

Hochverfügbarkeit von Anwendungen mit Oracle Solaris Cluster

**Hartmut Streppel
Oracle Deutschland
Kirchheim-Heimstetten**

Schlüsselworte:

Hochverfügbarkeit, Cluster, Oracle Solaris Cluster, Anwendungsintegration

Einleitung

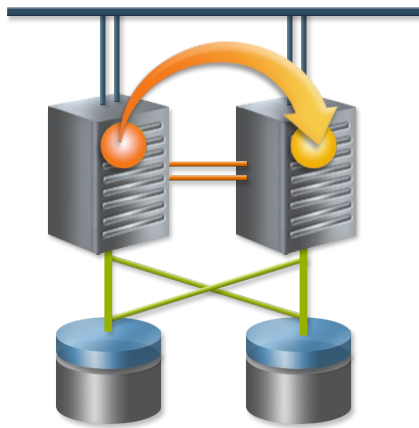
Die Verfügbarkeit eines Dienstes ist heutzutage eine Selbstverständlichkeit, seine Nichtverfügbarkeit eine Katastrophe. Ist ein Mobilfunknetz nicht verfügbar, wird dies am folgenden Tag in der Presse breitgetreten, kommt die S-Bahn zu spät, ist die Häme groß. Dass trotz dieser Katastrophen erheblicher Aufwand in die Hochverfügbarkeit solcher Systeme gesteckt wurde, ist meist nicht sichtbar. Häufig sind Clustersysteme im Einsatz, die die Verfügbarkeit der Dienste garantieren sollen.

Welches sind nun die Methoden, mit denen Anwendungen unter Oracle Solaris hoch-verfügbar gemacht werden können. Dazu steht das Produkt Oracle Solaris Cluster zur Verfügung, das zum Einen sogenannte Agenten zur Verfügung stellt, die Standard-Anwendungen in das Cluster integrieren, mit dessen Hilfe aber auch andere, nicht-Standard-Anwendungen hoch-verfügbar gemacht werden können.

Hochverfügbarkeit von Anwendungen mit Oracle Solaris Cluster

Eine der wichtigsten Technologien im Umfeld von Hochverfügbarkeit sind Cluster - mehrere gekoppelte Systeme, die sich gegenseitig und Anwendungen (Dienste) überwachen und in Fehlersituationen reagieren. Oracle Solaris Cluster, als Weiterentwicklung einer verteilten Betriebssystemtechnologie seit 1999 als Produkt (ursprünglich Sun Cluster) verfügbar, ist eines der verbreitetsten Cluster-Produkte am Markt. Seine wesentlichen Eigenschaften sind:

- Integration in den Solaris Betriebssystemkern
- enge Koppelung der Clusterknoten
- sicheres Quorum- und Membership-Konzept
- umfassendes Framework für die Anwendungsintegration
- große Anzahl vorgefertigter Agenten für Anwendungen
- Integration in Disaster Recovery Infrastrukturen mit der Oracle Solaris Cluster Geographic Edition



Diese typische Cluster-Konfiguration besitzt Redundanz auf fast allen Ebenen: redundante Cluster-Interconnects, redundante Anbindung an das „public net“, redundante Plattensubsysteme mit redundanten Pfaden und natürlich redundante Clusterknoten, auf denen eine Anwendung weiter betrieben werden kann, wenn ein Knoten ausgefallen ist.

Es ist extrem wichtig, Redundanzen auf allen Ebenen zu implementieren, d.h. die sogenannten „Single Points of Failures“ auch SPOF genannt, zu eliminieren. Bei der Definition solcher SPOFs kommt man schnell aus der Technik in andere Bereiche, da z.B. auch die Verfügbarkeit des Bedienpersonals kein SPOF sein darf.

Abb. 1: Eine typische Cluster Konfiguration

Anwendungsintegration mit dem Generic Data Service (GDS)

Ein wesentlicher, technischer Garant für den Erfolg eines Cluster-Frameworks ist die Verfügbarkeit einer einfachen Methode, Anwendungen sicher in das Cluster zu integrieren. D.h. eine Anwendung wird vom Cluster gestartet und gestoppt, überwacht, und beim Auftreten von Fehlern entweder neu gestartet oder auf einen anderen Clusterknoten migriert. Und das Ganze muss möglich sein, ohne die Anwendung in irgendeiner Art zu modifizieren.

Ein „Data Service“ ist in Oracle Solaris Cluster Terminologie eine Anwendung, die einen hoch-verfügbaren Dienst zur Verfügung stellt. HA Oracle ist z.B. ein hoch-verfügbarer Oracle Datenbank-Dienst, der mit Hilfe eines Agenten in Oracle Solaris Cluster integriert ist.

