabellen	<input checked="" type="checkbox"/>	2.680.708	22.251.331	24,11
Lesende Zugriffe	<input type="checkbox"/>	2.015.812	13.458.519	14,58
Ändernde Operationen	<input type="checkbox"/>	664.896	8.792.812	9,53
Datenzugriffe intern	<input checked="" type="checkbox"/>	601.021	4.524.173	4,90
Persistente Daten	<input type="checkbox"/>	600.864	4.517.376	4,89
Transiente Daten	<input type="checkbox"/>	157	6.797	0,01
Datenzugriffe extern	<input checked="" type="checkbox"/>	9.228	43.648.819	47,30
Datenbank	<input type="checkbox"/>	9.228	43.648.819	47,30
Sonstiges	<input type="checkbox"/>	3.366	137.359	0,15
Laden / Generieren	<input type="checkbox"/>	82	36.954	0,04
Laufzeitanalyse	<input type="checkbox"/>	2	0	0,00
Nicht zugeordnet	<input type="checkbox"/>	18.740	68.234	0,07

Abbildung 8.12 Transaktion SAT – Profil »Trace-Ereignisse«

Sicht »Hitliste«

Der ABAP Trace, bzw. der in der Profil-Sicht gefilterte Teil davon, wird in der HITLISTE dargestellt (siehe Abbildung 8.13).

Treffer	Brutto [µsec]	Netto [µsec]	Brutto [%]	Netto [%]	Anweisung / Ereignis	Gerufenes Programm	Rufendes Programm
733	18.087.920	18.087.920	19,60	19,60	DB: Fetch ONRKL	SAPFK21R	SAPFK21R
3	10.913.750	10.913.750	11,82	11,83	DB: Open COSS	GP478F7SKZM2LCFH...	GP478F7SKZM2LCF...
1	6.829.184	6.829.079	7,40	7,40	Sort IT_2011	SAPFK21R	SAPFK21R
3	5.879.071	5.879.071	6,37	6,37	DB: Open COSP	GP478F7SKZM2LCFH...	GP478F7SKZM2LCF...
1	8.132.304	5.635.580	8,81	6,11	Loop At IT_2011	SAPFK21R	SAPFK21R
1	20.833.419	2.734.292	22,57	2,96	Fetch ONRKL	SAPFK21R	SAPFK21R

Abbildung 8.13 Transaktion SAT – Profil »Hitliste«

Folgende Informationen werden hier angezeigt:

- ▶ TREFFER: Anzahl der Aufrufe der Anweisung/des Ereignisses
- ▶ BRUTTO [µSEC]/[%]: absolute und relative Bruttozeit

- ▶ Netto [μ SEC]/[%]: absolute und relative Nettozeit
- ▶ ANWEISUNG / EREIGNIS
- ▶ GERUFENES PROGRAMM
- ▶ RUFENDES PROGRAMM


Die Hitliste ist zunächst nach Nettozeiten sortiert, kann für die Auswertung aber nach beliebigen Spalten sortiert werden. Durch einen Doppelklick auf einen Eintrag wird der entsprechende Programmteil im ABAP Editor dargestellt (und mit der ZURÜCK-Taste (F3) kommen Sie wieder in die Transaktion SAT).

Sicht »Datenbanktabellen«

Diese Sicht zeigt eine Übersicht der verwendeten Tabellen und Zeiten, die auf diesen Tabellen verbraucht wurden. Im Detail sind dies:

- ▶ Tabellename
- ▶ Zugriffsart: Open SQL oder natives SQL
- ▶ Anzahl der Zugriffe
- ▶ Bruttozeit
- ▶ Pufferung: ein, aus
- ▶ Pufferungsart: Einzelsatz-, generisch oder vollständig gepuffert
- ▶ Tabellenart: transparente Tabelle, View
- ▶ Kurzbeschreibung

Sicht »Zeiten«

Ähnlich wie in der HITLISTE werden hier die Zeiten einzelner Anweisungen angezeigt. Zusätzlich gibt es noch einen Experten-Modus , der die Zeiten sehr detailliert pro Anweisung und nach Benutzer- und Systemzeit getrennt darstellt:

- ▶ Zeit im Verarbeitungsblock
- ▶ Zeit in der Datenbankschnittstelle
- ▶ Zeit auf internen Tabellen
- ▶ Zeit in der Dynproverarbeitung

Sicht »Verarbeitungsblöcke«

Die Sicht VERARBEITUNGSBLÖCKE steht nur bei nicht aggregierter Aufzeichnung zur Verfügung. Hier wird die Aufrufreihenfolge und -hierarchie von Verarbeitungsblöcken (Modularisierungseinheiten)

zusammen mit Brutto- und Nettozeiten dargestellt. Falls in der Messvariante aktiviert, wird auch der Speicherverbrauch angezeigt. Es gibt die Möglichkeit, einzelne Verarbeitungsblöcke farblich hervorzuheben, wenn z. B. ein Verarbeitungsblock für mehr als 5 % der Brutto- oder Nettozeit verantwortlich ist oder der Speicherbedarf ansteigt. Die Schwellenwerte können dabei eingestellt werden.

Sicht »Aufrufhierarchie«

Zusätzlich zur Aufrufreihenfolge von Verarbeitungsblöcken zeigt die Sicht AUFRUFHIERARCHIE sämtliche aufgezeichneten Ereignisse in der richtigen Reihenfolge. Für jedes Ereignis gibt es je einen Eintrag zu Beginn und am Ende des Ereignisses. Angezeigt werden hier ebenfalls Brutto- und Nettozeiten und, falls aktiviert, auch der Speicherverbrauch.

Aufzeichnung und Auswertung mit dem ABAP Profiler

Die ABAP-Laufzeitanalyse kann auch aus den ABAP Development Tools (ADT, ABAP in Eclipse, siehe Abschnitt 2.1, »Der ABAP-Anwendungsserver im Überblick«) heraus aufgezeichnet und analysiert werden. Der Trace wird dabei analog zur Aufzeichnung mit der Transaktion SAT erstellt. Zunächst definieren Sie eine Messvariante und führen anschließend die Laufzeitanalyse innerhalb der ABAP-Perspektive aus.

1. Öffnen Sie das Dropdown-Menü am Icon PROFILE.
2. Wählen Sie PROFILE CONFIGURATIONS...
3. Wechseln Sie zum Tab TRACING.
4. Setzen Sie die gewünschten Traceeinstellungen.

Messvariante im
ABAP Profiler
definieren

Der Trace kann jetzt über den Button PROFILE gestartet werden.

Die Auswertung des ABAP Traces erfolgt in der Perspektive ABAP PROFILING.

1. Wechseln Sie zur Sicht ABAP TRACES.
2. Aktualisieren Sie die Traceliste Ihres Projekts.
3. Starten Sie mit einem Doppelklick auf den Trace die Auswertung.

