OPITZ CONSULTING

Menschen. Innovationen. Lösungen.

Oracle Net Troubleshooting+Tuning

DOAG 2015

Uwe Küchler

Managing Consultant – Infrastructure

OPITZ CONSULTING Deutschland GmbH

ORACLE Platinum Partner

Specialized
Oracle Database

Nürnberg, 17.11.2015

© OPITZ CONSULTING GmbH 2015

Agenda

1. Einführung: Was ist Oracle Net (nicht)?
2. Tuning
3. Praxisbeispiel Troubleshooting: Sporadische Verbindungsabbrüche

© OPITZ CONSULTING GmbH 2015

OPITZ CONSULTING

Menschen. Innovationen. Lösungen.

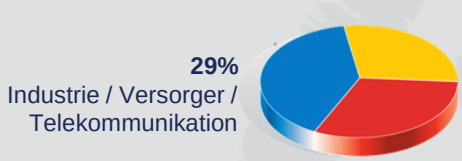
Mission

Wir entwickeln gemeinsam mit allen Branchen Lösungen, die dazu führen, dass sich diese Organisationen besser entwickeln als ihr Wettbewerb.

Unsere Dienstleistung erfolgt partnerschaftlich und ist auf eine langjährige Zusammenarbeit angelegt.

Märkte

- Branchenübergreifend
- Über 600 Kunden

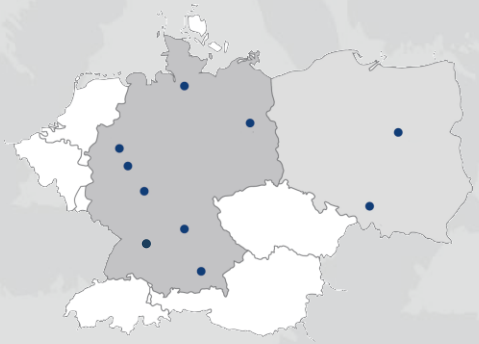


Leistungsangebot

- Application Lifecycle Management
- IT-Beratung
- Business-Lösungen
- Managed Services
- Training und Coaching
- IT-Trends

Eckdaten

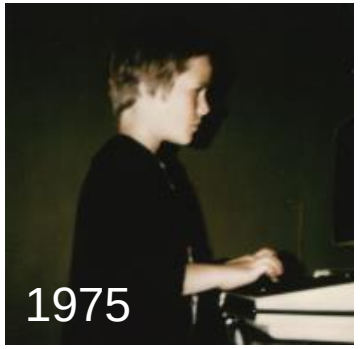
- Gründung 1990
- 400 Mitarbeiter
- 10 Standorte

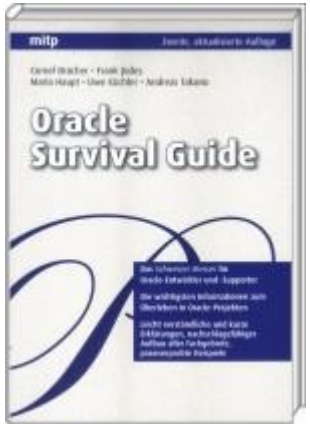


© OPITZ CONSULTING GmbH 2015

Zur Person

- Generation C=64
- Seit über 25 Jahren in der IT tätig
- 1997-2000 bei Oracle Deutschland
- Seither durchgehend Oracle-Berater, im DBA- und Entwicklungs-Umfeld, Tutor
- Seit 09/2013 bei OPITZ CONSULTING
- Buch- und Blogautor (oraculix.de)
- Performance als „Steckenpferd“





© OPITZ CONSULTING GmbH 2015

1

Was ist Oracle Net (nicht)?

© OPITZ CONSULTING GmbH 2015

Was ist Oracle Net (nicht)?

