

Lasttest für die Datenbank mit Swingbench

Matthias Reimann
GISA GmbH
Halle (Saale)

Schlüsselworte

Oracle, Datenbank, Lasttest, Swingbench.

Einleitung

Jeder Datenbankserver ist anders. Ressourcen können ausgewechselt und geändert werden. Nicht selten zeigt die Änderung nicht den gewünschten Erfolg. Die Datenbank ändert dabei einfach ihr Verhalten. Wer vorher testen möchte und auch mal die Leistungsgrenze des Datenbankservers wissen möchte kann dafür in 45 Minuten den Grundstein legen. Gezeigt wird mit dem Werkzeug Swingbench, was ein Lasttest ist und was dafür erforderlich ist. Und das dies in 45 Minuten einsatzbereit ist. Zur Sprache kommt aber auch was Sinn macht und was man lassen kann. Gezeigt wird auch, wie ein individueller Lasttest organisiert werden kann und welche Szenarien möglich sind.

Vortragsziel

Lasttests nehmen viel Zeit in Anspruch. Dieser Vortrag soll den Einstieg beschleunigen und erleichtern.

Grundlagen – Die Ressourcen einer Datenbank

Die drei Ressourcen, die einer Datenbank zur Verfügung stehen sind CPU, Memory und Disk I/O. Die Geschwindigkeit definiert sich aus einer Zusammenstellung dieser drei Ressourcen. Hier hat der DBA die Wahl, mal mehr mal weniger. Die Vielfalt verschiedener Hersteller und Geschwindigkeiten hat dabei in den letzten Jahren weiter zugenommen. Der Einfluss des DBA dadurch aber auch, denn die Verfügbarkeit und Möglichkeiten der Skalierbarkeit gilt es zu beeinflussen somit es geht.

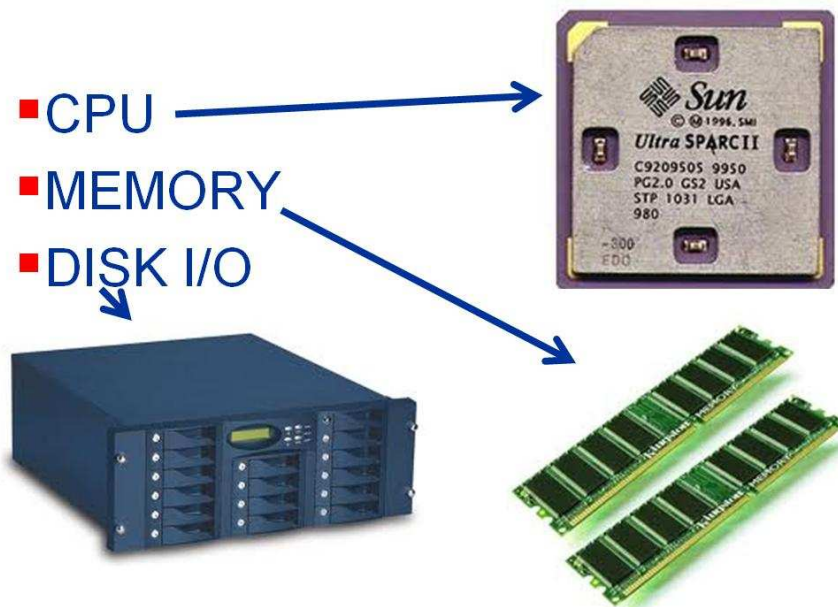


Abb. 1: Ressourcen einer Datenbank

Auch das Betriebssystem, welches diese Ressourcen zur Verfügung stellt spielt dabei eine nicht unwesentliche Rolle.

Grundlagen - Die Evolution der DB-Instanz

In den Anfangszeiten waren die Datenbanken oft noch klein und es reichten wenige Ressourcen aus um eine Datenbank zu betreiben. Nicht selten war die Datenbank und die Anwendung auf einem Server, auch aus Kostengründen. Ressourcenkontrolle war hier nicht möglich und auch noch nicht gefragt. Stabilität stand hier oft im Vordergrund.

Größer werdende Datenbanken sorgten nachfolgend dafür, das Datenbanken eigene Server bekamen und damit standen alle Ressourcen der Datenbank zur Verfügung. Das es sich dabei oft immer noch um Hardware mit eingebauten Festplattencontrollern handelte, war der Memory in Größe und Verwaltung die einzige Ressource, die gemanagt werden musste.

Die Arbeit des DBAs im Bezug auf Ressourcenverwaltung war bis hierher eher klein und übersichtlich. Wenn ein System nicht mehr Performanz ist, dann musste neue Hardware ran und die Datenbank umziehen.

