

High-End DB-Applikationen für alle Kunden ?

Möglichkeiten zur Lizenzkostenreduzierung

Dr. Kurt Franke

Cellent Finance Solutions AG

Stuttgart

Schlüsselworte

Enterprise Edition, Standard Edition, Embedded Software Licence

Einleitung

Für Applikationen mit Verarbeitung und Vorhaltung grosser Datemengen ist es oftmals notwendig, Features der Oracle Enterprise Edition (z. B. SQL-Parallelisierung und Datenkomprimierung) und darüber hinaus weitere Optionen einzusetzen, wie beispielsweise die für eine lineare Skalierung mit wachsenden Datenmengen nahezu unabdingbare Tabellen- und Index-Partitionierung. Schon die Enterprise Edition und natürlich auch darauf aufbauende Optionen sind mit deutlich höheren Lizenzkosten verbunden als eine einfache Standard Edition. Für Kunden, die selbst große Mengen an Kundendaten zu verwalten haben, ist der Einsatz von Enterprise Edition und zusätzlichen Optionen meist schon gegeben, so dass sich die Lizenzkosten einer weiteren Applikation im üblichen Rahmen halten.

Anders sieht es aus, wenn im Normalfall die Standard-Edition ausreichend ist, um alle Performanceanforderungen zu erfüllen. In diesem Falle stellen die Lizenzkosten einer derartigen fachlich durchaus gewünschten Applikation eventuell ein Hindernis gegen die Einführung dar.

Für den Hersteller der Applikation ist es natürlich wünschenswert, auch Kunden zu erreichen, die nicht unbedingt eine High-End-Variante benötigen. Im Prinzip bieten sich dafür zwei unterschiedliche Möglichkeiten an: Zum einen ein abgestuftes Downsizing der Applikation bis hinunter zur einfachen Standard Edition, zum anderen der Einsatz der Embedded Oracle Software Licence mit einer deutlichen Lizenzkostenreduzierung für den Kunden bei gleichbleibender Applikations-Technologie.

Im folgenden wird beschrieben, wie sich die beiden Möglichkeiten in der Herstellung und Bereitstellung der Applikation für die Kunden unterscheiden und wo die jeweiligen Vorteile und Nachteile liegen.

Möglichkeit A: abgestuftes Downsizing bis zur Standard-Edition

Bei dieser Art der Bereitstellung wird dem Kunden genau das angeboten, was in seine Umgebung passt. Wenn er normalerweise eine Standard-Edition einsetzt, bekommt er die Applikation als Standard-Edition-Ausführung, wenn er die Enterprise-Edition einsetzt, bekommt er die Applikation als Enterprise-Edition-Ausführung, und wenn er noch zusätzliche Optionen zur Enterprise-Edition wie Partitioning einsetzt, bekommt er die Applikation als Enterprise-Edition-Ausführung mit Verwendung dieser Optionen, soweit sie von der Applikation benötigt werden.

Wie man unschwer erkennt, kann dies erhebliche Mehraufwände bei der Erst-Entwicklung und vor allem auch bei der Fort-Entwicklung zu weiteren Versionen mit erweiterter Funktionalität bedeuten. Insbesondere bei der Weiterentwicklung besteht die Gefahr des unbeabsichtigten Auseinanderlaufens der Funktionalität in den Varianten für die verschiedenen Editionen.

Um die Gefahr dieses Auseinanderlaufens zu minimieren und die zusätzlichen Aufwände für die verschiedenen Varianten zu begrenzen, wird man versuchen, mit einer Kernversion auszukommen.

Für den Zugriff der Benutzer-Applikation auf die Datenbank ergibt sich keinerlei Unterschied. In der Datenbank selbst jedoch gibt es entscheidende Unterschiede. So können beispielsweise parallele Abfragen, Bitmap-Indexe, Daten-Komprimierung bei Batch-Befüllung bzw. sogar die Variante für OLTP-Komprimierung oder die Tabellen-Partitionierung eingesetzt werden oder auch nicht.