- **In früheren Versionen unter „SQL*Net“ bekannt**

Quelle: Tim Reckmann / flickr, CC by-nc-sa 2.0

- **… sonst ändert sich nix. ;-)**

© OPITZ CONSULTING GmbH 2015

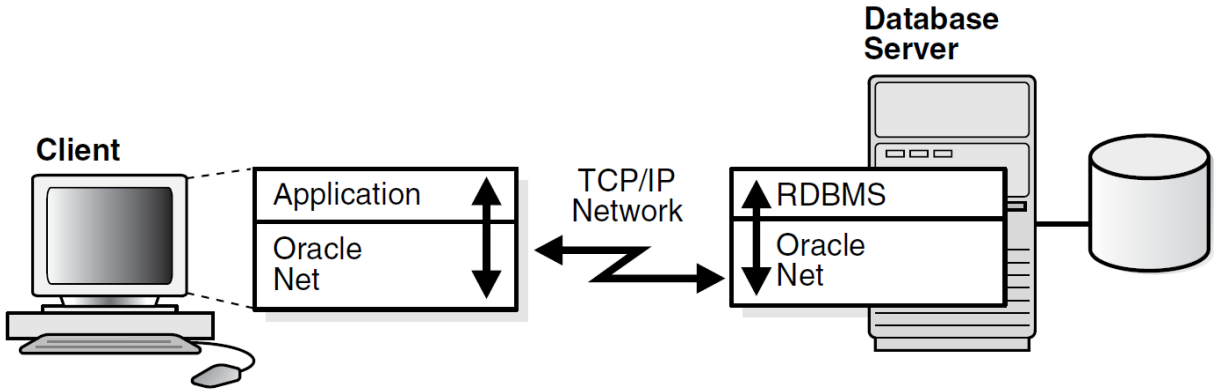
Was ist Oracle Net (nicht)?

- **Grundlage für die Kommunikation mit der Oracle-Datenbank über das Netzwerk**
- **Teil der „Oracle Net Services“**
  - Oracle Net
  - Listener
  - Connection Manager
  - Oracle Net Configuration Assistant
  - Oracle Net Manager
- **Oracle Net ist *kein* Netzwerkprotokoll!**
  - Verbindungen zwischen einem Client und einem DB-Server aufbauen
  - Verbindungen aufrecht erhalten
  - Mitteilungen zwischen Client und Server
  - Über bestehende Netzwerkprotokolle

© OPITZ CONSULTING GmbH 2015

Was ist Oracle Net (nicht)?

- **Allgemeine Aufgaben: „Oracle Net Foundation Layer“**
- **Abbildung auf das verwendete Netzwerkprotokoll: „Oracle Protocol Support“.**



Quelle: Oracle Doku

© OPITZ CONSULTING GmbH 2015

2 Tuning

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Woraufhin optimieren?

- **Datendurchsatz**
  - „Packe so viel Information wie möglich in so wenige Pakete wie nötig“
  - Z.B. für Ladeprozesse aus Vorsystemen in ein DWH
- **Verbindungsaufbau**
  - „Brich den Verbindungsaufbau so früh wie möglich ab und versuche ggf. eine alternative Verbindung“
  - Z.B. bei RAC oder Standby-Konfigurationen
- **Schutz vor Überlastung**
  - Logon Storms
  - Dead Connection Detection

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Datendurchsatz

- **Ziel: optimale Ausnutzung der TCP-Pakete**
- **Weg: Reduktion überflüssiger Kommunikation**
  - Einsparen von ACK-Paketen
  - Einsparen von TCP/IP-Overhead
- **Parameter: SDU_SIZE**
- **Wissenswerte Begriffe dazu:**
  - **SDU**: Session Data Unit; Puffergröße, die an das Netzwerkprotokoll übergeben wird (512 bis 2097152 Byte;
    Default: **8192 Byte** ab Oracle 11.2, vorher: 2048 Byte)
  - **MTU**: Maximale Paketgröße (1500 Byte bei Ethernet, 9000 Byte bei Jumbo Frames)
  - **MSS**: Maximale Segmentgröße für Nutzdaten

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Datendurchsatz

- **Wenn SDU voll ist, werden Daten an TCP übergeben**
- **TCP verpackt die Daten in Pakete, passend zur MTU**
- **➔ SDU wird meist über mehrere Pakete verteilt.**