Mit der Einführung der Virtualisierung konnte dann der DB-Instanz die Ressourcen eines Servers zugewiesen werden. Alle drei Ressourcen sind seit dem dynamisch und können je nach Betriebssystem und Datenbankversion auch Online geändert werden.

Seit dem ist die Aufgabe des DBAs komplexer geworden. Er kann nun die Ressourcen variieren. Wenn eine Datenbank nicht mehr ihre Leistung bringt, dann heißt es Anforderungen stellen, aber welche? Und das geschieht meistens erst im produktiven Betrieb, wenn die Zeit knapp ist.

Es gilt eine Möglichkeit zu schaffen, die Ressourcen eine DB-Instanz möglichst genau zu disponieren, so dass der produktive Betrieb so lange wie möglich ohne Änderungen geht. Oder eine neue Umgebung entsprechend geprüft an den Start geht.

Abgrenzung Performancetest, Benchmark, Lasttest

Ein Performancetest soll zeigen wie schnell eine Datenbank ist. Die normale Antwortzeit für die wichtigsten SQLs ist hier z.B. gefragt.

Ein Benchmark soll die Leistung eines Datenbanksystems vergleichbar machen. Nicht nur im Bezug auf Ressourcen, sondern sogar auf Systeme und Hersteller.

Mit einem Lasttest soll versucht werden das Datenbanksystem bis an die Leistungsgrenze zu belasten und gezeigt werden, wo die Grenzen sind. Wie viel ist maximal möglich und sind die Ressourcen dabei gleichmäßig ausgelastet.

Warum ausgerechnet Lasttest?

Es gibt viele Fragen, die sich ein DBA heute zu stellen hat, hier einige Beispiele:

- Ist mein System bei voller Belastung überhaupt noch administrierbar?
- Nehme ich 12c Multithreading Option?
- 12c nonCDB oder CDB+1xPDB?
- i86 oder SPARC?
- Verschiedene SAN, lohnt sich ein Besseres überhaupt?
- Habe ich nach jahrelangen Patches immer noch die gleiche Leistung?
- Ist mein neues System schneller als das Alte?
- Features von einzelnen Systemkomponenten testen.
(Virtualisierung, SAN usw.)
- Last erzeugen für Testsysteme?

Was wird gebraucht um Lasttests durchzuführen?

Als erstes werden Werkzeuge benötigt, um die Last zu messen. Das können einmal die mitgelieferten Werkzeuge der einzelnen Betriebssysteme sein, aber auch spezielle Tools. Um in eine Tabelle die Messwerte verschiedener Komponenten zu vergleichen sollten die Werkzeuge in der Lage sein

Aufzeichnungen zu machen. Nur, wenn die Messungen auch der Last zugeordnet werden können, macht die Messung auch Sinn.

Dann wird ein Werkzeug benötigt, welches die Last erzeugt. Dieses Werkzeug sollte in der Lage sein, die Last variabel zu gestalten und über mehrere Computer koordiniert diese zu erzeugen. Das kann natürlich auch die Applikation für die Datenbank sein. In der Regel wird die aber sehr aufwendig. Deshalb ist ein spezielles Tool hier die bessere Wahl.

Dazu gehört dann auch ein für diese Last geeigneter Datenbestand. Meistens steht einem kein Datenbestand einer Applikation in variabler Größe zur Verfügung.

Swingbench, als ein Beispiel für Lasttests

Swingbench ist ein Werkzeug, welches alle Kriterien für einen Lasttest erfüllt. In soll hier aber nicht die Kriterien für die Eignung dargestellt werden, sondern die Handhabung. Deshalb hier ein kurze Einführung, wie man in kürzester Zeit eine Lasttestumgebung aufbaut.

Demonstration

Praktische Vorführung für den Einstieg in Swingbench.

Fazit

- Der Zeitaufwand ist bei geplanter Vorgehensweise überschaubar.
- Vorgehensweise dokumentieren. Ein Lasttest den niemand nachvollziehen kann, macht wenig Sinn.
- Für grössere Datenbanken braucht man mehr als ein Client.
- Testabbruch führt zu nicht abgebauten Session.
- Swingbench ist eine geeignete Software um die Ressourcen seiner Datenbank gut zu testen.

Kontaktadresse:

Matthias Reimann
GISA GmbH
Leipziger Chaussee 191a
D-06108 Halle

Telefon: +49 (0) 345 585-2044
E-Mail matthias.reimann@gisa.de

```
ALTER USER scott  
IDENTIFIED BY User_Name  
REPLACE tiger
```