



**Menschen. Innovationen. Lösungen.**



**OPITZ CONSULTING**



# Das neue ORACLE-Monitoring von Check\_MK

---

**Thorsten Bruhns**  
Solution Architect | OCIMSI-Team  
OPITZ CONSULTING Deutschland GmbH



München, 21.10.2014

# Der Referent ☺

---

## Thorsten Bruhns

Solution Architect



**1999 – 2003**      **Oracle Deutschland GmbH**

**2003 –**            **OPITZ CONSULTING Deutschland GmbH**

Spezialist für:

Performance-Analysen und SQL-Tuning  
Hochverfügbarkeit mit RAC + Data-Guard

Nagios seit ~2003

Check\_MK seit 2012

# Agenda

---

- 1. Motivation**
- 2. Einsatzszenarien**
- 3. Tipps & Tricks**
- 4. Security in mk\_oracle**
- 5. Alternative Checks für Oracle**
- 6. Aussichten**
- 7. Fazit**

1

# Motivation



## Motivation – Entstehungsgeschichte bei OC

### ■ OMD als Lösung zur Überwachung von Oracle Datenbanken

- Kunden benötigen einfache Lösung
  - OMD + Check\_MK + mk\_oracle sehr einfach zu installieren und zu konfigurieren
- Workshop mit Götz Golla
- OC Newsletter zu OMD + Check\_MK
- Vortrag auf der DOAG 2013 (~200 Teilnehmer beim Vortrag)
  - Vorstellung von sqlplus.sh + Oracle Wallet

### ■ Neues Monitoring für Kunden bei OPITZ CONSULTING

- Existierende Eigenentwicklung soll durch Produkt abgelöst werden
  - Wir wollen das Rad nicht neu erfinden – wenn möglich
- Nagios Know-how vorhanden
- Administration des System muss für jeden Mitarbeiter ‚verständlich‘ sein
- Modulare Architektur, einfache Updatemöglichkeiten



## Motivation – Entstehungsgeschichte bei OC

### ■ **Check\_MK-Packages maßgeblich für die Entscheidung**

- Modulares Konzept ist wichtig
- Individuelle Erweiterungen möglich

### ■ **mk\_oracle (alt) als Basis**

- Ohne mk\_oracle wäre die Entscheidung nicht für Check\_MK gefallen!
- Erste Erfahrungen und Erweiterungen am Plugin

### ■ **Idee für mk\_oracle (aktuell mk\_oracle.new)**

- Partnerschaft zwischen MK und OPITZ CONSULTING (OC)
  - Überarbeitung vom mk\_oracle-Plugin von Mathias Kettner + Thorsten Bruhns
    - 1. Version von mk\_oracle.new (08. Juli 2014)
  - Zahlreiche Patches von Thorsten Bruhns
  - 1. Weiterentwicklungen für einen Kunden im August



## Motivation – Was hat sich geändert?

### ■ Abhängigkeiten zwischen Checks

- Neuer Instance-Check wird ‚führend‘
  - Alle weiteren Checks werden ‚Stale‘, wenn Login nicht möglich
  - Reduzierung von Notifications

### ■ Konsequentes Design für RAC und physical Standby-Database

- Beides wird nun unterstützt – clustered Tablespacechecks sind möglich.

### ■ Reduzierung der Logins mittels SYNC + ASYNC Checks

- Performancesteigerung durch Reduzierung der Logins
  - ASYNC-Checks mittels ‚run\_cached‘ verbessert
  - SYNC + ASYNC Checks benötigen 2 Logins pro Datenbank – für alle Checks!

### ■ Neue Checks

- Diverse neue Checks (‚besserer‘ RMAN-Check, Recovery-Area ...)