Der GDS ist nun ein generischer Dienst, der mit Hilfe von Properties so konfiguriert werden kann, dass er eine beliebige Anwendung in das Cluster integriert. Tatsächlich gibt es nur eine Eigenschaft, die gesetzt werden muss, die Start-Methode, alle anderen sind optional. Das klingt nun fast zu einfach, um wahr zu sein. Tatsächlich kann die Integration einer komplexen Anwendung immer noch Wochen in Anspruch nehmen, bis sie Produktreife erlangt hat. Auf der anderen Seite ist eine einfache „Anwendung“ wie die unter Unix bekannte xclock innerhalb von wenigen Minuten integriert mit den Kommandos:

```
clrg create xclock-rg
clrs create -g xclock-rg -t SUNW.gds -p Network_aware=false -p\
Start_command="DISPLAY=dummy:0.0 /usr/openwin/bin/xclock xclock-rs"
clrg online -M xclock-rg
```

(Dies lässt sich natürlich auch mit Hilfe einer grafischen Oberfläche und eines „Wizards“ konfigurieren.)

Warum ist nun trotz der vorhandenen Hilfsmittel eine Anwendungsintegration immer noch aufwändig? Man muss hier unterscheiden zwischen der

eigentlichen Integration in das Clusterframework, d.h. die korrekte Kommunikation zwischen Anwendung und Framework und die Definition, Implementierung und den Test der vielen möglichen Fehlerfälle, die vom Cluster überwacht werden sollen. Die Integration ist auf Grund der existierenden Werkzeuge recht einfach. Die umfassende Implementierung einer sorgfältigen Überwachung ist schwierig. Welche Fehlerfälle sind z.B. zu berücksichtigen? Die einfachsten sind:

- Serverausfall
- Netzwerkprobleme
- Storage-ausfall
- Anwendungsausfall.

Beim Anwendungsausfall wird es schon schwierig. Wie sieht der Algorithmus aus, der einen Ausfall der Anwendung eindeutig erkennt? Reicht es aus, die Existenz eines (Unix)-Prozesses zu überwachen? Die Antwort ist ganz klar: nein. Auch ein existierender Prozess garantiert nicht, dass die Anwendung etwas Sinnvolles tut. Also muss eine ausgefeiltere Überwachung her. Und genau hier wird es aufwändig.

Der Agent für HA Oracle, d.h. die Komponente in Oracle Solaris Cluster, die eine single instance Oracle Datenbank ins Cluster integriert mit Start-, Stopp- und Überwachungsmethoden, nutzt sehr ausgefeilte Methoden, um die Funktionsfähigkeit der Datenbank ständig zu überwachen: er beobachtet DB-eigene Statistiken, überwacht das alert log und schreibt und liest u.U. in und aus einer eigenen Tabelle (dies natürlich mit entsprechenden Timbuktus, um auch bei einer hängenden Datenbank noch reagieren zu können. Diese direkten Zugriffe auf die DB sind eine sehr gute Lösung, da es recht unwahrscheinlich ist, dass die Operationen des Agenten durchgeführt werden können, die anderer Benutzer aber nicht (obwohl auch das vorkommen kann).

Kritische Anwendungen schreiben im laufenden Betrieb Meldungen aller Art in ihre Log-Dateien. Eine Oracle-Datenbank nutzt für Meldungen die sog. Alert-log Datei. Der HA-Oracle Agent überwacht nun diese Log-Datei und reagiert auf Einträge mit Aktionen, die einer Tabelle festgelegt sind. Dort findet man z.B. den Eintrag:

```
{
  ERROR=470;
  ERROR_TYPE=DBMS_ERROR;
  ACTION=SWITCH;
  CONNECTION_STATE=*;
  NEW_STATE=die;
  MESSAGE="Oracle Background Prozess dies";
}
```

Diese Regel besagt, dass beim Auftreten des Fehlercodes 470 die Datenbank auf einen anderen Clusterknoten verlagert wird. Das kann in vielen Situationen vernünftig sein, muss es aber nicht. Also kann man z.B. die Zeile so ändern, dass die DB neu gestartet wird durch:

```
ACTION=RESTART;
```

I.A. Ist es nicht sinnvoll, die Standardregeln zu ändern.

