



Herzlich Willkommen

Private Label mit Oracle Advanced Queuing und Scheduling

Herrmann & Lenz Services GmbH

Herrmann & Lenz Solutions GmbH



- Erfolgreich seit 1996 am Markt
- Firmensitz: Burscheid (bei Leverkusen)
- Beratung, Schulung und Betrieb/Fernwartung rund um das Thema Oracle Datenbanken
- Schwerpunktthemen: Hochverfügbarkeit, Tuning, Migrationen und Troubleshooting
- Herrmann & Lenz Solutions GmbH
 - Produkt: Monitoring Module
 - Stand Nr. 212 auf Ebene 2

DIE DATENBANK-PROFIS.
DIE DATENBANK-PROFIS.

nobilis – Marktführer in Deutschland

- Größter Produzent von Einbauküchen in Europa
- Umsatz von 850,1 Millionen Euro in 2011
- Die Mitarbeiteranzahl beträgt 2.357
- Produktion ausschließlich am Standort Deutschland
- Täglich werden mehr als 2.400 Küchen produziert, das sind 535.000 Küchen pro Jahr



„Jede dritte in Deutschland verkaufte Küche ist eine nobilis.“

535.000 Küchen pro Jahr. Jede ein Unikat!

- Zwei Produktionsstandorte im Zwei-Schicht-Betrieb:
Werk I in Verl Sürenheide
Werk II in Verl Kaunitz

Leistungszahlen 2011

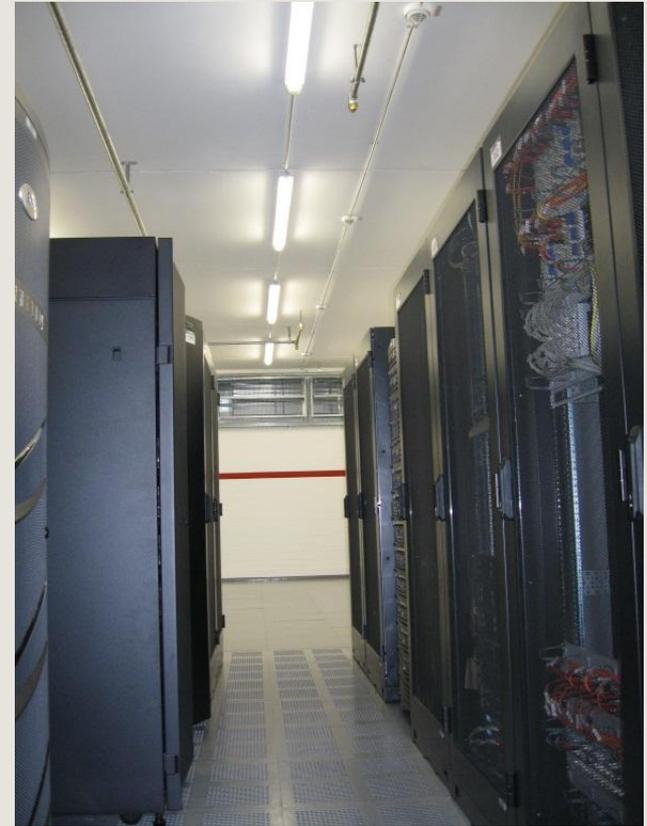
- 5,4 Millionen Schränke und 1,4 Millionen Arbeitsplatten pro Jahr
- 25.000 Schränke und 5.000 Arbeitsplatten pro Tag
- J.I.T.-Anlieferung von mehr als 5.000 Elektrogeräten pro Tag



Die IT im Unternehmen

Abteilung IV/ORG

- 37 Mitarbeiter
- 2 Rechenzentren
- 7x24 h Betrieb
- Nutzung von Unix und Windows als Betriebssystem
- Datenbankbetrieb unter Oracle und MS-SQL
- Einsatz von Exchange als Groupware- und Mailsystem
- 400 Server
- 2500 Clients und MDE-Terminals
- 150 Terabyte betriebenes Speichervolumen im SAN
- Verarbeitung von 60.000 IDocs pro Tag



Die IT im Unternehmen

Heterogene Umgebung mit den Kernbestandteilen:

COE (Cad OrdEr Entry) für die grafische Auftragserfassung

ERP SAP als Backbone für alle Geschäftsprozesse

VPL-Verplanung und Anwendungen Werk für die Anbindung der Produktion

IT in der Produktion, bestehend aus Systemen mit LVS-, PPS-, MES- und BDE-Funktionalitäten auf Basis von Oracle-Datenbanken



Aufgabenstellung

Ausgabe von Logos und internationalisierten
Artikeltexten auf den Fertigungsetiketten :=
„Private Label“ in der Produktion



Komise: 62301632 / Trong

Artikl: H60-01
Texty artiklú: VS, 5 poliček, 2 dveře, V 1930mm

Program: 343 Solo
Čelní plocha: 691 Bílá vysoký lesk
Korpus: 048 Dub žilnatý IM
Úchyt: 069 Úchytka barva nerez ocel / 01 Vodrovně

US	0	WS	0	HS	1	50	R	01	SH	15	B	60.0	S	326 00
													201210181306	Box: 18.10.2012

GESCHIRRSCHRANK 193 CM HOCH

T:31490 / R B:211

DO 538

E4 ZB3 G:069 01 QUER 160

048 SPLINTEICHE 691 SOLO WEISS HOCHGLANZ

AB: 1430 **8227** Pos: **07** **6/01**



98000

Ausgangslage

- Quelle der Informationen auf den Fertigungsetiketten sind das COE- und das SAP-System. Diese beiden Systeme sind vor Projektbeginn auf UNICODE umgestellt worden.
- Ausgegeben werden die Fertigungsetiketten im Rahmen der Produktion von 6 verschiedenen Systemen, mit denen die LVS, PPS, MES und BDE-Funktionalitäten im Betrieb abgebildet werden. Diese waren zu Projektbeginn alle Non-UNICODE-DB's.
- Die Fertigungsetiketten werden an vielen Stellen im Durchlauf gedruckt. Damit hängt dort die Produktionsgeschwindigkeit von der Druckgeschwindigkeit ab.

Konzept zur Umsetzung

Die Etiketten sollen im Vorfeld aufbereitet und auf die Systeme in der Produktion verteilt werden.

Vorteile dieses Konzepts:

- Die Druckgeschwindigkeit vor Ort wird durch den Prozess der Etikettenaufbereitung nicht beeinflusst.
- Durch die Verteilung der Daten auf die Systeme werden neue Abhängigkeiten vermieden. Jedes System kann für sich wie vorher die Etiketten ausgeben.
- Die Notwendigkeit, für den Projekterfolg die Produktionsdatenbanken auf UNICODE zu migrieren, entfällt.

Erster Lösungsansatz

1. Die Texte für die Fertigungsetiketten einzelner Artikel werden aus dem SAP-System übernommen und im Verplanungssystem um die Fertigungsinformationen ergänzt.
2. Als Ergebnis dieses Prozesses werden Nutzdaten und Steuerungsdaten für die zu erstellenden Fertigungsetiketten getrennt jeweils in einer Tabelle gespeichert.
3. Die Prozesse zur Erstellung der Fertigungsetiketten laufen auf Windows-Servern. Diese holen sich ihre Aufträge aus der Steuertabelle. Die Aufträge werden in der Steuertabelle jeweils durch Update auf eine Status-Spalte markiert und zurückgemeldet.

Auswertung des Lösungsansatzes

- Zu 1.:** Zur Steuerung der Jobs zu diesem Zeitpunkt wurde eine eigenentwickelte JCL auf Perl-Basis eingesetzt. Aufgrund der Abhängigkeiten vom Wert der Zeichensatz-Variablen der Umgebung beim Aufruf kam es zu Problemen bei der Übernahme der Texte im UNICODE-Format von SAP in die VPL. Aufgrund anderer Abhängigkeiten konnten die Einstellungen nicht ohne weiteres geändert werden.
- Zu 2.:** Diese Aufteilung erwies sich als praktikabel.
- Zu 3.:** Bei 10.000 Fertigungsetiketten und zwei Servern funktionierte dieser Ansatz. Bei 100.000 Etiketten und 5 Servern kam es zu Lock-Situationen auf der Tabelle. Dadurch reduzierte sich die Verarbeitungsgeschwindigkeit auf ein nicht tolerierbares Niveau.

