

Performance Analyse und Prognose mit Hilfe statistischer Methoden

Felix Castillo Sanchez

Freiberuflicher Berater

Glashütten-Schloßborn

Schlüsselwörter

Performance Analyse, explorative Datenanalyse, Statistiken, Wait-Events, Event-Histogramme

Abstract

Die Analyse der Oracle Performance gründet seit Version 10g auf der Auswertung der drei Informationsquellen Statistiken, Wartezeiten und CPU-Verwendung. Darauf aufsetzend werden verschiedene Auswertungen angeboten, automatisch erstellt oder nachträglich erstellbar. Doch sowohl der Aufbewahrungszeitraum als auch die Genauigkeit der zugrunde liegenden Daten sind für eine genaue Analyse nicht ausreichend. Die vorhandenen Möglichkeiten können zwar unmittelbare und auffällige Probleme gut darstellen. Sie können jedoch weder schleichenden Leistungsverlust erkennen lassen noch sind sie für eine Prognose nutzbar. Zudem können sie auch bei einer "regulären" Analyse zu falschen Schlussfolgerungen führen.

Mit statistischen Methoden wie der explorativen Datenanalyse können unbekannte Zusammenhänge entdeckt und damit Schlussfolgerungen bestätigt oder widerlegt werden. Mit Regressionsanalysen können Prognosen über die Skalierbarkeit erstellt werden.

Einleitung

Im Allgemeinen wird Performance Analyse und Tuning als ein zusammenhängender Prozess verstanden, der bei Leistungsproblemen zur Identifizierung der Ursache(n) ausgeführt wird. Je nach Situation – sprich Dringlichkeit und/oder Auswirkung – kann die Vorgehensweise und die Wahl der Hilfsmittel anders ausfallen. Grundsätzlich wird jedoch immer nach einem oder mehreren Verursachern gesucht. D.h., welche(r) SQL/Session /User/Job/etc. verbraucht am meisten welche Ressourcen. Dabei stellt sich immer die Frage *was* wird *wie oft* ausgeführt und *wie lange* wartet man darauf und *wie viel CPU-Zeit* wird dafür benötigt. Der erste Teil kann mit Hilfe der Statistiken (`v$sysstat`, `v$sesstat`, ...), der zweite mit Hilfe des Wait-Interfaces (`v$system_event`, `v$session_event`, `v$event_histogram`...) und der dritte mit dem DB-Time Interface (`v$sys_time_model`, `v$sess_time_model`, ...) beantwortet werden.

Was aber, wenn keine eindeutigen Verursacher für Performance-Probleme identifizierbar sind oder eine Analyse zur Bewertung der Leistungsfähigkeit eines Systems durchgeführt werden soll?

Wait-Interface: Wartezeit und Histogramme

Der in `v$system_event` mitgeführte Durchschnittswert eines Wait-Events ist das exakte arithmetische Mittel aller Wartezeiten. Das Problem ist, dass der isoliert betrachtete Durchschnittswert keine Rückschlüsse auf die Entstehung zulässt. Dazu hat Oracle mit der Version 10g die Wait-Histogramme eingeführt. Diese teilen die **Häufigkeit** der Events in Klassen ein.

Beispiel: Wartezeiten *log file sync*

Im Folgenden soll das Problem am Beispiel des Events *log file sync* dargestellt werden.

Gemäß `v$sysstat` beträgt die durchschnittliche Wartezeit 15.9 ms. (139.296.479 Millisekunden Wartezeit und 8.735.478 Events).

