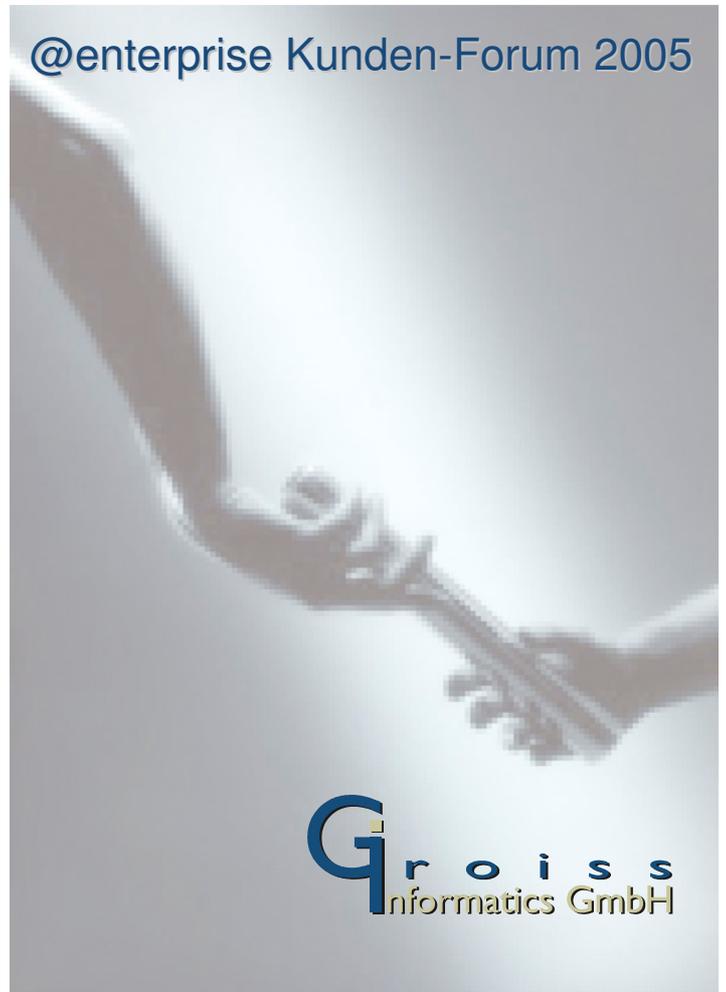




# Performance Tuning mit @enterprise

Herbert Groiss

@enterprise Kunden-Forum 2005



## Inhalt

- ◆ Datenbank
- ◆ RMI
- ◆ JAVA
- ◆ API
- ◆ HTTP
- ◆ Konfiguration
- ◆ Analyse



# Datenbank

- ◆ Anzahl der Statements reduzieren
- ◆ Prepared Statements verwenden
  - API: `store.list(x.class,"x=? and y=?", "orderAttr", new Object[]{a,b});`
- ◆ Index auf Attribute, auf die häufig selektiert werden
- ◆ BulkQuery für "in" statements
- ◆ Aufteilung auf die Platten
  - mehrere kleine Platten besser als eine große
  - Trennen von ep log, Datenbank logs, Datenbank Daten Files, Page Files

# BulkQuery Beispiel

```
List news = ServiceLocator.getStore().list(Note3.class, "oid in ...",
    "avwchangedat");
for (Iterator it = news.iterator(); it.hasNext();) {
    DMSObject o = (DMSObject)it.next();
    boolean b = store.hasRows(
        "select seen from avw_seenobject2 where userid=? and seen=?",
        new Object[]{u,o});
    ...
}
```

**100 items, 520 Millisekunden**

```
List news = ServiceLocator.getStore().list(Note3.class, "oid in ...",
    "avwchangedat");
BulkQuery bQ = new BulkQuery(news);
String query = "select seen from avw_seenobject2 where userid="+u.getId()+
    " and seen in (?)";
Map qres = bQ.execute(query,1);
for (Iterator it=news.iterator(); it.hasNext();) {
    DMSObject o = (DMSObject)it.next();
    boolean b = qres.containsKey(Long.toString(o.getId()));
    ...
}
```

**60 Millisekunden**

# Oracle Spezifika

- ◆ Oracle Parameter Einstellungen siehe Installationshandbuch
  - oder Oracle Doku.
- ◆ Cost-based Optimizer
  - Statistiken erzeugen:  
`execute dbms_utility.analyze_schema('USER','COMPUTE');`
  - Histogramme bei Schiefverteilung  
z.B. viele fertige, wenige aktive Prozesse  
`execute dbms_stats.gather_table_stats('USER','avw_stepinstance', method_opt => 'for columns size 10 status');`  
Querydauer in einem Fall auf 1/40 reduziert

