



Menschen. Innovationen. Lösungen.





Betrieb und Wartung "out of the box"

Monitoring von Oracle Datenbanken mit OMD

Thorsten Bruhns

Solution Architekt | OC|MSI-Team

OPITZ CONSULTING Deutschland GmbH



Hamburg, 17.06.2014

Agenda

- 1. Vorwort**
- 2. Was ist OMD?**
- 3. Check_MK**
- 4. Oracle-Plugin**
- 5. Aussichten**
- 6. Fazit**

1

Vorwort



Vorwort

- **Datenbankmonitoring ist wichtig**
 - Probleme erkennen bevor sie kritisch werden
 - Für kritische Systeme unverzichtbar
- **Wir brauchen ein einfaches Monitoring**
 - Cloud-Control ist viel ‚zu groß‘
 - Häufig fehlen Lizenzen für Enterprise Edition + Diagnostic Pack
 - Erweiterbarkeit nur mit viel Know-How
 - Wartungsarm
 - Flexibel, idealerweise können wir alles damit überwachen
 - Zukunftssicher, wie sieht es mit 12c aus?
- **=> Open Monitoring Distribution als Lösung**

2

Was ist OMD?



OMD (OpenSource Monitoring Distribution)

■ Was ist das?

- Fertiges Paket (RPM bzw. deb) aus diversen Nagios-Komponenten
 - Nagios, nagios-plugins, nsca, check_nrpe
 - PNP4Nagios
 - Check_MK, Livestatus, Multisite, MKEventd, MKNotifyd, WATO
 - Check_logfiles, check_oracle_health, check_mysql_health, check_multi ...
 - Und vieles mehr...

■ Wer steht hinter dem Projekt?

- Lars Michelsen, Matthias Kettner und Wolfgang Barth
- Entwickler: Gerhard Laußer, Joerg Linge, Matthias Flacke, Michael Friedrich, Stefan Hoesl, Sven Nierlein

■ Lizenz

- GPL



OMD (OpenSource Monitoring Distribution)

■ Was ist an OMD so innovativ?

- Installation aller Komponenten mit 1 RPM/deb Paket
 - Abhängigkeiten zu OS-Packages werden berücksichtigt!
- Alles aufeinander abgestimmt
 - ‚es funktioniert sofort nach der Installation‘
- Updatefähigkeit durch Trennung von Software ↔ Konfigurationsdateien
- Mandantenfähigkeit (mehrere Sites auf 1 Host möglich)
 - Jede Site startet eigene Nagios, httpd, npcd, crontabs ...
 - Mehrere OMD-Versionen installierbar, Site wird einer Version zugeordnet

■ OMD-Updates sind sehr einfach

- `omd update <site>`

■ Konfiguration der Ziele steht im Vordergrund

- Für alles andere ist OMD zuständig

3

Architektur von Check_MK



Warum Check_MK? Nagios reicht doch, oder?

- **Check_MK ist in OMD sofort einsatzbereit**
- **Multisite als Frontend deutlich moderner**
 - WATO als Konfigurationsfrontend sehr praktisch
- **Check_MK generiert Nagios-Konfigurationsdateien**
 - Installation des Agenten auf Zielsystem (1 Python-/Shellskript!)
 - Inventory über WATO starten
 - Check_MK erkennt die möglichen Checks aus dem Result vom Agenten
 - Individuelle Schwellwerte über Regelwerke konfigurierbar – optional per WATO
 - Nagios reloaden/restarten
 - Check_MK prüft die Konfiguration vor dem Neustart
 - Ziel ist in der Überwachung!



Warum Check_MK? Nagios reicht doch, oder?

■ Umfangreiche Checksammlung

- Standardchecks
 - CPU, Filesystem, Kernel, Netzwerk, NTP, Uptime ...
- Individuelle Checks programmierbar
 - Check_MK kann über Plugins im Agenten und Packages erweitert werden
 - Plattform für Plugins: <http://exchange.check-mk.org>
- Einbinden und automatisiertes Erkennen von NRPE-Checks über Agenten
 - NRPE-Modul installieren, in mrpe.cfg eintragen und dann automatisch inventarisieren
 - Migrationspfad beim Umstieg von Nagios auf Check_MK + Nagios
- Checks per SNMP möglich
 - snmpwalk als Inventory
 - Bibliothek für viele Komponenten bereits vorhanden (cmk -L | grep snmp)
- Aktive Nagios-Checks