## Motivation – Was hat sich geändert?

- **Automatische Erkennung der Oracle-Version**
  - Nur mögliche Checks werden ausgeführt
    - Plugin funktioniert ab 9.2
    - Recovery-Area geht erst ab 10.1
      - Plugin erkennt das automatisch!
  - Optional können Checks über `mk_oracle.cfg` konfiguriert werden
    - Mittels Bakery über WATO einfach konfigurierbar
- **Connect zur Instance immer über Listener**
  - Nur zuverlässig, wenn `sqlplus.sh` nicht genutzt wird
  - Bessere Überwachung einer Instance
  - Ohne Listener kann sich kein Endanwender anmelden
    - Warum sollte das Monitoring am Listener ‚vorbei‘ die Instance überwachen?



## Motivation – Was hat sich geändert?

### ■ **sqlplus.sh nicht mehr benötigt**

- Aus Kompatibilitätsgründen noch nutzbar
  - Es sollte dringend darauf verzichtet werden!
  - Erkennung der Oracle-Version möglicherweise fehleranfällig mit sqlplus.sh
  - Testmodus vom Plugin nicht mit sqlplus.sh möglich
- **mk\_oracle.cfg ersetzt sqlplus.sh**
  - Konfiguration der Connection zur Oracle erfolgt über mk\_oracle.cfg
  - Bisher sind keine Einschränkungen zum sqlplus.sh vorhanden
    - „su – oracle –c „sqlplus / as sysdba““ nicht empfehlenswert
      - Monitoring sollte mit möglichst wenig Rechten erfolgen
  - tnsnames.ora optional nutzbar
    - TNS-Eintrag muss der ORACLE\_SID entsprechen
- Bakery wird sqlplus.sh nicht unterstützen



## Motivation – Was hat sich geändert?

### ■ **mk\_oracle\_asm** wird überflüssig

- ASM wird über sqlplus statt asmcmd überwacht
  - Kein eigenes Plugin mehr erforderlich
    - Code von mk\_oracle wird verwendet
    - 1 Plugin weniger 😊
  - ASM wird ab 10.2 unterstützt
- Konfiguration sehr einfach
  - Userkonfiguration in ASM-Instance
  - 1 Eintrag in mk\_oracle.cfg genügt
- Überwachung von ASM-Diskgroups
- Überwachung von OFFLINE-Disks
- ‚Required Mirror Free Space‘ bei NORMAL oder HIGH Redundancy
  - Verlust einer Disk einer Failuregroup erhält nach Rebalancing die Failuregroup
  - Option führte zum ‚incompatible Change‘
    - Ursprünglicher Check oracle\_asm\_diskgroup nutzte ‚filesystem‘-Rule



# Motivation – Was hat sich geändert?

## ■ Geänderte Checknamen

- Geänderte Checknamen werden für RAC und physical Standby benötigt
  - Check\_MK benötigt bei clustered Checks ‚gleiche‘ Checknamen
- Checks nutzen ORACLE\_SID oder DB\_NAME
  - Beispiel für DB\_NAME: Tablespace, Recovery-Area ...
  - Beispiel für ORACLE\_SID: Instance, Undo ...
- ORACLE\_SID / DB\_NAME immer in Großbuchstaben
  - Ist notwendig, weil sonst ‚mixed case‘ entstehen können

## ■ Neue Checknamen für Jobs (DBMS\_SCHEDULER)

- Jobnamen waren nicht eindeutig
  - ‚incompatible Change‘!
  - Alt: ORA <DB\_NAME>.<JOB\_NAME>
  - Neu: ORA <DB\_NAME>.<JOB\_OWNER>.<JOB\_NAME>



# Motivation – Was hat sich geändert?

## ■ Windows-Plugin

- .bat ersetzt durch PowerShell
- Wird alle Checks von mk\_oracle für Linux/Unix unterstützen
- Derzeitig noch experimentell
  - Noch nicht offiziell in Check\_MK enthalten
- Nutzt mk\_oracle.cfg analog zu mk\_oracle