In den mit PL/SQL-Packages realisierten Batch-Verarbeitungen bietet sich an, genau eine Implementierung zu erstellen, in der auch Code-Zweige vorhanden sind, die nicht in jeder Variante ausgeführt werden müssen und dürfen. Zur Steuerung bietet sich ein spezielles Package an, das die entsprechenden Definitionen als Konstanten enthält. Als Beispiel sei hier das von Oracle bereitgestellte Package `dbms_dv_version` genannt, mit dem ermöglicht wird, spezielle Code-Zweige für unterschiedliche Oracle-Versionen bereitzustellen. Ermöglicht wird dies über bedingte Kompilierung, die von Konstanten in diesem Package gesteuert wird. Ein solches Steuerpackage wird etwa folgendermaßen aussehen:

```
CREATE OR REPLACE PACKAGE my_app_variant AS
BEGIN
    use_compression CONSTANT BOOLEAN := TRUE;
    use_partitioning CONSTANT BOOLEAN := FALSE;
    -- further constants as needed ...
END my_app_variant;
```

Dieses Steuerpackage wird für die jeweiligen Kunden jeweils in einer passenden Variante bereitgestellt. Die Verwendung funktioniert dann etwa so:

```
CREATE OR REPLACE PACKAGE BODY my_app_batch AS
BEGIN
    PROCEDURE do_it
    AS
    BEGIN
        $IF my_app_variant.use_partitioning $THEN
            INSERT into TEMP-Table
```

```

        EXCHANGE PARTITION with target table
    $ELSE
        INSERT into target table directly
    $END
END;
END my_app_batch;

```

Durch eine derartige Methodik ist sichergestellt, dass nicht versehentlich eine Option wie Partitioning eingesetzt wird, die dann teuer nachlizenzieren muss. Außerdem bietet sich der Vorteil, dass für alle nicht varianten-spezifischen Code-Teile wirklich derselbe Code eingesetzt wird, womit für diese Funktionalität ein Auseinanderlaufen grundsätzlich ausgeschlossen ist.

Etwas komplizierter wird es mit SQL-Skripten zur Installation von Tabellen etc., die mit einem Daten-Modellierungs-Tool generiert werden. Dafür bietet sich an, mit diesem Tool grundsätzlich die komplexeste Variante zu generieren, also mit Partitionen, Daten-Komprimierung etc. Für die erforderlichen „Vereinfachungen“ – z.B. Entfernen der Partitionierung – werden dann Skripte bereitgestellt, die dies für die erste und auch alle nachfolgenden Versionen durchführen. Sinnvollerweise wird die Ausführung dieser Skripte gleich in den jeweiligen kundenspezifischen Build-Prozess integriert.

Ein einfacher Switch zwischen den verschiedenen Varianten ist schon allein deshalb nicht möglich, weil die verwendete Edition nicht einfach geändert werden kann und weil die Tabellen nicht einfach von partitioniert auf unpartitioniert und vice versa geändert werden können. Die Änderung der Oracle Edition erfolgt durch Einspielen eines Backups in eine Datenbank der gewünschten Edition. Hiefür kann beispielsweise Transportable Tablespaces genutzt werden. Die Umwandlung von partitionierten in unpartitionierte Tabellen erfolgt durch Anwendung der DBMS-Re-Definition Methodik, wobei diese Methodik unabhängig von der Richtung der Umwandlung immer in der Enterprise-Edition erfolgen muss, weil nur dort Partitionierung zur Verfügung steht. Bei der Umwandlung von einer EE mit Partitioning in Richtung SE bietet sich als weitere Möglichkeit an, ein identisch konfiguriertes Duplikat der Applikation zu installieren und alle Tabellen der Applikation zu leeren. Dann wird durch Einspielen nur der Dateninhalte eines Backups alles automatisch unkomprimiert in die nicht partitionierten Tabellen geschrieben.