ABAP Trace
auswerten

Bei einem aggregierten Trace stehen die Karteireiter OVERVIEW, HITLIST und DATABASE ACCESSES mit der gleichen Funktionalität wie in

der Transaktion SAT zur Verfügung. Bei einem nicht aggregiert aufgezeichneten Trace gibt es außerdem noch die Reiter CALL TREE und CALL TIMELINE. Der CALL TREE entspricht den Verarbeitungsblöcken in der Transaktion SAT. Die CALL TIMELINE dagegen ist eine Sicht, die es so nicht in der Transaktion SAT gibt. Sie stellt die Aufrufhierarchie im Gegensatz zur Transaktion SAT grafisch dar und zeigt damit anschaulich den zeitlichen Programmablauf (siehe Abbildung 8.14).

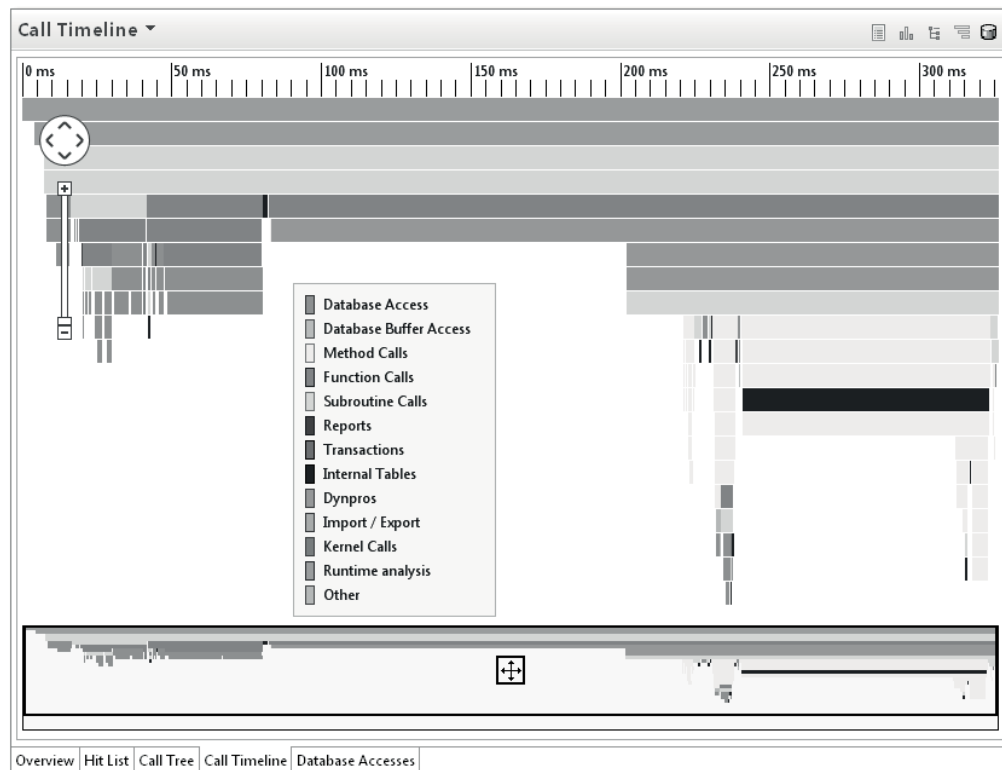


Abbildung 8.14 ABAP Profiler – Call Timeline

8.2.4 Single Activity Trace (ST12)

Die in den vorigen Abschnitten beschriebenen Werkzeuge Laufzeitstatistiken (STAD), Performance Trace (ST05) und Laufzeitanalyse (SAT) werden in der Praxis häufig gemeinsam eingesetzt, um alle Laufzeitinformationen eines Programms, sowohl auf der Datenbank als auch in ABAP, zu bestimmen. Dafür wurde die Single Transaction Analysis (Transaktion ST12) entwickelt, ein Werkzeug, das ursprünglich nur vom SAP-Support in der Servicelieferung eingesetzt wurde.

Die Transaktion ST12 wird mit dem Plug-in *Service Tools for Applications (ST-A/PI)* ausgeliefert und daher nicht offiziell als Teil des Standardprodukts unterstützt und ist nur in Englisch verfügbar. SAP-Hinweis 69455 beschreibt die Installation des Plug-ins.

Eine Einführung in die Bedienung der Transaktion ST12 finden Sie in SAP-Hinweis 755977. Eine ausführlichere Beschreibung der Funktionen gibt es im SAP Community Network unter: <http://wiki.scn.sap.com/wiki/display/ABAP/Single+Transaction+Analysis>.

In diesem Wiki und den dort aufgeführten Blogs werden die verschiedenen Traceszenarien beschrieben.

Im Vergleich zu den einzelnen Transaktionen SAT und ST05 bietet die Transaktion ST12 ein paar nützliche Erweiterungen, die wir im Folgenden beschreiben.

ST12 im Vergleich zu SAT und ST05

Zentrale Administration der Traces

Traces werden zentral über eine Oberfläche gestartet und gestoppt und lassen sich damit einfacher bedienen. Die Traceergebnisse und Einzelsatzstatistiken werden in der Datenbank gespeichert (Transaktion ST05, Daten in aggregierter Form) und stehen dauerhaft zur Verfügung.

Integration von ABAP- und SQL-Trace-Ergebnissen

Aus dem zentralen ABAP Trace heraus können Sie bei SQL-Anweisungen in die entsprechende SQL-Trace-Aufzeichnung wechseln und haben damit Zugriff auf Detailinformationen des SQL Traces, wie z. B. den SQL-Ausführungsplan oder die Anzahl identischer Datensätze.

Verknüpfung mit Ergebnissen des Code Inspectors (statische Prüfung)

Für einen aufgezeichneten ABAP Trace können Sie aus der Transaktion ST12 heraus eine Code-Inspector-Prüfung starten. Die Ergebnisse werden für die geprüften Anweisungen im ABAP Trace dargestellt und bieten damit eine einfach zu nutzende Hilfe bei der Programmoptimierung. Abbildung 8.15 zeigt beispielhaft eine solche Meldung. Die Code-Inspector-Prüfung kann optional über das Menü SHOW/HIDE • CODE INSPECTOR CHECKS aktiviert werden.

Call	Number	Gross	Net	Gross in %	Net (%)	Program (called program)	Code Inspector Checks
DELETE GT_SY...	6.606	12.897.531	12.897.531	1,8	1,8	YFSCMO_SYNC_BUSAB	Sequential Read Access for a Standard Table
Select KNB5	1	425.999	425.999	0,1	0,1	SAPLYFCMO_GENERAL...	Large table KNB5: No first field of a table index
Select KNB1	1	584.242	584.242	0,1	0,1	SAPLYFCMO_GENERAL...	Large table KNB1: No first field of a table index

Abbildung 8.15 Integration des Code Inspectors in Transaktion ST12

Integration der Statistiksätze (STAD)

Nach Aufzeichnung der Messung werden von der Transaktion ST12 automatisch zugehörige Statistiksätze zusammen mit dem Trace abgespeichert. Diese können später mit dem Trace ausgewertet werden.