## Motivation – Neues Plugin `mk_oracle_crs`

- **2013 als GridInfrastructure-Package für Linux veröffentlicht**
  - Vorstellung im Rahmen des DOAG-Vortrages
  - mein 1. selbst geschriebenes Plugin. 😊
- **August 2014 Überarbeitung und Portierung nach AIX**
- **=> Nun als `mk_oracle_crs` in `Check_MK` verfügbar**
- **Checks**
  - Clusterware-Ressourcen
  - Voting-Disks

# 2

## Einsatzszenarien



# mk\_oracle - Einsatzszenarien

## ■ Unterstützte Plattformen

- Linux
- AIX (nur mit bash)
- Solaris, HP-UX (nicht getestet, nur mit bash)

## ■ Oracle-Versionen

- RDBMS 9.2
- RDBMS 10.1, 10.2, 11.1, 11.2, 12.1
- ASM 10.2, 11.1, 11.2, 12.1
- Durch automatische Versionserkennung werden nur mögliche Checks ausgeführt





# mk\_oracle\_crs - Einsatzszenarien

## ■ Unterstützte Plattformen

- Linux
- AIX (nur mit bash)

## ■ Oracle-Versionen

- Grid-Infrastructure 11.2 + 12.1



## mk\_oracle - Einsatzszenarien

- **Single Instance**
  - Die häufigste Variante
- **Physical Standby Database incl. Data-Guard**
  - Connect mit ‚sysdba‘ oder ‚sysoper‘ notwendig!
- **Failover-Cluster**
  - Beim Connect zur Instance die VIP und nicht die Host-IP nutzen!
- **Real Application Cluster (RAC)**
  - Beim Connect zur Instance die VIP und nicht die Host-IP nutzen!



## mk\_oracle - Einsatzszenarien

### ■ Logical Standby Database

- Ist nur extrem selten in der Praxis zu finden
- Standby darf nicht auf gleichem Host wie Primary oder physical Standby laufen

### ■ RAC One-Node geht nicht

- One-Node ‚verschiebt‘ ORACLE\_SIDs zwischen Clusterknoten
- Ist die ‚offizielle‘ Alternative zum Failover-Cluster von Oracle
  - Aus Kostengründen so gut wie nie genutzt

### ■ ASM

- Als Single Instance, Oracle Restart oder RAC



## mk\_oracle - Einsatzszenarien

- **Mischkonfigurationen sind möglich**
  - RAC + physical Standby als RAC geht auch!
  - Danke für die Clusterfunktionalität in Check\_MK!
- **mk\_oracle ist somit für die Zukunft gut gerüstet!**
  - Einmalige Migration auf neue Checknamen beim Umstieg auf mk\_oracle erforderlich
    - RAC und Data Guard machen diesen Wechsel zwingend notwendig
      - Wegen zu hoher Komplexität wurde auf einen Parameter für neue/alte Checknamen verzichtet. Spätestens mit RAC oder Data Guard muss der Wechsel vollzogen werden.



# mk\_oracle – Check Instance

## ■ oracle\_instance

- Neuer Check, der eine zentrale Aufgabe übernimmt
- Alle anderen Checks werden STALEs, wenn Login nicht möglich
  - Reduzierung von Alarmierungen, wenn Instance nicht aktiv
  - Dependency-Regel nicht mehr notwendig
- Prüft:
  - OPEN\_MODE der Instance
    - Primary => OPEN
    - Physical Standby oder ASM => MOUNTED
  - Logins erlaubt ja/nein
  - Archivelogmodus ja/nein
  - Force Logging bei Archivelogmodus ja/nein
  - Uptime der Instance



# mk\_oracle - Checks

## ■ Neue Checks

- oracle\_processes
  - Überwachung der Prozessanzahl
- oracle\_recovery\_area
  - Idee von [Exchange.mathias-kettner.de](http://Exchange.mathias-kettner.de)
- oracle\_recovery\_status
  - Überwachung letzter recovery Checkpoint
- oracle\_rman
  - Prüft letztes Level0/1 und Archivelog-Backup
- oracle\_undostat
  - Überwachung v\$undostat
  - Ersetzt die Überwachung von UNDO-Tablespace



