



DB2 Datenbank– Kosten senken, Performance erhöhen



Vielleicht denken Sie daran...

- die Datenbank zu optimieren
- den Quellcode all Ihrer Anwendungen zu prüfen/verändern/testen
- jede Möglichkeit zu nutzen, Performance in DB2 einzusparen
- die Hardware aufzurüsten

Es gibt eine schnellere und bessere Lösung:

Qsel/DB2

Qsel/DB2 gibt ihnen mehr Performance - ohne Veränderung an der Datenbank oder den Anwendungen

QuickSelect für DB2 - Qsel/DB2

Die Herausforderung: Genügend Ressourcen zur Verfügung zu stellen, um SLAs mit minimalen Kosten und so geringen Auswirkungen wie möglich sicherzustellen.

Der Einsatz von DB2 – Wohin verschwinden die Ressourcen?

DB2 ist heute eine weit verbreitete relationale Datenbank auf dem Mainframe. Die fortlaufende Überwachung und Optimierung der Performance von DB2 ist dementsprechend eine häufige Aufgabe von DBAs, Application-Managers und Production-Managers.

Sie sind verantwortlich für das Einhalten der SLAs, müssen das Meiste aus der aktuellen IT-Infrastruktur herausholen. Es stellt sich die Frage: Wohin verschwinden die Ressourcen?

Eine Analyse ergibt, dass Ressourcen sowohl in der Anwendung, als auch im DB2 adress space verbraucht werden. Die folgende Tabelle verdeutlicht dies.

Ressource	Wo wird sie verbraucht?		Kommentar
	Anwendung	DB2	
Die SQL-Abfrage analysieren	☒		Dies verbraucht viel CPU Leistung in der Anwendung
Kontrolle an DB2 übergeben (und Rückgewinnung)	☒		Dies hat großen Effekt auf belastete Systeme und auf Laufzeiten.
Cross-memory Aktivierung und Transfer		☒	
I/O		☒	
Bufferpool Management		☒	
Deadlock/Queue Management		☒	
Wettbewerb um DB2 Ressourcen		☒	



- Plug-in Performance Optimierung
- Reduzierte I/O- weniger CPU Last
- Einzigartiges „dynamic fingerprinting“ und caching von SQL-Abfragen
- Keine Auswirkungen auf den Speicher der Anwendung, keine regionalen Anforderungen
- Innovative Nutzung der z/OS 64 Bit Technologie
- Einfache Installation
- Kompatibel mit z/OS 1.6 und DB2 v 7 und höher

Eine deutliche Reduzierung der Workload auf dem DB2 System: Qsel/DB2 wartet darauf, dies für Sie zu realisieren.

Was wurde bereits getan und was kann noch getan werden?

Das Optimieren von DB2 ist aufgeteilt zwischen DBAs und Application-Managers.

Der Fokus liegt auf der Vermeidung von I/O Operationen. Großes Einsparpotenzial bietet die Optimierung von Indizes. Durch Konformität zwischen Abfragen und Indizes kann viel I/O eingespart werden.

Bufferpools und das Zwischenspeichern von Daten findet sowohl im Adressbereich der Anwendung als auch der DB2 Datenbank statt. Dies reduziert weiter I/O. Auch andere Ressourcen werden dadurch reduziert, möglicherweise manche andere für die Verarbeitung erhöht. In der Regel finden große Einsparungen vor allem im I/O statt.

Für CPU Ressourcen sowie konkurrierend genutzte Ressourcen wird wenig getan.

Es gibt eine weitere Möglichkeit, die darauf wartet, genutzt zu werden:

Intelligentes Zwischenspeichern von Abfragen.

Worum handelt es sich dabei?

Die meisten Anwendungen nutzen unter anderem einen Satz von Tabellen, der selten verändert und häufig adressiert wird.

Abfragen auf diese Tabellen kommen sehr häufig vor.

Wesentlich dabei ist, dass das Resultat von diesen wiederholenden Abfragen stets identisch ist. Beispiele dafür sind:

Fehlermeldungen, Informationsmeldungen, Städtenamen, Namen von Staaten, Postleitzahlen, Zinssätze etc.

Das Zwischenspeichern von Abfragen („query caching“) bedingt die Lokalisierung und die eindeutige Identifikation von Abfragen, das Zwischenspeichern („over the 64-bit BAR“) und das schnelle Auffinden und Zurückgeben des Resultats.

Bei Benchmark-Durchläufen hat sich gezeigt, dass der Effekt auf alle DB2 Ressourcen erheblich ist.

Qsel/DB2 – Das Zwischenspeichern von Abfragen bringt sehr große Einsparungen

Qsel/DB2 wendet mehrere Strategien und Technologien beim Zwischenspeichern von Abfragen an. Qsel/DB2 fängt alle SQL Befehle ab, bevor sie an DB2 gerichtet werden.

