Entwicklung einer Informix-Administrationsdatenbank mit ER*win*

Ausgangslage

- Ein oder mehrere Informix-Datenbankserver
- Mehrere Datenbanken
- Sehr viele Tabellen

Problemstellung

- Fehlerprävention statt Fehlerbehebung
- Stabile Performance
- Datenbankmonitoring

Datenbankkennzahlen

Plattenspeicherreserven

- Extent-Überwachung
- DBSpace-Überwachung

I/O-Leistung

- Cache-Rate
- Read-Ahead
- Foreground-Writes

Datenbankmonitoring und Systemtuning

- Transaktionslängen & LOG-Space-Bedarf
- Wait-Statistics
- SQL-Optimizer

Datenbankkennzahlen

Plattenspeicherreserven

- Extent-Überwachung
- DBSpace-Überwachung

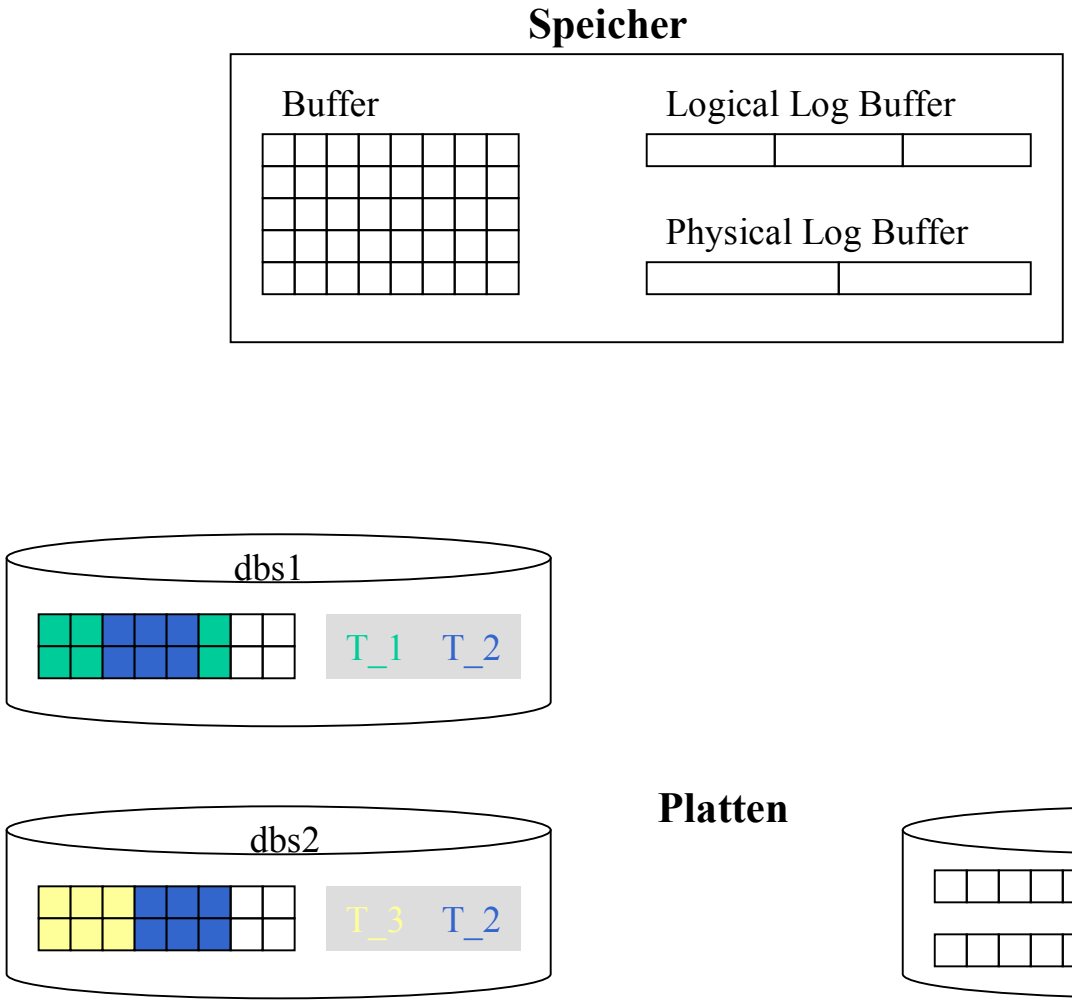
I/O-Leistung

- Cache-Rate
- Read-Ahead
- Foreground-Writes

Datenbankmonitoring und Systemtuning

- Transaktionslängen & LOG-Space-Bedarf
- Wait-Statistics
- SQL-Optimizer

Informix-Architektur



Die logischen Protokolle dienen der Transaktions-protokollierung

Die physikalischen Protokolle speichern die Before-Images der veränderten Pages im Buffer, welche noch nicht auf Platte gespeichert wurden.