Maximum Transport Unit (MTU)
1500 Bytes im Ethernet

- **Bsp.: SDU 8192 Bytes → 6 Pakete**
  - 5*1460 Bytes + 892 Bytes

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Datendurchsatz

<u>Rechenbeispiel:</u> Laden von 2 GB Daten

- **SDU 2048 → 2 Pakete / 2 kB ➔ 2097152 Pakete**
- **SDU 8192 → 6 Pakete / 8 kB ➔ 1572864 Pakete**
- **SDU 32768 → 23 Pakete / 32 kB ➔ 1507328 Pakete**
- **SDU 2097152 → 1437 Pakete / 2 MB ➔ 1494422 Pakete**
- **➔ Vom aktuellen Default zur max. SDU_SIZE können nur ca. 5% Pakete eingespart werden!**
  - Bei älteren Clients < 11.2 ist der Default aber noch 2048, hier liegt die Ersparnis bei bis zu 40%!

Spezialfall Jumbo Frames:

- **SDU 2097152 → 235 Pakete / 2 MB ➔ 240640 Pakete**

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Datendurchsatz

Warum setzt man die SDU nicht gleich auf das Maximum?

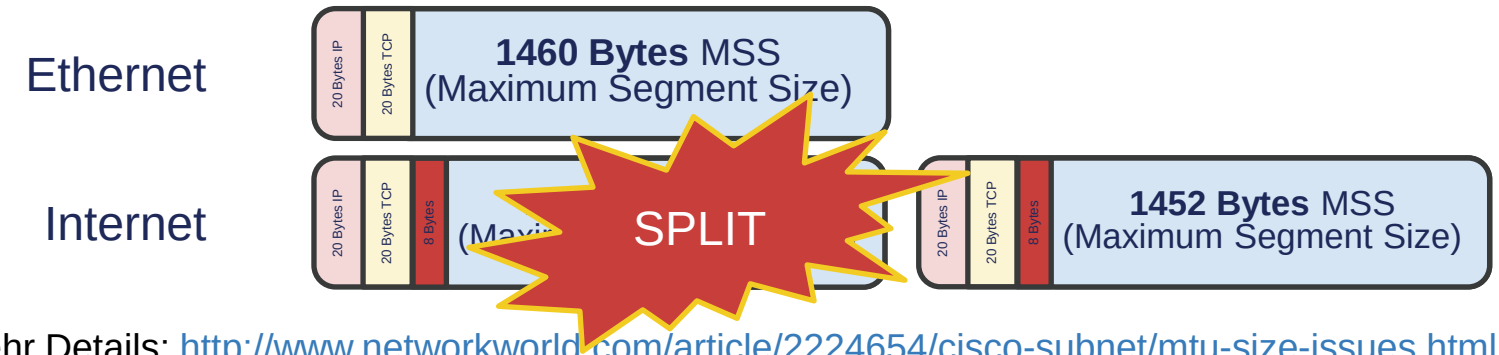
- **SDU wird pro Session angelegt und verbraucht RAM.**
- **➔ Bei vielen Sessions hoher RAM-Verbrauch**
  - 500 Sessions bei 2 MB SDU → 1 GB RAM
- **TCP Window Size des Betriebssystems limitiert**
  - Derzeitiger Standard: 64 kB
  - Bei Window Scaling bis 2 GB

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning:
Vorsicht im WAN – Packet Fragmentation!

Aufspaltung von Paketen beim Übergang in andere Netze, wenn Ziel-MSS < Quell-MSS, z.B. bei

- **VLAN (- 4 Byte pro Layer)**
- **VXLAN (Cloud-Umgebungen, - 50 Byte)**
- **PPPoE (z.B. DSL, - 8 Byte)**
- **GRE (RFC 2784, IPsec, VPN, - 24 Byte)**



Mehr Details: http://www.networkworld.com/article/2224654/cisco-subnet/mtu-size-issues.html

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Vorsicht im WAN – Packet Fragmentation!

