

Benchmarking für Oracle-Datenbank-Server

Thomas Kalb, ITGAIN GmbH

Wie kann die Leistungsfähigkeit eines Datenbank-Server-Systems gemessen werden? Welche Möglichkeiten gibt es, um Auswirkungen einer Migration auf neue Versionen des Betriebssystems oder einer anderen Datenbank-Version zu ermitteln? Dieser Artikel beantwortet diese Fragen und verrät, wie ein Kanarienvogel in die Datenbank kommt.

In der Praxis sind es meist ganz konkrete Fragestellungen, die den Datenbank-Verantwortlichen beschäftigen und ein Benchmarking sinnvoll erscheinen lassen: „Wie schneidet mein Datenbanksystem im Vergleich zu anderen Systemen ab?“, „Welchen Preis haben andere für Virtualisierung zu zahlen?“ Eine andere Fragestellung zielt darauf, das Filesystem zu identifizieren, das unter der Berücksichtigung der Workload-Charakteristik und der eingesetzten Hardware den größten Nutzen in Hinblick auf die Performance verspricht. Um gezielt Antworten auf diese Fragen zu erhalten, steht eine ganze Reihe von differenzierten Benchmarking-Tools zur Auswahl.

Benchmarking zur Standortbestimmung

Einen Schritt in Richtung Vergleichbarkeit von Transaktions- und Datenbankmanagement-Systemen (DBMS) verschiedener Hersteller bietet der TPC-Benchmark. Das durchaus bekannte Verfahren liefert begrenzte Antworten auf die aus der Praxis genannten Fragestellungen, ist allerdings recht aufwändig in der Durchführung. So wundert es nicht, dass seine Bedeutung selbst in seiner ursprünglichen Domäne, der Vergleichs-Argumentation der Hersteller, langsam abnimmt.

Grundsätzlich kann Benchmarking singular oder permanent ausgeführt werden. Der TPC-Benchmark ist ein Beispiel für ein Werkzeug, das sich eher für das singuläre Benchmarking anbietet, da der Aufwand zur

Durchführung vergleichsweise hoch ist. Im Kern zielt Benchmarking immer darauf ab, Transparenz über das Leistungspotenzial des eigenen Datenbank-Servers zu vermitteln. Es dient der eigenen Standortbestimmung: Wie leistungsfähig ist mein System eigentlich? Wie skaliert es? Wo liegen die Leistungsgrenzen und wie verhält sich das System, wenn es an seine Grenzen stößt?

Erst durch den Vergleich der Leistungsdaten mit anderen Systemen ergibt sich eine Transparenz der Leistung. Sie bildet die Basis für die Bewertung und die Erkennbarkeit möglicher Schwachstellen. Aus dem Vergleich leiten sich Grundlagen für die Optimierung von Serverkonfigurationen ab. Zusätzlich helfen die Werte aus dem Benchmarking bei einer validen Kapazitätsplanung.

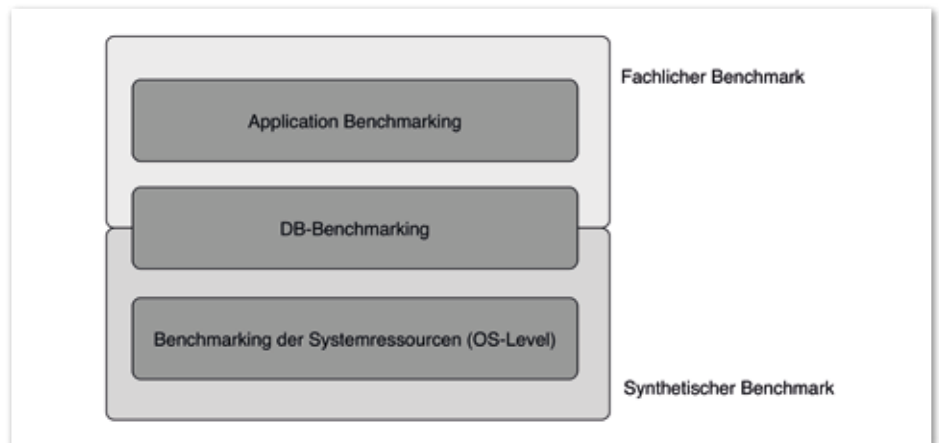


Abbildung 1: Kategorisierung der Benchmarks

I/O Benchmarking Tool	Quelle
Fio	freecode.com/projects/fio
Iometer	www.iometer.org
IOzone	www.iozone.org
Bonnie	www.textuality.com/bonnie/
Bonnie++	www.coker.com.au/bonnie++/
Xdd	www.ioperformance.com/

Tabelle 1: Benchmarking-Tools

Im Gegensatz zum einmaligen Durchführen eines Benchmarkings, das zur Standortbestimmung eines Systems verwendet wird, ermöglicht permanentes Benchmarking die laufende Beurteilung der aktuellen Leistungsfähigkeit.