Ein weiterer schwieriger Fehlerfall sind Timeouts. Alle Aktionen in einer Cluster-Konfiguration werden durch Timer überwacht, um zu verhindern, dass Überwachungsprozesse hängen bleiben. Läuft der Timer ab, kann das Cluster entsprechend reagieren. Die im Agenten eingestellten default Timeouts für Start-, Stopp- und Überwachungsmethoden sollten für eine Standard-Installation ausreichend sein. Wenn sich aber die Bedingungen für die Anwendung entscheidend verändern, müssen auch die Timeouts angepasst werden. Ein typischer Fehlerfall sind nicht angepasste Timeouts bei erheblich erhöhter Last, die dann zu unerwünschten Neustarts oder Verlagerung der Datenbank führen.

Zum Schluss sei für die Zweifler aller automatischen Überwachungsmethoden und automatischer Reaktionen des Clusters gesagt, dass das gesamte Monitoring einer Anwendung sehr einfach auf LOG_ONLY gesetzt oder gar ganz ausgeschaltet werden kann, so dass keinerlei Reaktion des Clusters auf Fehler, die der Monitor entdeckt, erfolgt.

Agenten für Oracle Infrastruktur und Anwendungen

Da Oracle Produkte wie die Oracle DB, Oracle Applications, Oracle Real Application Cluster (oder früher: Oracle Parallel Server) immer schon die waren, die am häufigsten auf unternehmenskritischen Sun Solaris Systemen anzutreffen waren, ist es nicht überraschend, dass gerade für diese bewährte und ausgefeilte Agenten zur Verfügung stehen.

Der am häufigsten eingesetzte Agent ist der HA Oracle Agent; d.h. Eine Failover-Lösung für eine „single instance“ Oracle DB. Gerade bei diesem Agenten gibt es immer wieder Diskussionen, ob ein versierter DBA nicht schnell und effizient einen eigenen Agenten auf der Basis des Generic Data Service (GDS) implementieren und damit die Anschaffungs- und Wartungskosten für den HA Oracle Agenten gespart werden könnten. Im Prinzip ist das richtig; aber wer erstmal anfängt, einen eigenen Agenten zu implementieren, merkt sehr schnell, dass der Aufwand, einen guten Agenten zu schreiben, nicht unerheblich ist, und die Kosten für einen eigenen Agenten sehr schnell die Kosten für den kommerziellen Agenten übersteigen würden.

Daneben gibt es noch den Agenten oder besser die Infrastruktur für die Integration von Oracle Solaris Cluster mit Oracle Real Application Cluster. Die Wurzeln für dieses Produkt reichen bis in die Zeiten Oracle PDB (Parallel DB) und Oracle OPS zurück. Seit RAC mit einer eigenen Clusterschicht ausgeliefert wird, ist die Oracle Solaris Cluster Infrastruktur für Oracle RAC mehrfach erweitert worden und liegt seit Mitte September 2010 auch für die Integration mit Oracle 11gR2 RAC vor. Eine Diskussion der Vorzüge einer solchen Integration würde den Rahmen dieses kurzen Papiers sprengen.

Die Oracle E-Business Suite ist eine komplexe multi-Tier Infrastruktur, die natürlich nicht durch einen einzigen Agenten abgesichert werden kann. Stattdessen werden existierende Agenten, z.B. HA Apache und spezielle E-Business Suite Agenten, z.B. für Forms Server, Concurrent Manager Server und Reports Server kombiniert, um eine Absicherung der E-Business Suite zu gewährleisten.

Darüber hinaus existieren mit Oracle Solaris Cluster 3.3 Agenten für folgende Oracle Technologien:

- Oracle Communication Messaging Exchange Server and Enterprise Mobility Server,
- Oracle Business Intelligence Suite Enterprise Edition Plus,
- Oracle WebLogic Server,
- Oracle Grid Engine,
- Oracle TimesTen,
- Oracle Solaris Containers,
- Oracle VM Server for SPARC,
- PeopleSoft, PostgreSQL,
- Siebel CRM,
- MySQL Server,
- MySQL Cluster.

Zusammenfassung

Oracle Solaris Cluster ist ein in Oracle Solaris integriertes Framework, das existierende Anwendungen problemlos durch sog. Agenten in das Cluster integriert. Zur Integration eigener Anwendungen stehen Programmierschnittstellen, Werkzeuge und der Generic Data Service (GDS) zur Verfügung. Agenten für die wichtigsten Oracle Anwendungen wie Oracle DB, Oracle RAC, Oracle E-Business Suite stehen als Produkt zur Verfügung.

Kontaktadresse:

Hartmut Streppel

Oracle Deutschland B.V. & Co. KG
Sonnenallee 1
D-85551 Kirchheim-Heimstetten

Telefon: +49 (0) 89-460082563
Fax: +49 (0) 89-460082572
E-Mail: Hartmut.Streppel@oracle.com
Internet: www.oracle.com