



Eine Einführung in **Workload Tuning**

Ein Whitepaper von Ami Levin
September 2010

„Das Konzept ist interessant und wohl formuliert, aber um eine bessere Note als „C“ zu erhalten, muss die Idee auch durchführbar sein.“

-- Ein Management-Professor der Yale Universität über den Vorschlag von Fred Smith zu einem verlässlichen Über-Nacht Liefer-Service. Smith gründete daraufhin FedEx.

Inhaltsverzeichnis

Einführung	i
Datenbank-Performance Management.....	1
Query Tuning - Der vertraute Standard.....	1
Wie Query Tuning funktioniert.....	2
1. Kontinuierliches Monitoring der Produktions-Umgebung	2
2. Flaschenhalse identifizieren.....	2
3. Optimieren	2
4. In die Produktions-Umgebung einspielen.....	3
Query Tuning – Pro und Kontra	3
Workload Tuning – Das vielversprechende Kraftwerk	4
Wie Workload Tuning funktioniert.....	5
1. Vorbereiten einer dedizierten Offline Analyse Umgebung	5
2. Erstellen einer Trace-Datei des zu analysierenden Produktions-Workloads.....	5
3. Starten einer Workload Tuning Session	6
4. Überprüfung von Qure's Empfehlungen, Begründungen und Ergebnissen	8
5. Qualitätssicherung durchführen und Änderungen in die Produktion einspielen.....	10
Vor- und Nachteile von Workload Tuning	11
Fazit.....	13
Weitere Informationen.....	14

Einführung

Workload Tuning ist genau das, wonach es klingt: Tuning sämtlicher Prozesse aus denen sich der Workload einer Datenbank zusammensetzt, so dass sich der Effekt des Tunings über alle Prozesse hinweg positiv auswirkt.

Unmöglich dachten Sie?! Dank Qure ist es nun Realität!

Das Workload Tuning System wird bisher bereits eingesetzt, um mehr als 200 Produktions-Datenbank Workloads weltweit zu optimieren. Dabei lag die durchschnittliche Verbesserung der Gesamt-Performance für den Workload bei über 60%. Dies schließt auch Datenbanken ein, welche bereits mit Standard-Tuning Techniken und Tools optimiert wurden. Darüber hinaus ist es möglich, das Maß der Verbesserung auf den Workload exakt vorherzusagen, bevor die Änderungen überhaupt in die Produktions-Umgebung eingespielt worden sind. Ermöglicht wird dies durch den Umstand, dass Workload Tuning exakte Messwerte für jeden einzelnen Prozess vor und nach Optimierung in einer kontrollierten Nicht-Produktions-Umgebung bereitstellt.

Dieses Dokument beschreibt anhand von Qure™ für SQL Server, dem ersten kommerziell verfügbaren Workload Tuning Tool, detailliert, was Workload Tuning ist und wie es arbeitet. Außerdem vergleicht es Workload Tuning mit traditionellen Abfrage-für-Abfrage Tuning Ansätzen.

Datenbank-Performance Management

Performance von IT-Systemen ist seit jeher eine kritische Angelegenheit für viele Organisationen. IT ist die Lebensader von nahezu allen Organisationen. Sie ermöglicht Prozess-Effizienz und Service-Levels, die sonst nicht möglich wären. Dies hat jedoch dazu geführt, dass ein ständig wachsender Druck auf die IT entstand, mehr Daten von höherer relationaler Komplexität immer schneller bereitzustellen bei gleichzeitig sinkenden Budgets.

Eine natürliche Nebenerscheinung dieser Herausforderungen ist die Evolution automatisierter Software-Lösungen, die geschrieben wurden, um IT Experten zu unterstützen. Workload Tuning ist gerade durch diesen Fortschritt in Automatisierung möglich geworden. Es versetzt den DBA in die Lage, noch höhere Datenbank-Performance abzuliefern und signifikant IT-Kosten einzusparen.

Bevor wir Workload Tuning erläutern, sollten wir zunächst einmal das traditionelle Query Tuning beleuchten. Dies wird uns auch den Hintergrund liefern, warum die Entwicklung von Workload Tuning überfällig war.

Query Tuning - Der vertraute Standard

***„Ein Computer hat mich mal im Schach geschlagen,
beim Kick-Boxing jedoch war er kein Gegner für mich.“***

-- Emo Philips

Datenbank-Applikationen starten ihren Lebenszyklus in einem hochoptimierten Zustand. Vielfach jedoch verlieren sie mit der Zeit diesen Biss. Das mag mit zahlreichen natürlichen evolutionären Veränderungen in der Applikation, im Business oder in den Daten zusammenhängen. Erweiterungen, welche unausgewogene Belastungen für die existierenden Datenstrukturen mit sich bringen, werden in die Applikation eingefügt. Das Verhältnis zwischen Daten und Abfragen verschiebt sich mit den Anforderungen des Business und erhöht den Druck auf den Workload. Umfangreiche ETL Jobs werden auf frühere oder spätere Zeiten verschoben. Neue Anforderungen an das Reporting erwachsen. Den Rest mag sich der Leser selber denken. Es gibt immer neue Einflüsse, welche die Natur des Workloads beeinflussen und neue Flaschenhälse schaffen können. Wenn dies geschieht, wird es notwendig, ein Mittel zur Performance-Diagnose und Wiederherstellung zur Verfügung zu haben.

Traditionell hat man sich der Herausforderung nach Verbesserung einer sich stetig verändernden Performance von Datenbank-Applikationen derart genähert, dass man hauptsächlich versucht hat, diejenigen Prozesse und Abfragen zu identifizieren, die die meisten Ressourcen verbrauchen. In diesem Abschnitt betrachten wir diesen Ansatz näher und diskutieren seine Vor- und Nachteile.

Wie Query Tuning funktioniert

Wie Ihnen zweifelsohne bekannt ist, sieht der traditionelle Query Tuning Ansatz in etwa folgendermaßen aus:



Figure 1 - Traditional Tuning Process

1. Kontinuierliches Monitoring der Produktions-Umgebung

Monitoring kann aktiv oder passiv erfolgen. Erfolgt es aktiv, sammelt in der Regel ein Monitoring Tool unentwegt Fakten auf dem Produktions-Server und vergleicht diesen mit einem vorab definierten Grenzwert. Wird der Grenzwert erreicht oder überschritten, benachrichtigt ein automatischer Alarm den Support. Erfolgt das Monitoring passiv, benutzen wir unsere Kunden als unbezahltes Monitoring Personal und sie bedanken sich dafür bei uns mit wütenden Telefonaten, wenn sie ein Performance-Problem entdeckt haben.