# mk\_oracle - Checks

---

## ■ Neue Checks

- oracle\_dataguard\_stats
  - Überprüft Transport und Apply Lag
- oracle\_locks
  - Lang anhaltende Locks in der Instance
- oracle\_longactivesessions
  - Überprüfung auf lang laufende Sessions
  - Per Default nicht aktiv



# mk\_oracle - Checks

## ■ Erweiterte Checks

- oracle\_jobs
  - Überprüfung der Scheduler-Jobs
  - Check wurde erweitert wegen fehlenden OWNER beim Job-Namen
- oracle\_tablespaces
  - Zusätzliche Überprüfungen
    - UNDO und TEMP-Tablespaces werden nicht mehr überwacht!
    - Im Moment nur Überwachung mit ‚free‘-Space möglich
- oracle\_asm\_diskgroup
  - Zusätzliche Überprüfungen
    - OFFLINE-Disks
    - ‚Required Mirror Free Space‘ bei NORMAL oder HIGH Redundancy





# mk\_oracle - Checks

## ■ Alte Checks

- oracle\_logswitches
  - Anzahl Logswitches
- oracle\_rman\_backups
  - Check wurde durch oracle\_rman ersetzt
  - Check ist ungünstig, da nur auf mögliche Fehler in letzten n-Tagen geprüft wird
    - Wie soll man den Status im Fehlerfall korrigieren?
- oracle\_sessions
  - Überwachung der Sessions in der Instance
  - Überwachung der Prozesse ist mehr zu empfehlen
- oracle\_version
  - Diente früher als Logincheck
  - Ist überflüssig geworden und wurde den Instancecheck abgelöst

# 3

## Tipps & Tricks



# mk\_oracle – Konfiguration des Plugin

## ■ Installation

- Kopiere `mk_oracle` nach `/usr/lib/check_mk/plugins`
- Kopiere `sqlnet.ora` nach `$MK_CONFDIR`
- Es wird auch wieder ein RPM für Linux geben. 😊

## ■ Konfiguration von Check\_MK Plugin

- `mk_oracle.cfg`
  - `ASMUSER=„<username>:<password>:<sysdba>:<hostname>:<port>“`
  - `DBUSER=„<username>:<password>:<sysdba>:<hostname>:<port>“`
  - `DBUSER_<ORACLE_SID>=„<username>:<password>:<sysdba>:<hostname>:<port>“`
  - Defaults
    - `username = /`
    - `password` und `sysdba` nicht gefüllt
    - `hostname = localhost`
    - `port = 1521`



# mk\_oracle – Konfiguration des Plugin

## ■ Konfiguration von Check\_MK Plugin

- `vi $MK_CONFDIR/mk_oracle.cfg`
  - `ASMUSER="asmsnmp:asmsnmp:sysdba"`
  - `DBUSER="dbsnmp:dbsnmp:"`



## mk\_oracle - Testmodus

### ■ Verbindungstest nach der Konfiguration

- Nicht mit sqlplus.sh nutzbar!
- Debugging von Konfigurationsproblemen
- Sollte nach jeder Konfiguration genutzt werden
  - Erspart die Fehlersuche
  - Funktioniert der Testmodus, dann funktioniert auch das Plugin
- Bei Fehlern erfolgt die Ausgabe der genutzten Verbindungsdaten