Möglichkeit B: Einsatz der Embedded Oracle Software Licence

Bei der Bereitstellung der Applikation mit einer Oracle Embedded Software Licence für die Kunden ist die Vorgehensweise eine komplett andere. Hier gibt es nur eine einzige Variante der Applikation mit Nutzung der Enterprise Edition und allen erforderlichen Optionen. Der Kunde erhält die erforderliche Oracle Lizenz direkt als Teil der Applikation und darf die für die Applikation eingesetzte Oracle-Datenbank nicht anderweitig nutzen. Dies muss auch sichergestellt werden, indem der Kunde keine Kenntnis von irgendwelchen Datenbank-Passwörtern erhält.

Daraus ergeben sich einige Einschränkungen für die Applikation, insbesondere für die Installation und Upgrades auf neuere Versionen für den Datenbankteil sowie natürlich sämtliche datenbankbezogenen administrativen Tätigkeiten, zu denen natürlich auch die Installation der Datenbank und deren Versions-Upgrades und andere übliche DBA-Aufgaben gehören.

Oracle schreibt vor, dass für die Installation der Datenbank der Oracle Universal Installer nicht sichtbar sein darf. Dies lässt sich mit Responsefiles geeignet handhaben. Die Installation der Oracle

Software und der Datenbank erscheint dabei als Teil der Applikations-Installation. Da es in heutiger Zeit kaum noch üblich ist, eine Datenbank von Grund auf neu mit „CREATE DATABASE“ zu erzeugen, sondern auch von der Oracle Installation selbst meist eine als Backup bereitgestellte Datenbank verwendet wird, kann man hier auch noch einen Schritt weitergehen. Man installiert den Datenbankteil der Applikation bereits im Hause in eine Datenbank und stellt die ggf mit „rman convert“ konvertierten Datefiles oder ein rman Backup für die Auslieferung bereit. Da mit „rman convert“ keine Endianess-Konvertierung des SYSTEM-Tablespaces durchgeführt werden kann, benötigt man mindestens eine Datenbank-Quelle mit Big-Endian sowie eine mit Little-Endian, soweit man entsprechende Plattformen beim Kunden unterstützen will. Rein kundespezifische Anpassungen des Datenbankteils der Applikation wie spezielle Steuerdaten etc. werden dann nach der Grundinstallation der kompletten Datenbank entweder mit der Applikation selbst oder mit einem geeigneten Administrationstool durchgeführt.

Da eine Datenbank aber auch administrative Aufgaben bereithält, die nicht mit einem Applikationsuser durchgeführt werden können, sondern einen DBA-Account benötigen, ist es in den meisten Fällen sinnvoll, dafür ein separates Tool bereitzustellen. Natürlich wird man zunächst versuchen, die administrativen Aufwände weitestgehend durch Features von Oracle – Self Managing Database, z. B. Oracle Managed Files – zu minimieren, aber zu 100 % wird das wohl kaum gelingen. Um hier weitgehend flexibel zu bleiben, empfiehlt sich ein Mechanismus, der es erlaubt, beliebige SQL-Statements als Files für den Kunden zur Ausführung bereitzustellen. Natürlich darf der Kunde solche Files nicht in modifizierter Form und auch keine selbst erstellten SQL-Files ausführen können. Dies wird erreicht, indem eine passende Signaturdatei vorhanden sein muss, die der Kunde nicht selbst erstellen kann. Das Tool selbst benötigt Zugriff auf einen DBA-Account, für dessen Erstellung idealerweise ein Script verwendet wird, um immer aller erforderlichen Rechte zu haben. Abgesichert ist dieser Zugriff durch eine verschlüsselte Hinterlegung des Passwords. Bei der Auslieferung einer Applikation mit Oracle Embedded Software Licence ist dieser Account natürlich auch schon in der vorinstallierten Datenbank so vorhanden. Da dieses Tool von seinen DBA-Aufgaben her sowieso schon weitgehende Rechte benötigt, wird es auch zum Upgrade des Datenbankteils der Applikation auf nachfolgende Versionen eingesetzt, wobei noch zusätzlich eine geeignete Steuerung integriert wird, die dafür sorgt, dass ausgehend von der vorgefundenen Version die verschiedenen Versionen bis zum aktuellen Stand der Auslieferung in der korrekten Reihenfolge eingespielt werden.