Erweiterte Laufzeitinformationen: Time/Execution und Time/Record

Für Anweisungen auf interne und Datenbanktabellen lassen sich durchschnittliche Zugriffszeiten einblenden:

- ▶ Interne Tabellen: Time/Execution
- ▶ Datenbanktabellen: Time/Execution und Time/Record

Die Informationen lassen sich über das Menü SHOW/HIDE • TIME PER RECORD OR EXEC einblenden.

Aufrufhierarchie auch bei aggregierter Traceaufzeichnung verfügbar

Um die Aufrufhierarchie einzelner Anweisungen anzeigen zu können, muss ein nicht aggregierter ABAP Trace aufgezeichnet werden, was in der Transaktion ST12 aus Performancegründen nicht möglich ist. Die Transaktion kann allerdings aus aggregierten Traceaufzeichnungen eine *wahrscheinliche* Aufrufhierarchie bestimmen, die in den meisten Fällen der tatsächlichen entspricht. Abbildung 8.16 zeigt die Aufrufhierarchie einer DELETE-Anweisung, die über die Buttons ein- bzw. ausgeschaltet werden kann. Zur besseren Lesbarkeit können Sie die nicht zur Aufrufhierarchie gehörenden Zeilen über den Button ausblenden.

Call	Number	Call Hierarchy	Gross	Net	Gross (%)	Net (%)	Program (called program)	Code Inspector Checks
PERFORM USER_COMMAND_NEW	1	↓	709.434.226	54	100,0	0,0	SAPLKKBL	
PERFORM (ext) USER_COMMAND	1	↓	709.433.810	38	100,0	0,0	SAPLBTCB	
PERFORM DEBUG_JOB_SM37B	1	↓	709.433.419	21	100,0	0,0	SAPLBTCB	
LOOP AT DBG_STEPLIST	1	↓	709.423.660	10.016	100,0	0,0	SAPLBTCB	
SUBMIT REPORT YFSCMO_SYNC_	1	↓	709.413.588	25.045	100,0	0,0	SAPLBTCB	
Program YFSCMO_SYNC_BUSAB	1	↓	709.381.773	3.279	100,0	0,0		
Dynpro Entry	1	↓	693.935.966	19	97,8	0,0	SAPMSSYO	
System Event LDB-Processing	1	↓	691.523.893	6.192.610	97,5	0,9	YFSCMO_SYNC_BUSAB	
LOOP AT GT_BUKRS	1	↓	683.850.395	608.092.681	96,4	85,7	YFSCMO_SYNC_BUSAB	
PERFORM BUSAB_CLEAN	1	↓	65.517.653	38.684.484	9,2	5,5	YFSCMO_SYNC_BUSAB	
LOOP AT GT_SYNC_BUSAB	1	↓	25.260.646	11.658.041	3,6	1,6	YFSCMO_SYNC_BUSAB	
Dynpro Entry	2	↓	15.445.721	116	2,2	0,0	YFSCMO_SYNC_BUSAB	
PBO Dynpro SAPMSSYO	2	↓	14.084.716	60	2,0	0,0	SAPMSSYO	
DELETE GT_SYNC_BUSAB	6.606	↓	12.897.531	12.897.531	1,8	1,8	YFSCMO_SYNC_BUSAB	Sequential Read Access

Abbildung 8.16 Die Aufrufhierarchie in der Transaktion ST12

Automatischer Start von Folgetraces

Für Batchprozesse, die länger als die maximale Tracezeit (4294 s) laufen und getraced werden sollen, können Folgetraces eingeplant werden. Diese werden automatisch mit Ende des Vorgängertraces gestartet. Es handelt sich hierbei jeweils um eigenständige Traceaufzeichnungen.

8.2.5 SQL Monitor (SQLM)

Die Grundidee des SQL Monitors ist es, Laufzeitinformationen zu SQL-Anweisungen im Datenbankinterface DBI zu aggregieren und zu persistieren.

Im Gegensatz zum SQL-Cache der Datenbank, der weitere datenbankspezifische Informationen zur SQL-Anweisung, wie etwa die Anzahl der gelesenen Seiten oder die benötigte Zeit für I/O oder CPU und den Ausführungsplan (Execution Plan) bietet, werden im SQL Monitor die Daten mit weiteren Informationen zum ABAP-Programm und dessen Laufzeit und der Aufrufkontext, in dem die Anweisung ausgeführt wurde, festgehalten. Diese beiden Datenquellen ergänzen sich also und bieten jeweils eigene Zusatzinformationen zu SQL-Anweisungen:

Abgrenzung zum SQL-Cache der Datenbank

- ▶ der SQL Monitor im Kontext zum ABAP-Programm
- ▶ der SQL-Cache im Kontext zur Datenbank und SQL-Ausführung

[*] **Verfügbarkeit des SQL Monitors**

Beim SQL Monitor handelt es sich um ein relativ neues Werkzeug, das seit dem SAP-NetWeaver-Release 7.02 (mit Kernel 7.21) zur Verfügung steht. Beachten Sie dazu die SAP-Hinweise 1855676, 1855676 und 1824769.

Aufzeichnung aktivieren

Der SQL Monitor wird über die Transaktion SQLM aktiviert. Hier können Sie den SQL Monitor auf allen oder nur bestimmten Applikationsservern aktivieren und den Zeitraum festlegen (standardmäßig eine Woche). In Abbildung 8.17 sehen Sie den Einstiegsbildschirm der Transaktion SQLM nach der Aktivierung.