Dabei tritt gemäß `v$event_histogram` folgende Verteilung auf:

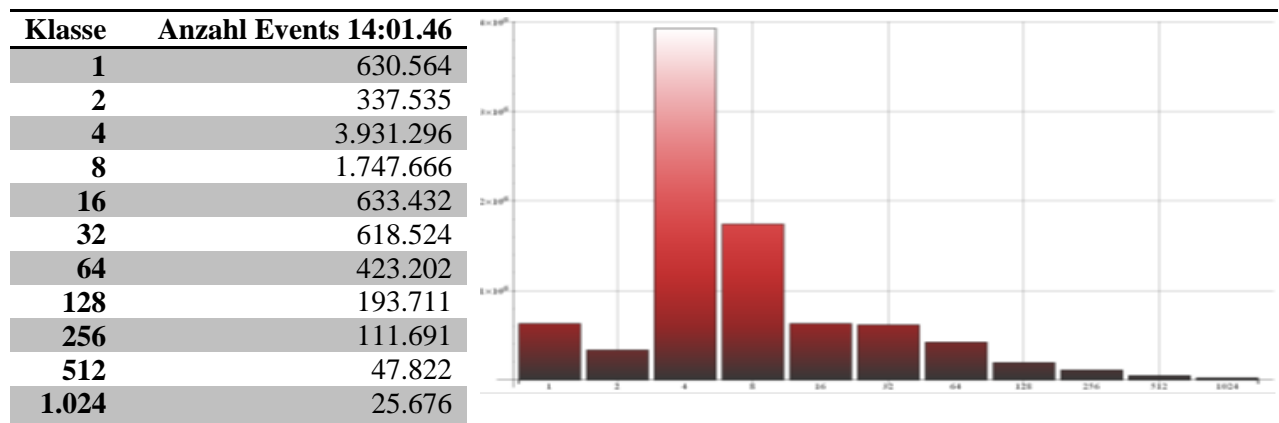


Tabelle 1: Häufigkeit der Wartezeiten über den gesamten Zeitraum

Da Histogramme die **Häufigkeit** der Events beschreiben, ist keine Aussage über die **Wartezeit** möglich. So kann z.B. Klasse 4, die die Zeiten von > 2 bis 4 ms beschreibt, eine Wartezeit von 7.863 – 15.725 Sekunden zusammenfassen. Alle Klassen zusammen können eine Wartezeit von 95.780 bis 192.189 Sekunden abbilden. (De facto liegt sie in diesem Fall bei 139.296 Sekunden).

Wie sieht es aus, wenn ein kurzer Zeitraum isoliert betrachtet wird (z.B. der AWR-Standardintervall von einer Stunde)?

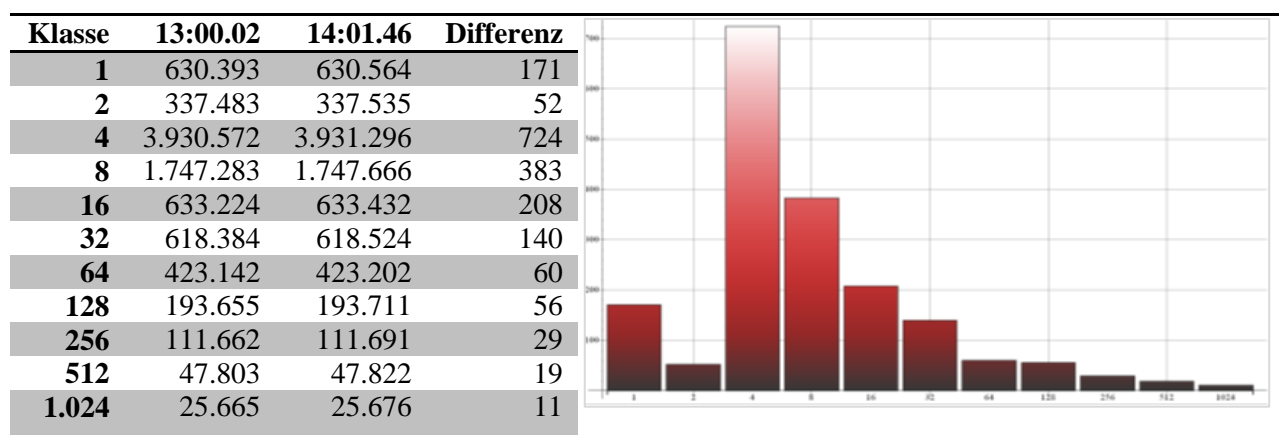


Tabelle 2: Häufigkeit der Wartezeiten über eine Stunde

Optisch betrachtet ist der Unterschied minimal. Gemäß *AWR-Report* beträgt die durchschnittliche Wartezeit jedoch **20 ms!**

