



# Oracle-Reports in Enterprise-Projekten: Erfahrungsbericht über Architektur, Performance und weitere Aspekte



Jan-Peter Timmermann • Bereichsleiter  
OPITZ CONSULTING Hamburg GmbH

Hamburg, 18.06.2009





- **Die Herausforderung**
- **Welches Betriebssystem**
- **Reports Server Ansprache und Loadbalancer**
- **Ein bisschen Tuning**
- **Die Landschaft**
- **Zusammenfassung**

# 1

## Die Herausforderung





- **Formsmodule** : **1450**
- **Reports Module** : **1600**
- **Benutzer** : **2000**
- **Rechenzentrum** : **Zentral in Hamburg**
- **Niederlassungen** : **über 12 Länder in Europa verteilt**
- **Druck Volumen** : **ca. 4000 Reports pro Stunde**



- **Druckerauswahl** : wie gewohnt aus der Anwendung
- **Druckarten** : direkt und mit Vorschau
- **Druckerplatz** : am Arbeitsplatz in der Niederlassung
- **Was für Drucke** : Lieferscheine, Rechnungen, Bestellscheine, Statistiken
- **Zeichensatz** : UNICODE
- **FaxVersand** : direkt aus Reports heraus



- **Für welches Betriebssystem entscheide ich mich**
  - Windows , Linux, AIX, Unix
- **32 oder 64 Bit System**
- **Webcache oder Hardware Loadbalancer**



- **Was ist in Ihrem Hause das bevorzugte Betriebssystem**
- **Gibt es Vorteile / Nachteile wenn ja welche**
  - Gross Klein Schreibung auf Unix
  - Speicherverwaltung
  - Prozess Liste
  - Security
  - Austauschen von Modulen im laufendem Betrieb
- **Einbinden von Druckern**
  - Print Server oder alle Drucker dem Reports Server bekannt geben
  - Probleme bei unterschiedlichen Versionen (32 / 64 Bit)



- **Antwort von Oracle zum Thema UNICODE**

**Reports 10g unter Linux benutzt noch das alte Toolkit (guicommon, TK) zum Aufbereiten der Reports - und das kann nur singlebyte-Code. Daher ist eine saubere Aufbereitung von multibyte-Codes nicht möglich (Zeilenverschiebungen....). ORACLE empfiehlt, die Reportserver auf Windows aufzusetzen (hier ist eine saubere Aufbereitung garantiert, da das Toolkit hier nicht verwendet wird).**





- **Gibt es eine Möglichkeit aus Forms heraus den Drucker auszuwählen**
- **Ich möchte keine „Web Vorschau“**
- **Printer.dll erstellt und die über ora\_ffi in Forms eingebunden**
- **Möglichkeit damit alle Drucker die dem Client bekannt sind anzuzeigen und auszuwählen**



# 2

## Reports Server Ansprache und Load Balancer





- **Server Umgebung mit n-Servern**
  - Webcache oder HW-Loadbalancer
- **Loadbalancer verteilt die Forms Session**
  - Auf welchem Server lande ich
- **Starten des Reports mit run\_report\_object**
  - Welcher Server ?
- **Darstellen des Dokumentes mit Web.Show\_document**
  - Von welchem Server



- **Report Server Name meistens in den Modulen hinterlegt**
- **Ich muss eindeutigen Report Server Namen in meinem Netz haben**
- **Wie kann ich aus Forms heraus meinen Server Namen erkennen**



- **Erweitern der `rwServlet.properties`**
  - `REPORTS_SERVERMAP = maptest:<reportservername>`
- **Erweitern der `<XXX>.env`**
  - `REPORTS_SERVERMAP = maptest:<reportservername>`
- **Forms Modul erweitern**

```
tool_env.getvar('REPORTS_SERVERMAP',mapped_entry);
reportServer :=substr(mapped_entry,instr(mapped_entry,':')+1);

WEB.SHOW_DOCUMENT('http://machine:port/reports/rwServlet/getjobid'||
substr(v_rep,instr(v_rep,'_',-1)+1)||'?'||'server='||reportServer,'_blank');
```

- **Zuweisen in einem DB-Package**
  - Ermitteln des Servers auf dem die Forms Anwendung läuft
    - `tmpVar := sys_context('USERENV', 'HOST');`
  - Zuweisen des Reports Servers
  
- **Verwaltung von über Reports Verwaltungs Tabelle**
  - Als Steuerung Mechanismus um bei Bedarf Ausdrücke auf gezielte Server zu leiten



# 3 Ein bisschen Tuning





- **SQL Tuning**
  - Reports nutzt SQL PL/Sql optimalen Code
- **Schalten Sie Funktionen die Sie nicht brauchen aus**
  - UrlEngine kann ausgeschaltet werden wenn alles nur über `run_report_object`





