

Datenbanken Konsolidiert in der Cloud

Heiko Eitner
Landesbetrieb Daten und Information Rheinland-Pfalz (LDI)
Mainz

Marco Mischke
Robotron Datenbank-Software GmbH
Dresden

Schlüsselworte:

Cloud, Database, Grid Infrastructure, RAC, Service, Serverpool

Einleitung:

Cloud-Computing ist heute eines der spannendsten Themen in der IT-Branche. Sie umfasst eine elastische Skalierbarkeit, schnelle Servicebereitstellung und eine höhere Effizienz durch gemeinsame Nutzung von IT-Ressourcen. Aufgrund des Anstiegens der Aufgaben der IT innerhalb von Landesforsten RLP, ist durch den Landesbetrieb (LDI) ein Projekt zur Prüfung der Möglichkeiten einer Konsolidierung der Datenbanken ins Leben gerufen worden. Ziel dieses Projektes war eine Vereinfachung der Administration, die Konsolidierung der Hardware und die Möglichkeit, die Infrastruktur schnell an dynamische Anforderungen und Geschäftsprozesse anpassen zu können. Aus diesem Grund wurde die Benutzung von unterschiedlichen Serverpools oder Container Datenbanken interessant. Dieses Konzept wurde in Zusammenarbeit mit der Firma Robotron erfolgreich erarbeitet.

Ziel dieser Machbarkeitsstudie war die zahlreich bestehenden Datenbanken auf verschiedenen Servern in einem bestehenden 5-Knoten-Cluster zusammenzuführen um physische Ressourcen optimal auszunutzen und finanzielle Aufwände möglichst zu minimieren. Es wurden dabei Themen wie Verfügbarkeit und Lastpriorisierung betrachtet ebenso wie eine möglichst große Flexibilität und Administrierbarkeit. Dies wurde durch den Einsatz von Server Pools und Child Server Pools und die Definition von entsprechenden Services getestet. Der Vortrag beschreibt den Fortschritt des Projektes von der Idee über die Testphase hin bis zur geplanten Produktivsetzung und beleuchtet dabei insbesondere die aufgetretenen Probleme und deren Lösungsansätze.

Idee:

Bei der Lösungsfindung kristallisierten sind drei mögliche Umsetzungsvarianten heraus.

1. Benutzung von Parent- und Child Serverpools
2. Nutzung der Option Multitenant (Container and Pluggable Database)
3. Benutzung policy managed RAC inkl. Serverpools

Dabei ist die Lösungsvariante der Benutzung von Parent- und Child Serverpools als die eleganteste Lösung angesehen worden. Serverpools bieten in einem Oracle Cluster die Möglichkeit Server zu gruppieren. Es sind dabei zwei Ebenen vorgesehen: Top Level Serverpools (Parent) bilden die oberste Ebene, ein physikalischer Server kann nur genau einem Top Level Serverpool zugeordnet werden. Die zweite Ebene bilden die Child Serverpools, diese können mehrere Top Level Serverpools als Parent

zugeordnet bekommen und laufen dann entsprechend auf allen Servern die allen zugeordneten Top Level Serverpools angehören. Services können im Cluster jeweils genau einem Serverpool zugeordnet werden. Die größte Flexibilität würde daher die Benutzung von Child Serverpools für Datenbanken bieten. Dieses Herangehen wurde im Folgenden geprüft. Der Lösungsansatz Container and Pluggable Database ergab, dass dieser nicht den Anforderungen der Kostenminimierung gerecht wird. Dieser Ansatz wurde nicht weiter verfolgt.

Die Lösung:

Es werden so viele Parent Serverpools eingerichtet wie sich Server im Cluster befinden. Jeder dieser Serverpools darf maximal einen Server beinhalten. Die Serverpools werden mit absteigenden Prioritäten versehen.

Für jede Datenbank wird ein Child Serverpool erzeugt, dem als Parent die Serverpools entsprechend der Priorität und Last der Datenbank zugeordnet werden.

Dies ermöglicht beim Hochfahren des Clusters das Starten aller Datenbanken in der gewünschten Reihenfolge ihrer Priorität und das schrittweise Zuschalten von weiteren Ressourcen wenn der Cluster sich erweitert.