Dieser Wert lässt sich nicht direkt auslesen, da er aus der Differenz der Häufigkeit und der Wartezeit berechnet wird.

$$\frac{139\,296\,479 - 139\,258\,988}{8\,735\,478 - 8\,733\,622} = 20.19 \text{ ms/Event}$$

Fazit

Der große Unterschied zwischen dem Allzeit-Durchschnitt und dem eines zufälligen Zeitintervalls zeigt, dass es wohl zu großen *Fluktuationen* kommt und es deshalb wichtig ist, sowohl die Intervalle drastisch zu verkürzen als auch die Methodik der Untersuchung zu verbessern.

Datensammlung – *Performance Data Warehouse*

Die von Oracle im AWR-Repository historisierten Daten haben neben der Lizenzfrage auch den Nachteil, standardmäßig nur jede Stunde und dazu auch nur für zwei Wochen aufbewahrt zu werden. Sowohl die Auswahl als auch die Tiefe der historisierten Daten sind versionsabhängig.

Werden AWR-Snapshots häufiger und ausführlicher angelegt und zudem länger aufbewahrt, wird das SYSAUX-Tablespace unverhältnismäßig groß. (Beispiel: Oracle 11gR1, *statistics_level* auf *ALL*, Intervall 10 Minuten: nach 3 Monaten ⇒ SYSAUX über 90 GByte!)

Um die aufgezählten Probleme zu umgehen, wurden eigene Abfrage-Routinen erstellt. Diese speichern u.a. alle Daten aus *v\$sysstat*, *v\$system_event*, *v\$event_histogram* in 30 Sekunden-Intervallen in Dateien. Das komprimierte Datenvolumen hierfür liegt bei ca 500 Mbyte/Woche. Es besteht aber noch erhebliches Potenzial, die Datenmenge ohne Informationsverlust zu optimieren.

Zusätzlich kann jede gewünschte Information in Form eigener Abfragen Werte liefern, um diese isoliert oder im Zusammenhang mit allgemeinen Daten zu analysieren.

Auswertung

Folgender Graph zeigt alle berechneten Wartezeiten über einen Zeitraum von 9 Tagen in 30 Sekunden-Intervallen.

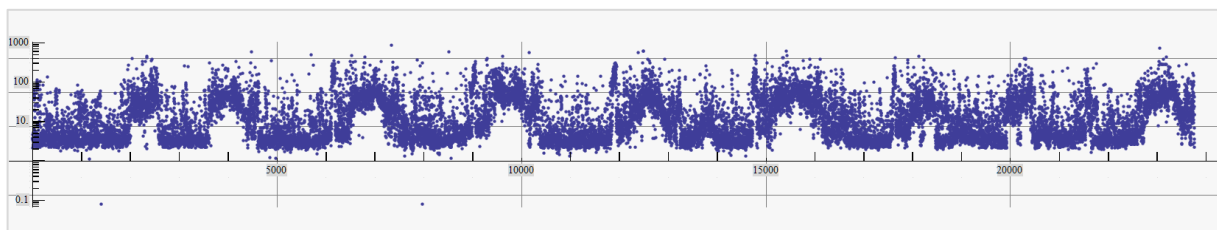


Abb. 1: Wartezeiten über 9 Tage (*Darstellung logarithmisch*)

Wie vermutet variieren die einzelnen Werte sehr stark und werden massiv von Ausreißern beeinflusst. Deshalb ist eine Bewertung resp. Analyse mit Hilfe allgemeiner Durchschnittszeiten und der zugehörigen Histogramme in der von Oracle vorgelegten Form nicht aussagekräftig genug.

Die Statistik bietet hierfür verschiedene Methoden.

Median, Oberes und Unteres Quartil – Boxplot

Um die Gewichtung der Ausreißer einzuschränken bietet sich die Verwendung des Medians und der oberen und unteren Quartile an. Diese Werte werden dann idealerweise über einen *Boxplot* dargestellt.