Benchmarking-Tools im Überblick

Vor dem Benchmarking steht immer die Frage, was überhaupt gemessen und verglichen werden soll. Erst danach folgt die Wahl des passenden Werkzeugs. Die Anzahl der Sockets und Cores sowie deren Taktung oder auch die Größe des Hauptspeichers bilden die rein technische Spezifikation des Datenbank-Servers. Überträgt man dies auf die Automobilbranche, lautet die eigentliche Fragestellung nicht, wie viel PS das Auto besitzt, sondern wie schnell es ist. Und zur Beantwortung dieser Frage trägt die reine Spezifikation wenig bei.

Aufschlussreicher und relevanter für die Praxis sind die Leistungsdaten des Servers, etwa darüber, wie viele Trans-

aktionen pro Sekunde möglich sind (was der Easy Benchmark messbar macht), wie viele I/O-Operationen pro Sekunde (IOPS) oder wie viele Aufträge pro Minute (TPC-C) verarbeitet werden können.

Um die Leistungsfähigkeit zu ermitteln, steht ein breites Portfolio an Benchmarking-Verfahren zur Auswahl, die nachfolgend kategorisiert werden. Im ersten Schritt werden die Benchmarking-Verfahren nach den Schichten einer Anwendung untergliedert. Hierbei erkennt man, dass Application Benchmarks immer fachliche Benchmarks sind, während Benchmarking des OS-Systems immer synthetisch ist. Das Datenbank-Benchmarking kann beides sein (siehe Abbildung 1).

Fachliche Benchmarks beziehen sich auf konkrete Geschäftsprozesse. Synthetische Benchmarks zielen dagegen darauf ab, bestimmte Komponenten eines Serversystems einem Test zu unterziehen. Es sind die synthetischen Benchmarks, die eine Vergleichbarkeit zwischen Systemen zulassen. Normierte fachliche Benchmarks beurteilen nur die Leistungsfähigkeit des gesam-

ten Systems. Nachfolgend werden bekannte Benchmarking-Verfahren gegliedert nach den Schichten einer Anwendung vorgestellt:

- Application Level
- Datenbank
- System (Systemressourcen)

Application-Level-Benchmarks

Auf Basis der Anwendung („Application Level“) arbeiten zum Beispiel Oracle Load Testing, Oracle Real Application Testing, HP Load Runner oder auch JMeter. Diese Benchmarks eignen sich zur Ermittlung der Leistungsfähigkeit eines Gesamtsystems. Dazu werden üblicherweise Geschäftsprozesse und Anwendungsverläufe aufgezeichnet und stets erneut ausgeführt. Mittels solcher „Capture & Replay“-Läufe unterziehen sie die Anwendungssysteme einem Test auf Funktionalität, Geschwindigkeit, Skalierung oder Limits.

Der von Oracle selbst empfohlene Real-Application-Benchmark ermöglicht die Bewertung der Auswirkungen von Systemänderungen. Denn damit lassen sich



Unsere Vorträge zur DOAG 2014 inkl. Download

1)

„Einführung in die Statistik mit R“

mit unserem Sprecher Bernd Weiler
am Messe-Tag 2, Mittwoch der 19.11
von 15.00 – 15.45 Uhr, Raum ‚Helsinki‘

2)

„Graphen in Apex“

mit dem Referenten Thomas Hernando
am Messe-Tag 3, Donnerstag der 20.11
von 16.00 – 16.45 Uhr im Raum ‚Oslo‘

Zu gewinnen!



1. Asus Transformer Pad



2. Sony SmartWatch



3. Kindle Paperwhite



reale Workloads aus der bestehenden Datenbank aufzeichnen und auf dem neuen System wiedergeben. Eventuelle Risiken für den Produktionsbetrieb können so aller Voraussicht nach vermieden werden.