2. Flaschenhalse identifizieren

Tritt ein Problem auf, besteht einer der ersten Maßnahmen darin, ein zuverlässiges Ressourcen-Management Tool einzuschalten, um die physikalische Ressource (I/O, CPU, Arbeitsspeicher oder Netzwerk) zu identifizieren, die diesen Flaschenhals verursacht. Dieser Teil ist nicht sonderlich schwierig. Dann setzen wir unseren Zauberhut auf und versuchen die Ursache für diesen Flaschenhals sichtbar zu machen. Dieser Teil ist mal leichter, mal schwieriger. Je mehr die physikalischen Ressourcen ausgelastet sind und je mehr gleichzeitige Prozesse um diese Ressourcen wetteifern, umso schwieriger ist es, den oder die Hauptverursacher zu identifizieren.

3. Optimieren

Ist die Ursache für den Flaschenhals identifiziert, werden ein oder mehrere Lösungsansätze vorgeschlagen, um die Performance zu optimieren, den Druck auf die Ressource

aufzuheben und den Flaschenhals zu beseitigen. Im Hinblick auf den Zweck dieses Whitepapers ignorieren wir aber solche kurzfristigen Mittel zur Entlastung und fokussieren lieber auf Maßnahmen zur anhaltenden Entlastung. Vielfach werden Indices angepasst oder neu hinzugefügt, um Abfragen schneller beantworten zu können. Unterschiedliche Abfragetechniken werden vielleicht getestet, um einen effizienteren Zugriff auf die Daten zu finden. Unter gänzlich unglücklichen Umständen wird die Hardware upgegraded oder Teile der Applikation redesigned.

Jede vorgeschlagene Langzeitlösung für den identifizierten Flaschenhals wird in einer Nicht-Produktions-Umgebung getestet und mit den „Bevor“ Performance-Kennzahlen verglichen, um diejenige zu finden, welche die konsistenteste und verlässlichste Reduktion bezüglich Ressourcenverbrauch mit sich bringt.

4. In die Produktions-Umgebung einspielen

Nach ausreichenden (oder auch nicht) QA-Tests werden die Änderungen in die Produktions-Umgebung eingespielt. Das Performance Monitoring wird wieder aufgenommen und versucht zu verifizieren, dass die Änderung wie erwartet funktioniert, ebenso um neue Performance-Kopfschmerzen abzufangen, welche vielleicht schon an der nächsten Ecke lauern.

Query Tuning – Pro und Kontra

Die Mechaniken von Query Tuning sind weithin bekannt. So weit, dass wir uns vielleicht sogar manchmal dabei ertappen, wie wir mit Familienmitgliedern oder Nachbarn über unsere letzten Tuning-Abenteuer sprechen, bevor wir dann nach einigen Minuten deren seltsamen Gesichtsausdruck bemerken der zu sagen scheint: „Hilfe! Wovon spricht er...?!“

Die Vor- und Nachteile von Query Tuning hingegen sind uns nicht andauernd präsent. Es liegt nicht in der menschlichen Natur ein Lösungskonzept zu hinterfragen, wenn dieses Konzept historisch gesehen das Einzige für eine lange Zeit war.

Nehmen wir uns jedoch die Zeit, einmal darüber nachzudenken, fallen folgende Aspekte auf:

Query Tuning ist von Natur aus reaktiv. Query Tuning wird eingesetzt, nachdem ein Problem aufgetreten ist. Wir beten nur, dass unser Monitoring System das Problem entdeckt, bevor unsere Kunden es tun.

Da Query Tuning von DBAs und Entwicklern angetrieben wird, die sich manuell nur um die Prozesse kümmern können, die als Quelle des aktuellen Flaschenhalses identifiziert wurden, ist es von Natur aus nur in der Lage, ein paar wenige Prozesse auf einmal zu optimieren. Die Benutzer verstehen, dass qualitativ hochwertige Lösungen Zeit brauchen und so akzeptieren wir, dass es schlicht und einfach nicht möglich ist, diese Art von erhöhter Aufmerksamkeit mehr als einigen wenigen Problemfeldern zu widmen. Diese Praktik wird als normal erachtet und akzeptiert, aber nur aus dem Grund, weil die Kosten um einen Großteil der Produktions-Prozesse zu optimieren nahezu untragbar waren.

Hinter den meisten Performance-Krisen steht üblicherweise nur eine kleine Anzahl von Abfragen, und DBAs sind vielfach kundig in der Fehlersuche und im Bereitstellen von praktikablen Lösungen. Query Tuning, das beim Tunen einer kleinen Anzahl von Abfragen häufig ins Schwarze trifft, ist daher perfekt geeignet, um eine Krise zu lösen.

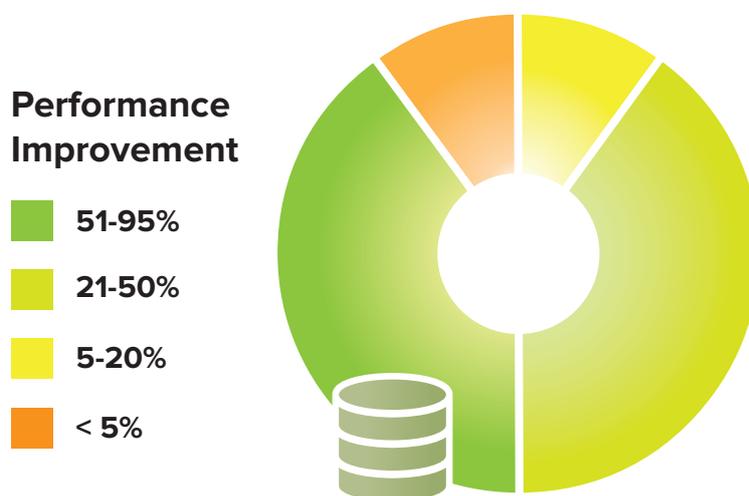
In solch einer Krisensituation, in der ein System kurz vor dem Stillstand ist oder bereits stillsteht, ist die oberste Priorität, das System in einen operationalen Zustand zu versetzen, **und zwar**

umgehend. Dies verleitet einen manchmal, Abkürzungen zu verwenden. Wir werden gelobt für die Geschwindigkeit, mit der wir ein System reparieren, nicht notwendigerweise für die Qualität der Reparatur. Manchmal kann unser Eingriff den Druck von einem Prozess zu nehmen, unbeabsichtigter Weise den Druck auf einen anderen Teil der Datenbank lenken. Je weniger Zeit wir haben, um einen realistischen Belastungstest auf die ressourcen-teilenden Prozesse durchzuführen, desto weniger wissen wir über den potentiellen Begleitschaden. Üblicherweise ist aber unsere Freude darüber, es doch noch vor Mitternacht nach Hause zu schaffen, genug Grund, um diese Bedenken beiseite zu schieben.