Datenbankkennzahlen

Plattenspeicherreserven

- Extent-Überwachung
- DBSpace-Überwachung

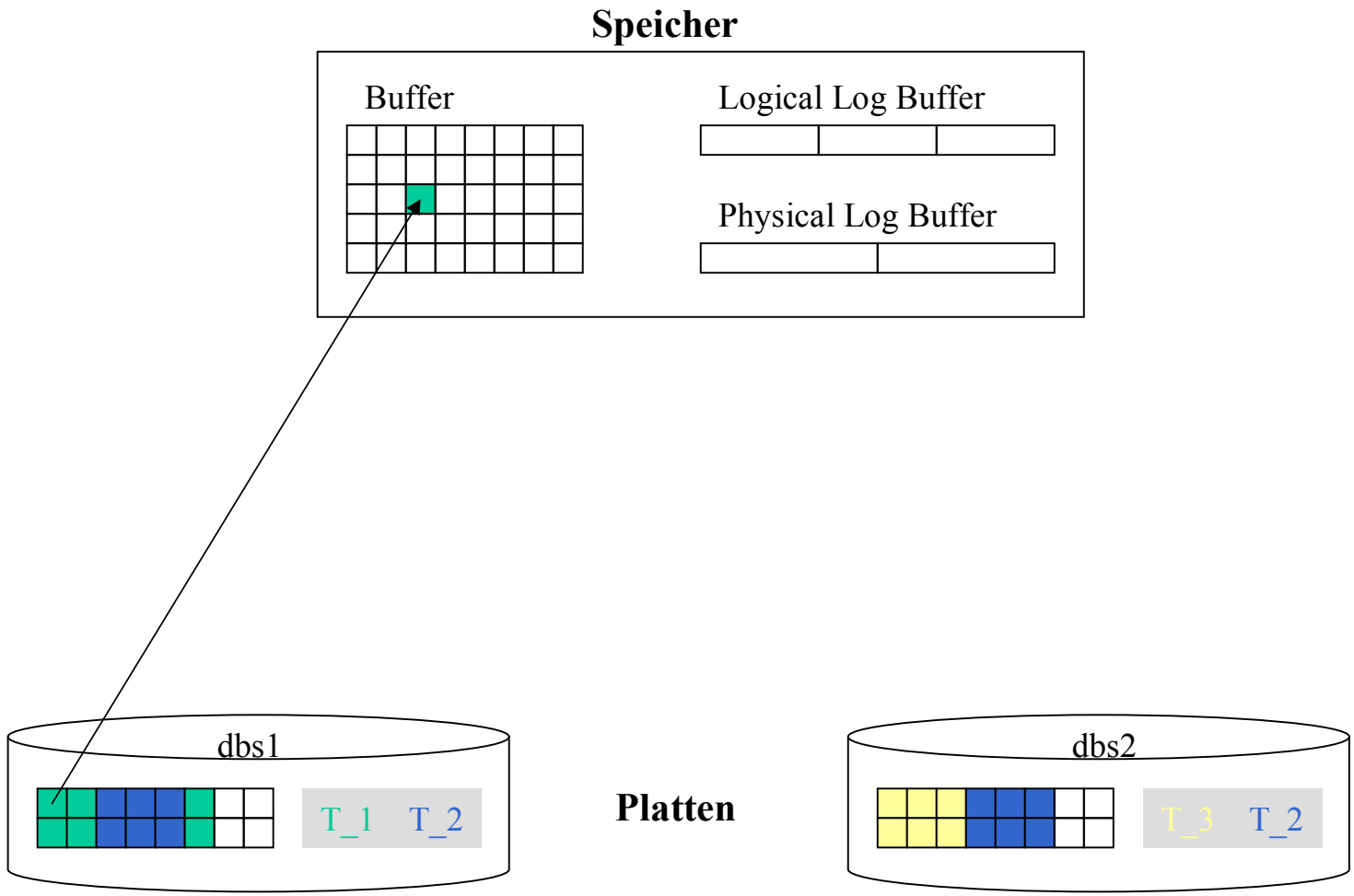