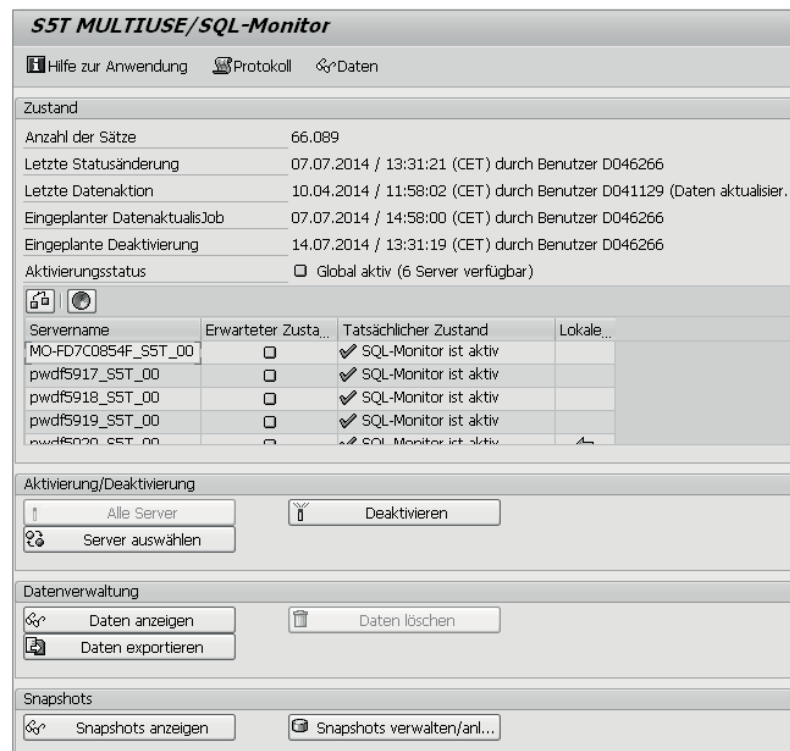


Abbildung 8.17 SQL Monitor nach Aktivierung

Nach der Aktivierung werden für jede ausgeführte SQL-Anweisung Daten gesammelt und aggregiert. Die Datensammlung geschieht im Hauptspeicher, und die Daten werden asynchron in eine Datenbank-

tabelle geschrieben. Nach ca. einer Stunde stehen sie zur Auswertung in der Transaktion SQLM bereit. Die Daten werden dabei über einen Hintergrundjob bereitgestellt, um die Laufzeitauswirkungen dieser Messungen minimal zu halten.

Aufwand für SQL Monitor

Der SQL Monitor ist bei SAP und vielen Kunden in großen produktiven Systemen rund um die Uhr aktiv, ohne zu Beeinträchtigungen zu führen. Der Aufwand für dieses Monitoring kann vernachlässigt werden.

[+]

Daten auswerten

Um die Daten des SQL Monitors auszuwerten, klicken Sie auf DATEN ANZEIGEN (siehe Abbildung 8.17). Sie erhalten dann eine Selektionsmaske, wie sie in Abbildung 8.18 gezeigt ist.

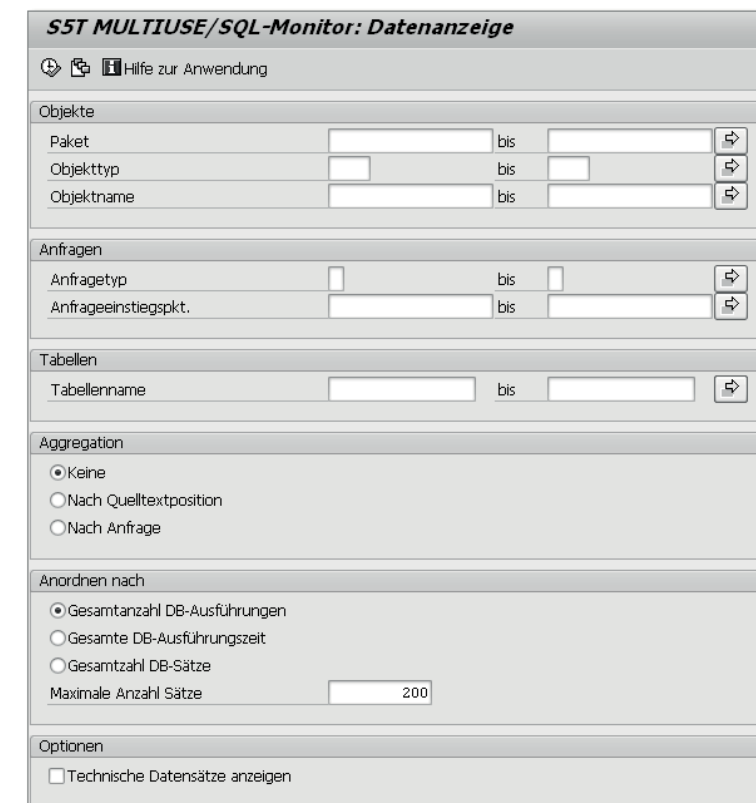


Abbildung 8.18 SQL Monitor – Selektionsmaske

Wir beschreiben Ihnen nun die wichtigsten Einstellungen und Filteroptionen:

- Filtermöglichkeiten**
- ▶ Im Bereich **OBJEKTE** können Sie auf ABAP-Entwicklungsobjekte filtern.
 - ▶ Im Bereich **ANFRAGEN** können Sie auf Hauptprogramme filtern, dabei handelt es sich um die ersten Einträge im ABAP-Call-Stack (also um einen Report, eine Transaktion, eine URL, ...).
 - ▶ Im Bereich **TABELLEN** können Sie nach Tabellen filtern.
 - ▶ Bei der **AGGREGATION** stellen Sie ein, ob die SQL-Anweisungen ohne Aggregation (**KEINE**), **NACH** der **QUELLEXTPOSITION** oder der **ANFRAGE** (Hauptprogramm) aggregiert dargestellt werden sollen.
 - ▶ Im Bereich **ANORDNEN NACH** können Sie die Sortierung der Ausgabe nach verschiedenen Kategorien einstellen und diese limitieren.
 - ▶ Mit der Option **TECHNISCHE DATENSÄTZE ANZEIGEN** werden auch SQL-Anweisungen dargestellt, die nicht direkt aus dem ABAP-Programm, sondern vom SAP-Kernel stammen (wie z. B. das Laden gepufferter Tabellen).

Im Folgenden beschreiben wir Ihnen detailliert, wie Sie ein ABAP-Programm mit dem SQL Monitor analysieren können. Anschließend gehen wir noch kurz auf weitere Analysemöglichkeiten mit dem SQL Monitor ein.

Analyse eines ABAP-Programms

Um die teuersten SQL-Anweisungen in einem bestimmten ABAP-Programm zu analysieren, können Sie wie folgt vorgehen:

- ▶ Geben Sie das zu untersuchende Programm im Bereich der **ANFRAGEN** an.
- ▶ Wählen Sie bei der **AGGREGATION NACH ANFRAGE** aus.
- ▶ Sortieren Sie nach **GESAMTE DB-AUSFÜHRUNGSZEIT**
- ▶ Führen Sie die Selektion aus.

DB-Zeit	Gesamtzeit	DB-Zt/GZt	Ges.DBAusf	Sätze	Mittl.DB-Z	Sessions	Anfragetyp	EinstPunkt	DB-Anweis.	Max.DBZeit	Min.DB-Z	DB-AbwZeit	Max. Sätze
1.531,495	3.626,770	42,23	1.648	23,995	0,929	2	Transaktion	ZkBX	53	33,248	0,543	1,434	7.097

Abbildung 8.19 SQL Monitor – nach Anfrage aggregiert

Unterschied zwischen Anfrage und Objekt

[*]

Im SQL Monitor unterscheiden wir die Selektion nach einem Objekt (ABAP-Entwicklungsobjekt wie ein Report, eine Klasse usw.) und einer Anfrage (Einstiegspunkt in eine Aufrufhierarchie, wie z. B. ein Report, eine Transaktion, ein Hintergrundjob usw.).