Eine Auslieferung für eine Neu-Installation mit einer Oracle Embedded Software Licence enthält also die eigentliche Applikation, ein Administrations-Tool für die Datenbank, die passende Oracle Datenbank-Software für die Zielplattform, die in der Endianess passenden und mit „rman convert“ an die Zielplattform angepassten Datefiles der Quelldatenbank, und natürlich die dafür erforderliche Installationssoftware in Form von Scripten oder Programmen. Die Installationssoftware ist dafür zuständig, ggf. User und Gruppen auf OS-Ebene anzulegen und die Ownership und Rechte von Dateien und Directories zu ändern. Sie muss deshalb mit einem hochprivilegierten User ausgeführt werden – root unter Unix, und als Mitglied der Administratorengruppe unter Windows. Unter Unix werden dann alle Aktionen, die weniger Rechte benötigen, unter dem neu erzeugten User für die Oracle-Datenbank ausgeführt.

Eine weitere Aufgabe der Installationssoftware ist es, die mitgelieferten Konfigurationsdateien – dazu gehören das Responsefile des Oracle Universal Installers, eine init.ora Datei, eine listener.ora Datei usw. – an die Zielumgebung anzupassen und an die richtige Position im Dateibaum zu kopieren. Die Anpassung erfolgt dabei durch geeigneten Textersatz, wobei zuvor festgelegte spezielle Zeichenketten durch die aktuellen Werte von bestimmten Variablen wie bzw. ORACLE_HOME ersetzt werden.

Nicht zuletzt ist die Installationssoftware auch dafür zuständig, sicherzustellen, dass Datenbank und Listener nach dem Booten der Maschine automatisch hochfahren und vor dem Herunterfahren der

Maschine zuerst heruntergefahren werden. Dabei werden für Unix und Windows jeweils die platform-spezifischen Mechanismen eingesetzt, also Windows Services bzw. der rc-Mechanismus unter Unix Derivaten. Auch eine automatische tägliche rman Backup wird an dieser Stelle bereits eingerichtet, das aus der Datenbank heraus scheduled wird und damit unabhängig von der jeweiligen Umgebung ist. In der Datenbank definierte Directories der Applikation sowie einige Oracle Standard Directories werden automatisch auf OS-Ebene in der zu installierenden Maschine angelegt und mit geeigneter Ownership, Groupmembership und geeigneten Rechten versehen, insbesondere werden spezielle neu erzeugte Applikations-Gruppen als primäre Gruppe dieser Directories verwendet, damit zu der jeweiligen Gruppe gehörige Applikationsuser auf OS-Ebene darauf Zugriff haben.

Ein möglicher späterer Übergang von einer Oracle Embedded Software Licence auf eine normale Oracle Licence ist denkbar einfach: Es werden einfach die Zugriffsbeschränkungen aufgehoben, so dass der Kunde unbeschränkten Zugang zur Applikationsdatenbank hat. Sonst kann alles in der Datenbank weiterlaufen wie bisher. Der umgekehrte Weg ist etwas aufwändiger, weil hier auch sichergestellt werden muss, dass die Zugriffsbeschränkungen greifen. Eventuell sind auch Optimierungen zur Verringerung des Administrationsaufwandes notwendig.

Gegenüberstellung der Vor- und Nachteile der beiden Methoden

Die beiden Methoden zur Lizenzkostenreduzierung beim Kunden sind vom Konzept her völlig unterschiedlich und benötigen deshalb auch an unterschiedlichen Stellen Mehraufwände in der Entwicklung.