- **Wenn man in der richtigen Verhandlungsposition ist, kann man mit dem Provider auch eine größere MTU aushandeln.**
- **Ansonsten wird die kleinste MTU entlang des Weges genommen („Path MTU“).**
- **In diesen Fällen ist es sinnvoll, die SDU_SIZE *kleiner* einzustellen.**
  - Platz für zusätzlichen Overhead verhindert Fragmentierung
- **Praxisbeispiel im IPsec WAN: SDU_SIZE=1360**

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: SDU_SIZE einstellen

- **SDU wird zwischen Client und Server ausgehandelt**
- **Die *kleinere* Einstellung gewinnt!**
  - Z.B. Client 10.2 gegen DB 11.2 → 2048 Byte SDU
- **Einstellung im Client**
  - sqlnet.ora: DEFAULT_SDU_SIZE=x
  - tnsnames.ora: (SDU=x)
- **Einstellung im Server**
  - sqlnet.ora: DEFAULT_SDU_SIZE=x
  - listener.ora: (SDU=x)
  - SPfile bei Shared Server:
    ```
    DISPATCHERS="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp))(SDU=32768))"
    ```
  - Es gibt Hinweise darauf, dass *sowohl* sqlnet.ora *als auch* listener.ora konfiguriert sein müssen (s. SAP on Oracle Blog).

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: SDU_SIZE einstellen

- **Beispiel tnsnames.ora**

```
sales.us.example.com=
(DESCRIPTION=
     (SDU=32767)
       (ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
     (CONNECT_DATA=
      (SERVICE_NAME=sales.us.example.com))
)
```

- **Beispiel listener.ora**

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SDU = 32767)
      (SID_NAME = orcl)
      (ORACLE_HOME = /u01/app/oracle/product/12.1)
    )
  )
```


© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: SDU_SIZE einstellen

SDU anpassen, wenn

- **Daten vom Server in fragmentierten Paketen**
  - „sqlnet more data to client“
- **WAN mit langen Laufzeiten**
- **Paketgröße immer gleich**
- **Sehr große Datenmengen übertragen werden.**

nicht anpassen, wenn

- **Die Anwendung noch nicht optimiert wurde**
  - Array Fetch Size!!!
- **Bandbreite und Latenzen kein Problem sind**
- **Nur kleine Datenmengen übertragen werden.**

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Weitere Parameter

- **SEND_BUF_SIZE und RECV_BUF_SIZE**
  - Parameter ändern Verhalten von TCP, nicht Oracle Net
  - Puffer zur Unterstützung eines kontinuierlichen Datenflusses
  - Anpassung nur im WAN wirklich interessant
  - Anpassen an die Gegebenheiten des Netzwerks (Bandbreite und Latenzen)
  - Sollten auf Client und Server identisch sein
  - Ermitteln der Bandbreite über Tracing oder Tools wie iperf
  - Weitere Infos im Oracle Net Administrator‘s Guide
    oder MOS Doc ID 1377929.1 (am Beispiel Streams)

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Verbindungsaufbau

- **SQLNET.OUTBOUND_CONNECT_TIMEOUT**
  - In sqlnet.ora auf dem Client zu setzen.
  - Innerhalb der eingestellten Zeit in Sekunden muss die Verbindung zu einer DB-Instanz hergestellt sein.
  - Andernfalls wird der Verbindungsversuch abgebrochen.
  - Sollte größer als Listener-Timeout sein.
  - Sollte kleiner als TCP-Timeout von 60 s sein.
  - Guter Startwert im LAN: 10s
- **CONNECT_TIMEOUT**
  - In tnsnames.ora auf dem Client zu setzen.
  - Übersteuert SQLNET.OUTBOUND_CONNECT_TIMEOUT.
  - Guter Startwert im LAN: 10s.
- **Werte im WAN ggf. größer wählen!**

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Verbindungsaufbau

- **Beispiel: tnsnames.ora für Failover-Konfiguration**
  - Im RAC: Besser SCAN-Listener verwenden