Wenn Sie eine Selektion nach Objekt vornehmen, werden nur SQL-Anweisungen angezeigt, die direkt in diesem Entwicklungsobjekt stehen. Beispiel: Sie sehen alle SQL-Anweisungen, die im ABAP-Report vorkommen. SQL-Anweisungen, die in einem Funktionsbaustein ausgeführt werden, der von diesem Report aufgerufen wird, werden nicht angezeigt.

Wenn Sie eine Selektion nach Anfrage vornehmen, werden alle SQL-Anweisungen, die innerhalb und unterhalb des Objekts vorkommen, angezeigt, das heißt in allen von diesem Objekt aufgerufenen Modularisierungseinheiten wie Funktionsbausteinen und Methoden usw. Beispiel: Sie sehen alle SQL-Anweisungen, die in einem ABAP-Report vorkommen, und alle SQL-Anweisungen, die in Funktionsbausteinen, Methoden usw. ausgeführt werden, die von diesem Report aufgerufen werden.

Wir empfehlen Ihnen daher, die Suche nach Anfrage zu aggregieren, um alle SQL-Anweisungen innerhalb und unterhalb der Anwendung zu sehen, die Sie analysieren wollen.

Wenn Sie die Selektion nach Anfrage aggregiert ausführen, erhalten Sie zunächst eine Zeile mit aggregierten Werten (siehe Abbildung 8.19). Diese Liste enthält die in Tabelle 8.8 beschriebenen Spalten.

Zusammenfassung pro Anfrage

Spalte	Beschreibung
DB-ZEIT	gesamte Datenbankzeit der Anwendung seit Aktivierung des SQL Monitors
GESAMTZEIT	gesamte Laufzeit der Anwendung seit Aktivierung des SQL Monitors
DB-Zt/GZt	Anteil der Datenbankzeit von der Gesamtzeit in Prozent
Ges.DBAUSF	Summe der ausgeführten SQL-Anweisungen in dieser Anwendung
SÄTZE	Summe aller gelesenen Datensätze für diese Anwendung
MITTL.DB-Z	durchschnittliche Laufzeit aller ausgeführten SQL-Anweisungen

Tabelle 8.8 Angezeigte Spalten bei Selektion nach Anfrage

Spalte	Beschreibung
SESSIONS	Anzahl der Ausführungen für diese Anfrage (Transaktion, Report, Hintergrundjob usw.). Bei RFC-Bausteinen handelt es sich um die Anzahl der Aufrufe, bei denen eine RFC-Verbindung geöffnet werden musste; Aufrufe innerhalb einer geöffneten RFC-Verbindung werden nicht gezählt.
ANFRAGETYP	Art der Anfrage (Transaktion, Report, Hintergrundjob usw.). Von hier aus können Sie direkt zum Objekt in der Entwicklungsumgebung navigieren.
EINSTPUNKT	erster Eintrag in der ABAP-Aufrufhierarchie
DB-ANWEIS.	Anzahl der unterschiedlichen SQL-Anweisungen in der Anwendung (innerhalb und unterhalb, wenn nach Anfrage selektiert wurde, nur innerhalb, wenn nach Objekt selektiert wurde). Bei einem Doppelklick navigieren Sie zu der detaillierten Liste dieser SQL-Anweisungen.
MAX.DBZEIT/ MIN.DBZEIT/ DB-ABWZEIT	Zeit für die langsamste/schnellste SQL-Anweisung in der Anwendung; Standardabweichung der Laufzeit
MAX.SÄTZE/ MIN.SÄTZE/ MWE.SÄTZE/ ABWEISÄTZE	maximale, minimale, durchschnittliche Anzahl und Standardabweichung der verarbeiteten Datensätze für die SQL-Anweisungen in der Anwendung

Tabelle 8.8 Angezeigte Spalten bei Selektion nach Anfrage (Forts.)

SQL-Profil Wenn Sie auf die Spalte DB-ANWEIS. klicken, gelangen Sie auf die nächste Ebene, auf der alle SQL-Anweisungen dieser Anwendung aufgelistet sind (siehe Abbildung 8.20).

DB-Zeit	DB-Zeit[%]	DB-Zt/GZt	DB-Ausf.	DB-Auff[%]	Sätze	Sätze [%]	Mittl.DB-Z	Tabellennamen	SQL-Operationstyp	Obj-Typ	Objektname	Include-Name
316,092	20,64	8,72	322	19,54	1,438	5,99	0,982	PAT03	SELECT (Open SQL)	FUGR	OCS_CRM	LOCS_CRMU08
259,722	16,96	7,16	322	19,54	0	0,00	0,807	PAT03	SELECT (Open SQL)	FUGR	OCS_CRM	LOCS_CRMU08
251,509	16,42	6,93	322	19,54	0	0,00	0,781	PAT03_PRE	SELECT (Open SQL)	FUGR	OCS_CRM	LOCS_CRMF01
249,061	16,26	6,87	322	19,54	2	0,01	0,773	PAT03_PRE	SELECT (Open SQL)	FUGR	OCS_CRM	LOCS_CRMF01
96,294	6,29	2,66	75	4,55	3,525	14,69	1,284	SFW_ACTIVE_B2	SELECT (Open SQL)	CLAS	CL_SFW_HELPER	CL_SFW_HELPER
70,170	4,58	1,93	75	4,55	71	0,30	0,936	PAT06	SELECT (Open SQL)	FUGR	OCS_CRM	LOCS_CRMU08
63,499	4,15	1,75	2	0,12	4,146	17,28	31,750	PAT03	SELECT (Open SQL)	CLAS	CL_OCS_ENVIRONMENT	CL_OCS_ENVIRON
49,324	3,22	1,36	75	4,55	0	0,00	0,658	SFW_ACTIVE_BFUNC	SELECT (Open SQL)	CLAS	CL_SFW_HELPER	CL_SFW_HELPER
47,248	3,09	1,30	75	4,55	75	0,31	0,630	SFW_ACTIVE_B1	SELECT (Open SQL)	CLAS	CL_SFW_HELPER	CL_SFW_HELPER
30,807	2,01	0,85	1	0,06	2,073	8,64	30,807	PAT03	SELECT (Open SQL)	FUGR	OCS_CORE	LOCS_COREU32
22,536	1,47	0,62	1	0,06	7,097	29,58	22,536	EUOBJIT	SELECT (Open SQL)	CLAS	CL_WB_REGISTRY_SHM	CL_WB_REGISTRY
4,122	0,27	0,11	1	0,06	462	1,93	4,122	EUOBJEDIT	SELECT (Open SQL)	CLAS	CL_WB_REGISTRY_SHM	CL_WB_REGISTRY
3,952	0,26	0,11	1	0,06	462	1,93	3,952	EUOBJEDIT	SELECT (Open SQL)	CLAS	CL_WB_OBJTYPE_PERSI	CL_WB_OBJTYPE
3,793	0,25	0,10	1	0,06	1,088	4,53	3,793	OBJT	SELECT (Open SQL)	FUGR	TR_OBJECTS	LTR_OBJECTSP02

Abbildung 8.20 SQL Monitor – Liste der SQL-Anweisungen für eine Anfrage

Bei diesem SQL-Profil handelt es sich im Prinzip um eine ähnliche Darstellung wie bei der Liste der strukturgleichen Anweisungen im SQL Trace der Transaktion ST05. Sie sehen hier, welche SQL-Anweisungen den größten Teil der Datenbankzeit ausmachen.