Der eindeutige Vorteil von Query Tuning liegt darin, dass es etwas Bekanntes ist. Viele von uns DBAs und Software Engineers haben eine Menge Erfahrung mit Query Tuning und so ist es eine bequeme und berechenbare Methode, um den Karren aus dem Dreck zu ziehen. Darüber hinaus gibt es eine Vielzahl von Tools von Dritt-Anbietern, die uns helfen, die Methodologie sehr effizient einzusetzen.

Auf der anderen Seite kann selbst mit den richtigen Performance Monitoring Tools die Diagnose der Kernursache sehr schwierig werden, wenn verschiedene gleichzeitige Prozesse in Echtzeitmultiple Effekte aufeinander haben. Gott bewahre, dass dieses Problem erst auftritt, wenn die Auslastung nahe 100% ist, da es dann genauso einfach ist, das Problem zu identifizieren, wie die berühmte Nadel im Heuhaufen zu finden.

Figure 2 - Workload Improvement Distribution



Average, workload-wide improvement in Duration, CPU and IO
Based on ~200 analyzed production workloads

Workload Tuning – Das vielversprechende Kraftwerk

„Die Fähigkeit, Defekte zu reparieren, macht keinen Senior DBA aus.

Vielmehr zeigt sich ein Senior DBA darin, dass er Sachen macht, die nicht repariert werden müssen.“

-- Thomas LaRock

Workload Tuning stellt einen komplett neuartigen Ansatz für das Datenbank Performance Management dar. Es komplettiert Query Tuning als ein Mittel zur Diagnose und Reparatur von Produktions-Problemen.

Die Neuartigkeit kann in zwei Adjektiven zusammengefasst werden: Ganzheitlich und umfassend. Zunächst einmal zum ganzheitlichen Aspekt.

Workload Tuning ist für Query Tuning, was ganzheitliche Medizin für Schulmedizin ist. Beklagt man sich über Rückenschmerzen, behandelt die Schulmedizin üblicherweise genau die Stelle, an der die Schmerzen auftreten, durch Verschreiben von Arzneimitteln zur Entspannung der Muskeln und Physiotherapie zur Dehnung und Kräftigung der Muskeln. Schmerzlinderung kann auf diese Weise sicher erreicht werden, aber die Schmerzen können wieder auftreten, wenn die Ursache nicht behoben wird. Ganzheitliche Medizin sucht nach dieser Ursache. Der ganzheitliche Therapeut wird unter Umständen feststellen, dass der Gang ungleichmäßig ist und dass ein Problem mit dem linken Fuß in den Rücken ausstrahlt. Durch Behandlung des Fußleidens kann so das Rückenleiden langfristig geheilt werden.

Die zweite neue Charakteristik ist „umfassend“.

Workload Tuning ist für Query Tuning, was automobiler Massenproduktion für die individuelle Einzelherstellung eines Autos ist. Nach Jahren, in denen alle Arbeit Stück für Stück durchgeführt und der gesamte Herstellungsprozess sorgfältig analysiert wurde, wurde ein automatisierter Ansatz entwickelt, welcher tausende von Stücken bei gleichbleibend hoher Qualität herstellen kann. Eine neue Fachkompetenz entwickelte sich: Die des Erstellens und Wartens dieser umfassenden Operation.

Ein ähnlicher Fortschritt ist jetzt auch im Bereich des Datenbank Performance Tunings eingetreten. Zum ersten Mal ist es nun kosteneffizient möglich, jedes einzelne SQL Statement in einem Workload zu tunen.

In über 200 Produktion-Workloads und Datenbanken, die bisher durch Qure analysiert und optimiert wurden (siehe Abbildung 2), haben 80% eine Verbesserung der Performance über den gesamten Workload hinweg zwischen 20% und 95% erzielt.

Zum Verständnis dieser 95% Zahl: Sie bedeutet, dass, nachdem die von Qure empfohlenen Änderungen angewendet wurden, der Kunde festgestellt hat, dass 95% weniger Ressourcen (CPU, Laufzeiten von Abfragen, Lese- und Schreibvorgänge) benötigt wurden, um den gleichen Workload zu bewältigen. Das bedeutet, dass der gesamte Workload durchschnittlich eine 20-fache Verbesserung der Abfragegeschwindigkeit und dabei nur 1/20 der CPU-Zeit und Lese- und Schreibvorgänge verbrauchte.

Dieser Punkt verdient es, besonders hervorgehoben zu werden. Derartige Performance-Verbesserungen beziehen sich nicht nur auf eine Handvoll von Abfragen oder nur eine der gemessenen Ressourcen. Sie betreffen die durchschnittliche Verbesserung über die Tausende oder gar Millionen von Prozessen, die den Workload ausmachen, und greifen über alle Ressourcen hinweg.

Weitere Beispiele von analysierten Datenbanken und Workloads können zusammen mit den Ergebnissen auf der <http://www.dbsophic.com/case-studies> von DBSophic abgerufen werden.

Um nun das Wesen dieses neuartigen Tuning-Ansatzes zu beleuchten, werden wir zunächst seine Arbeitsweise skizzieren, bevor wir dann seine Vor- und Nachteile erläutern.

Wie Workload Tuning funktioniert

„Jede Lösung sollte so einfach wie möglich sein, aber nicht einfacher“

-- Albert Einstein

Workload Tuning, so wie es von Qure implementiert wird, funktioniert folgendermaßen:

1. Vorbereiten einer dedizierten Offline Analyse Umgebung

Qure wurde entwickelt, um mit einer Kopie der Produktions-Datenbank komplett offline zu arbeiten, die in einer dedizierten Analyse-Umgebung bereitgestellt wird. Dieses Offline-Arbeiten ist aus 2 Gründen wichtig:

- Es ermöglicht eine ausführliche analytische Diagnose, die in dieser Art nahezu unmöglich durchführbar in einer Produktions-Umgebung ist.
- Es ermöglicht es Qure, automatisch Objekte und Code in der Datenbank-Kopie zu modifizieren, um dadurch automatisch seine empfohlenen Änderungen zu validieren. Dies kann nicht in Produktion durchgeführt werden.

Die Kopie der Produktions-Datenbank enthält alles (alle Tabellen, Indices, Gespeicherte Prozeduren, Funktionen, Sichten, Trigger und Daten). Dies gibt Qure die Möglichkeit alle Aspekte der Datenbank vollständig zu verstehen und alle diese so gewonnenen Informationen in heuristische Algorithmen einzuspeisen, die ganzheitlich den gesamten Workload optimieren.

2. Erstellen einer Trace-Datei des zu analysierenden Produktions-Workloads

Der zweite Input für Qure's Analyse Ihrer Datenbank ist eine umfangreiche Workload-Trace Datei. Der zur Verfügung gestellte Workload sollte möglichst realitätsnah den gesamten Bereich der Aktivitäten abdecken, welchen die Applikation und die Datenbank durchführen. In der Trace-Datei sind die Anhaltspunkte für die größten Flaschenhälse, CPU-Belastungen, Häufigkeiten der verschiedenen SQL Statement Typen, Variationen in den Parametern für jeden Typ von SQL Statement und all die anderen kleinen Hinweise enthalten, die Qure analysiert, um die Performance Geheimnisse zu lüften, die durch diese Hinweise offensichtlich werden.