```
sales.us.example.com=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=10)(FAILOVER=on)
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales1-svr)(PORT=1521))
   (ADDRESS=(PROTOCOL=tcp)(HOST=sales2-svr)(PORT=1521)))
  (CONNECT_DATA=
   (SERVICE_NAME=sales.us.example.com)
   )
  )
```

Übrigens:
12c-Clients merken sich den letzten erfolgreichen Verbindungsversuch und priorisieren die Reihenfolge intern um.

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Schutz vor Überlastung

- **INBOUND_CONNECT_TIMEOUT_*listener_name***
  - In listener.ora zu setzen
  - Teilt dem Listener mit, wie schnell er Verbindungsversuche abbrechen soll, wenn sich der Client bis dahin seine Anfrage nicht beendet hat.
  - Abbrüche werden im Alert Log protokolliert (abschaltbar!)
- **SQLNET.INBOUND_CONNECT_TIMEOUT**
  - In sqlnet.ora auf dem Server zu setzen
  - Teilt der Datenbank mit, wie schnell sie Verbindungsversuche abbrechen soll, wenn sich der Client bis dahin nicht authentifiziert hat.
  - Sollte größer als Listener-Timeout sein.
  - Zeit für die Eingabe eines Passworts lassen!
- **Werte im WAN deutlich größer wählen!**

© OPITZ CONSULTING GmbH 2015

Oracle Net Tuning: Schutz vor Überlastung

- **SQLNET.EXPIRE_TIME= <*Minuten*>**
  - In sqlnet.ora auf dem Server zu setzen
  - Server sendet alle n Minuten ein Testpaket.
  - Erfolgt keine Antwort, wird die Session geschlossen und zurückgerollt.
  - Guter Startwert: 10
  - *Kein Ersatz für ordentliches Beenden durch Applikation!*
- **RATE_LIMIT und CONNECTION_RATE_*listener_name***
  - In listener.ora zu setzen
  - Begrenzt die Anzahl akzeptierter Verbindungsversuche pro Sekunde.
  - Hintergrund: Hochfahren von Connection Pools oder gezielte Attacken können die Datenbank lahmlegen.
  - Guter Startwert: 10

© OPITZ CONSULTING GmbH 2015

3 Praxisbeispiel: Troubleshooting

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Problemszenario

- **Batch-Job zum Datenabgleich per JDBC**
- **Baut kurz hintereinander bis zu 5 DB-Verbindungen auf**
- **Sporadische Timeouts beim Verbindungsaufbau**
  - Im Logfile der Applikation nachvollziehbar
  - Die meisten Verbindungen stehen innerhalb 1 s
  - Manche Verbindungen stehen erst nach 20 s
  - Abbrüche nach 60s TCP Timeout
- **DB-Server: Oracle Linux 5, Oracle EE 11.2.0.3**
- **JDBC: Oracle ojdbc6.jar, Version 11.2.0.3**

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Problemszenario

```
Alert Log

...

TNS-12535: TNS:operation timed out

 ns secondary err code: 12606

 nt main err code: 0

 nt secondary err code: 0

 nt OS err code: 0

 Client address:
(ADDRESS=(PROTOCOL=tcp)(HOST=10.20.30.40)(PORT=33600))