Datenbank-Benchmarks

Um die Leistungsfähigkeit des Datenbank-servers in seiner Kernaufgabe zu messen, sind Benchmarks am besten geeignet, die unmittelbar auf der Datenbank aufsetzen. Ihr Einsatz ist dann unerlässlich, wenn zum Verwalten des Datenbank-Storage ASM zum Einsatz kommt, das mit I/O-Benchmarking-Tools nicht vernünftig erreicht werden kann. Hier kann nur mit Datenbank-Operationen valide gearbeitet werden. Zu dieser Gruppe gehören:

- SLOB (Silly Little Oracle Benchmark)**
 Frei verfügbar; der Kommandozeilen-basierte Client wird für Windows und Linux angeboten. Der synthetische Benchmark eignet sich speziell für den I/O-Test.
- SwingBench**
 Der frei verfügbare Benchmark arbeitet mit Geschäftsprozessvarianten, zum einen „OrderEntry“ (wie „TPC-C“), zum anderen „SalesHistory“ (wie „TPC-H“). Er kann auch für Stress-Tests eingesetzt werden.
- HammerDB**
 Vormalig als „Hammerora“ bekannt. Gehört zu den Open-Source-Benchmarks und kann auch für Datenbanken anderer Hersteller (MySQL, PostgreSQL) eingesetzt werden.
- DELL Benchmark Factory for Databases**
 Ist kostenpflichtig und lässt sich auch für die Datenbanken anderer Hersteller nutzen. Erlaubt das Mitschneiden von Datenbank-Workloads. Umfasst auch Analysen zur Skalierbarkeit und „Was wäre wenn“. Er bildet eine Alternative zu Oracle-Real-Application-Testing.
- ORION - ORacle I/O Numbers**
 Eignet sich für das Stress-Testing des I/O-Subsystems. Dazu kann ein I/O-Workload synthetisch generiert werden, der dann typische Zugriffsmuster auf die Datenbank enthält. Der Benchmark ist über viele Parameter anpassbar.

System-Benchmarks

Bei den System-Benchmarks gibt es solche, die in erster Linie die Rechengeschwindigkeit der CPU ermitteln und dabei auf dem Level des Operation-Systems arbeiten. Dazu gehören Prime95/MPrime, LinX und SysBench. Die zweite Gruppe der System-Benchmarks bilden die Block-Device-Benchmarking-Tools, die explizit die Leistungsfähigkeit des I/O-Subsystems untersuchen. Stellvertretend seien hier Iometer, IOzone, und bonnie++ genannt. Einen kompletten Eindruck hinterlässt Fio (siehe „freecode.com/projects/fio“), der einfach in der Bedienung und verständlich ist (siehe Tabelle 1).

Für Insider: Benchmarking mit „xdd“

„xdd“ (siehe „www.ioperformance.com“) gilt als Geheimtipp für das I/O-Benchmarking. Das Werkzeug ist einfach aufzusetzen (Dokumentation siehe „www.mindwerks.net/wp-content/uploads/2010/09/xdd6.5.pdf“). Sein wesentlicher Vorteil ergibt sich aus der Möglichkeit, die Performance der Anbindung (Daten sind im Cache des I/O-Systems) von der Lese- und Schreib-Performance der Storage-Einheit (HDD, SSD etc.) differenziert zu messen.

```
create table.... storage (buffer_pool recycle) nocache;
create index.... storage (buffer_pool recycle);
```

Listing 1

```
alter system set DB_RECYCLE_CACHE_SIZE=0;
alter system set DB_RECYCLE_CACHE_SIZE=128;
```

Listing 2

```
SQL> show parameter filesystemio_options
NAME                                TYPE                                VALUE
-----                                -
filesystemio_options                string                               none

SQL> ALTER SYSTEM SET FILESYSTEMIO_OPTIONS=SETALL SCOPE=SPFILE;

System wurde geändert.

SQL> shutdown immediate
Datenbank geschlossen.
Datenbank dismounted.
ORACLE-Instanz heruntergefahren.

SQL> startup
ORACLE-Instanz hochgefahren.

Total System Global Area 2471931904 bytes
Fixed Size 2255752 bytes
Variable Size 603980920 bytes
Database Buffers 1862270976 bytes
Redo Buffers 3424256 bytes
Datenbank mounted.
Datenbank geöffnet.

SQL> show parameter filesystemio_options

NAME                                TYPE                                VALUE
-----                                -
filesystemio_options                string                               SETALL

SQL>
```