Sollten sie nicht für das Zwischenspeichern geeignet sein, werden sie an DB2 weitergeleitet.

Bei zwischenspeicherbaren Abfragen wird überprüft, ob sie zu den Tabellen und Anwendungen gehören, von denen wiederholende Abfragen kommen. Geeignete Abfragen werden mit „Fingerabdrücken“ versehen.



Dies stellt sicher, dass jede wiederholbare Abfrage eindeutig dem Resultat zugeordnet wird.

Qsel/DB2 bietet sowohl Einsparungen der CPU-Zeit als auch im I/O, sowohl für die Anwendung, als auch für DB2.

Da der gesamte Prozess auf einem syntaktischen Level ausgeführt wird, beinhaltet er nur sehr wenig Overhead.

Für die Batch-Umgebung bietet *Qsel/DB2* eine Shared Memory Option (SMO). SMO spart den Speicher, der zwischen wiederholtem Aktivieren von Programmen angesammelt wird.



Kurzfristiger ROI durch

- gesparte CPU-Zeit
- weniger I/O
- kürzere Laufzeiten

= Deutlich reduzierte Kosten

Wer profitiert davon?

- Der CIO/ IT-Manager
- der Application-Manager
- Der Production-Manager
- Der DB-Administrator
- Die User

= Das gesamte Unternehmen

Es werden Selects, Cursors und Joins unterstützt. Beim Zwischenspeichern der Abfrage wird die 64-Bit-Technologie genutzt. Es finden keine Änderungen an den Adressbereichen, kein SVC und keine Programmaufrufe statt – die Logik ist superschnell und gradlinig.

SMO hält die zwischengespeicherten Daten in einem gemeinsam genutzten Bereich "above the Bar" und ermöglicht die konstante Verwendung des Zwischenspeichers bis zur Freigabe oder bis zum nächsten Initial Program Load (IPL).

Um *Qsel/DB2* einzusetzen, werden keine Änderungen an Ihren Programmen benötigt. Wenn beim Aufruf von DB2 die Serviceroutine *dsnqli* dynamisch ausgeführt wird, muss nichts weiter getan werden, um das Modul oder DBRM zu laden.

Diese Option bringt zusätzliche Einsparungen, zieht aber die Installation von SVC und einer Sub-System Definition nach sich.

Die Funktionsweise von *Qsel/DB2* ähnelt der eines Plug-Ins – einfach Einklinken. Die Standard-Version von *Qsel/DB2* benötigt keine SVC und keine System-Installation.

Die folgende Tabelle dient der Demonstration:

	Ressource	Wo wird sie aufgewendet?		Eingespart durch <i>Qsel/DB2</i>
		Anwendung	DB2	
1.	Die SQL-Abfrage analysieren	☒		☒
2.	Kontrolle an DB2 übergeben (und Rückgewinnung)	☒		☒
3.	Übergreifende Adressbereiche (Aktivierung und Transfer)		☒	☒
4.	I/O		☒	☒
5.	Bufferpool Management		☒	☒
6.	Deadlock / Queue Management		☒	☒
7.	Wettbewerb um DB2 Ressourcen		☒	☒



Qsel/DB2 - Die Optimierung für
ihre Datenbank!

Einsparungen durch *Qsel/DB2*

	Art der Einsparung	Hochgerechnete und gemessene Einsparungen
1.	Einsparungen an CPU bei wiederholten SQL-Abfragen	40% und mehr
2.	Einsparungen an I/O bei wiederholten SQL-Abfragen	99.9%
3.	Hochgerechnete Einsparungen am CPU-Bedarf der Anwendung	40 – 55% in Abhängigkeit von den Eigenschaften der Anwendung.
4.	Hochgerechnete Einsparungen in der Laufzeit	Schwierig zu Quantifizieren. Testmessungen haben ergeben, dass 40 - 50% eingespart werden kann. Grade bei hochbelasteten Systemen lassen sich die größten Einsparungen messen.

Zum Schluss

- *Qsel/DB2* unterstützt sowohl TSO (DSN) Verbindungen zur DB2 als auch CAF Verbindungen. Natural /SQL wird unterstützt.
- *Qsel/DB* ist zurzeit für Batch GA (General Availability) Release.
- *Qsel/DB2* unterstützt alle Versionen von z/OS 1.5 und höher sowie alle Versionen von DB2 Version 7 und höher.

Kontakt:



OAK-Software GmbH
Gesellschaft für Management-Lösungen

Kalscheurener Strasse 19
Atelier 40
D-50354 Hürth

Telefon: 0 22 33 - 96 55 63
Telefax: 0 22 33 - 96 55 62

E-Mail: info@oak-software.de
www.oak-software.de