In Tabelle 8.9 gehen wir auf die wichtigsten Spalten dieser Ansicht ein.

Spalte	Beschreibung
DB-ZEIT	gesamte Datenbankzeit der SQL-Anweisung seit Aktivierung des SQL Monitors
DB-ZEIT [%]	Zeitanteil dieser SQL-Anweisung an der gesamten Datenbankzeit
DB-ZT/GZT	Zeitanteil dieser SQL-Anweisung an der gesamten Antwortzeit
DB-AUSFÜH.	Summe der Ausführungen dieser SQL-Anweisung
DB-AUSF [%]	Anteil der Ausführungen dieser SQL-Anweisung an den gesamten Ausführungen der Anwendung
SÄTZE	Summe aller verarbeiteten Datensätze dieser SQL-Anweisung
SÄTZE [%]	Anteil der verarbeiteten Datensätze dieser SQL-Anweisung an den insgesamt verarbeiteten Datensätzen der Anwendung
MITTL.DB-Z	durchschnittliche Antwortzeit dieser SQL-Anweisung
TABELLENNAMEN	Name der Tabellen oder Views, auf die sich die SQL-Anweisung bezieht. Bei Joins werden die Tabellen kommasepariert gelistet.
SQL-OPERATIONSTYP	Typ der SQL-Anweisung, wie SELECT, INSERT, UPDATE, DELETE, SELECT (FAE), Load Buffer (Tabellenpuffer), Native SQL usw.
ABAP-QUELLTEXT-FRAGMENT	Text der SQL-Anweisung im ABAP-Programm
GEÄNDERT	Kennzeichen, ob die SQL-Anweisung oder das Programm seit Beginn der Aufzeichnung geändert wurde
INT.SESS.	Anzahl der Sessions (Anmeldungen, interne Modi), in denen die SQL-Anweisung ausgeführt wurde
AUSF/SESS.	Anzahl der Ausführungen in einer Session

Tabelle 8.9 Angezeigte Spalten bei DB-Anweisungen (SQL-Profil einer Anfrage)

Spalte	Beschreibung
MAX.DBZEIT/ MIN.DBZEIT/ DB-ABWZEIT	langsamste/schnellste Zeit und Standardabweichung für diese SQL-Anweisung
MAX.SÄTZE/ MIN.SÄTZE/ MWE.SÄTZE/ ABWEISÄTZE	maximale, minimale, durchschnittliche Anzahl und Standardabweichung der verarbeiteten Datensätze für diese SQL-Anweisung

Tabelle 8.9 Angezeigte Spalten bei DB-Anweisungen (SQL-Profil einer Anfrage) (Forts.)

[+] Schneller Überblick über typische Probleme bei Datenbankzugriffen

Mit der hier vorgestellten Analysemethode können Sie recht schnell und komfortabel (ohne, dass Sie einen Trace erstellen müssen) die typischen Probleme von vielen verarbeiteten Datensätzen (siehe TBD), vielen ausgeführten SQL-Anweisungen (siehe TBD) und hohe Zeiten pro Ausführung/Datensatz (siehe TBD) erkennen. Im SQL Monitor können Sie also schnell sehen, ob solche Probleme vorliegen. Um das Problem der identischen SQL-Anweisungen zu erkennen, brauchen Sie allerdings den SQL Trace (siehe Abschnitt 8.2.2, »Performance Trace (ST05)«). Für die anderen Probleme hängt dann vom konkreten Fall ab, ob eine Analyse des Quelltextes zur Problembhebung genügt oder ob ein weiterer SQL Trace mit weiteren Daten, wie z. B. Parameterwerten oder Aufrufhierarchien, erstellt werden muss, um die notwendigen Daten für die Problemlösung zu bekommen.

Weitere Analysemöglichkeiten

Mit dem SQL Monitor können Sie neben der Analyse eines ABAP-Programms auch noch weitere Analysen vornehmen. Solche Analysen werden üblicherweise von Systemadministratoren oder Qualitätsmanagern durchgeführt, da sie statt nur einer Anwendung das gesamte System oder größere Entwicklungsbereiche betrachten. Sie haben hier folgende Möglichkeiten:

- ▶ Analyse teurer SQL-Anweisungen in einem bestimmten Entwicklungsobjekt (Report, Transaktion, Paket, ...). Füllen Sie dazu die entsprechenden Filter bei OBJEKTE aus.
- ▶ Systemweite Analyse teurer SQL-Anweisungen. Wählen Sie dazu bei der AGGREGATION, je nachdem, in welcher Aggregation Sie die Daten betrachten wollen, NACH QUELLTEXTPOSITON oder KEINE.

Wenn Sie keine Aggregation gewählt haben, erhalten Sie je Aufrufer (erster Eintrag in der Aufrufhierarchie) eine Zeile. Diese können aber auch von der Aggregation nach Quelltextposition in die nicht aggregierte Liste verzweigen (Drill-down).

- ▶ Systemweite Analyse von Zugriffen auf eine bestimmte Tabelle. Füllen Sie dazu die entsprechenden Filter bei TABELLEN.
- ▶ Systemweite Analyse der teuersten Prozesse (Anfragen/Hauptprogramme). Wählen Sie bei der AGGREGATION NACH ANFRAGE, und sortieren Sie z. B. nach GESAMTE DB-AUSFÜHRUNGSZEIT.

8.2.6 Laufzeitprüfungs-Monitor (SRTCM)

Mit diesem Monitor können Sie bestimmte Prüfungen aktivieren, die während der Laufzeit von ABAP-Programmen durchgeführt werden. Anschließend können Sie die Ergebnisse dieser Prüfungen analysieren.