Mit dieser Darstellung werden viele Informationen kompakt dargestellt. So zeigt der rote Block – der *Interquartilsabstand IQR* – die mittleren 50 % aller Events, das untere Quartil links, das obere rechts. Der dicke weiße Strich markiert den Median und die gestrichelte Linie den Durchschnitt. Die „Antennen“ – Whisker – können frei gewählt werden, im Beispiel sind sie auf Faktor 0,5 des IQR-Abstandes gesetzt.

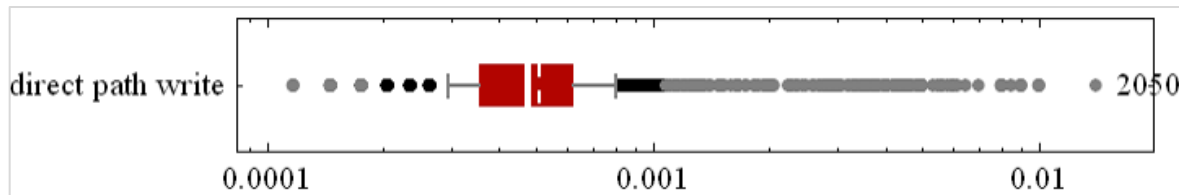


Abb. 2: BoxPlot-Beispiel

Alle Messpunkte, die außerhalb der Whisker liegen, werden als schwache (schwarze Punkte) und starke (graue Punkte) Ausreißer bezeichnet. Die Zahl 2.050 bezeichnet die Anzahl der berücksichtigten Messpunkte insgesamt. (*Anm.:* In der Statistik werden üblicherweise Werte erst ab 1,5 IQR als schwache und ab 3 IQR als starke Ausreißer bezeichnet).

Mit Hilfe der selbst gesammelten Daten können auch Wartezeiten erfasst werden, die mit Oracle-Mitteln nicht analysiert werden können, da diese weit unter 1 ms liegen.

Diese Auflösungsproblematik kommt überall dort vor, wo Wartezeiten entweder zu klein sind (z.B. Latenzzeiten einer RAC-Interconnects mit Myrinet, Festplattenzugriffe bei SSDs, etc.) oder bei Wertebereichen, die am unteren Ende als gut zu bezeichnen sind, jedoch am oberen Ende als inakzeptabel (z.B. mechanische Festplattenzugriffe im Bereich von 8-16ms).

Histogramme

Werden nun die selbst berechneten Werte als Histogramm dargestellt, erhält man einen wesentlich detaillierteren Einblick zu einem Event. Die Grafiken zeigen dieselben Werte!

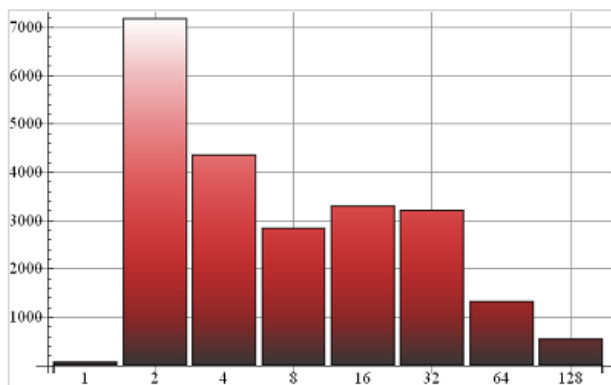


Abb. 3 – Klassenbreite in 2er-Potenz-Schritten

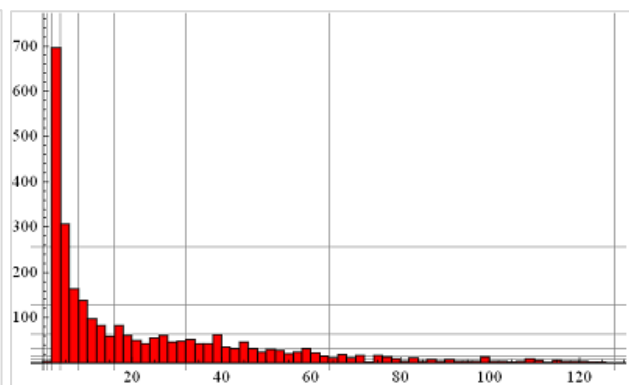


Abb. 4 – Klassenbreite 2ms

Eine grafische Darstellung der Histogramm-Informationen aus `v$event_histogram` zeigt sehr grob die Verteilung der Wartezustände. Mit den selbst berechneten Werten lassen sich wesentlich genauere Histogramme erstellen, die die **Über-/Unterbewertung** von Wertebereichen verhindern, da die Klassenbreite selbst definiert werden kann..