Bei Methode A gibt es einen einmalig zu entwickelnden Anteil an Scripten für die „Vereinfachung“ der aus dem Datenmodell generierten Scripte zur Installation desselben. Diese Scripte zur Entfernung von Partitionierungs-, Compress-, Parallel-Klauseln usw. sind so allgemein gehalten, dass sie im Regelfall für zukünftige Versionen der Applikation und meist sogar für zukünftige Versionen der Datenbank unverändert eingesetzt werden können, solange nicht völlig neue Datenbank-Features in der Applikation eingesetzt werden, die auch auf die Tabellendefinition Einfluß haben.

Anders sieht es hier bei der Weiterentwicklung von Batch-Verarbeitungen aus. Hier müssen beim Einsatz von Oracle High-End-Features zur Erweiterung Abstufungs-Varianten entwickelt werden, und zwar für die gesamte Lebensdauer der Applikation. Diese Mehraufwände lassen sich durch die Zusammenfassung der Varianten zu einer einzigen Implementierung zwar reduzieren, aber nicht komplett vermeiden. Desweiteren sind hier erhöhte Testaufwände notwendig, weil die Tests für jede einzelne Variante durchgeführt werden müssen, natürlich ebenfalls für die gesamte Lebensdauer der Applikation.

Desweiteren ist zu erwarten, dass bei Datenbank-Änderungen für Applikations-Upgrades unterschiedliche SQL-Statements für die verschiedenen Implementierungsvarianten notwendig werden, die über eine geeignete Steuerung aufgerufen werden müssen. Die Upgrades sind deshalb ebenfalls über die gesamte Lebensdauer der Applikation aufwändiger zu implementieren und zu testen.

Bei Methode B hingegen gibt es nur eine einzige Variante des Datenbankteils der Applikation, so dass über die gesamte Lebensdauer der Applikation keine zusätzlichen Aufwände für die Weiterentwicklung zu einer neuen Version und deren Tests entstehen. Dies gilt ebenso für die Erstellung der Upgrade-Scripte für die neuen Versionen.

Andererseits muss ein Installationsmechanismus für die Komplettinstallation einer Applikation mit einer Oracle Embedded Software Licence implementiert werden, wobei hier spätere Änderungen nur dann zu erwarten sind, wenn Oracle Änderungen am Universal Installer oder am grundlegenden Design der Datenbank vornimmt. Für die Erstellung eines Datenbank-Admin-Tools sind im wesentlichen auch nur einmalige Aufwände zu erwarten, insbesondere wenn man die Ausführung von beliebigen SQL-Dateien implementiert hat. Eine darin integrierte Steuerung für die Upgrade-Vorgänge ist ebenfalls nur einmalig zu entwickeln.

Es ist also so, das bei Verwendung der Oracle Embedded Software Licence die Mehraufwände am Anfang entstehen, während beim abgestuften Downsizing nur ein Teil der Aufwände am Anfang entsteht, ein weiterer Teil aber während der gesamten Lebenszeit der Applikation. Der Umfang der Mehraufwänden beim abgestuften Downsizing hängt auch davon ab, wie groß die Breite der abzudeckenden Varianten ist. Zusammen mit der erwarteten Lebensdauer der Applikation läßt sich daraus dann für einen gegebenen Fall ableiten, ob die Methode A oder die Methode B besser geeignet ist. Berücksichtigt werden sollte dabei aber auch, dass wegen des umfangreicheren Codes beim abgestuften Downsizing mehr Fehler und deshalb auch mehr Fehlerbehebungsaufwände zu erwarten sind.

Kontaktadresse:

Dr. Kurt Franke

Cellent Finance Solutions AG

Calwer Straße 33

D-70173 Stuttgart

Telefon: +49 (0) 711-222992-676

Fax: +49 (0) 711-222992-899

E-Mail Kurt.Franke@cellent-fs.de

Internet: www.cellent-fs.de