Rules und Tags

■ Rules und Tags machen Check_MK genial

- Hostsystemen werden 1-n ‚Tags‘ zugewiesen
- Rules können Tags, Hosts, Services ... als Ziel nutzen
 - Automatische Generierung von Host-/Servicegruppen möglich
 - Entternen/Hinzufügen generiert automatisch gültige Nagioskonfigurationen
- Rules können global genutzt werden
 - Allgemeine Rule für ‚df /‘
 - Pro Host/Tag können individuelle Schwellwerte definiert werden.

■ Rules erleichtern die Administration

- WATO als Web-Frontend ermöglicht die Erstellung und Verwaltung von Rules und Tags

■ Directories als weiteres Strukturelement

- Wird im Vortrag nicht betrachtet



Überwachung von geclusterten Systemen

■ Check_MK kann mit Clustern arbeiten

- Überwachung von Failover- oder Active/Active-Cluster ist möglich
- Für Checkentwicklung ist alles vorhanden, um effektive Clusterchecks schreiben zu können
 - Da sind alle denkbaren Kombinationen möglich
- Check_MK cached den Agentoutput
- Jeder Result vom Agent wird dem Zielservice zugeordnet
 - Zielservice kann auf anderen Host liegen!

■ Agentbasierte Checks sind extrem flexibel

- Es ist praktisch alles möglich
 - Service muß auf allen Knoten aktiv sein, nur auf 1 oder auf 2 von 3 usw.