I/O-Leistung

- Cache-Rate
- Read-Ahead
- Foreground-Writes

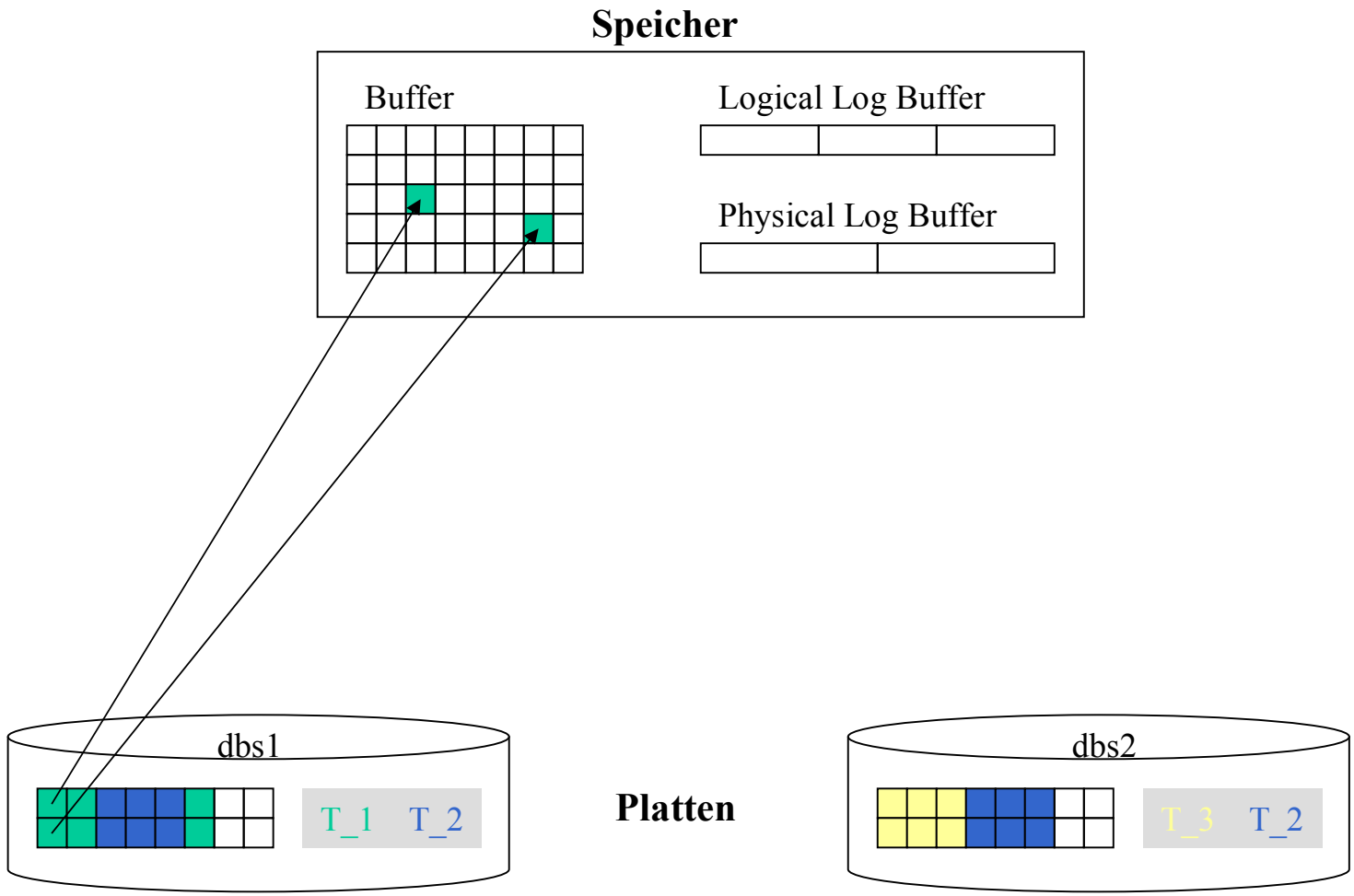
Datenbankmonitoring und Systemtuning