Diskussion im Projektteam

Zu 1.: **Mögliche Lösungen des Problems der Datenübernahme:**

- Redesign der bestehenden Steuerung und Angleichung der Parameter
- Beschaffung und Einsatz eines neuen Tools zur Jobsteuerung
- Nutzung von Oracle Scheduled Chains zur Steuerung

Zu 3. **Mögliche Lösungen des Problems der Lock-Situationen**

- Ausbau der vorhandenen Job-Steuerungslogik
- Beschaffung und Einsatz einer Middleware-Lösung zur Kommunikation
- Einsatz von Oracle Advanced Queuing zu Kommunikation

Ergebnis der Diskussion

Zu 1.: **Nutzung von Oracle Job Scheduling zur Steuerung**

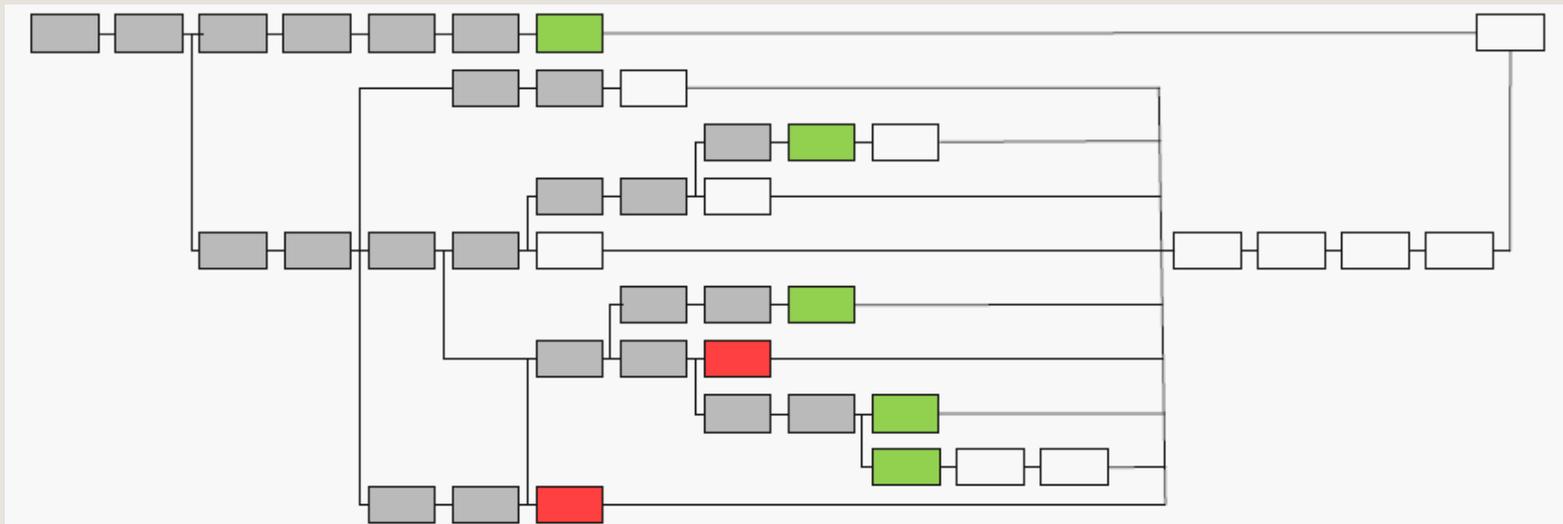
- Ablauf der Jobs mit den Zeichensatz und –code-Einstellungen der Datenbank
- Möglichkeit, Abhängigkeiten zwischen Jobs abzubilden
- Vorhandenes Tool in Oracle, keine Beschaffung notwendig
- Betrieb durch vorhandene Mitarbeiter im RZ möglich
- Keine zusätzlichen Kosten

Zu 3. **Einsatz von Oracle Advanced Queuing zu Kommunikation**

- Vermeidung von Dead-Lock-Situationen
- Keine eigene Entwicklung einer Steuerung notwendig
- Vorhandenes Tool in Oracle, keine Beschaffung notwendig
- Betrieb durch vorhandene Mitarbeiter im RZ möglich
- Keine zusätzlichen Kosten

Umsetzung auf der Datenbank

- ⇒ Eine Job-Kette kann aus beliebig vielen Steps bestehen
- ⇒ Steps können vordefinierte Programme, andere Jobketten oder Event Schedules sein