# Oracle Spezifika

- ◆ Partitionierung der Tabellen  
z.B. StepInstance nach status  

```
create table avw_StepInstance ( ... )
PARTITION BY LIST (status) (
  PARTITION avw_si_curr VALUES(0,1,5),
  PARTITION avw_si_old VALUES(2,4,7,8,9),
  PARTITION avw_si_paper VALUES(3,10),
  PARTITION avw_si_pfinp VALUES(5),
  PARTITION avw_si_defa VALUES(DEFAULT) )
ENABLE ROW MOVEMENT;
```
- ◆ Vergleiche mit NULL verwenden keinen Index
  - z.B.: `finished is null`
- ◆ oci Treiber ist schneller als der thin Treiber

# RMI

---

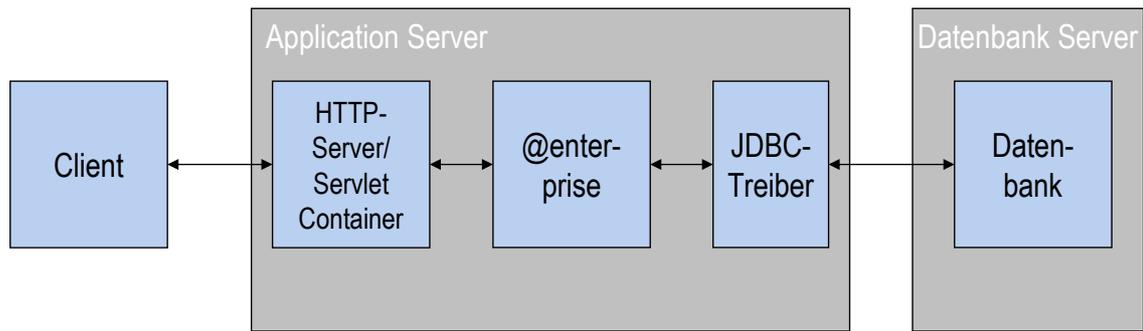
- ◆ wenig roundtrips
  - eigene Services schreiben
- ◆ kleine Datenmengen verschicken
  - oids statt Objekte verschicken, Felder transient machen
  - Caches am Client aufbauen
- ◆ kurze Transaktionen
  - um Ressourcen freizugeben und Locks zu vermeiden
  - am besten: ein Request = eine Transaktion
- ◆ finalize für Remote Objekte
  - um bei client Absturz Ressourcen freizugeben
  - Unreferenced implementieren

# Java

---

- ◆ StringBuffer verwenden, wenn Strings zusammengebaut werden
- ◆ synchronized nur wenn nötig
  - Vector und Hashtable sind synchronized
- ◆ Objekt erzeugen ist teuer
- ◆ Threads aus Pool nehmen
- ◆ buffered Streams verwenden
- ◆ java Optionen beachten
  - Xmx maximum memory, 64MB is default !

# Hardwarekonfiguration



- zwei schnelle Netzwerkkarten
- viel Memory
- ev. mehrere Application Server

- schnelles Netzwerk
- viel Memory
- viele Platten

# API

- ◆ Worklist Cache verwenden bei Abfrage auf Work-Items
  - Forms in Worklist, über XML Konfiguration hinzufügen
- ◆ Thread-Cache: Objekte werden aus der Datenbank pro Transaktion nur einmal gelesen.
  - `store.get(class, oid)` sieht zuerst im cache nach
  - `store.get(class, "oid="+oid)` macht das select und prüft danach, ob das Objekt bereits im Cache ist
- ◆ File IO: Falls Files mehrmals gelesen werden, FileCache verwenden
- ◆ Dinge in der Session speichern
  - nicht in Cookies
- ◆ Applikationsklasse mit static Variablen für Tasks, FormTypes, etc.

# HTTP + HTML

---

- ◆ Größe der HTML Seiten
  - Javascript und Styles auslagern
- ◆ Caching
  - `lastModified` Header mitschicken, und auf `If-Modified-Since` reagieren
  - ev. `Expires` Header mitschicken, bringt beste Performance
  - Browsereinstellungen prüfen, nicht immer alle gifs neu laden
  - roundtrips vermeiden

# Konfiguration

---

- ◆ *Klassen neu laden* ausschalten
- ◆ log level im Betrieb runtersetzen
  - Beachte: `Settings.log(foo(x), 3)`  
foo wird unabhängig vom log-level ausgewertet
  - Besser: `if (Settings.getLogger().getLevel() >2)`  
`Settings.log(foo(x), 3)`
- ◆ Tuning Optionen
  - Referenzrollen ignorieren, Hierarchische Rollen ignorieren, Persönliche Vertretungen ignorieren, Rollenvertretungen ignorieren
  - haben kaum Auswirkungen, wenn Worklist cache eingeschaltet, aber auf Rechteprüfung
- ◆ Suchen begrenzen
  - max. Tabellengröße, gleichzeitige Suchen