Solche Trace-Dateien sind der Output von SQL Servers' Trace-Infrastruktur und können optional durch die von Qure zur Verfügung gestellten Trace Vorlagen gesteuert werden. Diese Vorlagen maximieren die Erfassung von hilfreichen Informationen und minimieren die Erfassung nutzloser Informationen, damit so der Overhead auf den Produktions-Server während der Generierung des Traces verringert wird.

3. Starten einer Workload Tuning Session

Qure kann auf jedem Computer (Workstation oder Server) installiert werden, der Zugriff auf die Analyse-Umgebung hat. Man kann die Analyse starten, nachdem man ein paar simple Informationen eingegeben hat: Den Server, der für die Analyse verwendet werden

soll, die Datenbank(en) und die Trace-Datei(en), die analysiert werden sollen, sowie einige ausgewählte Analyse-Einstellungen.

Automatisierte Datensammlung

Ist der Analyse-Prozess einmal gestartet, arbeitet Qure vollkommen selbstständig. Die Dauer der Analyse hängt von der Größe der analysierten Datenbank ab und vom analysierten Workload, von dessen Eigenschaften, den Ressourcen des Analyse-Servers und diversen anderen Faktoren ab. So kann eine Analyse zwischen einigen Stunden und einigen Tagen dauern. Um einen Eindruck von der Tiefe der Analyse zu erhalten, betrachten Sie einmal die Abbildung 3. Sie zeigt den Report-Fortschritts Bildschirm.

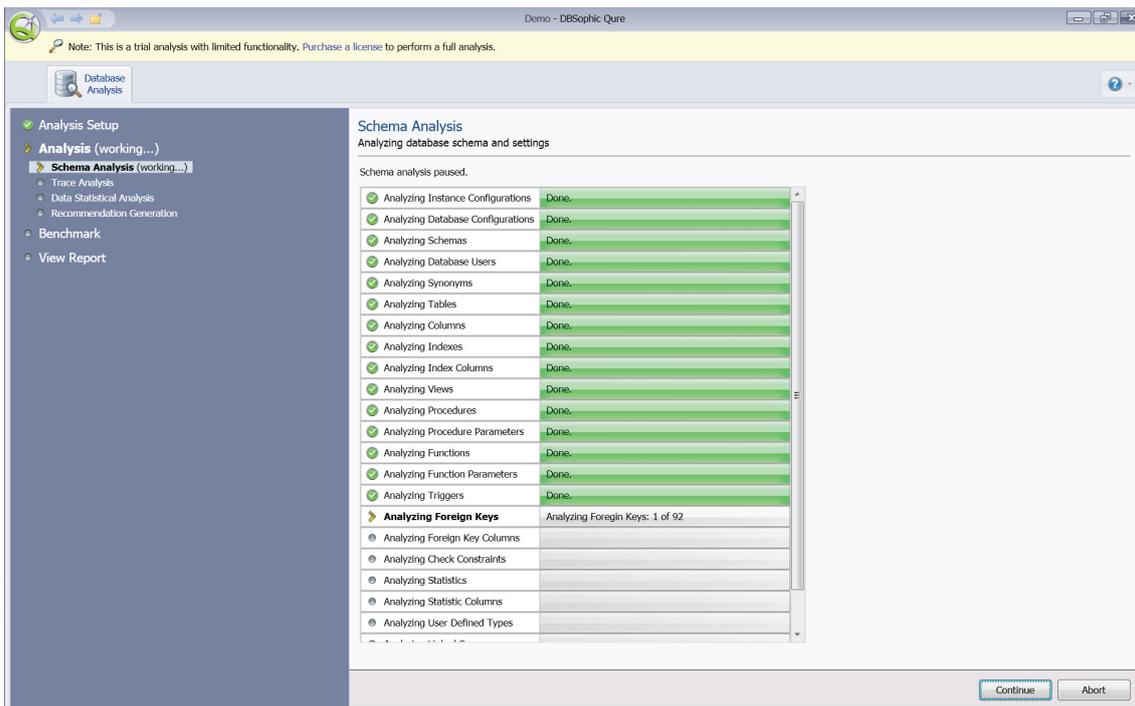


Figure 3 –Automated Analysis Process

Während der Daten-Sammelungsphase:

- beurteilt Qure alle Instanzen jeder Abfrage (wobei als Instanz einer Abfrage die einzelne Ausführung der Abfrage mit einer spezifischen Kombination von Eingabeparametern verstanden wird).
- untersucht Qure jede Abfrage nicht als individuelle Entität, sondern als Teil einer Gruppe von Entitäten, die nach dem Herunterbrechen auf die variablen Parameter eine gemeinsame Funktion und Syntax teilen. Solch eine Gruppe wird als „Batch“ bezeichnet. Natürlich kann jede einzelne Instanz eines Batches sich anders verhalten als andere Instanzen, abhängig von den verwendeten spezifischen Input Parametern.
- gewichten Qure's Algorithmen jede Abfrage mit Hinblick auf den gewichteten Verbrauch von Ressourcen. Dabei berücksichtigt Qure nicht nur die durchschnittliche Performance jeder Abfrage, sondern auch die Anzahl der Ausführungen jeder Abfrage innerhalb eines normalen Workloads.

Automatische Generierung der empfohlenen Änderungen

Basierend auf den gesammelten Daten, generiert Qure automatisch Empfehlungen, um jeden Prozess, der innerhalb der Datenbank getuned werden kann, zu optimieren. Diese Empfehlungen können verschiedene Formen annehmen:

- Erstellen, Ändern oder Löschen von Indices
- Weiterentwickeln von Objekt-Code
- Modifizieren des Schemas
- Adjustieren der Konfiguration
- Weiterentwickeln der Abfrage-Syntax
- Diverse Empfehlungen

Die Magie hinter diesen Empfehlungen stammt aus einer einmaligen Kombination von Faktoren:

1. Eine eingebaute Wissensdatenbank, basierend auf dem Erfahrungsschatz von weltbekannten Microsoft SQL Server Gurus, welche die eine oder mehrere mögliche Lösungen für jede Abfrage ermitteln kann, die den spezifischen Anforderungen des Workloads am besten gerecht wird.
2. Der beste Index-Optimierer auf dem Markt. Was ihn besonders macht ist, dass er nicht jeden Index auf einer Abfrage-für-Abfrage Basis (individuell) optimiert, sondern ganzheitlich. Er empfiehlt weniger Indices die aber mehreren Prozessen nützlich sind. Durch heuristische Verfahren wird dabei ermittelt, wie der größte Gesamtgewinn erreicht werden kann. Da er Kenntnisse hat von den miteinander konkurrierenden Bedürfnissen der Abfragen, die auf die gleichen Daten zugreifen, kann er eine ausgewogene Balance erreichen zwischen dem Erhöhen der Abfragegeschwindigkeit einerseits und dem Minimieren des Overheads den die Index-Pflege mit sich bringt andererseits. Das resultiert in einer kleinen Anzahl von neu definierten Indices und einer ganzheitlichen allgemeinen Verbesserung der Performance.
3. Die Fähigkeit Informationen von verschiedenen Aspekten einer Datenbank miteinander in Beziehung zu setzen, wenn Empfehlungen für jeden Prozess gemacht werden.
Beispiel:
 - Qure weiß, welche Daten häufig modifiziert werden, und kann so ein „Über“-indizieren dieser Daten vermeiden, um den damit verbundenen Wartungsaufwand zu vermeiden.
 - Qure kann das statistische Profil nutzen, um die Aktionen des Index-Optimierer feinzusteuern. Hat zum Beispiel eine Spalte eine sehr geringe Kardinalität, bedeutet dies, dass es nicht sehr effektiv ist, diese Spalte zu indexieren und daher kann diese Spalte vielleicht eher als eine „Included“ Spalte Sinn machen anstatt als Schlüsselspalte.
 - Objekte und Abfrageveränderungen basieren auf der Kenntnis des Schemas. Nehmen wir zum Beispiel die Empfehlung in Abbildung 6 weiter unten (Diskrepanz zwischen den Datentypen eines Prädikats). Solche Diskrepanzen können nur

gefunden werden, wenn man Kenntnis sowohl über das Schema als auch über die Struktur der Abfrage hat.

4. Die Fähigkeit, automatisch Objekte, Batches und Indices zu verändern, um neue ausführbare Versionen zu erstellen.

Weitere Details zu den Empfehlungen werden in Punkt 4 erläutert.

Automatische Baseline Erstellung

Nachdem die Phase der Generierung der Empfehlungen abgeschlossen ist, erstellt Qure automatisch eine Performance-Baseline für den gegebenen Workload durch erneutes Abspielen des Workloads in der Kopie der Produktions-Datenbank.

Messwerte werden während jedes Benchmark-Laufes ermittelt. Dies schließt physikalische Lesevorgänge, logische Lesevorgänge, Schreibvorgänge, CPU Zeiten und Abfrage-Laufzeiten ein.

Automatische Optimierung der Analyse-Umgebung

Qure wendet seine Empfehlung (Änderungen an Abfrage-Syntax, Objekte, Indexen und Konfiguration) automatisch auf die Kopie der Produktions-Datenbank an. Schemaänderungen können optional ebenfalls automatisch angewendet werden und damit in den Benchmark-Prozess einfließen.

Bei hunderten oder gar tausenden von potentiellen Empfehlungen ist es sehr praktisch, dass sämtliche Revisionen getestet und validiert werden, dass sie funktional identisch mit dem Original, syntaktisch korrekt, fehlerfrei und bereit zur Ausführung sind. Es leuchtet ein, dass dies eine enorme Zeitersparnis für den DBA bedeutet, der ansonsten Empfehlungen manuell schreiben, anwenden und ändern müsste.

Automatisches Benchmarking der Auswirkungen auf die Performance

Sind die empfohlenen Änderungen an der Datenbank und ihren Objekten vorgenommen, spielt Qure den gesamten Workload erneut ab, um die erreichten Performance-Verbesserungen genau messen zu können. Diese Ergebnisse wiederum werden verwendet, um Empfehlungen zu einzelnen Batches, die es erfordern, feiner zu steuern. Qure kann solche Nachprüfungen und erneutes Benchmarking so lange durchführen, bis eine optimale Verbesserung erreicht wurde.

4. Überprüfung von Qure's Empfehlungen, Begründungen und Ergebnissen

Die empirischen „Vorher“ und „Nachher“ Messwerte werden dem User in einer leicht zu verstehenden Reihe von interaktiven Berichten präsentiert, die die Größenordnungen und die Natur der Performance-Verbesserungen detailliert ausweisen (siehe Abbildung 4).

Eine weitverbreitete Frage zu Workload Tuning ist die, wie die Empfehlungen generiert werden und wie diese aussehen.

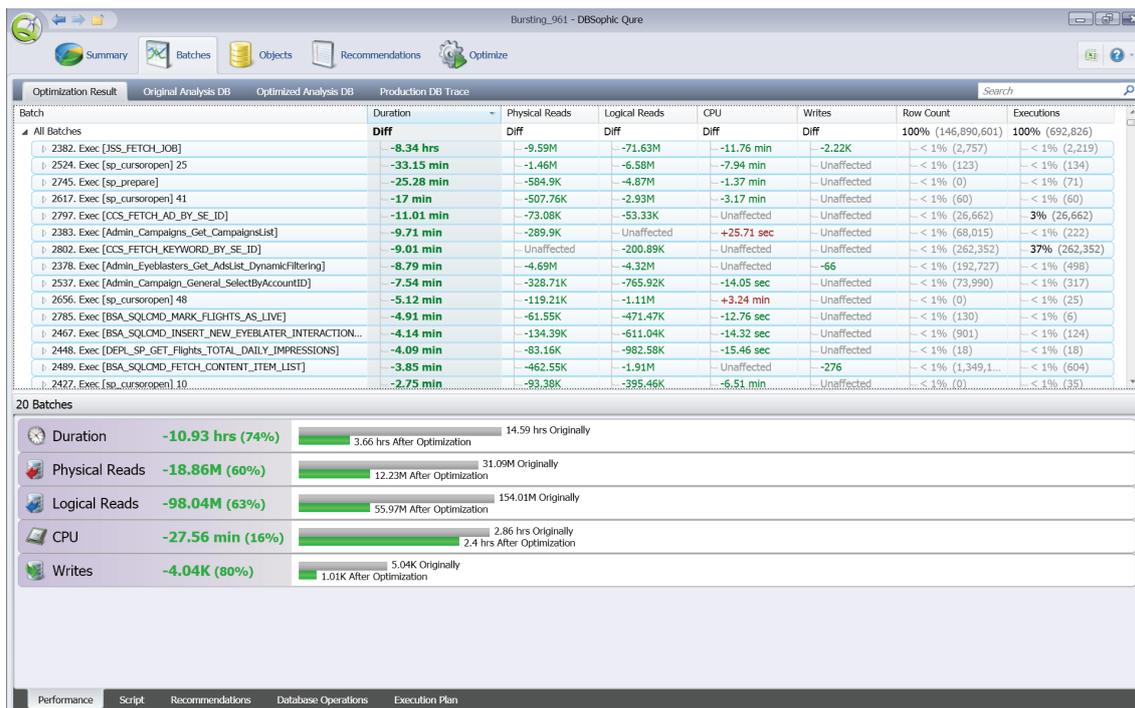


Figure 4 - Workload Performance Improvements

Die Empfehlungen

Empfehlungen basieren auf Informationen zu zahlreichen Aspekten einer Datenbank und ihres Workloads, sowie den Querverbindungen zwischen Abfrage-Strukturen, Tabellen-Strukturen, Index-Strukturen, Zugriffsmuster, Statistiken und Metadaten.