Im Folgenden ist die Aufteilung der Datenbanken auf die Serverpools verdeutlicht:

Childpool	Server Serverpool	1 S1	2 S2	3 S3	4 S4	5 S5
Database 1				X		
Database 2		X				X
Database 3			X			
Database 4				X		X
Database 5			X			
Database 6					X	
Database 7					X	
Database 8		X	X	X	X	X
Summe		2	3	3	3	3

Tabelle 1: Aufteilung der Child Serverpools

Zuerst startet der Top Level Serverpool „S1“ dem die Child Serverpools für Database 2 und Database 8 angehören. Danach startet der Top Level Serverpool „S2“ mit den zugehörigen Child Serverpools Database 3, Database 5 und nochmals Database 8. Dies setzt sich entsprechend der Tabelle bis zum vollständigen Start aller Cluster Server fort.

Für die Verwendung mit policy-managed Datenbanken werden die Serverpools per „srvctl“ angelegt. Dies unterstützt aber die Angabe des Attributes „PARENT_POOLS“ nicht, dies muss per „crsctl“ separat gesetzt werden.

```

[oracle@server2 ~]$ srvctl add srvpool -g database1 -i 10 -l 1 -u 3
[oracle@server2 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=S1
ACTIVE_SERVERS=server2

NAME=S2
ACTIVE_SERVERS=server3

NAME=ora.database1
ACTIVE_SERVERS=server1

[oracle@server2 ~]$ crsctl modify serverpool ora.database1 -attr
"PARENT_POOLS=S1"
[oracle@server2 ~]$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=S1
ACTIVE_SERVERS=server1 server2

NAME=S2
ACTIVE_SERVERS=server3

NAME=ora.database1
ACTIVE_SERVERS=server1

```

Leider kann der Serverpool dann nicht mehr für Datenbanken ausgewählt werden. Er lässt sich auch per „srvctl“ nicht mehr entfernen. Dies ist bei Oracle unter SR 3-6789975161 und Bug 16369884 gemeldet.

```

[oracle@server2 ~]$ srvctl remove srvpool -g database1
PRKO-3161 : Server pool database1 is internally managed as part of
administrator-managed database configuration and therefore cannot be
removed directly via srvpool object.
[oracle@server2 ~]$ crsctl delete serverpool ora.database1

```

Ergebnis der Auswertung des Servicerequest ergab, dass dieser Lösungsansatz nicht von Oracle supported ist.

Alternative Lösung:

Es wurden drei Top Level Serverpools entsprechend drei unterschiedlichen Prioritäten eingerichtet, mit den Namen High, Medium, Low. Die Serverpools werden dadurch in Reihenfolge ihrer Priorität zuerst mit einem Server gestartet. Sind alle Serverpools mit einem Server aktiv, so werden weitere Server ebenfalls in Reihenfolge der Priorität den Serverpools zugeordnet bis diese ihre maximale Größe erreicht haben.

Daraus ergibt sich folgende Umsetzung:

Datenbank	Server Serverpool	1 High	2 Medium	3 Low	4 High	5 High	Service
Database 1			X				Uniform
Database 2		X			X	X	Singleton
Database 3		X			X	X	Singleton
Database 4			X				Uniform
Database 5			X				Uniform
Database 6				X			Uniform
Database 7				X			Uniform
Database 8		X	X	X	X	X	Uniform
Summe		3	4	3	1	1	