Listing 3

```
oracle@SLES11SP3:~/dbhome_1/oracle_easy_bench_basis> lsof +fg | grep -i "bench_tab_01"
oracle8339 oracle 261uW REG RW,SYN,DIR,0x88000 8,1 1073750016 5104498 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01.dbf
oracle8339 oracle 262uW REG RW,SYN,DIR,0x88000 8,1 204808192 5104499 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01_idx.dbf
oracle8341 oracle 264u REG RW,SYN,DIR,0x88000 8,1 1073750016 5104498 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01.dbf
oracle8341 oracle 265u REG RW,SYN,DIR,0x88000 8,1 204808192 5104499 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01_idx.dbf
oracle8345 oracle 260u REG RW,SYN,DIR,0x88000 8,1 1073750016 5104498 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01.dbf
oracle8345 oracle 261u REG RW,SYN,DIR,0x88000 8,1 204808192 5104499 /data/oracle/oradata/ORCL01/data-
file/bench_tab_01_idx.dbf
oracle@SLESHSP3:~/dbhome_1/oracle_easy_bench_basis>
```

Listing 4

Die Spezifikation des Benchmarkings kann komfortabel in einer Konfigurationsdatei erfolgen. Ein zentraler Parameter ist dabei „mbytes“. Dessen Vorgabe definiert, wie groß der Bereich ist, aus dem die zufälligen Zugriffe erfolgen sollen. Wird ein sehr kleiner Wert gewählt, befinden sich spätestens nach dem ersten Durchlauf sämtliche Informationen im Cache. Bei besonders großen Werten erfolgt der Zugriff über die physikalische Storage-Einheit.

Das Benchmarking der ITGAIN

Der Easy Database Benchmark ist ein synthetischer Benchmark, der somit einen direkten Leistungsvergleich zwischen verschiedenen Systemen erlaubt. Der Benchmark ist

für Oracle und DB/2 verfügbar. Er läuft unter AIX, Sun Solaris, Linux, HP-UX, Windows sowie z/OS und benötigt wenig Ressourcen. Die schnelle Durchführung bildet die Grundlage dafür, dass der Test regelmäßig und in kurzen Abständen aufgerufen werden kann.

Im Basisteil werden die Leistung der CPU sowie die Leistungsfähigkeit des I/O-Subsystems differenziert nach synchronen (Random) und asynchronen (read ahead) I/O-Operationen gemessen. Der Transaktions-Benchmark führt 10x100.000 Transaktionen als Standardwert aus. Die daraus ermittelten Ergebnisse lassen sich dann unmittelbar auf verschiedene Systeme anwenden, um eine direkte Vergleichbarkeit zu ermöglichen. Der aktuelle Highscore liegt derzeit bei siebzehn Sekunden für die-

se Aufgabe. Der Benchmark kann natürlich auch dazu genutzt werden, die Skalierbarkeit zu testen, indem statt des Faktors zehn höhere Werte fortgeschrieben werden.

Der Limit-Benchmark schließlich führt das System dann an seine Grenzen und beantwortet damit die Frage, wie robust das System unter extremen Leistungsanforderungen ist. Ist die Eingabe von Befehlen noch möglich?

Der Kanarienvogel in der Datenbank

Im vorigen Jahrhundert züchteten Bergarbeiter Kanarienvögel, die sie auch in die Tiefe mitnahmen. Die Arbeiter mögen zwar auch an der Schönheit der Tiere interessiert gewesen sein, konkret ging es