Die Transaktion SRTCM beinhaltet im Moment die folgenden beiden Prüfungen:

- ▶ **Leere Tabelle in der Klausel FOR ALL ENTRIES**
Diese Prüfung zeichnet alle Ausführungen von SELECT-Anweisungen auf, bei denen die interne Tabelle der FOR ALL ENTRIES-Klausel leer ist. Mit dieser Prüfung können Sie Probleme identifizieren, bei denen zu viele Datensätze von der Datenbank gelesen werden (siehe Abschnitt 7.1.1, »Verarbeitung großer Datenmengen«).
- ▶ **ORDER BY oder SORT fehlt nach SELECT**
Diese Prüfung zeichnet alle Ausführungen von SELECT-Anweisungen ohne ORDER BY oder nachfolgende SORT-Anweisung auf, bei denen die interne Tabelle anschließend eine Sortierung erfordert. Dies ist z. B. der Fall, wenn die Anweisung READ . . BINARY SEARCH verwendet wird, aber auch bei anderen Anweisungen.

Im Folgenden zeigen wir Ihnen, wie Sie die Datenaufzeichnung aktivieren und anschließend die Daten auswerten können.

Monitor aktivieren

Nachdem Sie die Transaktion SRTCM gestartet haben, erhalten Sie einen Überblick über den Status der verfügbaren Prüfungen (siehe

Abbildung 8.21). Sie sehen dort den Aktivierungsstatus und weitere Informationen zu den zwei Prüfungen. Über den Button INFORMATION können Sie die Dokumentation zur Prüfung aufrufen. Um eine Prüfung zu aktivieren, klicken Sie auf GLOBAL AKTIVIEREN. Im folgenden Popup-Menü geben Sie den Zeitpunkt an, an dem die Prüfung wieder deaktiviert werden soll.



Abbildung 8.21 SRTCM – Übersicht

Sie können die Prüfung auch nur auf ausgewählten Applikationsservern aktivieren, indem Sie die entsprechende Drucktaste wählen.

Daten auswerten

Um die Daten auszuwerten, klicken Sie auf ERGEBNISSE ANZEIGEN (siehe Abbildung 8.21). Anschließend wählen Sie die gewünschte Prüfung im Bereich LAUFZEITPRÜFUNGS-AUSWAHL aus und wählen ERGEBNISSE ANZEIGEN (siehe Abbildung 8.22). Alternativ können Sie die Transaktion SRTCMD aufrufen, um die Daten direkt zu selektieren.

Im Ergebnis sehen Sie dann eine Liste der Prüfergebnisse (siehe Abbildung 8.23). Mit einem Klick auf den Link in den Spalten INCLUDE und INCL.ZEILE können Sie direkt zum ABAP-Quelltext navigieren. Die Spalte GEÄNDERT zeigt an, ob der ABAP-Quelltext in der Zeit zwischen der Prüfung und der Auswertung geändert wurde.

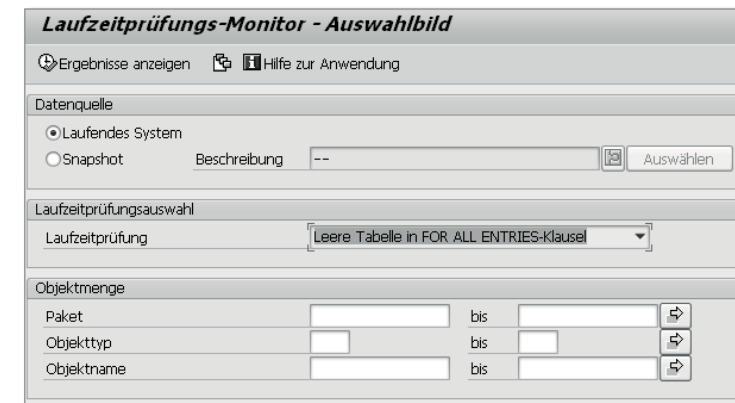


Abbildung 8.22 Transaktion SRTCM – Ergebnisse auswählen

Vorkommen	Vork.Datum	VorkomZeit	Paket	Objekttyp	Objektname	Include	Incl.Zeile	Geändert
8	08.07.2014	15:27:57	YFDM	PROG	YF_INVOICE_DUE_ALERT	YF_INVOICE_DUE_ALERT_F01	451	
2	24.06.2014	12:33:25	YIDM_AUTH	PROG	YIDM_AUTH_DEL_UNUSED_AUTHS	YIDM_AUTH_DEL_UNUSED_AUTHS	601	
1	03.07.2014	10:19:33	YMMO	PROG	ZM_IT_MAT_INF	ZM_IT_MAT_INF	955	
1	03.07.2014	10:19:34	YMMO	PROG	ZM_IT_MAT_INF	ZM_IT_MAT_INF	995	
1	03.07.2014	10:19:34	YMMO	PROG	ZM_IT_MAT_INF	ZM_IT_MAT_INF	1.015	
4	06.06.2014	12:09:09	YMMO	PROG	ZMMMAT30	ZMFORM30	109	

Abbildung 8.23 Transaktion SRTCM – Ergebnisse anzeigen

8.3 Kombinierte Auswertung statischer und dynamischer Prüfungen (SWLT)

Die SQL Performance Tuning Worklist (SWLT) ist seit dem Release SAP NetWeaver AS ABAP 7.4 verfügbar. Mit diesem Werkzeug können Sie Daten aus einer statischen Codeanalyse mit Laufzeitmessungen kombinieren. Dies erlaubt Ihnen, schnell zu erkennen, wo Optimierungspotenzial vorhanden und eine Performanceoptimierung am zielführendsten ist.


Als statische Datenquellen stehen der Code Inspector (SCI, siehe Abschnitt 5.1, »Qualitätssicherung mit dem Code Inspector«) oder das ABAP Test Cockpit (ATC, siehe Abschnitt 5.2, »Qualitätssicherung mit dem ABAP Test Cockpit«) zur Verfügung. An dynamischen Datenquellen stehen der SQL Monitor (SQLM, siehe Abschnitt 8.2.5, »SQL Monitor (SQLM)«) und der Coverage Analyzer (SCOV) zur Ver-

Datenquellen

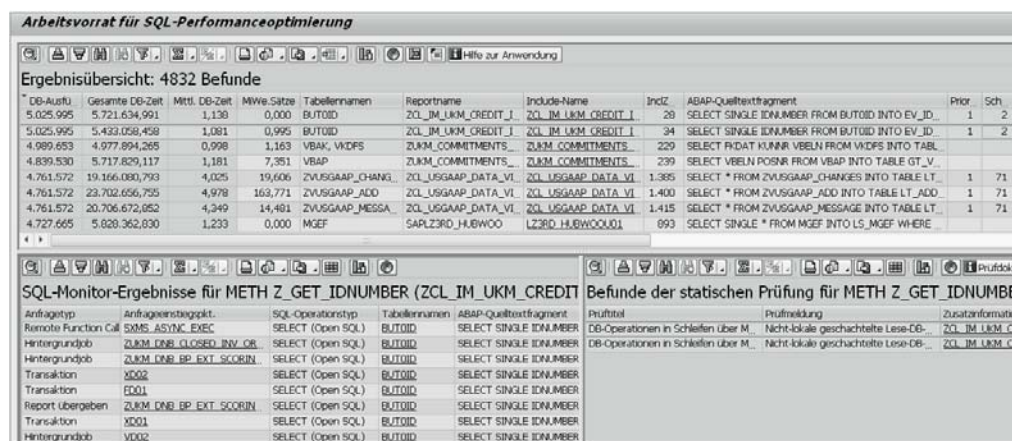
fügung. Der Coverage Analyzer liefert die Anzahl der Ausführungen der Verarbeitungsblöcke. Diese liefern zusätzliche Laufzeitinformationen und können genutzt werden, wenn keine SQL-Monitor-Daten verfügbar sind.