Tabelle 2: Größenverteilung der Serverpools

Die Startreihenfolge der Serverpools und somit der Datenbanken entspricht im Vorgehen wie oben beschrieben. Zusätzlich wird über die Definition der zur Datenbank gehörenden Service gesteuert, ob Instanzen auf allen Servern im Serverpool (uniform) oder nur auf einem Server (singleton) benutzt werden.

```
srvctl add srvpool -g high -i 10 -l 1 -u 3
srvctl add srvpool -g medium -i 8 -l 1 -u 2
srvctl add srvpool -g low -i 6 -l 1 -u 1
```

```
srvctl add database -d database8 -g high,medium,low
srvctl add database -d database2 -g high
srvctl add database -d database3 -g high
srvctl add database -d database1 -g medium
srvctl add database -d database4 -g medium
srvctl add database -d database5 -g medium
srvctl add database -d database6 -g low
srvctl add database -d database7 -g low
```

```
srvctl add service -d database8 -s database8s1 -g high -u uniform
srvctl add service -d database8 -s database8s2 -g medium -u uniform
srvctl add service -d database8 -s database8s3 -g low -u uniform
srvctl add service -d database2 -s database2s -g high -u singleton
srvctl add service -d database3 -s database3s -g high -u singleton
srvctl add service -d database1 -s databasels -g medium -u uniform
```

```

srvctl add service -d database4 -s database4s -g medium -u uniform
srvctl add service -d database5 -s database5s -g medium -u uniform
srvctl add service -d database6 -s database6s -g low -u uniform
srvctl add service -d database7 -s database7s -g low -u uniform

```

Um eine optimale Ausnutzung der Ressourcen zu erreichen, wurde für die database8 pro Serverpool ein Service (database8s1, database8s2, databases3) konfiguriert. Da alle Services der Database8 benutzt werden sollen, ist ein Alias per Oracle NET Description eingerichtet wurden.

```

Database8s=
(DESCRIPTION_LIST=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP) (HOST=scan_listener) (PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=database8s1.zebit.de))
  )
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP) (HOST=scan_listener) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=database8s2.zebit.de))
)
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP) (HOST=scan_listener) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=database8s3.zebit.de))
))

```

Alternativ kann man auch den Default Service (database8) benutzen, dieser sollte aber nur für administrative Zwecke verwendet werden.

Diese Lösung zeichnet sich durch folgende Merkmale aus:

- Elastizität: die Ressourcen können in kürzester Zeit umkonfiguriert werden
- Hochverfügbar und skalierbar: mit Hilfe der Server Pools ist ein automatisches Failover und ein Lastausgleich innerhalb der Cloud gewährleistet

Zusammenfassung

Cloud Computing ist nichts grundlegendes Neues, sondern fasst bereits in der Praxis eingesetzte technische Möglichkeiten zusammen. Bei der Untersuchung wurde schnell klar, dass der Begriff Cloud kein fest definiertes Konzept oder gar ein fertiges Produkt ist, sondern für unterschiedliche Anforderungen und Geschäftsprozesse jeweils eigene Lösungen gefunden werden müssen. Der erarbeitete Lösungsansatz reduziert die Anzahl der zu lizensierenden Server auf dann 5 Server. Dadurch wird sich eine erhebliche Einsparung in den Lizenzkosten ergeben.

Die Konsolidierung mit Hilfe von Pluggable Databases (CDB & PDB) ist die flexiblere Lösung, da hier auch die Möglichkeit des automatisierten Clonings besteht. Aufgrund des zwingenden Einsatzes von Version 12.1.0.1 und der zusätzlichen Lizenzkosten wurde diese Variante zwar für die Zukunft getestet aber nicht weiter verfolgt.

Im Zuge der Lösungsfindung hat sich gezeigt, dass die Komplexität der Administration und der Verwaltungsaufwand für die einzelnen Services deutlich abnehmen wird. Voraussetzung für eine erfolgreiche Implementierung in eine Produktivumgebung, ist es besonderes Augenmerk auf das Design der Serverpools und der Services zu legen.

Weiterhin ist der Landesbetrieb jetzt in der Lage, sehr effektiv und schnell auf sich ändernde Anforderungen von Landesforsten Rheinland-Pfalz zu reagieren.

Kontaktadresse:

Heiko Eitner

Landesbetrieb Daten und Information Rheinland-Pfalz (LDI)

Valencia Platz 6

D-55118 Mainz

Telefon: +49 (0) 6131-605 191

E-Mail: heiko.eitner@ldi.rlp.de

Internet: www.ldi.rlp.de

Marco Mischke

Robotron Datenbank-Software GmbH

Stuttgarter Str. 29

D-01189 Dresden

Telefon: +49 (0) 351- 258 59 28 84

E-Mail: marco.mischke@robotron.de

Internet: www.robotron.de