Die von Qure generierten Empfehlungen enthalten drei Teile:

1. Eine Begründung für die empfohlene Änderung in verständlichem Englisch.
2. Ein generiertes Änderungs-Skript.
3. Eine Liste von Objekten und Batches, die durch diese Änderung betroffen sind.

Eine Begründung für die empfohlene Änderung in verständlichem Englisch

Qure liefert eine klare und umfassende Erklärung für die Hintergründe, die jeder Empfehlung Kompetenz des DBA-Teams zu erhöhen und, nach und nach, auch die Kompetenz des Entwicklungs-Teams.

Ein generiertes Änderungs-Skript

Für (fast) alle Empfehlungen erstellt Qure ein fertiges Änderungs-Skript, das direkt ausgeführt werden kann und die Empfehlung implementiert. Solche Skripte können zum Beispiel für Änderungen an Gespeicherten Prozeduren, ALTER INDEX Statements, CREATE INDEX Statements, etc. generiert werden. Weitere Beispiele werden in dem freien Webinar „Working with Qure’s Recommendations“ unter <http://www.dbsophic.com/resources/> aufgeführt. Diese Skripte sind selbstverständlich notwendig für Qure, um die Empfehlungen während der Benchmarking Phase automatisch einzuspielen. Gleichzeitig aber erlauben es diese Skripte

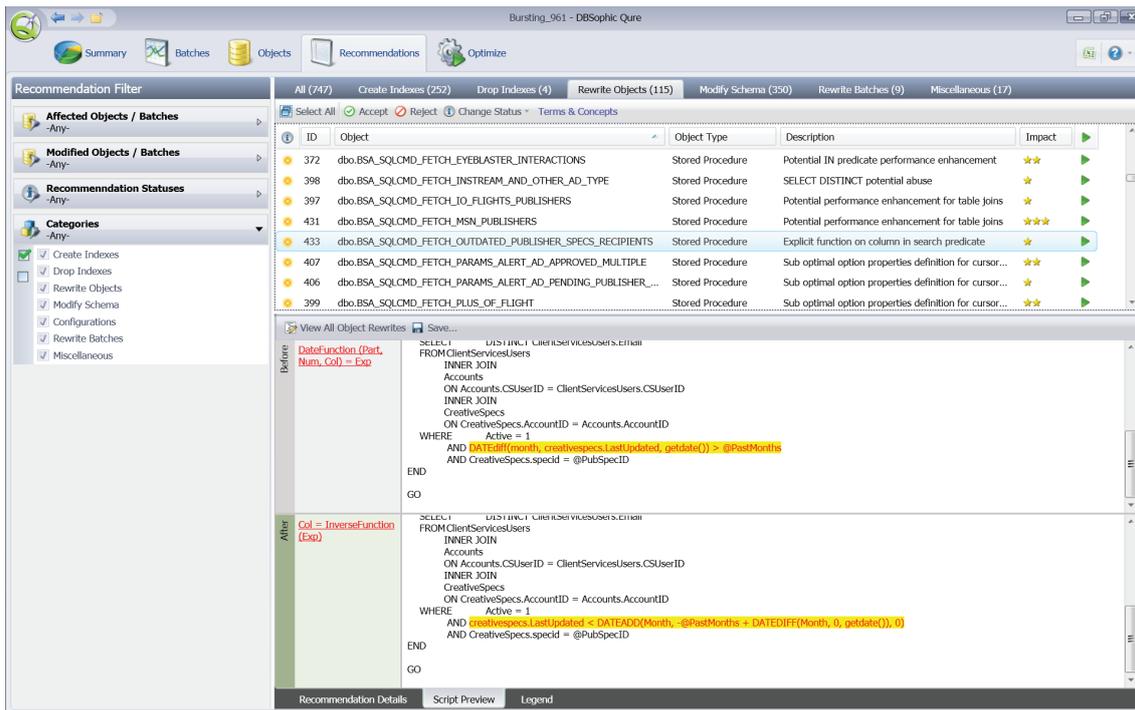


Figure 5 - Executable Scripts for Deploying Recommendations

dem User, die Empfehlungen sehr schnell in eine QA-Umgebung oder in die Produktions-Umgebung zu deployen.

Eine Liste von Objekten und Batches, die durch diese Änderung betroffen sind

Da es multiple Abhängigkeiten zwischen den Objekten in einer Datenbank geben kann, und weil DBAs aus verständlichen Gründen besorgt über eine mögliche Downstream Auswirkung bei Änderung eines Objektes auf andere Objekte sind, listet Qure sämtliche Objekte und Batches auf, die durch die empfohlene Änderung betroffen sind.

Zum Beispiel Abbildung 6: Dies ist ein interaktiver Bericht, der es dem Benutzer erlaubt, schnell zwischen allen betroffenen Objekten hin- und her zu navigieren, um diese zu inspizieren.

Akzeptieren oder Ablehnen einzelner Empfehlungen

Qure bietet dem Benutzer eine Oberfläche, mit der Empfehlungen nach Objekt-Typ, Batch Namen, Empfehlungs-Typ, Verbesserungs-Level sowie anderen Kriterien leicht sortiert und gefiltert werden können. Nachdem man eine oder zwei Empfehlungen eines bestimmten Typs begutachtet hat, kann man optional alle Empfehlungen dieses Typs akzeptieren. Dieses Feature hilft dem DBA-Team, sich schnell durch die Liste der Empfehlungen zu arbeiten, die üblicherweise aus hunderten von Einzelempfehlungen besteht.

5. Qualitätssicherung durchführen und Änderungen in die Produktion einspielen

Die gespeicherte Sammlung von Empfehlungen, die die Änderungs-Skripte für Objekte und Batches beinhaltet, kann anschließend kopiert und in eine QA-Umgebung eingespielt werden, um gegen die eigenen QA-Prozesse getestet zu werden. Sind dann die Empfehlungen in die Produktion deployed, kann man von sofortigen Verbesserungen der Performance profitieren.