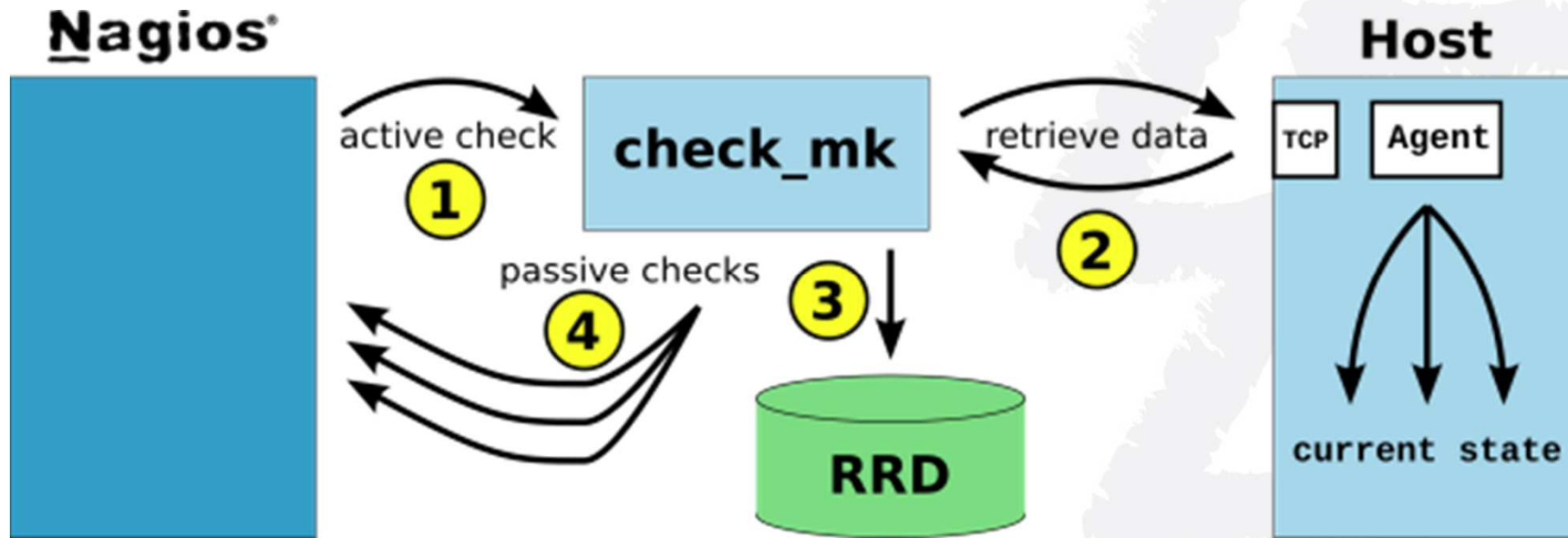


Performancegewinn durch passive Checks

- **1 aktiver Check generiert viele passive Checks**
 - Pro Hostsystem 1 aktiver Nagioscheck (Abfrage des Check_MK Agenten)
 - Cacheing des Results
 - für hostübergreifende Results
 - Schnelles Inventory ohne Agentaufruf
 - Checkergebnis wird an Check_MK übergeben
 - Splitt des Ergebnisses auf alle notwendigen Zielservices
 - Check_MK generiert passive Checks die an Nagios übergeben werden
- **=> Nagios-Core wird signifikant entlastet**



Schaubild





Installation OMD-Server

■ RPM und DEB-Package für Linux

- `yum install -y omd-1.10-rh61-31.x86_64.rpm`
 - Alle Paketabhängigkeiten werden automatisch aufgelöst
 - Yum-Repository unter: <https://labs.consol.de/repo/stable/>

■ Site anlegen und konfigurieren

- `omd create doaghh`
- `omd config doaghh set DEFAULT_GUI check_mk`
 - Alternativ: `omd config doaghh`
- `omd start doaghh`

■ Die Site in OMD läuft!



Installation Check_MK Agent unter Linux

■ RPM und DEB-Package für Linux

- Besteht aus 1 Python-Skript + Konfiguration xinetd
- Alternativ auch über ssh ansprechbar
 - Sinnvoll wenn Hosts im Internet überwacht werden sollen

■ Plattformen:

- SuSE, Debian, OracleLinux, RedHat, CentOS, ESXi, Solaris, AIX, HP-UX
 - Eigene Agenten pro Plattform
- Windows-Agent verfügbar
 - Als Installer und Source verfügbar
 - Erweiterbar wie die anderen Agenten

■ Download:

- http://mathias-kettner.de/check_mk_download.html



Installation Check_MK Agent unter Linux

■ Installation Check_MK Agent

- `yum install -y rpm/check_mk-agent-1.2.3i6-1.noarch.rpm`

■ Inventory in WATO

- `http://192.168.68.100/ocdemo`
- => WATO Configuration => Hosts & Folders => New host
 - Hostname: omd
- => Save & go to Services

■ Activate Changes nicht vergessen

- Generiert neue Nagioskonfiguration
- Validiert und rollt ggf. auf alte Konfiguration zurück
- Reload von Nagios-Core

4

Oracle Plugin(s)



Oracle Check_MK Plugin

- **In Check_MK integriertes Plugin**
 - Agent muß für Zielsystem installiert und konfiguriert werden
- **Über WATO konfigurierbare Schwellwerte**
- **Aktiv bei einigen Kunden im Einsatz**
 - praxiserprobtes Plugin!
- **Konfiguration gewöhnungsbedürftig**
 - Flexibel gestaltetes Plugin erfordert Anpassung eines Shellskriptes
 - Ist nicht schlimm nur sollte man etwas nachdenken bevor es konfiguriert wird.
 - Geliefertes Beispiel von Check_MK nicht schön
 - Ich zeige Alternativen



Oracle Check_MK Plugin - Checks

- **Login**
- **Session-Limit**
- **Tablespace-Monitoring**
- **Logswitches**

- **Alert.log**
 - Monitoring erfolgt über logwatch-Plugin von Check_MK!

- **OC-Agent als optimale Alternative im Einsatz**



Oracle oracle_health Plugin

■ Process-Usage

- % process Usage

■ Fast-Recovery-Area Usage

- % Usage

■ Download

- http://exchange.checkmk.org/index.php?option=com_remository&Itemid=53&func=fileinfo&id=97

■ OC-Agent als optimale Alternative im Einsatz



Oracle RMAN Plugin

■ RMAN-Backups

■ Download

- http://exchange.checkmk.org/index.php?option=com_remository&Itemid=53&func=fileinfo&id=95

■ OC-Agent als optimale Alternative im Einsatz



OC-Agent für Oracle-Überwachung

■ Refactoring des Shell-Agenten von Check_MK

- reduzierte Anzahl an Connects
 - Schnellere Antwortzeiten vom Agenten – weniger Last für die Instance(n)
 - Originalplugin hat massive Performanceprobleme bei vielen Instanzen
- Mehr Details zu einer Datenbank
- RAC-Fähigkeit in Arbeit
- Standby-Datenbank in Vorbereitung
- Pluggable Database in Planung