Schema der Umsetzung



Darstellung der fertigen Job-Kette Verplanung mit SQL Developer

Umsetzung auf der Datenbank

Problem des mehrfachen Aufrufes einer Job-Kette

- ⇒ Start der Job-Ketten jeweils unter einem eigenen Namen. Damit läuft jede Job-Kette in einem eigenen Kontext
- ⇒ Übergabe der Parameter aus den vorgelagerten Prozessen über eine Steuertabelle. Gleichzeitig wird damit ein Mehrfachaufruf verhindert, falls eine Job-Kette nicht parallel laufen darf

```
PROCEDURE run_jit_scheduler (p_paketnr IN NUMBER DEFAULT NULL)
AS
  --job_number BINARY_INTEGER;
  vJobName VARCHAR2 (100);
BEGIN
  vJobName :=
    'JIT_DATEN_'
    || p_paketnr
    || '-'
    || SUBSTR (TO_CHAR (DBMS_RANDOM.random), -3);
  --vJobName := 'JIT_DATENAUFBAU' || vJobName;
  DBMS_SCHEDULER.RUN_CHAIN (
    chain_name => 'ZETTELDRUCK_JOB_PVL_DATEN_JIT',
    job_name => vJobName,
    start_steps => NULL);
END run_jit_scheduler;
```

```
43 | .program_name => 'ZETTELDRUCK_P_VORGEBE_LISTEN';
44 | SYS.DBMS_SCHEDULER.DEFINE_CHAIN_STEP (
45 |   .chain_name => 'ZETTELDRUCK_JOB_PVL_DATEN_JIT'
46 |   .step_name => 'JIT_SHELLVERTEILUNG'
47 |   .program_name => 'ZETTELDRUCK_P_JIT_SHELLVERARBEITUNG');
48 |
49 | SYS.DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
50 |   .chain_name => 'ZETTELDRUCK_JOB_PVL_DATEN_JIT'
51 |   .condition => 'TRUE'
52 |   .action => 'START DO_NOTHING'
53 |   .rule_name => 'ZETTELDRUCK_R1000_START_JIT'
54 |   .comments => 'Parameter für die Programme setzen');
55 |
56 | SYS.DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
57 |   .chain_name => 'ZETTELDRUCK_JOB_PVL_DATEN_JIT'
58 |   .condition => '(select distinct(verarbtyp) FROM n010_verarbeitungs_parameter
59 |   WHERE status = 0) = 'JIT''
60 |   .action => 'START JIT_PARAMETER_SETZEN'
61 |   .rule_name => 'ZETTELDRUCK_R4000_JUST_IN_TIME'
62 |   .comments => NULL);
```

Umsetzung Advanced Queuing

- ⇒ Definition der Queue
- ⇒ 4 Queues für die Prozesse:
 - Standard-Verarbeitung
 - Just-In-Time-Verarbeitung
 - Storno und Nachverplanung
 - APL-Verarbeitung
- ⇒ Die jeweiligen Queues bilden die Grundlage für die Verarbeitungsprozesse

```
BEGIN
SYS DBMS_AQADM.STOP_QUEUE ( QUEUE_NAME => 'ZETTELDRUCK.Q_WEBSERVICE2' );
SYS DBMS_AQADM.DROP_QUEUE ( QUEUE_NAME => 'ZETTELDRUCK.Q_WEBSERVICE2' );
END;
/

BEGIN
SYS DBMS_AQADM.CREATE_QUEUE
(
  QUEUE_NAME           => 'ZETTELDRUCK.Q_WEBSERVICE2'
  , QUEUE_TABLE        => 'ZETTELDRUCK.TAB_WEBSERVICE2'
  , QUEUE_TYPE         => SYS.DBMS_AQADM.NORMAL_QUEUE
  , MAX_RETRIES        => 0
  , RETRY_DELAY        => 0
  , RETENTION_TIME     => 172800
);
END;
/

DECLARE
aSubscriber sys.aq$_agent;
BEGIN
aSubscriber := sys.aq$_agent('NOBILIA',
                             0);
dbms_aqadm.add_subscriber
( queue_name => 'ZETTELDRUCK.Q_WEBSERVICE2'
  , subscriber => aSubscriber);
END;
/

BEGIN
SYS DBMS_AQADM.START_QUEUE
(
  QUEUE_NAME => 'ZETTELDRUCK.Q_WEBSERVICE2'
  , ENQUEUE => TRUE
  , DEQUEUE => TRUE
);
END;
/
```

