

Oracle auf Mac

Wie läuft das?

Wie kann Oracle auf Mac installiert werden? Wie kann die Performance verglichen werden? Funktioniert die Oracle Installation als Server und als Client? Für wen ist es interessant, Oracle auf MAC einzusetzen?

Installation Oracle auf Mac

Die Oracle Version 10 Release 1 (10.1.0.3) ist seit längerem für Apple Mac OS X Tiger (10.4.1) erhältlich. In diesem Artikel erhalten Sie einige wertvolle Tipps zur Installation.

Bevor man überhaupt mit der Installation der Datenbank beginnen kann, müssen die XCODE Tools Version 2.1 von Apple installiert sein. Diese Tools können von der Homepage von Apple (<http://developer.apple.com/macosx>) heruntergeladen werden. Anschliessend kann mit der normalen Datenbankinstallation begonnen werden. Es müssen einige manuelle Schritte durchgeführt werden wie das Anlegen diverser Gruppen und eines Users Oracle. Unter diesem User wird dann die entsprechende Datenbankinstallation durchgeführt. Weiter muss noch eine Directory Struktur aufgebaut sowie verschiedene Pfade wie ORACLE_HOME und ORACLE_SID gesetzt werden. Eine genaue Beschreibung finden Sie in der Dokumentation von Oracle „Oracle Database Quick Installation Guide 10g Release 1 (10.1.0.3) for Apple Mac OS X“.

Da Oracle noch den Compiler gcc3.3 verwendet und Apple Mac OS X Tiger bereits den Compiler gcc 4 benutzt, muss dieser zurückgesetzt werden. Dies kann mit folgendem Befehl überprüft werden:

```
localhost:/usr oracle$ gcc -v
Reading specs from /usr/libexec/gcc/darwin/ppc/3.3/specs
Thread model: posix
gcc version 3.3 20030304 (Apple Computer, Inc. build 1809)
```

Das setzen des gcc3.3 Compilers funktioniert dann wie folgt:

```
sudo nicl . -append /groups/admin users oracle
sudo gcc_select 3.3
```

Anschliessend müssen die Libraries von Oracle nochmals relinkt werden. Dieses Thema und weitere Informationen finden sie im Dokument „Journal of a Journeyman“ unter folgendem Link: <http://journeymanjournal.blogspot.com/2005/06/set-up-oracle-10g-on-tiger.html>.

Nachdem die oben genannten Schritte durchgeführt wurden, kann nun mit ./runInstaller der Oracle Universal Installer gestartet werden. Wir haben die Enterprise Edition ausgewählt. Die Installation der Software läuft problemlos bis zum Erstellen der Oracle Datenbank. Das Programm „Database Configuration Assistant“ (dbca) kann die gewünschte Datenbank nicht erstellen. Der letzte Schritt kann nicht durchgeführt werden. Beim Startup der Datenbank erscheint folgende Fehlermeldung:

```
SQL> startup
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file
'/Volumes/u01/app/oracle/product/
10.1.1.0/db_1/dbs/initORCL.ora'
```

Wir nahmen das Default init.ora File und erstellte ein initSID.ora File, welches beim Startup der Datenbank benutzt wird. Da einige Defaulteinträge nicht korrekt waren, musste dieses Parameter File angepasst werden (shared_pool_size erhöhen, Pfade von cdump, bdump und updump anpassen, db_recovery_file anpassen). Nach diesen Korrekturen konnte die Datenbank manuell erstellt werden. Das Datenbankdictionary musste anschliessend ebenfalls manuell erstellt werden.

```
SQL> startup
ORACLE instance started.

Total System Global Area  167772160 bytes
Fixed Size                  778284 bytes
Variable Size               57679828 bytes
Database Buffers           109051904 bytes
Redo Buffers                 262144 bytes
Database mounted.
Database opened.
```

Damit der Datenbankzugriff von einem anderen Datenbank-Client erfolgreich ist, muss der Datenbank Listener gestartet sein. Mit dem Befehl lsnrctl start wird dieser gestartet.

Dabei erhielten wir folgende Fehlermeldung:

```
192:/etc oracle$ lsnrctl start

LSNRCTL for MacOS X Server: Version 10.1.0.3.0 - Production on 23-AUG-
2005
00:08:20

Copyright (c) 1991, 2004, Oracle. All rights reserved.

Starting /Volumes/u01/app/oracle/product/10.1.0/db_1/bin/tnslsnr:
please
wait...

TNSLSNR for MacOS X Server: Version 10.1.0.3.0 - Production
System parameter file is /Volumes/u01/app/oracle/product/10.1.0/db_1/
network/admin/listener.ora
Log messages written to /Volumes/u01/app/oracle/product/10.1.0/db_1/
network/log/listener.log
Error listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)
(PORT=1521)))
TNS-12545: Connect failed because target host or object does not exist
TNS-12560: TNS:protocol adapter error
TNS-00515: Connect failed because target host or object does not
exist
MacOS X Server Error: 22: Invalid argument

Listener failed to start. See the error message(s) above...
```

Der Fehler konnte behoben werden, indem die IP-Adresse des iBook im File /etc/hosts eingetragen wurde. Die richtigen Definitionen müssen zudem im tnsnames.ora und listener.ora gespeichert sein. Diese beiden Files befinden sich im ORACLE_HOME/network/admin Verzeichnis.