■ Bleibt kompatibel zum Originalagenten

- ORACLE_SID in Multisite immer in uppercase
 - Kann beim Wechsel zum OC-Agenten zu neuen Checks führen



OC-Agent für Oracle-Überwachung

■ Neue Funktionen

- Neuer Instancecheck
 - Wenn Check <> OK, dann gehen alle anderen Checks auf UNKNOWN
 - Reduzierung von Notifications!
- Undo- und Temp-Tablespace nicht mehr überwacht
- Überwachung v\$undostat als Alternative Undoüberwachung
- Synchroner / asynchroner Checks
 - Asynchrone Checks mit Cacheing zur Lastreduzierung (Tablespaceüberwachung)
 - Synchroner Checks (,einfache SQLs auf v\$-Views)
- Und einiges mehr

■ Was fehlt noch

- Physical Standby
- Oracle XE-Support
 - Einfacher Workarround vorhanden



Konfiguration Oracle-RDBMS – Variante 1

■ DBSNMP-User reicht

- `alter user dbsnmp identified by geheim account unlock;`

■ 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 "localhost/" dbsnmp geheim`

■ `vi /etc/check_mk/sqlnet.ora`

```
SQLNET.WALLET_OVERRIDE = TRUE
WALLET_LOCATION =
  (SOURCE=
    (METHOD = FILE)
    (METHOD_DATA = (DIRECTORY=/etc/check_mk/oracle_wallet))
  )
```



Konfiguration Oracle-RDBMS – Variante 2-3

- **Variante 2 ist wie Variante 1 aber zusätzlich,**
 - tnsnames.ora
 - Eintrag für ORACLE_SID muß vorhanden sein
 - Connect über spezielle Services und/oder Listener möglich

- **Variante 3 ist wie Variante 2 aber zusätzlich**
 - Login mit Hilfe einer Konfigurationsdatei
 - Über Konfigurationsdatei /etc/check_mk/mk_oracle_dbuser.conf
 - Login ‚as sysdba‘ möglich
 - wird für Standby-Database benötigt
 - Check_MK erkennt eine Instance als <> OK, wenn PRIMARY und <> OPEN



Oracle Grid-Infrastructure Plugin

- **Grid-Infrastructure Ressourcen**
 - Critical wenn ‚target‘ und ‚state‘ ungleich sind
- **Voting-Disks**
 - Anzahl Voting-Disks muß 1,3 oder 5 sein
- **ASM-Diskgruppen**
 - Experimentell
- ‚mein‘ 1. **Check_MK-Plugin.** 😊

5

Fazit



Fazit

- **Oracle-Monitoring ist schnell aufgebaut**
 - Warum gibt es überhaupt noch Umgebungen ohne Monitoring?
- **OMD ist schnell installiert sowie site eingerichtet**
 - Gut durchdachtes Konzept von Matthias Kettner
 - Software + Konfiguration sind konsequent getrennt
 - Alle Komponenten sind optimal aufeinander abgestimmt
- **Check_MK + Multisite + WATO**
 - Eine starke Kombination die keine Wünsche offen läßt
 - Check_MK kann noch viel mehr – Vortragszeit ist leider begrenzt ...
- **OMD gibt es auch als Subscription mit Support**
 - Matthias Kettner bietet Support und Individualentwicklung für OMD an



Fazit

- **Neuer Agent von OPITZ CONSULTING**
 - Einfache Konfiguration – kein lästiges Anpassen von sqlplus.sh mehr
- **Managed Service Infrastructure Monitoring von OPITZ CONSULTING**
 - Verteiltes Monitoring bei OPITZ CONSULTING im Aufbau
- **Nagioskenntnisse nicht zwingend erforderlich**
 - Check_MK ‚hängt‘ sich quasi ein – parallel kann bestehende Konfiguration genutzt werden.
- **OMD macht süchtig**
 - Wo sind noch mehr Systeme die ein Monitoring benötigen?
- **OPITZ CONSULTING hilft gerne bei der Suchtbehandlung. 😊**

Kontakt

Thorsten Bruhns

Solution Architekt | OC|MSI-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)