Advanced Queuing - Priorisierung

⇒ Über `PROPERTIES.priority` wird gesteuert, welches Etikett als erstes verarbeitet wird.

```
BEGIN
-- vrecipient (1) := sys.AQ$AGENT ('WEBSERVICE', NULL, NULL);
-- vMESSAGE_PROPERTIES.recipient_list := vrecipient;

vMESSAGE_PROPERTIES.expiration := 18000; -- Nach 3 Stunden in Errorqueue verschieben
vMESSAGE_PROPERTIES.correlation := 'NORMAL';
vMessageQueue := 'Q_WEBSERVICE2';
vBoxtag := TO_DATE (pBoxtag, 'YYYYMMDD');

CASE UPPER (pVerarbTyp)
WHEN 'VORG'
THEN
    vMESSAGE_PROPERTIES.priority := 3;           --- Priotität 2
    vMESSAGE_PROPERTIES.correlation := 'VORGEZOGEN';
    vMessageQueue := 'Q_PVL_VORGEZOGEN';
WHEN 'DV'
THEN
    vMESSAGE_PROPERTIES.priority := 4;           --- Priotität 3
    vMESSAGE_PROPERTIES.correlation := 'NORMAL';
    vMessageQueue := 'Q_WEBSERVICE2';
```

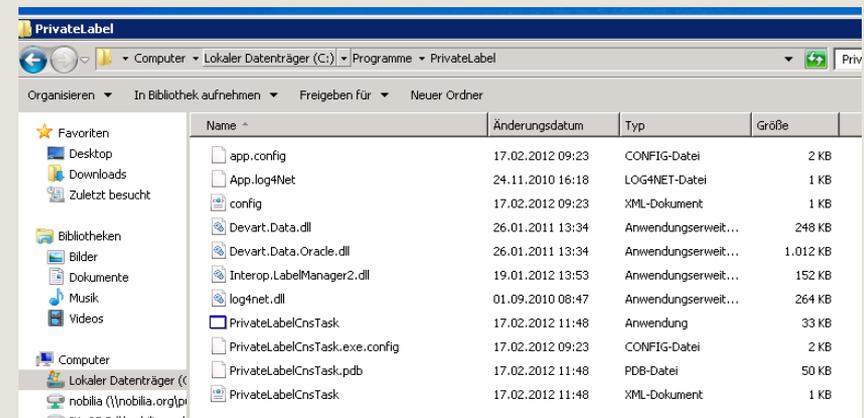
Mit den eigenen Queues und der Priorisierung wurde im Projekt das Puffern der Nachrichten umgangen, da es Etiketten gibt, die sofort verarbeitet werden müssen.

Umsetzung auf den Windows-Servern

Basis ist das Layout-Tool CODESOFT (Fa. TEKLYNX)

Die Verarbeitung selbst ist in VB 6 geschrieben

Die Kommunikation mit den Advanced Queuing-Methoden wird über dotConnect für Oracle der Fa. devart realisiert



Anmerkung: Aktuell wird das Programm auf .NET und die Nutzung von XML portiert.

Fazit

- (+) - Das Projekt konnte durch Nutzung von Oracle Advanced Queuing und Scheduling erfolgreich abgeschlossen werden.**
 - Im Betrieb laufen die damit erstellten Prozesse stabil . Durch das Advanced Queuing konnte die Laufzeit soweit reduziert werden, dass das Zeitfenster zur Erstellung der Etiketten eingehalten wird.
 - Der administrative Aufwand ist gering, da die Prozesse im Kontext der Oracle-Datenbanken laufen und genügend Oracle-Skill zur Administration in der Firma vorhanden ist.
 - Oracle Advanced Queuing und Scheduling wird bei neuen Projekten wie der Einführung von RFID im Betrieb eingesetzt.

- (-) - Die Beschaffung von geeigneten Beschreibungen und Dokumenten zur Implementierung der Techniken gestaltete sich schwieriger, als bei der Auswahl von Oracle Advanced Queuing und Scheduling angenommen.**

Vielen Dank für Ihre Aufmerksamkeit