- **maxEngine** = **max. Prozesse zur Ausführung**
  - Durchschnittliche Laufzeit \* max. Anfragen  
 $\text{maxEngine} = (\text{average time to run report}) * (\text{maximum report requests inputrate}) = (10/60) * 12 = 2$
- **initEngine** = **bereitstellen von Prozessen**
  - Durchschnittliche Laufzeit \* min. Anfragen  
 $\text{initEngine} = (\text{average time to run report}) * (\text{minimum report requests inputrate}) = (10/60) * 6 = 1$
- **minEngine** = **irgendwas zwischen 0 und initEngine**
- **Empfehlung** = **2 bis 4 Engines per CPU**



- **engineResponseTimeOut**
  - Hängende Jobs erkennen und „aufräumen“
  - Sollte auf einen Wert gesetzt werden der Ihre Laufzeit widerspiegelt
- **Batch Jobs von Online Jobs trennen**
- **maxConnect**
  - Die maximale Anzahl von gleichzeitigen Anfragen
  - Ein zu geringer Wert verursacht einen Java Fehler



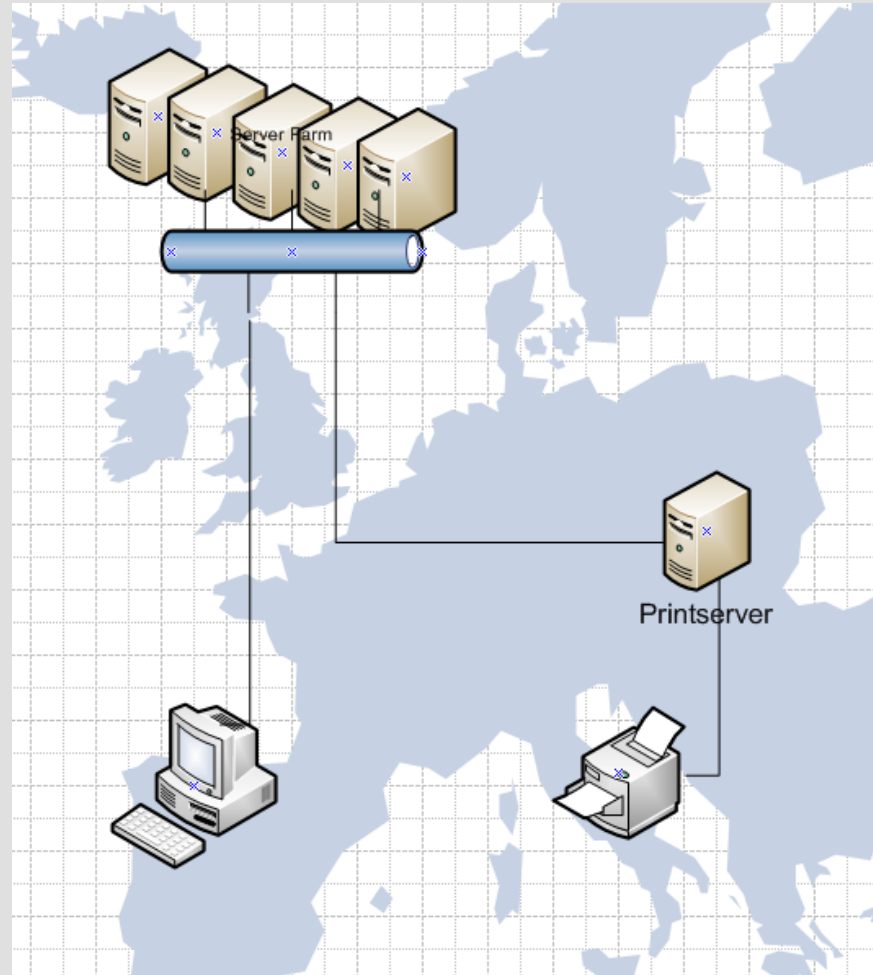
- **Jede Engine stellt eigenen Prozess da**
- **Zu viele Engine haben Einfluss auf das Betriebssystem**
- **Zu viel Heap Speicher beeinflusst den Server**



# 4

## Die Landschaft







- **8 Oracle Applikation Server in der Version 10gR2**
- **2 Print Server**
- **2 HW Load Balancer**
  
- **Report Server laufen mit auf dem Forms Server**
- **Online Reports über eigene Reports Server**
- **Batch / langlaufende Reports auf seperate Reports Server**
  
- **Pro CPU 2 Engine**



# 5

## Zusammenfassung



- **Konzeption zum Anfang ist wichtig**
- **UNICODE leider nur auf Windows (bis heute morgen)**
- **Bei Printservern unter Windows Treiber verträglichkeiten beachten**
- **HA nicht sauber unterstützt (bis heute morgen)**
- **Grund Konfiguration läuft stabil nicht über tunen**
- **Reports ist für mich eine Hochperformante Umgebung**





# Fragen & Antworten



## Kontakt:

Jan-Peter Timmermann  
OPITZ CONSULTING Gummersbach GmbH  
[Jan.timmermann@opitz-consulting.com](mailto:Jan.timmermann@opitz-consulting.com)  
+49 40 74 11 22 0

**Vielen Dank für Ihre  
Aufmerksamkeit !**