- Transaktionslängen & LOG-Space-Bedarf
- Wait-Statistics
- SQL-Optimizer

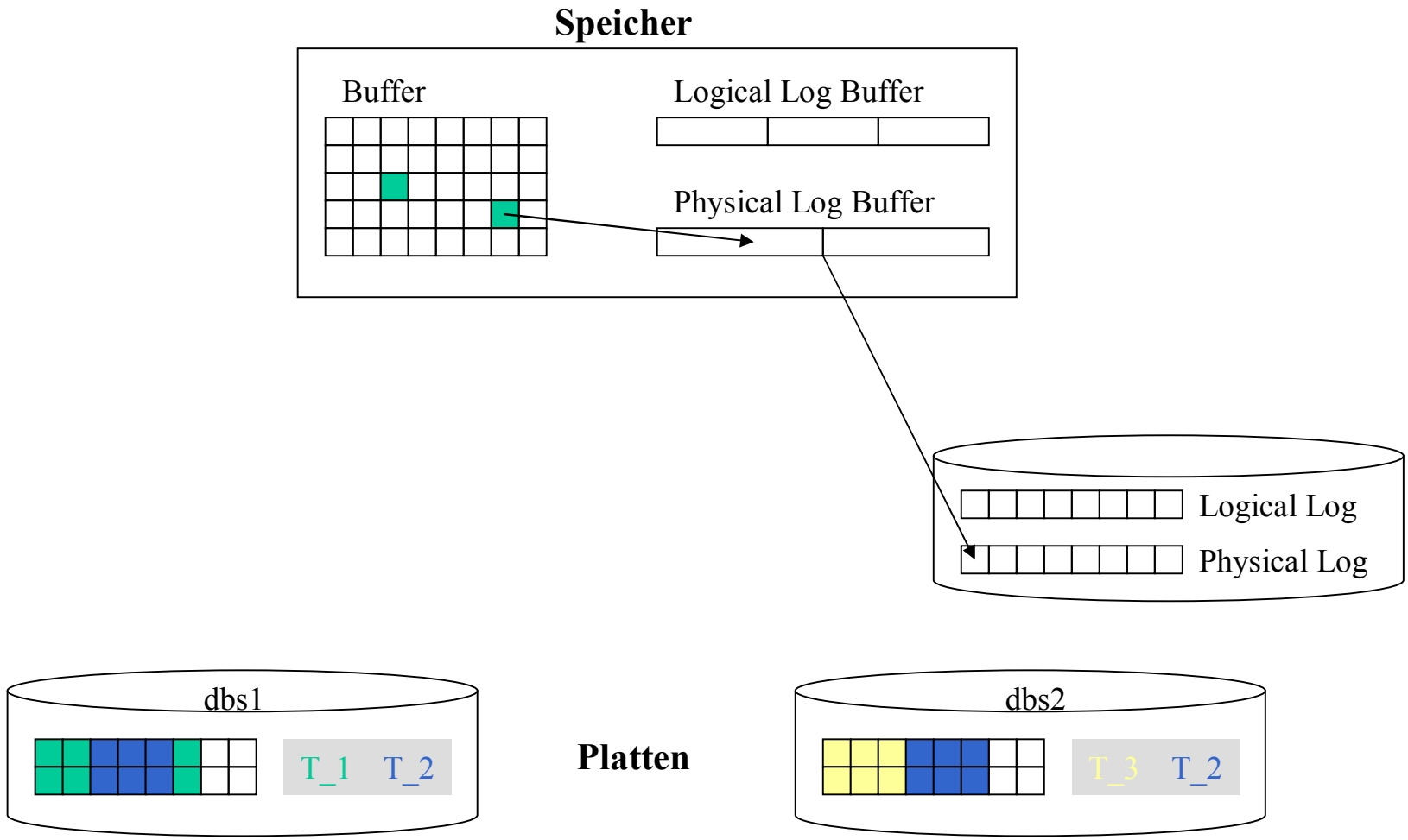
Daten-Caching



Read-Ahead