WARNING: inbound connection timed out (ORA-3136)
```


© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Vorbemerkungen

Der Aufbau einer Oracle-Session läuft wie folgt:

1. **Client schickt einen Request an den Oracle-Listener**
2. **Listener prüft, ob er eine DB-Instanz bzw. einen Service für den Request kennt**
3. **Listener benachrichtigt pmon-Hintergrundprozeß der Datenbankinstanz**
4. **pmon startet einen Serverprozess auf dem Datenbankserver (sichtbarer Prozeß auf OS-Ebene)**
5. **Der Serverprozess allokiert einen freien IP-Port auf dem Datenbankserver im Bereich von üblicherweise 9000-64000**

© OPITZ CONSULTING GmbH 2015

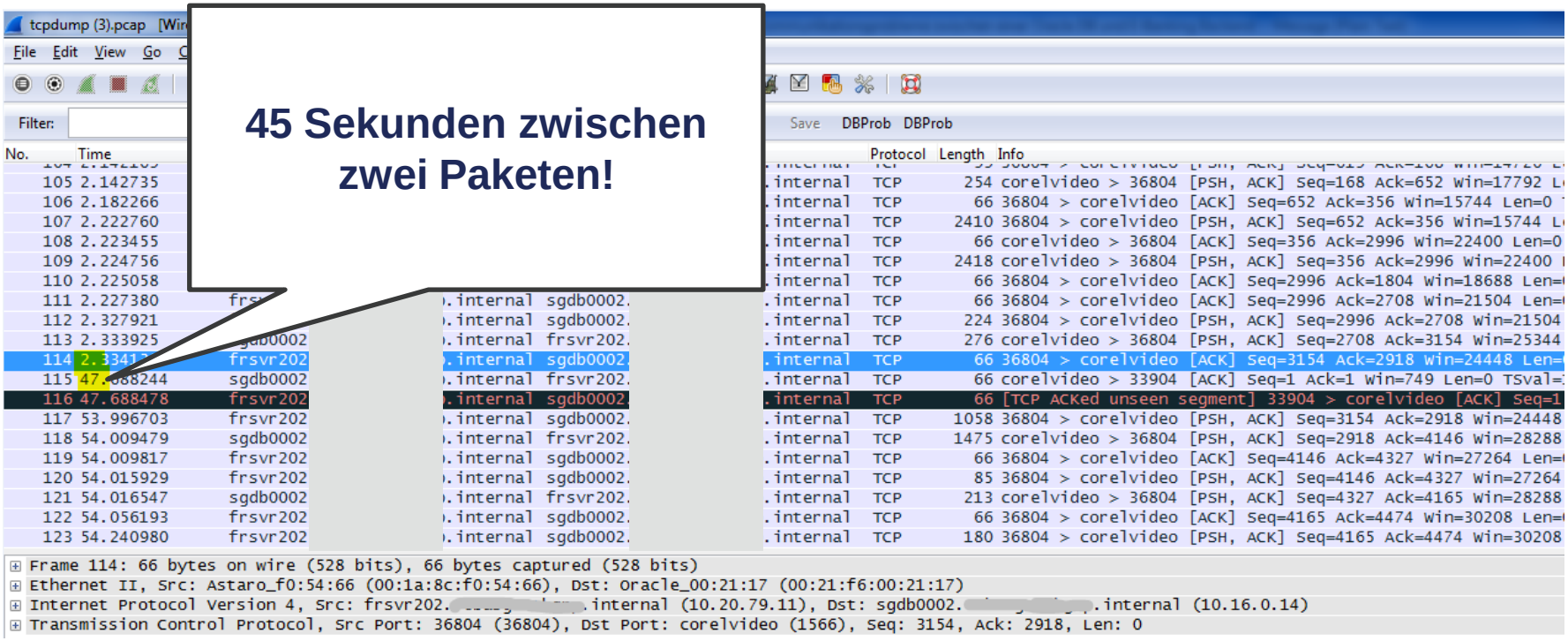
Oracle Net Troubleshooting: Vorbemerkungen

Aufbau einer Oracle-Session (Forts.)

6. **pmon schickt die Information über den gestarteten Serverprozeß mit OS-Port an den Listener**
7. **Listener schickt die Information an den Client**
   - Ab jetzt hat der Listener mit der Datenbankverbindung nichts mehr zu tun!
8. **Client startet eine neue IP-Verbindung DIREKT zum Serverprozeß mit dem vom Listener ausgehandelten Port.**
9. **Serverprozess fordert den Client zur Authentifizierung auf.**

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Untersuchung mit Wireshark



© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Differenzialdiagnose

- **Wie beweisen, dass das Problem nicht beim DB-Server liegt?**
  ➔ Mit Oracle Net Tracing
- **Tracing ist am Listener und am Serverprozess möglich**
  - Listener hat hier gut funktioniert (sonst gäbe es keinen Eintrag im Alert Log)
  - Also Server-Trace
- **Server-Trace in der sqlnet.ora einschalten:**
  `TRACE_LEVEL_SERVER=16`
  - ACHTUNG: ab jetzt Tracing für jede Session!
- **Mit dem Trace Assistant lesbare Form des Traces erzeugen**
  `trcasst -oc meintracefile.trc`

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Differenzialdiagnose