# Analyse

- ◆ Prüfen, ob ausreichend Ressourcen vorhanden sind.
- ◆ Fehler log
- ◆ Datenbankverbindungen und Threads: hängende oder lang dauernde Aktionen
- ◆ Statement Statistik: häufige und lang dauernde DB-Statements finden

Servermonitor			
@enterprise Server: herb1, Systemstatus am 20.09.2005 22:48			
Datenbankverbindungen:	1/2-3	[Detail] [Histogramm]	
HTTP Threads:	1/25	[Detail]	
Benutzer-Sessions:	2	[Detail]	
Speicher:	19,46/22,18 MB		
Fehler:	481	[Detail ...]	letzte: 20.09.2005 22:48
Läuft seit:	19.09.2005 16:09		
Anfragen bisher:			
HTTP:	3230		
RMI:	0		
DB-Anweisungen:	4631	[Detail]	
Aktive Schritte: 13			
[Logfile ansehen ...] [Konfiguration ansehen ...] [Server Info] [Client Info] [Http-Sessions]			
Aktualisierungsintervall: <input type="text"/> Sekunden			

# Analyse(2)

- ◆ Log Viewer

#	Le...	Thread	Timestamp	Message
71	2	EventDis...	2005-09-22 10:14:29.658	select ur.oid,ur.dept,ur."ROLE",ur.userid,ur."ACTIVE",ur.locked,ur.transactionid,ur.aci from aww_userro...
72	2	EventDis...	2005-09-22 10:14:29.668	Select * from aww_UserRole t0 where "ACTIVE"=1 order by userid
73	2	EventDis...	2005-09-22 10:14:29.688	Select * from aww_UserSubstitute t0 where iscurrent=1 order by substitute
74	2	EventDis...	2005-09-22 10:14:29.698	Select * from aww_UserSubstitute t0 where iscurrent=1 and reptime=1 order by substitute
75	2	EventDis...	2005-09-22 10:14:29.698	Select * from aww_RoleSubstitute t0 where iscurrent=1 order by substitute
76	2	EventDis...	2005-09-22 10:14:29.708	select so.oid,so.userid,so.seen from aww_seenobject so, aww_stepinstance si where so.seen=si.oid ...
77	2	EventDis...	2005-09-22 10:14:29.718	Select * from aww_ProcDefinition t0 where "ACTIVE"=1
78	2	EventDis...	2005-09-22 10:14:29.988	select * from aww_stepinstance where (t0.status = 1 or t0.finished is null and t0.status in (0,5) ) and...
79	1	EventDis...	2005-09-22 10:14:30.509	WLCache startup: handling 13 queued events
80	1	EventDis...	2005-09-22 10:14:30.509	WLCache startup: completed handling of 13 queued events
81	1	EventDis...	2005-09-22 10:14:30.509	WLCache startup: Complete
82	1	http-10	2005-09-22 10:14:37.108	10.205.112.21 herb GET /wtf/servlet.methodtest.Test63.showNews2
83	2	http-10	2005-09-22 10:14:37.259	Select * from aww_StandardFldr where oid=279
84	2	http-10	2005-09-22 10:14:37.269	select t1.oid,t1.folder,t1.folder_class,t1.item,t1.item_class,t1.itemServer from aww_dmsfldritemrel t1, av...
85	2	http-10	2005-09-22 10:14:37.279	Select * from aww_StandardFldr where oid=4295275006
86	2	http-10	2005-09-22 10:14:37.289	Select * from aww_note3 t0 where oid in (select item from aww_dmsfldritemrel where folder=?) order ...
87	2	http-10	2005-09-22 10:14:37.369	select seen from aww_seenobject2 where userid=? and seen=?
88	2	http-10	2005-09-22 10:14:37.399	select seen from aww_seenobject2 where userid=? and seen=?
89	2	http-10	2005-09-22 10:14:37.399	select seen from aww_seenobject2 where userid=? and seen=?
90	2	http-10	2005-09-22 10:14:37.409	select seen from aww_seenobject2 where userid=? and seen=?
91	2	http-10	2005-09-22 10:14:37.409	select seen from aww_seenobject2 where userid=? and seen=?

Detail:  
Select \* from aww\_note3 t0 where oid in (select item from aww\_dmsfldritemrel where folder=?) order by awwchangedat

# In Planung

---

- ◆ Object Cache threadübergreifend
- ◆ Http Verkehr zippen, reduziert Netzverkehr
- ◆ System Monitoring
  - Systemzustand über JMX abfragbar

