Migration von RMS nach ORACLE

http://138.245.152.20/EDV/DECUS2003_WEB.HTML

E.W.Raschner, Max von Pettenkofer-Institut, LMU München, raschner@m3401.mpk.med.uni-muenchen.de

Voraussetzungen, Umgebung
Grundlegende Überlegungen
Aufbau der ORACLE-Datenbank
Programm-Schnittstelle (OCI)
Probleme
Ergebnisse
Ausblick

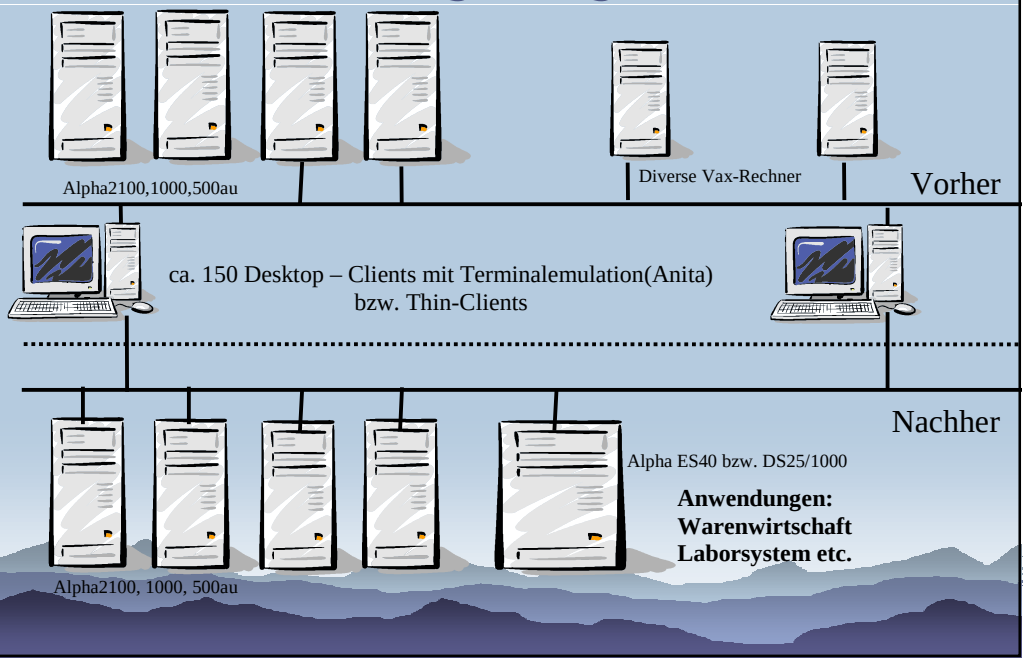
Migration von RMS nach ORACLE Einführung

- Gründe für die Migration:
- Von proprietären Lösungen zu Standards (eigener DB-Server, RMS, VMS)
- Plattformunabhängigkeit
- Öffnung für Tools anderer SW-Lieferanten (ODBC, SAS, Office, BI-Produkte)
- Ausgangspunkt für neue Anwendungen
- Kenntnisse der Mitarbeiter

Migration von RMS nach ORACLE Voraussetzungen

- Alle Anwendungen haben eine einheitliche, relativ einfache Schnittstelle zur Datenbank
- Diese ist Ansatzpunkt für die Umstellung der Datenbankschnittstelle von RMS-basierende- auf Oracle-basierende Zugriffe
- Programmierung mit OCI (Oracle Call Interface)
- Anwendungen bleiben zunächst auf VMS
- VMS-ALPHA-Cluster (Oracle-Server auf ES40 bzw. DS25/1000 mit 2 Prozessoren und RAID)

Migration von RMS nach ORACLE Umgebung



Migration von RMS nach ORACLE Grundlegende Überlegungen

- Oracle ist heute performant genug, um keinen Engpass gegenüber RMS zu bedeuten, allerdings
- ist ein 'starker' Rechner mit mind. 2 Prozessoren, und ausreichend Hauptspeicher
- Schneller Plattenzugriff und sinnvolle Verteilung der Oracle-Dateien (nicht nur Tables) nötig
- Die Schnittstelle zwischen Anwendungen und Datenbank muss eindeutig, relativ einfach sein, und von allen Programmen eingehalten werden