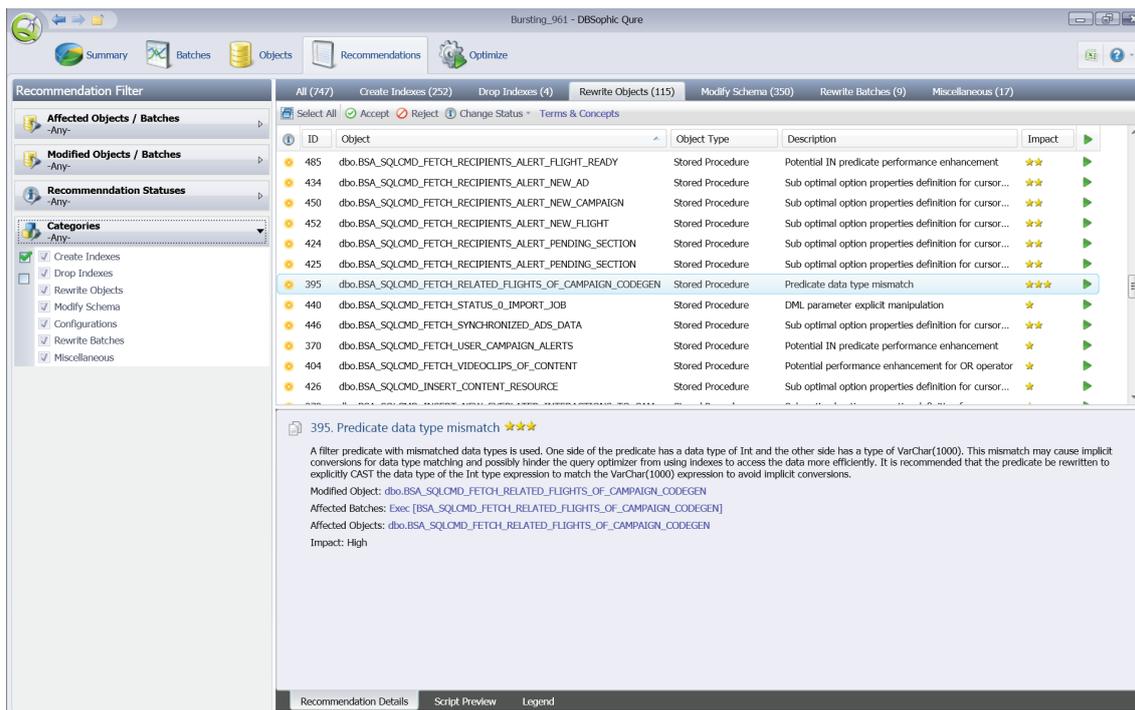


Figure 6 - Recommendation Details

Vor- und Nachteile von Workload Tuning

Nachdem wir nun die Arbeitsweise von Workload Tuning erläutert haben und wir die Vor- und Nachteile von Query Tuning besprochen haben, folgen nun einige Vor- und Nachteile von Workload Tuning, die ins Auge springen:

Workload Tuning ist proaktiv. Es findet „Probleme“, bevor diese entstehen.

Die Anzahl an Problemen, die gefunden werden kann ist nahezu unbegrenzt. Da Qure eine automatisierte Vorrichtung ist, die eine gesamte Datenbank inklusive aller Abfragen und Prozesse, die im Workload enthalten sind, analysiert, kann es Abfragen optimieren, für die ein Mensch sich nie die Zeit nehmen könnte. Gespeicherter Prozeduren, die nur gelegentlich mehrere Sekunden Laufzeit benötigen, werden unter Umständen nicht als „schlimm“ genug erachtet, um durch die Query-für-Query Methode optimiert zu werden. Qure hingegen bemerkt diese sofort und wird eine Empfehlung für einen optimierten Datenzugriff generieren, falls es einen gibt.

Qure analysiert ganzheitlich, nicht individuell. Qure verwendet heuristische Algorithmen, um den Workload einer Datenbank als Ganzes zu tunen und wägt jede potentielle Adjustierung gegen jede andere in einer Art und Weise ab, wie es nur eine Software machen kann, da es bei weitem zu viele Vergleiche für einen Menschen sind.

Beispiel:

- Findet Qure zwei (oder mehr) effizient designte Strukturen, die nicht immer optimal arbeiten, vergibt Qure eine Gewichtung, bei der die höchste Performance-Priorität auf die größten Ressourcen - Verbraucher und die geringste Priorität auf die geringsten Ressourcen - Verbraucher gelegt wird.

- Empfehlungen zu Änderungen erfolgen nicht isoliert. Wird ein Index oder eine Gespeicherte Prozedur optimiert, berücksichtigt Workload Tuning die projizierte Auswirkung auf alle Abfragen, die sich auf dieses Objekt beziehen, bevor es sich auf eine Empfehlung festlegt.
- Nachdem jede Empfehlung in Form eines veränderten Objektes, Batches oder einer Konfiguration, angewendet wurde, wird diese Empfehlung empirisch gemessen gegen den Workload für das geänderte Objekt, um zu verifizieren, dass diese Änderung tatsächlich optimal war.
- Qure's Benchmark testet viele verschiedene Kombinationen von Parametern, die im Produktion-Workload für jeden Batch gefunden wurden, nicht nur eine Handvoll davon. Dies hilft dabei, gerade diejenigen schwer zu findenden Parameter Kombinationen zu finden, die während des Testens in QA übersehen werden, später aber in der Produktion Schwierigkeiten bereiten können.

Qure führt intensive Analysen durch. Um derartige intensive Prozesse zu unterstützen, ist ein dedizierter Analyse-Server erforderlich, der so ausgestattet sein muss, dass er die gesamte Kopie der Produktions-Datenbank hosten kann und darüber hinaus noch Speicherplatz bietet, um die Empfehlungen anzuwenden. Ebenfalls ist ein Satz von Trace-Datei(en) notwendig. Alle diese Voraussetzungen haben zur Folge, dass Workload Tuning nicht unbedingt der beste Ansatz ist, um eine akute Krise schnellstmöglich zu beheben.

Da Workload Tuning ganzheitlich ist und heuristische Verfahren nutzt, um die Ressourcenauslastung auf ein Minimum zu reduzieren, findet eine bewusste Neuausrichtung statt. Diese Neuausrichtung nutzt gerade den Top-Ressourcen-Verbrauchern. Damit ist gemeint, dass es keine Garantie gibt, dass jeder einzelne Prozess optimiert wird. Vielmehr wird man feststellen, dass der gesamte Workload als solcher signifikant verbessert wurde.

Die Auswirkungen von Workload Tuning sind nahezu immer positiv und häufig geradezu beeindruckend. Durch diese Art von Verbesserung kann die Nutzungsdauer von existierender Hardware und Speicherplatz signifikant verlängert werden. Die daraus resultierenden Einsparungen durch den Einsatz von Qure führen zu einem soliden ROI für ein Unternehmen. Da ist ein Lächeln durchaus erlaubt.