Ein paar Test

Import

Damit die Performance der MAC-Installation mit einer PC-Installation verglichen werden kann, eignet sich das Schema SH mit der Tabelle Sales, die fast eine Million Datensätze enthält. Da wir die DB nicht mit dem Assistent installieren konnten, wurde das Schema SH auch nicht automatisch aufgesetzt. Zusätzlich mussten wir feststellen, dass die entsprechenden Files zum Aufsetzen der Beispiel Schemata leider fehlen. Daher mussten wir zuerst die Daten in das Schema SH importieren. Diese haben wir von der PC-Installation in ein dump-file exportiert. Dieses dump-file konnte dann ohne Probleme in die Oracle Datenbank auf dem MAC importiert werden.

Performance

Bei diesem Test geht es lediglich darum zu prüfen, ob die Performance auf MAC mit anderen Installationen vergleichbar ist. Dazu definierten wir auf dem SH-Schema eine View und eine Abfrage, die sicher mehrere Sekunden in Anspruch nehmen wird. Hier die View und Abfrage:

```
CREATE OR REPLACE VIEW sales_view AS
SELECT country_name country, prod_name product, calendar_year year,
       SUM(amount_sold) sales, COUNT(amount_sold) cnt,
       MAX(calendar_year) KEEP (DENSE_RANK FIRST ORDER BY SUM(amount_sold)
DESC)
OVER (PARTITION BY country_name, prod_name) best_year,
       MAX(calendar_year)
       KEEP (DENSE_RANK LAST ORDER BY SUM(amount_sold) DESC)
OVER (PARTITION BY country_name, prod_name) worst_year
FROM sh.sales, sh.times, sh.customers, sh.countries, sh.products
WHERE sales.time_id = times.time_id
      AND sales.prod_id = products.prod_id
      AND sales.cust_id = customers.cust_id
      AND customers.country_id=countries.country_id
GROUP BY country_name, prod_name, calendar_year;

SELECT year, sum(sales)
FROM sales_view
GROUP BY year;
```

Codebeispiel für Performancetest

Bei einem Vergleich ist es sicher auch wichtig zu wissen, welche Hardware im Einsatz ist. Hier die wichtigsten Angaben:

	iBook	Notebook
CPU	G4 1.2 Prozessor	Intel 1.6 GHz
RAM	512 MB	1 GB
Betriebssystem	Apple Mac OS X Tiger	Windows XP Professional

Die Abfrage läuft auf dem iBook etwa 10% langsamer als auf dem Notebook. Dabei ist aber zu berücksichtigen, dass das Notebook doppelt soviel RAM hat, wie das iBook. So betrachtet kann man sagen, dass die Performance vergleichbar mit anderen Installationen ist, aber nicht schneller.

Client und Server

Oracle unterstützt die Client/Server-Architektur. Sowohl der Server als auch der Client kann auf einer beliebigen Maschine laufen. Beim Test zu Client Server geht es darum zu testen, ob von einem anderen Client auf die Server-Installation auf dem iBook zugegriffen werden kann und ob das iBook auch als Client eingesetzt werden kann.

Betrachten wir zuerst das iBook als Client. Damit auf eine Datenbank eines anderen Systems zugegriffen werden kann, braucht es einen entsprechenden Eintrag im tnsnames.ora. Im folgenden Beispiel wird auf eine Oracle Instanz auf einem Notebook zugegriffen.

```
ORA10G.localhost.ch =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.0.7)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = ora10g)  
    )  
  )
```

Codebeispiel für Eintrag im tnsnames.ora

Beim Zugriff auf die Instanz des iBooks braucht es einen entsprechenden Eintrag im tnsnames.ora des Clients.

Wie zu erwarten war, haben beide Arten von Zugriff problemlos funktioniert. So kann Oracle auf Mac sowohl als Client als auch als Server eingesetzt werden.

Dies hat verschiedene Vorteile. In einer heterogenen Systemlandschaft kann also auch problemlos von einem Mac aus auf eine beliebige Oracle Datenbank zugegriffen werden. Weiter kann auch eine Firma, die ausschliesslich Mac im Einsatz hat, Oracle als Datenbank einsetzen.

Schlussfolgerung

Wieso soll Oracle auf Apple installiert werden? Hier zuerst die Antwort von Apple:

"Die Plattform von Apple bietet eine stabile Infrastruktur für die auf offenen Standards basierende Software Oracle Fusion Middleware 10g und Oracle Datenbank 10g. Viele Kunden setzen bereits auf die Apple Plattform und möchten dies auch nicht ändern -

Oracle unterstützt diese Entscheidung. Eine Apple Infrastruktur ist oft einfacher zu warten als eine Windows oder Linux Infrastruktur."

Von grossem Vorteil ist sicher, dass Oracle auf unterschiedlichen Plattformen läuft. So kann die vorhandene Infrastruktur besser genutzt werden. Bei einer Neubeschaffung ist die Auswahl an Betriebssystemen grösser.

Apple hat auch den grossen Vorteil, dass die Systeme tatsächlich für ihre einfache Wartung bekannt sind. So sind auch Installationen in der Regel unproblematisch. Hier hat Oracle einen anderen Ruf: Installationen können aufwändig und kompliziert sein. Leider mussten wir feststellen, dass die Installation von Oracle auf Apple noch lange nicht den Apple-Ruf erfüllt. Im Gegenteil, die Installation ist nicht nur recht aufwändig, sondern verlangt Fachwissen sowohl im Bereich Unix als auch im Bereich Oracle. Sollten Sie hier Unterstützung brauchen, so können wir gerne helfen. Ist Oracle einmal installiert, so läuft es problemlos und mit einer akzeptablen Performance.

Andrea Haymoz und Andrea Kennel, InfoPunkt Kennel GmbH

info@infokennel.ch

www.infokennel.ch