Migration von RMS nach ORACLE Aufbau der OracleDB

- Ausgehend von der Manager-View der RMS-DB werden DCL-und SQL Prozeduren erzeugt, die folgende Aufgaben erledigen:
- Erzeugen aller nötigen Tabellen und Indizes
- Export/Import Scripte
- Einrichten des System-Tablespace, Redo-Log-Dateien, Rollback-Segmente mit entsprechender Verteilung auf möglichst viele Platten
- Synonyme, Anwenderprofile

Migration von RMS nach ORACLE Programm-Schnittstelle (OCI)

- Die Schnittstelle zur RMS-DB ist ein „logischer String“ in reverse polish Darstellung der Art operand operand operator aufgelöst auf einzelne Relationen
- Wird mittels eines Stack-basierten Algorithmus (endlicher Automat) in SQLStatements umgesetzt
- Programmierung in C unter Verwendung der Oracle Call Schnittstelle (OCI) als Standard-Interface zur Oracle Datenbank

Migration von RMS nach ORACLE Probleme

- Implizite Sortierung der RMS-basierten DB
- Behandlung von NULL-Werten
- Gross/Kleinschreibung bei Such-Befehlen
- Datumstypen
- Funktionsumfang Oracle versus RMS
- Aufwand für Datenbank-Aufbau etwa gleich gross wie der Programmieraufwand
- OCI ist relativ gut dokumentiert, aber kaum jemand hat Erfahrung

Migration von RMS nach ORACLE Ergebnisse 1 / 2

- Performance ausreichend, aber deutlich stärkerer Rechner für Oracle als für RMS
- Grösse der Tables und Indizes nahezu gleich
- Umgestellte Anwendungen laufen praktisch unverändert weiter, neue Oracle-basierende
- Anwendungen wurden bereits hinzu gefügt
- OCI-Interface ist für diese Art der Umstellung gut geeignet

Migration von RMS nach ORACLE Ergebnisse 2/2

- Aufwand Programmierung geringer als angenommen, trotz einiger unerwarteter Probleme wie NULL-Wert Behandlung
- Aufwand für Datenbank-Aufbau und Verwaltung sehr hoch im Vergleich zu RMS
- Erstaunliche Performance des Oracle Loaders Import mit Loader und Index-Erzeugung sind >Faktor 10 schneller als das Auslesen der Daten aus der RMS-Datenbank
- Fragen: raschner@m3401.mpk.med.uni-muenchen.de, astrid.kals@chello.at

Migration von RMS nach ORACLE Ausblick

- Einführung von Bind-Variablen bei Erzeugung der SQL-Abfragen
- Anstelle der relativ einfachen SQLStatements aufgelöst nach Relationen „gejointes“ SQL
- Optimierung mit Hilfe von Oracle Tools
- Erweiterung der bestehenden Anwendung und gleichzeitiges Hinzufügen neuer Anwendungen, damit wird Oracle zur Plattform der Integration mit aller eingesetzten Produkte/Programme

Migration von RMS nach ORACLE

Anhang:Manager-View der RMS-DB, Ausgang für automatisierten Aufbau der ORACLE-DB

Relation	Attr.Nr	Name	Typ	Nr./Bez.	Attr.Länge	Key	KeyLänge
AUFTRAG	1	ABTEILG	1	ASCII	15	0	0
AUFTRAG	2	USERNAM	1	ASCII	15	0	0
AUFTRAG	3	DATUM	12	DAT2	8	0	0
AUFTRAG	4	AUFTR_NR	1	ASCII	16	0	0
AUFTRAG	5	PAT_IDET	1	ASCII	16	2	16
AUFTRAG	6	AUFN_NR	1	ASCII	13	0	0
AUFTRAG	7	RUECKR	1	ASCII	1	0	0
AUFTRAG	8	KE_IDENT	1	ASCII	16	0	0
AUFTRAG	9	EFWI	1	ASCII	1	0	0
AUFTRAG	10	STUDIE	1	ASCII	13	0	0
AUFTRAG	11	AUF_UKL	1	ASCII	12	0	0
AUFTRAG	12	ZEICHEN	1	ASCII	25	0	0
AUFTRAG	13	AMB/STAT	1	ASCII	1	0	0
AUFTRAG	14	RISIKO	1	ASCII	10	4	6
AUFTRAG	15	KR_SEIT	1	ASCII	8	0	0
AUFTRAG	16	EX/IKT	1	ASCII	2	0	0
AUFTRAG	17	SSW	1	ASCII	2	0	0
AUFTRAG	18	E_IDENT	1	ASCII	16	0	0
AUFTRAG	19	A_IDENT	1	ASCII	16	1	16