Histogrammen fehlt jedoch weiterhin ein wesentlicher Aspekt, um Wartezeiten besser bewerten zu können: **Der zeitliche Zusammenhang**

Oberes/Unteres Quartil, Median – zeitliche Darstellung

Durch den *Plot* der Quartile (der Median ist das mittlere Quartil) wird die Bandbreite¹, in der sich 50% aller Wartezustände abspielen, ersichtlich.

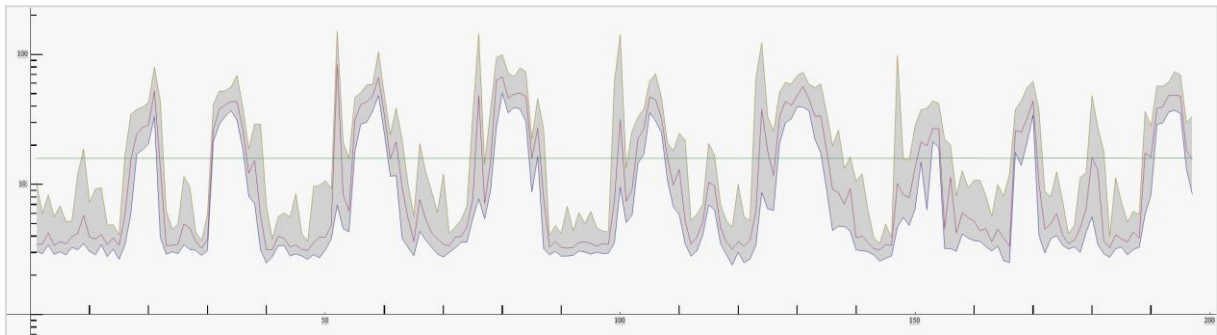


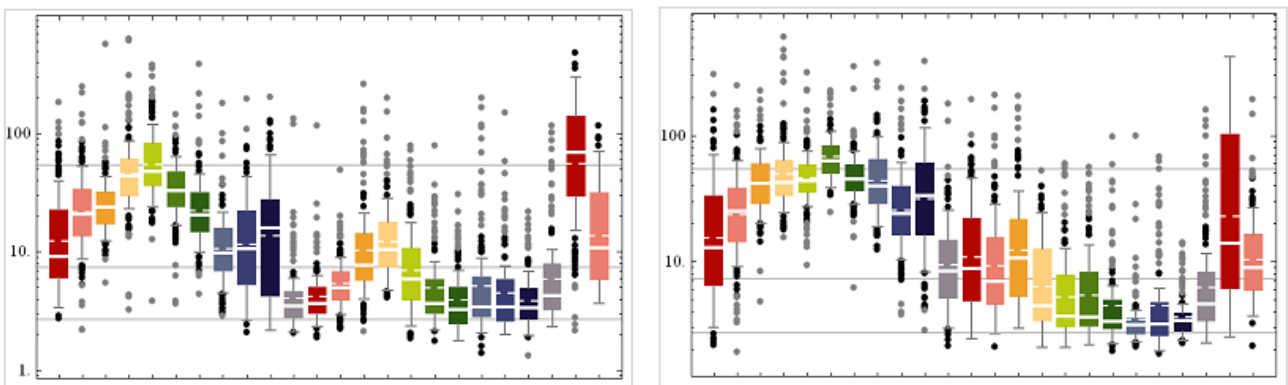
Abb. 5 - Zeitlicher Verlauf der Wartezeiten (9 Tage) - Oberes/Unteres Quartil/Median und Oracle Durchschnitt

Nun ist erkennbar, dass der Zeitpunkt einer Analyse absolut relevant ist. (Das gilt genauso für die Auswertung von AWR-Reports.)