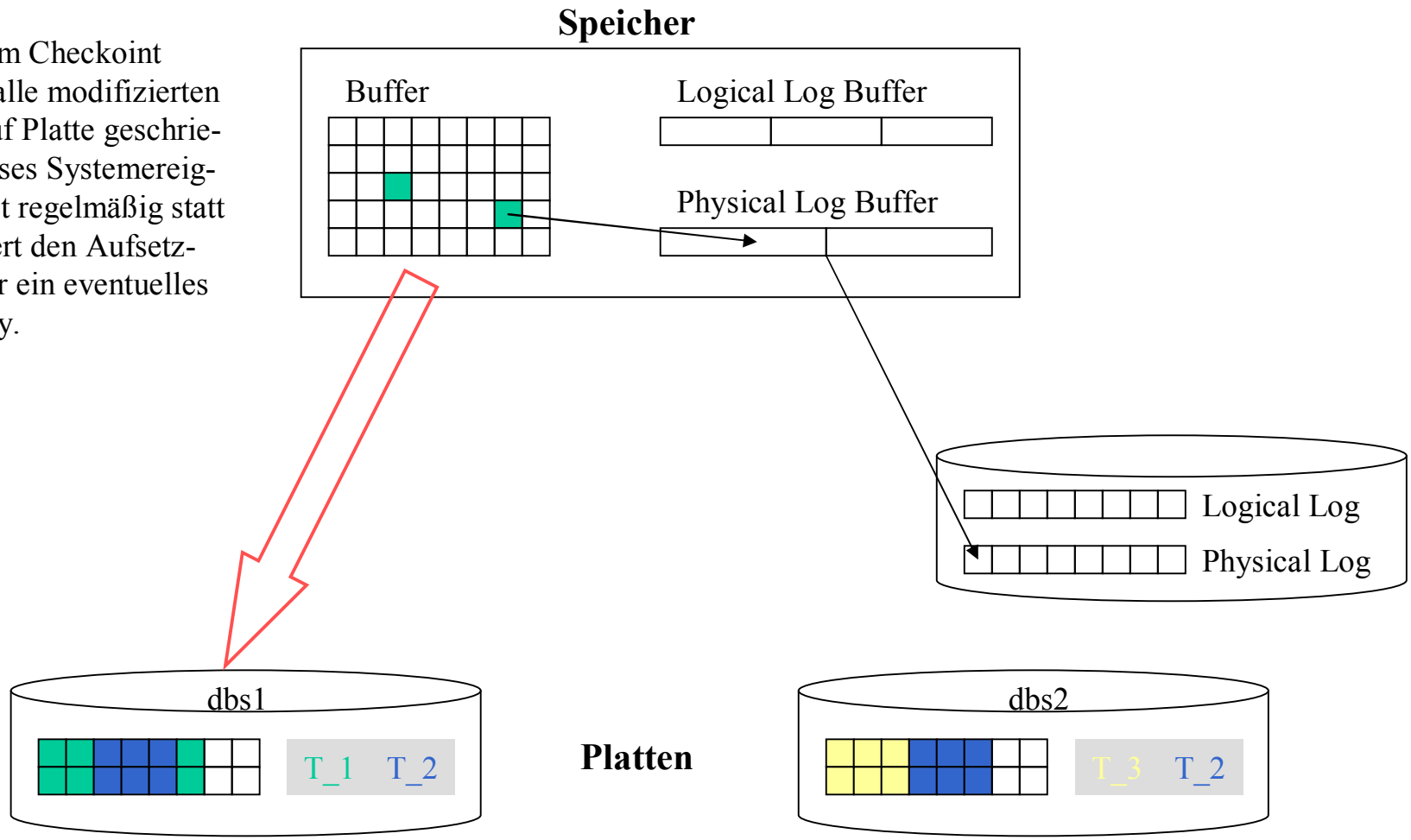

Speicher
Buffer
Logical Log Buffer
Physical Log Buffer
dbs1
T_1 T_2
Platten
dbs2
T_3 T_2