Migration von RMS nach ORACLE
Anhang: reverse polish nach Sql

Logstring als Characterstring mit folgendem Aufbau:

```
64| Nr.Attr1 |65| Laenge des Attr1 | Wert Attribut1| Vglop |
64| Nr.Attr2 |65| Laenge des Attr2 | Wert Attribut2| Vglop | Logop |
```

vglop =, <, > usw. Logop AND,OR, usw.

Beispiel:

```
64 4 65 10 0304091010 = 64 5 65 16 8610111230123765 = AND
```

Auftr_Nr 0304091010 = Pat_Idet 8610111230123765 = AND

Resultat der Umsetzung:

```
Select from Auftrag where Auftr_nr='0304091010' and
                          Pat_Idet = 8610111230123765;
```

Migration von RMS nach ORACLE
Anhang: Größen und Zeiten

Tabelle	sekun. Index	Anzahl Tupel	Groesse RMS/ Export-Datei	Groesse ORACLE Data/Index	Export Zeit	Import Zeit	create IndexZeit
AUFTRAG	3	1.5Mio	823700/1712918	536576/ 167936	0:28.00	??	??
BEFTEXT	1	12.0Mio	2969300/2900416	1974272/ 761856	0:36.00	0:06.30	1:10.00
ERGEBNIS	1	6.0Mio	1091450/1382602	696320/ 778240	1:01.00	0:03.00	0:21.00
MATERIAL	2	2.4Mio	747750/804223	421888/ 405632	0:27.00	0:01.40	0:17.00
UNTSUCH	3	5.9Mio	2394700/1694848	1310720/1310848	1:09.00	0:04.30	1:13.00
RECHBUCH	0	8.3Mio	2932736/3060200	2506752/ 516096	0:35.00	??	--

jeweils 1 Primaer-Index, Datei Groessen in VMS Bloecken, Versuch 1 :
Verteilung Oracle-Daten 4 Platten, 1 Kontroller, Plattentyp RZ1ED-LS (10K rpm)

Tabelle	sekun. Index	Anzahl Tupel	Groesse RMS/ Export-Datei	Groesse ORACLE Data/Index	Export Zeit	Import Zeit	create IndexZeit
AUFTRAG	3	1.5Mio	823700/1712918	536576/ 167936	0:28.00	0:01.20	0:01.33
BEFTEXT	1	12.0Mio	2969300/2900416	1974272/ 761856	0:36.00	0:03.51	0:07.18
ERGEBNIS	1	6.0Mio	1091450/1382602	696320/ 778240	1:01.00	0:02.16	0:03.02
MATERIAL	2	2.4Mio	747750/804223	421888/ 405632	0:27.00	0:01.15	0:02.41
UNTSUCH	3	5.9Mio	2394700/1694848	1310720/1310848	1:09.00	0:03.36	0:09.08
RECHBUCH	0	8.3Mio	2932736/3060200	2506752/ 516096	0:35.00	0:04.47	--

jeweils 1 primaer-Index, Datei Groessen in VMS Bloecken, Versuch 2 :
Plattentyp Oracle-Data 5 Platten 2x Compaq (15K rpm) fuer Daten und Index
3 Platten typ RZ1ED-LS (10K rpm), 2 Kontroller

Migration von RMS nach ORACLE Anhang: Oracle9i und VMS

OUI Oracle Universal Installer, basiert auf Java, ODS5-Disk nötig (weitgehend identisch mit Struktur auf UNIX/LINUX =/= 8i)

DBCA DataBaseCreationAssistant für Erzeugen/Verwalten von Dbs

Scripts für DB-Erzeugen können erstellt werden

Installationsskripte produzieren zum Teil schwer nachvollziehbare VMS-Probleme (set def, set display)

ältere Alpha-Prozessoren nicht unterstützt