Liegen zudem diese Daten auch über lange Zeiträume vor, können auch Trends erkannt werden, die zeitlich über die sonst üblichen Aufbewahrungsfristen hinausgehen.

Boxplot – zeitliche Darstellung

Die folgenden Grafiken zeigen den Verlauf zweier aufeinander folgender Tage mit Boxplots an. Dabei werden die Daten stundenweise zusammengefasst:



Vergleicht man die Grafiken, so wiederholt ist nicht nur das Wellenmuster, sondern selbst die Interquartil-Abstände.

¹ IQR – Inter Quartil Range

Punktewolke

Die Punktewolke zur Darstellung von Wartezeiten eröffnet zusätzliche Möglichkeiten der Schlussfolgerungen. Bei der Gegenüberstellung von Wartezeit und Häufigkeit, mit der diese vorkommen, können Bereiche erkannt werden, die als nicht akzeptabel gelten.

So liegen im vorliegenden Beispiel alle Zugriffszeiten für *db file sequential read* in einem guten Bereich, da Wartezeiten von > 10ms fast nicht vorkommen. Selbst Ausreißer sind nur sporadisch vorhanden. (*Anm:* Diese Annahme gilt jedoch nur für mechanische Platten, bei SSDs wäre der Plot anders zu interpretieren.)

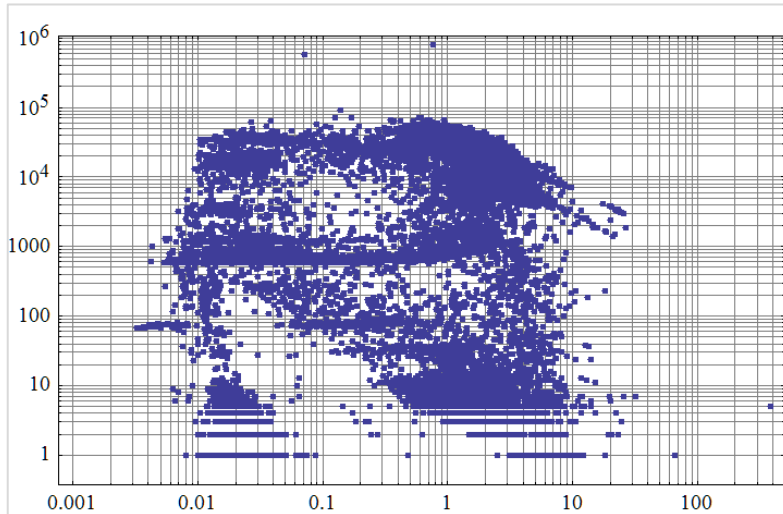


Abb. 6 - Punktewolke: Vergleich Wartezeit und Häufigkeit

Mit Hilfe eines solchen Plots lassen sich schnell problematische Events ermitteln.

Statistiken / Ratios

Ratios sind seit der Einführung und der allgemeinen Akzeptanz der Performance Analyse mit Hilfe von Wait-Events fast vollständig verschwunden. Aber auch bei Ratios ist das grundlegende Problem, dass bei der Betrachtung der Werte der zeitliche Zusammenhang fehlt und/oder die zeitliche Auflösung zu grob ist.

So hilft selbst der *Buffer Cache Hit Ratio*, zusammen mit anderen Werten, in einem zeitlichen Kontext potenzielle Probleme und Zusammenhänge zu erkennen.