Aus den verschiedenen Kombinationsmöglichkeiten zeigen wir Ihnen hier die Verbindung von Code-Inspector- und SQLM-Datenquellen. Dazu gehen Sie wie folgt vor:

SCI- und SQLM-Daten verknüpfen

- ▶ Auf der Registerkarte ALLGEMEIN nehmen Sie keine Einschränkungen oder Änderungen vor.
- ▶ Auf der Registerkarte STATISCHE PRÜFUNGEN setzen Sie den Haken bei DATEN DER STATISCHEN PRÜFUNGEN VERWENDEN. Anschließend wählen Sie eine vorher ausgeführte Code-Inspector- oder ABAP-Test-Cockpit-(ATC-)Prüfung aus. Diese kann sowohl aus dem lokalen als auch aus einem entfernten System (über das Feld RFC-DESTINATION) verwendet werden.
- ▶ Auf der Registerkarte SQL MONITOR setzen Sie den Haken bei SQL-MONITOR-DATEN VERWENDEN. Zunächst muss dazu ein SQL-Monitor-Snapshot im System erstellt werden. Als Datenquelle können Sie das lokale oder ein per RFC verbundenes entferntes System verwenden. Zusätzlich haben Sie noch die Möglichkeit, einen Snapshot per Dateiimport zu erstellen.
- ▶ Führen Sie die Analyse über das Symbol  aus.

Das Ergebnis wird angezeigt (siehe Abbildung 8.24).



DB-Auftf.	Gesamte DB-Zeit	Mittl. DB-Zeit	MWe.Sätze	Tabellennamen	Reportname	Include-Name	IncZ	ABAP-Quelltextfragment	Prior.	Sch.
5.025.995	5.721.634,991	1,138	0,000	BUT0ID	ZCL_IM_UKM_CREDIT_I	ZCL_IM_UKM_CREDIT_I	28	SELECT SINGLE IDNUMBER FROM BUT0ID INTO EV_ID_	1	2
5.025.995	5.433.058,458	1,081	0,995	BUT0ID	ZCL_IM_UKM_CREDIT_I	ZCL_IM_UKM_CREDIT_I	24	SELECT SINGLE IDNUMBER FROM BUT0ID INTO EV_ID_	1	2
4.989.653	4.077.894,265	0,998	1,163	VBAP, VKDPS	ZUKM_COMMITMENTS_	ZUKM_COMMITMENTS_	229	SELECT FKDAT KUNNR VBELN FROM VKDPS INTO TABL		
4.839.530	5.717.829,117	1,181	7,351	VBAP	ZUKM_COMMITMENTS_	ZUKM_COMMITMENTS_	239	SELECT VBELN POSNR FROM VBAP INTO TABLE GT_V		
4.761.572	19.166.080,793	4,025	19,606	ZVUSGAAP_CHANG	ZCL_USGAAP_DATA_VI	ZCL_USGAAP_DATA_VI	1.385	SELECT * FROM ZVUSGAAP_CHANGES INTO TABLE LT_	1	71
4.761.572	23.702.656,755	4,978	162,771	ZVUSGAAP_ADD	ZCL_USGAAP_DATA_VI	ZCL_USGAAP_DATA_VI	1.400	SELECT * FROM ZVUSGAAP_ADD INTO TABLE LT_ADD	1	71
4.761.572	20.706.672,652	4,349	14,481	ZVUSGAAP_MESSA	ZCL_USGAAP_DATA_VI	ZCL_USGAAP_DATA_VI	1.415	SELECT * FROM ZVUSGAAP_MESSAGE INTO TABLE LT_	1	71
4.727.665	5.828.362,830	1,233	0,000	MGEF	SAPL3RD_HUBWOOD	LZ3RD_HUBWOOD01	893	SELECT SINGLE * FROM MGEF INTO LS_MGEF WHERE		

Anfragetyp	Remote Function Call	Hintergrundjob	Transaktion	Report übergeben	Transaktion	Hintergrundjob	SQL-Operatortyp	Tabellennamen	ABAP-Quelltextfragment	Prüftitel	Prüfmeldung	Zusatzinformation
Remote Function Call	SXMS_ASYNC_EXEC						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER	DB-Operationen in Schleifen über M...	Nicht-lokale geschachtelte Lese-DB...	ZCL_IM_UKM_C...
Hintergrundjob	ZUKM_DNB_CLOSED_INV_CR						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER	DB-Operationen in Schleifen über M...	Nicht-lokale geschachtelte Lese-DB...	ZCL_IM_UKM_C...
Hintergrundjob	ZUKM_DNB_BP_EXT_SCORIN						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			
Transaktion	XD02						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			
Transaktion	ED01						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			
Report übergeben	ZUKM_DNB_BP_EXT_SCORIN						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			
Transaktion	XD01						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			
Hintergrundjob	VD02						SELECT (Open SQL)	BUT0ID	SELECT SINGLE IDNUMBER			

Abbildung 8.24 Transaktion SWLT – Ergebnis

Im oberen Bereich des Bildschirms werden die Daten aus dem SQL Monitor und die entsprechenden Ergebnisse der Code-Inspector-Prüfung (beides aggregiert nach Aufrufstelle) angezeigt, siehe Abbildung 8.24 rechts oben. Ein Klick auf das Feld INCLUDE-NAME führt Sie direkt zur Aufrufstelle der SQL-Anweisung. Wenn Sie eine Zeile aus dieser Ergebnisübersicht doppelklicken, erhalten Sie unten links die SQL-Monitor-Ergebnisse (nicht aggregiert). Ein Klick auf ANFRAGEINSTIEGSPKT oder TABELLENNAMEN führt Sie zum Hauptprogramm oder DDIC. Im rechten Bereich werden die Code-Inspector-Ergebnisse ebenfalls nicht aggregiert dargestellt. Über das Feld ZUSATZINFORMATIONEN kommen Sie zu den Ergebnissen der Code-Inspector-Prüfung, wie z. B. der Aufrufhierarchie bei einer SQL-Anweisung in Schleifen.

Durch die Kombination statischer und dynamischer Prüfungen bieten sich somit effizientere Auswertemöglichkeiten.