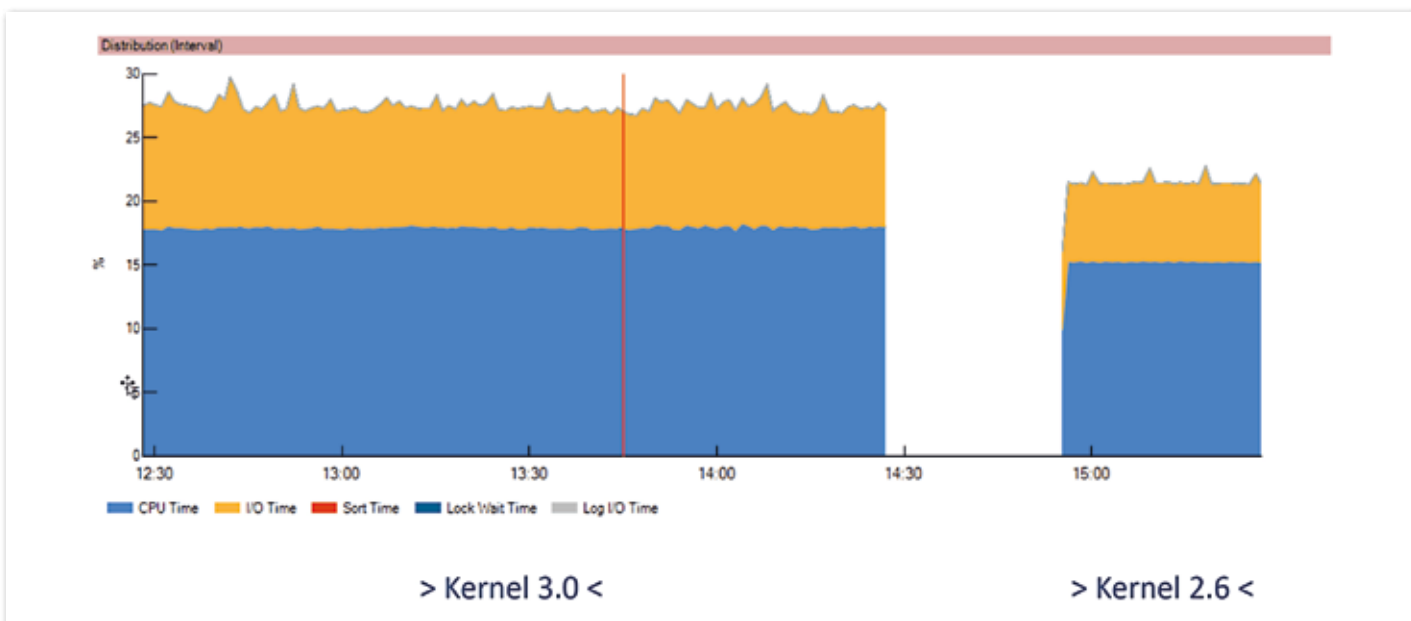


Abbildung 2: Messreihen des permanenten Benchmarkings eines Systems mit Kernel 3.0 & Kernel 2.6