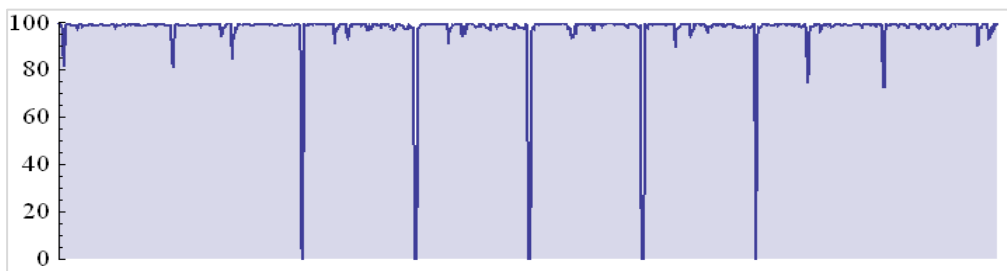


Abb. 7 - *Buffer Cache Hit Ratio*, Zeitraum: 9 Tage, Intervall: Durchschnitt über 12 Minuten

So ist im vorliegenden Beispiel ersichtlich, dass der *Buffer Cache* zu fünf, offensichtlich zusammenhängenden, Zeitpunkten komplett ausgetauscht wird.

Zudem hat die allgemeine Aufbewahrung von Oracle-Statistiken den Vorteil, jederzeit über Informationen zu verfügen, die weiterführende Analysen ermöglichen.

So ist beispielsweise die Transaktionsrate (trx/s) ein wesentlicher Parameter bei Performance-Prognosen. Je umfassender und genauer diese Werte bekannt sind, desto besser sind die daraus ableitbaren Prognosen.

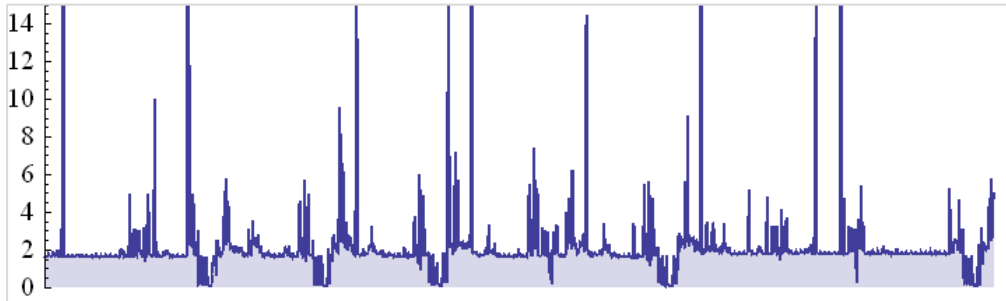


Abb. 8 - Transaktionsrate pro Sekunde, Zeitraum: 9 Tage, Intervall: 30 Sekunden

Zusammenfassung

Die Oracle Statistiken, das Wait- und DB Time-Interface bieten alle für eine Performance Analyse notwendigen Informationen. Bei der Untersuchung bestehender Performance Probleme sind die existierenden Möglichkeiten ausreichend. Eine Analyse von Systemen ohne Probleme ist jedoch eingeschränkt möglich, da sich alle üblichen Methoden auf die *Suche nach dem Verursacher* konzentrieren.

Eine problemunabhängige Performance Analyse ist nur durch statistische Methoden möglich. Hierzu ist gerade die grafische Aufbereitung der Daten als Methode der *explorativen* oder auch *hypothesengenerierenden Statistik* anzuwenden.

Oracle Statistiken können hierfür unverändert übernommen werden. Die Auswertung der Wait-Informationen jedoch stellt sich problematisch dar, da die abrufbaren Durchschnittswerte als auch die entsprechenden Histogramme zu ungenau sind.

Jedoch lassen sich mit einfachen Mitteln die für eine Bewertung notwendigen Informationen selbst ermitteln und darstellen. Die Rohdaten können in einem *Performance Data Warehouse* über lange Zeiträume aufbewahrt und für spätere Analysen verwendet werden, z.B. Trendanalyse, Zeitreihenanalyse, Regressionsanalysen.

Die gezeigten Auswertungsmöglichkeiten beschränken sich nicht nur das Oracle Wait-Interface und den -Statistiken, sondern können auch für beliebige Auswertungen im Datenbank-Kontext eingesetzt werden.

Kontaktadresse

Felix Castillo Sanchez
Akazienweg 6
D-61479 Glashütten-Schloßborn

Telefon: +49 (0) 174 900 49 89
E-Mail: felix.castillo@oraconsult.de
Internet: <http://blog.oraconsult.de>