### ■ Wie funktioniert der Testmodus?

- `export MK_CONFDIR=/etc/check_mk`
- `/usr/lib/check_mk/plugins/mk_oracle -t`



## mk\_oracle - Testmodus: Beispielausgabe

```
/usr/lib/check_mk_agent/plugins/mk_oracle.new -t
```

```
<<<oracle_instance>>>
```

```
... Ausgabe aller Agentsections von mk_oracle
```

```
<<<oracle_asm_diskgroup>>>
```

```
-----  
Logincheck to Instance: +ASM Version: 12.1
```

```
Login ok User: SYS on ora12c.local
```

```
SYNC_SECTIONS= instance
```

```
ASYNC_SECTIONS= asm_diskgroup  
-----
```

```
Logincheck to Instance: tux2 Version: 12.1
```

```
Login ok User: SYS on ora12c.local
```

```
SYNC_SECTIONS= instance sessions logswitches undostat recovery_area
```

```
processes recovery_status longactivesessions dataguard_stats performance
```

```
ASYNC_SECTIONS= tablespaces rman jobs ts_quotas resumable locks
```



## mk\_oracle - Debugmodus

- **Debugmodus vom Agenten übernommen**
  - ASYNC-Checks werden im Debugmodus als SYNC-Checks ausgeführt
    - Bessere Debugmöglichkeit im Fehlerfall
- **Wie wird der Debugmodus aufgerufen?**
  - `export MK_CONFDIR=/etc/check_mk`
  - `/usr/lib/check_mk/plugins/mk_oracle -d`

# 4

## Security in mk\_oracle





# mk\_oracle - Security

## ■ User & Rechte

- Connect als dedizierter User
  - Kein ‚/ as sysdba‘ wie bei sqlplus.sh empfohlen
  - ‚physical Standby‘ benötigt zwingend ‚sysdba/sysoper‘-Recht

## ■ Wallet

- Anpassung der Advanced Security Option Lizenz
  - Seit Juli 2013 darf das Wallet als Passwordstore für Datenbankverbindungen ab Standard-Edition genutzt werden – auch in DBs ab 10.1.
- Keine Klartextkennwörter in Konfigurationsdateien
- mk\_oracle.cfg ist kompatibel zum Wallet
  - Username=/ => Walletnutzung
  - tnsnames.ora empfohlen
    - andernfalls wird die Walletkonfiguration ‚unübersichtlich‘



## mk\_oracle - Wallet

### ■ Wallet für sicheres Login konfigurieren

- Als root Environment der DB-Software setzen
- `mkstore -wrl /etc/check_mk/oracle_wallet -create`
- `mkstore -wrl /etc/check_mk/oracle_wallet -createCredential „<TNS_ALIAS>“ dbsnmp geheim`
- `vi /etc/check_mk/sqlnet.ora`  
`SQLNET.WALLET_OVERRIDE = TRUE`

**5**

## **Vergleich zu alternativen Monitoringlösungen**





# Vergleich

---

## ■ **check\_oracle\_health**

- Bekannteste Lösung
  - Sehr umfangreich
  - Modular
  - Bestandteil von OMD
  - Wird meist mit NRPE kombiniert

## ■ **Diverse Nagios-Plugins**

- Typischerweise als NRPE-Plugin
  - durch check\_oracle\_health obsolete geworden



## Vergleich `mk_oracle` ↔ `check_oracle_health`

### ■ `check_oracle_health`

- + bietet viel mehr Checkmöglichkeiten
- + bewährtes Plugin in zahlreichen Umgebungen
- + modular erweiterbar
- + individuelle SQL-Statements möglich
- + individuelle Checkintervalle pro Check
- - aktive Checks in Nagios
- - jeder Check benötigt 1 Connect
- - individuelle Tablespacechecks ein Alptraum
  - Jeder Tablespace braucht 1 Check => erhebliche Anzahl an Connects notwendig
  - Reguläre Ausdrücke für Tablespacenamen machen es nur bedingt besser



## Vergleich `mk_oracle` ↔ `check_oracle_health`

### ■ `mk_oracle`

- + Gruppierung der Checks
  - SYNC-Check pro Aufruf des Agenten
  - ASYNC-Checks nur alle n-Sekunden
  - Massiver Performancegewinn durch Reduzierung von Connects
- + einfache Konfiguration
  - Nur Bakery oder `mk_oracle.cfg` – Rest passiert automatisch
- + passive Checks in `Check_MK`
  - Regelwerke in `Check_MK` erleichtern die Konfiguration
- - weniger Checkmöglichkeiten
- - ‚aufwändiger‘ zu erweitern
- - aktuell nur 1 Intervall für ASYNC-Checks konfigurierbar
  - Ist dem KISS-Prinzip geschuldet