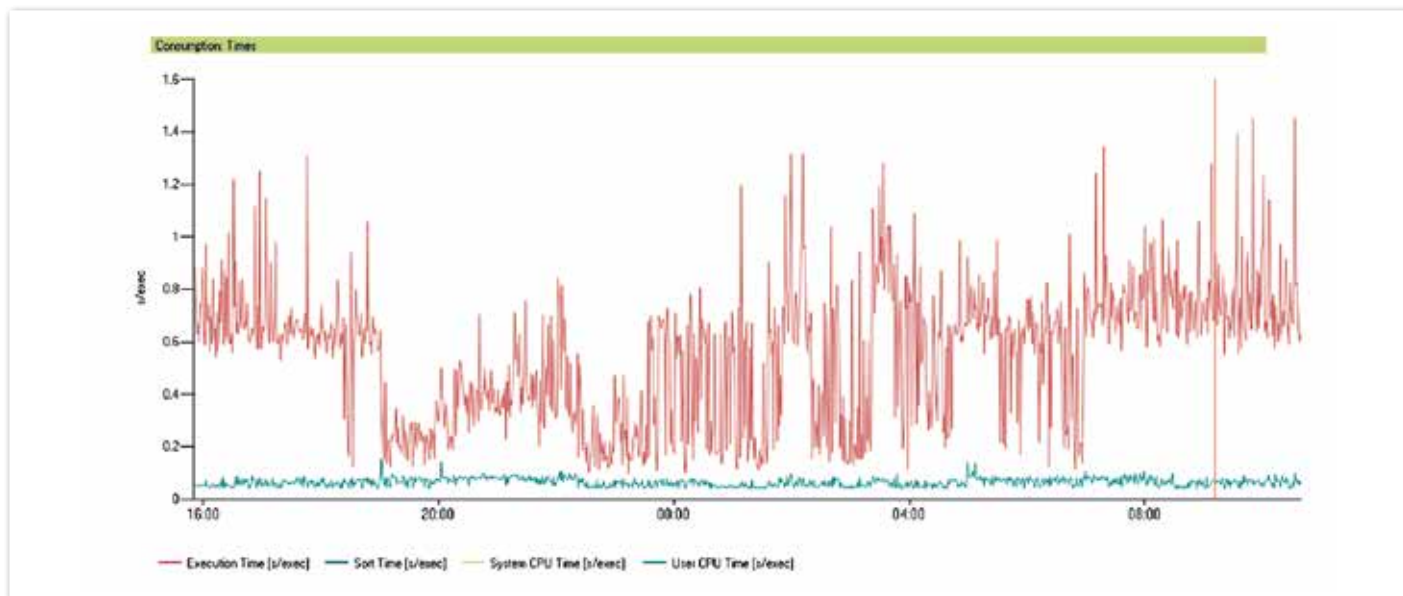


Abbildung 3: Messreihen eines Systems mit CPU- und Ausführungszeit

aber um etwas anderes. Denn so lange die empfindlichen Tiere sangen, bestand keine Gefahr. Verhielten sich die Vögel indes still oder fielen gar von der Stange, mussten die Bergleute auf dem schnellsten Wege den Schacht verlassen. Die Luft war vergiftet und ernsthafte Probleme drohten. Die Bergarbeiter hatten sich ein individuelles Frühwarnsystem geschaffen.

Mit einem permanenten Benchmarking und dessen Monitoring schafft man sich seinen eigenen Kanarienvogel für sein Datenbanksystem und erhält damit die Chance, rechtzeitig auf Leistungsdefizite und andersgeartete Probleme zu reagieren. Das permanente Benchmarking deckt Kapazitäts-Engpässe auf. Verschlechtern sich die Benchmarking-Werte bei Erhöhung des Workloads (etwa Anzahl I/O pro Sekunde), deutet dies darauf hin, dass der Server an seine Leistungsfähigkeit stößt.

Die Herausforderung bei der Implementierung eines permanenten Benchmarkings besteht in der Schaffung vergleichbarer Bedingungen. Wie jeder Datenbank-Administrator weiß, variiert der interne Ressourcenverbrauch beim Abarbeiten von SQL stark. Die benötigten Blöcke liegen entweder bereits im Cache vor oder müssen erst aus dem Storage beschafft werden. Um eine Vergleichbarkeit zu gewährleisten, gilt es, eine Messmethode zu implementieren, die einen identischen Workload mit einem identischen Ressourcenverbrauch verwendet.

Die Lösung für diese Herausforderung besteht in einem Satz von SQL-Anweisungen mit einem fest definierten Ressourcenverbrauch. Die einzelnen SQL-Statements prüfen dabei immer nur eine bestimmte Ressource des Servers (CPU, Netzwerk, I/O etc.).

Um die optimalen Bedingungen zu implementieren, müssen dedizierte Datenbank-Ressourcen geschaffen werden. Benötigt werden ein dedizierter Buffer Cache, zwei Tablespaces und eine Table. Für die Schaffung des Buffer Pools bietet sich unter Oracle der Recycle Cache an. Er kommt in der Praxis eher selten zum Einsatz und wird für Tabellen genutzt, die nur selten gelesen werden, wie zum Beispiel Log-Tabellen. Der Recycle Cache verhindert, dass der Buffer Cache mit Blöcken vollgeschrieben wird, die kein zweites Mal benötigt werden. Er entlastet somit den zentralen Buffer Cache. *Listing 1* zeigt die Definition des Buffer Cache. Eine Variante zum Initialisieren des Buffer Cache ist in *Listing 2* dargestellt.

Der Cache des Dateisystems muss ausgeschaltet sein, damit „Direct IO“ verwendet wird. Das wird mit dem Kommando „ALTER SYSTEM SET FILESYSTEMIO_OPTIONS=SETALL SCOPE=SPFILE;“ erreicht. Für Linuxsysteme kann dies mit dem Kommando „lsdf +fg“ überprüft werden (*siehe Listings 3 und 4*). Erst danach können mit den Benchmark-SQL-Kommandos gezielt einzelne Teilbereiche des Gesamtsystems geprüft werden. Dazu

zählen synchrone Lesezeit, asynchrone Lesezeit und die Netzwerk-Latenz.