Workload Tuning ist ein natürlicher evolutionärer Fortschritt bei Werkzeugen für DBAs – die Routinearbeit wird automatisch erledigt und schafft Raum für Expertenarbeit. Betrachtet man Qure's Arbeit als rohe Tischlereiarbeit, so kann man die Arbeit, die dem DBA verbleibt, damit vergleichen, aus dem rohen Handwerk ein fertiges Möbelstück zu erschaffen. Es ist die Arbeit, die ein höheres Maß an Expertise und Wissen erfordert. Daher ermöglicht es Qure dem DBA, mehr Zeit am oberen Ende der Qualifikationskurve zu verbringen.

Ja, Qure erzeugt Änderungs-Skripte, die messbar die Performance von Datenbank-Objekten und Batches verbessern, und die direkt verwendbar sind, nachdem sie die unternehmensinternen QA Prozesse durchlaufen haben.

Nein, Qure ist nicht in der Lage, sämtlichen Input in Betracht zu ziehen, über den ein erfahrener DBA verfügt:

- Qure verwendet eine feste Anzahl von Optimierungsmethoden. DBAs kennen üblicherweise zusätzliche und können dadurch sogar noch deutlichere Verbesserungen erreichen, wenn sie die Zeit dazu haben. Sie wollen Ergebnisse? Fordern Sie Ihren DBA heraus.

- Qure hat keine Kenntnisse über die maximale Anzahl von Empfehlungen (Änderungen), die innerhalb des Change Management Prozesses eines Unternehmens und eines Projektplanes für einen Release enthalten sein dürfen. Manche Unternehmen beschränken unter Umständen die Anzahl an Änderungen, die mit einem einzigen Change eingespielt werden dürfen. Ein DBA muss diese Entscheidung treffen und die gewünschten Empfehlungen herausuchen. Qure stellt einen robusten Satz von Werkzeugen zur Verfügung, um schnell Empfehlungen zu einem bestimmten Objekt oder einen bestimmten Typen von Empfehlungen, usw. auszusieben.
- Qure's einziges Wissen einer Datenbank stammt aus der Kopie dieser Produktions-Datenbank und aus den Trace-Dateien des Workloads. Jeder DBA wird wissen, dass es jenseits davon viele weitere Variablen gibt, die die Empfehlungen für eine Änderung beeinflussen können.

Gleichwohl sind mit dem Aufkommen von Workload Tuning DBAs weniger belastet mit Störungsfällen und können ihren Fokus auf die Bereiche komplexer Aufgaben richten, die nur Menschen ausführen können. Beispielsweise dem Einschätzen eines Geschäftsprozesses auf die Performance des Workloads der Datenbank oder der Verbesserung der Architektur oder einem netten Gespräch bei einer Tasse Tee oder Kaffee.

	Traditionelles Query Tuning	Workload Tuning
Zeitwahl des Tunings	Reaktiv (nachher)	Proaktiv (vorher)
Vertraut, intuitiv	Ja	Kurze Eingewöhnungsphase
Bandbreite der Optimierungen	Klein	Groß
Testumgebung erforderlich	Nein	Ja
Kenntnis auf die Auswirkung auf andere Prozesse	Nein	Ja
Verbesserungen über mehrere Prozesse hinweg	Nein	Ja
Einfluss auf professionellen Kenntnisstand	In geringem Umfang	Erheblich
Hilfreich bei akuten Performance Krisen	Ja	In geringem Umfang
Hilft Kosten zur Krisenbewältigung zu verhindern	Nein	Ja

Fazit

Wie wir gesehen haben ergänzen sich Query Tuning und Workload Tuning folgenderweise:

Datenbank Performance Tuning Management Praktiker haben nun zwei Mittel zur Verfügung, um eine Datenbank zu tunen. Mit der Einführung von Workload Tuning hat das Query Tuning einen gleichwertigen Partner bekommen. DBAs haben nun ein breiteres Arsenal zur Verfügung, um den stetig wachsenden Anforderungen steigender Volumen mit höherer Geschwindigkeit bei sinkenden Kosten zu begegnen. Gestützt auf eine Wissensdatenbank von SQL Server Optimierungs-Techniken und basierend auf einer ganzheitlichen Analyse aller Aspekte eines Produktion-Workloads, ermöglicht es Workload Tuning einen automatischen Ausgleich zwischen sich entgegenstehenden Anforderungen in komplexen Workloads herzustellen. Der Benutzer übt die Kontrolle aus, in dem er festlegt, was einen Workload ausmacht, in dem er die Empfehlungen, die Begründungen und die Benchmark-Ergebnisse überprüft, in dem er die fertigen Änderungs-Skripte kontrolliert, und diese Skripte je nach Priorität für das Geschäft und seiner eigenen Expertenmeinung zu datenbank-spezifischen Charakteristika annimmt oder ablehnt. Qure arbeitet so vollautomatisiert, dass es Millionen von Abfragen oder Objekte ohne Problem optimieren kann. Dies macht es zur einzigen verfügbaren kosteneffizienten Methode um einen kompletten Produktions-Workload zu optimieren.

Weitere Informationen

Über Ami Levin:

Ami Levin, CTO und Mitbegründer von DBSophic, ist ein Microsoft SQL Server MVP mit mehr als 20 Jahren Erfahrung in der IT-Branche. In den letzten 12 Jahren hat er weltweit zu SQL Server Themen beraten, Seminare gehalten und referiert. Er leitet die israelische SQL Server User Group, betreibt ein lokales Support Forum und ist regelmäßig Sprecher auf internationalen SQL Server Konferenzen. Seine neuesten technischen Artikel finden Sie auf SQL-Server-Performance.com.

Über DBSophic:

DBSophic ist ein innovativer Hersteller für Performance-Management Produkte für datenzentrierte Anwendungen. Während traditionelle Tools das Augenmerk auf die Optimierung individueller Abfragen richten, liefert DBSophic automatisierte Workload Tuning Lösungen. Durch Optimierung des gesamten Applikation-zu-Datenbank Workloads, helfen DBSophic's Lösungen Unternehmen eine Steigerung in der Applikations-Performance zu realisieren, wie sie kein anderes Tool erreicht. Das Unternehmen hat seinen Hauptsitz in Jerusalem.

- > [Klicken Sie hier, um mehr zu Qure's Arbeitsweise zu erfahren](#)
- > [Downloaden Sie hier eine kostenlose Trial-Version von Qure und entdecken Sie, wie sehr Ihre Datenbank-Performance verbessert werden kann!](#)

„Hätte ich darüber nachgedacht, hätte ich mich nicht auf dieses Experiment eingelassen. Die Literatur war voll von Beispielen die besagten, dass es nicht machbar sei.“

-- Spencer Silver über seine Arbeit, die den einmaligen Klebstoff der 3M Post-It Merktzetteln hervorbrachte.



> www.dbsophic.com