## Vergleich `mk_oracle` ↔ `check_oracle_health`

### ■ Fazit

- `mk_oracle` für Standardmonitoring einfach konfigurierbar
  - Inventarisierung + `mk_oracle.cfg` / Bakery bieten hohen Automatisierungsgrad
- Server mit vielen DBs leiden unter hoher Belastung durch Monitoring
  - Gilt nicht für `mk_oracle`, da deutlich weniger Connects und SQLs notwendig
- `check_oracle_health` für ‚Spezialfälle‘ vorerst notwendig
  - Argument entfällt mit weiterem Ausbau von `mk_oracle`
- => **Umstieg auf `mk_oracle` immer zu empfehlen**

# 6

## Aussichten





# Aussichten – Anpassung bestehender Checks

## ■ Tablespace

- Alle Optionen vom Filesystemcheck per WATO ermöglichen
- Überwachung von TEMP-Tablespace ab Oracle 11.1
  - Danke für den Tipp eines MK-Kunden!

## ■ Recovery-Status

- Erweiterung notwendig, um Irritationen zu vermeiden
  - Check liefert zu häufig WARNINGS in Primary-Datenbanken

## ■ Oracle Performance-Daten

- Plugin liefert schon einige Daten 😊
- DB-Time, CPU-Time
- Buffer-Cache und Library-Cache hit-Ratio
- Für ASM Performance-Daten pro Diskgroup



# Aussichten – Anpassung bestehender Checks

## ■ Oracle Optionen

- Überwachung auf „aktive“ Optionen
  - Active-Data-Guard
    - Nur open read only + aktives Recovery im Data-Guard
  - Diagnostic+Tuning-Pack
    - Nur ab Oracle 11g
  - OLTP Compression



# Aussichten – Anpassung neue Checks

## ■ Far Sync Standby

- Neue Option in 12c

## ■ Individuelle SQL-Statements

- Hier sind noch einige Diskussionen notwendig
  - Das wird bestimmt kommen, da es sehr hilfreich ist.

## ■ Multitenant Database

- Pluggable Database ab 12c



## Aussichten - Sonstiges

---

### ■ **MAN-Pages**

- Es müssen aufgrund der neuen Logik alle MAN-Pages überarbeitet werden

### ■ **Dokumentation auf Homepage**

- Generelle Überarbeitung der alten Dokumentation
- Beispiele für RAC und physical Standby

### ■ **Bakery**

- Ist alles vorbereitet
- Mathias muss hier nur noch den Teil in WATO komplettieren



## Fazit

---

### ■ **Check\_MK ein geniales Projekt**

- Mir macht die Zusammenarbeit und Weiterentwicklung sehr viel Spaß
- Danke an das ganze Team von Mathias Kettner!

### ■ **es wird nicht langweilig**

- Ideen für Erweiterungen sind immer willkommen!
- Patches für Erweiterungen sind auch gerne gesehen
  - Sollten im Vorfeld mit MK angestimmt werden, da dort die endgültige Entscheidung für die Aufnahme erfolgt

### ■ **DOAG 2014 in Nürnberg**

- Ich bin dort– einfach am Stand von OPITZ CONSULTING fragen.

### ■ **Danke für die tolle Konferenz!**

# Kontakt

---

## Thorsten Bruhns

Solution Architect | OCIMSI-Team

OPITZ CONSULTING Deutschland GmbH  
Norsk-Data-Strasse 2 | 61352 Bad Homburg  
Tel. +49 (6172) 66 26 0 - 1541  
thorsten.bruhns@[opitz-consulting.com](mailto:thorsten.bruhns@opitz-consulting.com)