```
...
<--- Received 158 bytes - Data packet timestamp=04-NOV-2013
10:50:46:024

Start of user function (TTIFUN)

Get the session

---> Send 210 by                                    4-NOV-2013
10:50:46:031

Return opi para

<--- Received 9[illegible] bytes - Data packet timestamp=04-NOV-2013
10:51:13:023

Start of user function (TTIFUN)

Generic authentication call (OAUTH)

...
```

Hier vergingen 27 Sekunden!
(Der Rest lief dann zügig durch)

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Heiße Spur

- Nun wurde nochmal mit Wireshark nach den oben gesendeten 210 Bytes gesucht
- Diese wurden ohne besondere Verzögerung an den Client gesendet.
- ➔ Die Verzögerung findet auf dem Client statt!

- Das war nun genug Information für einen SR bei Oracle.

© OPITZ CONSULTING GmbH 2015

Oracle Net Troubleshooting: Ergebnis

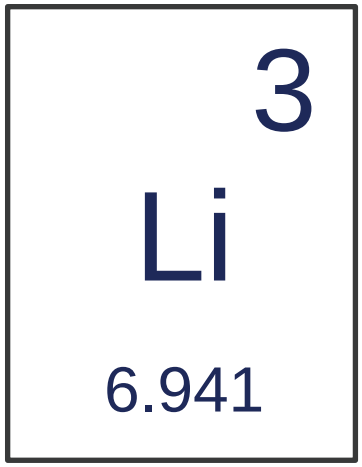
- „Schuld“ liegt beim Zufallszahlengenerator des Client-Betriebssystems.
- Ursache: schnell hintereinander aufgebaute Verbindungen
- Benötigen Zufallszahlen aus einem Pool
- Wenn Pool leer, dann bis zu 60 s Wartezeit auf neue Zahlen.

Beste Lösung:

- Applikation ändern, so dass nicht so viele Verbindungen gleichzeitig geöffnet werden.

© OPITZ CONSULTING GmbH 2015

Links + Literatur



© OPITZ CONSULTING GmbH 2015

Links + Literatur

- Oracle 12c Net Services Reference
- Oracle 12c Net Services Administrator's Guide
- Information Center: Oracle Net Services (Doc ID 1381244.2)
- Oracle Net Performance Tuning (Doc ID 67983.1)
- "Undocumented or Lesser Known SQL*Net/Net8/Net8i Features & Parameters" (Doc ID 39357.1)
- Stefan Koehler, SAP on Oracle Blog: „[Oracle] SQL*Net researching - Setting Session Data Unit (SDU) size and how it can go wrong"
- Setting Parameters for Scan and Node Listeners on RAC, Queuesize, SDU, Ports, etc (Doc ID 1292915.1)

© OPITZ CONSULTING GmbH 2015

Links + Literatur

- Wikipedia: Maximum Transmission Unit
- Scott Hogg: MTU Size Issues
- Richard Hay: IP Packet Overhead
- Cisco: Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPSEC
- How To Control the Amount of Connections Handled by the TNS Listener (Doc ID 443744.1)
- Dead Connection Detection (DCD) Explained (Doc ID 151972.1)
- Setting SEND_BUF_SIZE and RECV_BUF_SIZE (Doc ID 260984.1)

© OPITZ CONSULTING GmbH 2015

Kontaktdaten

Uwe Küchler

Managing Consultant

OPITZ CONSULTING Deutschland GmbH
uwe.kuechler@opitz-consulting.com
Telefon +49 6172 66260 – 0
Mobil +49 173 727 91 43

youtube.com/opitzconsulting

@OC_WIRE

slideshare.net/opitzconsulting

xing.com/net/opitzconsulting

© OPITZ CONSULTING GmbH 2015