Durch den Vergleich der aktuellen Benchmark-SQL-Statements mit der Baseline können dann Leistungsdefizite und Schwachstellen aufgedeckt werden. Handlungsbedarf besteht aber immer nur dann, wenn die Abweichung auch robust (über einen längeren Zeitraum, zum Beispiel fünf Minuten, bestehend) und signifikant (um den Faktor drei höher als die Basis) ist. Ist das der Fall, hat der Kanarienvogel in der Datenbank gerade seinen Gesang eingestellt.

Aus der Praxis für die Praxis

Eine fast klassisch zu nennende Diskussion in Unternehmen findet regelmäßig zwischen Datenbank- und System-Administratoren statt. Streitpunkt ist fast immer eine Diskrepanz zwischen der zur Verfügung gestellten und der empfangenen Leistung. Der Datenbank-Administrator ermittelt mit seinen Werkzeugen eine deutliche Verzögerung der I/O-Leistung. Der System-Administrator kann dies dagegen auf seiner Seite mit seinen Tools nicht bestätigen, wird im Zweifel aber für sich in Anspruch nehmen, dass die Datenbank wahrscheinlich auch stärker genutzt wurde. Permanentes Benchmarking kann den Streit entscheiden und befriedigende Antworten liefern. Denn das permanente Benchmarking objektiviert die Fragestellung, da mit einem synthetischen Bench-

mark zu jedem Messzeitpunkt exakt die gleichen Aktionen durchgeführt werden.

Wie *Abbildung 2* zeigt, ergab die Auswertung der Messreihen bei diesem Kunden eine deutliche Verschlechterung der Performance des Systems, nachdem auf den Kernel 3.0 umgestellt wurde. Unter identischem Workload gibt es bei Kernel 3.0 erhöhte Werte der CPU- und I/O-Zeiten. Ein zunächst überraschendes Ergebnis, das so wohl niemand erwartet hatte. In diesem konkreten Beispiel waren Transparent Huge Pages das Problem, die letztlich trotz des neueren Kernels zu einer Verschlechterung der Werte führten.

Beim Benchmarking in *Abbildung 3* wurden CPU-Zeit und Ausführungszeit mithilfe eines SQL-Statements untersucht, das eine besonders große Ergebnismenge erzeugte. Die Differenz zwischen beiden Werten liefert die Netzwerkzeiten zurück. Erst durch die regelmäßige Ausführung des Benchmarks und den langen Betrachtungszeitraum wurden die gravierenden

Unterschiede hinsichtlich der Zeiten sichtbar.

Die grafische Auswertung zeigt, dass es im Laufe des Tages während des Batchbetriebes (19 Uhr bis 7 Uhr) durchaus einige Zeitpunkte gibt, an denen die Differenzen recht kurz sind. Überwiegend (besonders in der Onlinezeit von 7 Uhr bis 19 Uhr) kommt es jedoch zu deutlichen Verzögerungen. Daraus kann geschlossen werden, dass die eigentliche Netzwerk-Anbindung des Systems in Ordnung ist. Problematisch ist offensichtlich die Bandbreite. Einzelne Benchmarks hätten dieses Analyse-Ergebnis kaum sichtbar gemacht, selbst wenn der Test zu verschiedenen Zeitpunkten wiederholt worden wäre. Erst das permanente Benchmarking und Monitoring über den größeren Zeitverlauf zeigte das Verbesserungspotenzial.

Fazit

Mit dem singulären Benchmarking lassen sich unterschiedliche Datenbank-Systeme miteinander vergleichen. Die Ergebnis-

se dienen damit der Standortbestimmung. Die Kombination von permanentem Benchmarking und Monitoring dagegen schafft laufend Transparenz und zeigt Leistungsdefizite des Serversystems auf. Es eignet sich damit als individuelles Frühwarnsystem, weil Veränderungen und Leistungseinbußen zeitnah und empirisch nachweisbar werden. So schafft man sich seinen Kanarienvogel in der Datenbank.



Thomas Kalb
thomas.kalb@itgain.de

ORACLE Gold Partner
Specialized
Oracle Database

MUNIQSOFT
Datenbanken mit iQ

IT-Consulting / Schulungen

Hinweis
Erfahren Sie mehr über **Oracle RAC 12c** als Basis für Ihre **internen Cloud-Lösungen**.
Unsere Experten stehen Ihnen im Rahmen von **Workshops, Schulungen** oder **klassischer IT-Beratung** von der Konzeption bis hin zu Implementierung und Betrieb gerne zur Seite.
www.muniqsoft.de