Dirty Pages / Flushing



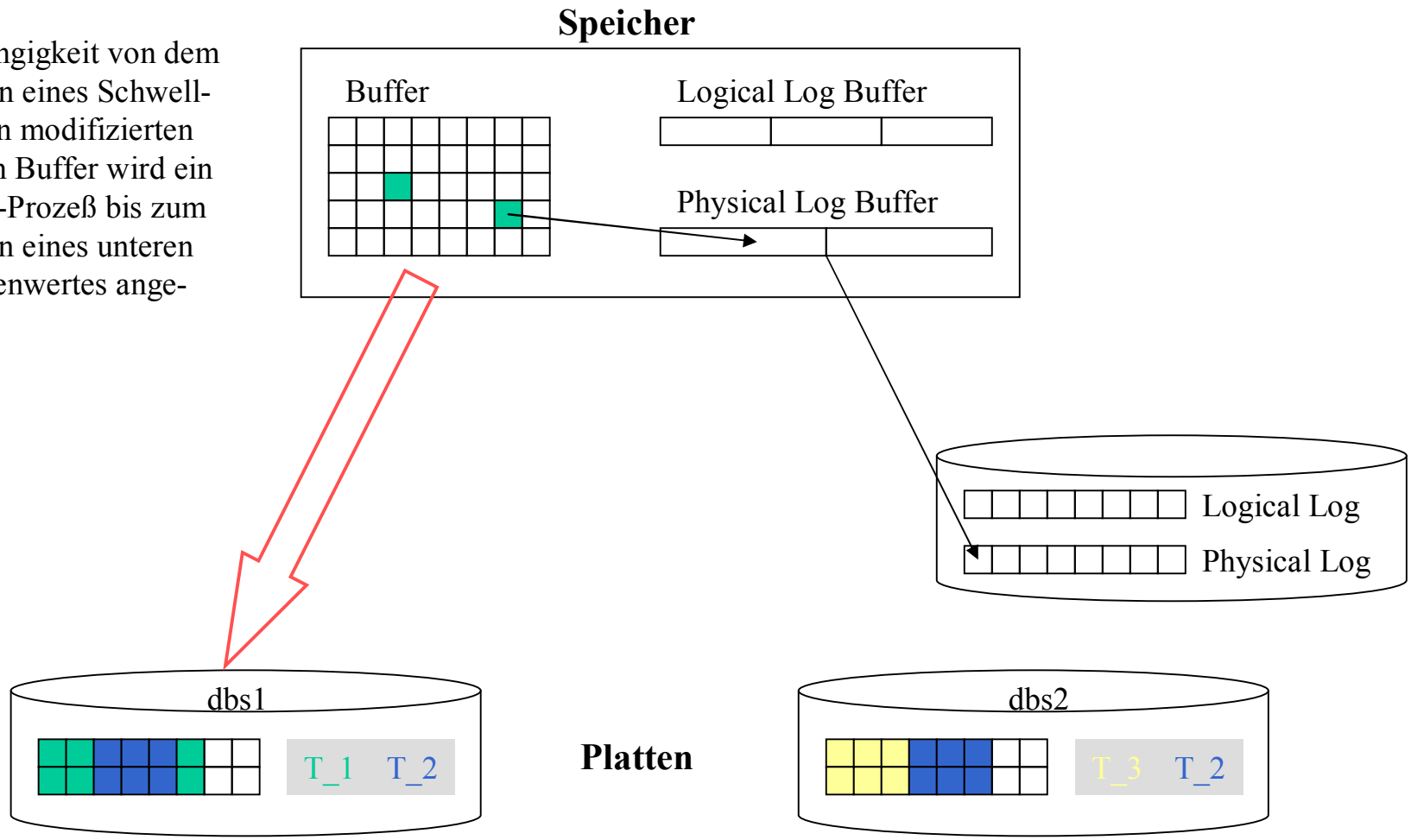
Flushing: Checkpoint-Writes

Bei einem Checkoint werden alle modifizierten Pages auf Platte geschrieben. Dieses Systemereignis findet regelmäßig statt und liefert den Aufsetzpunkt für ein eventuelles Recovery.



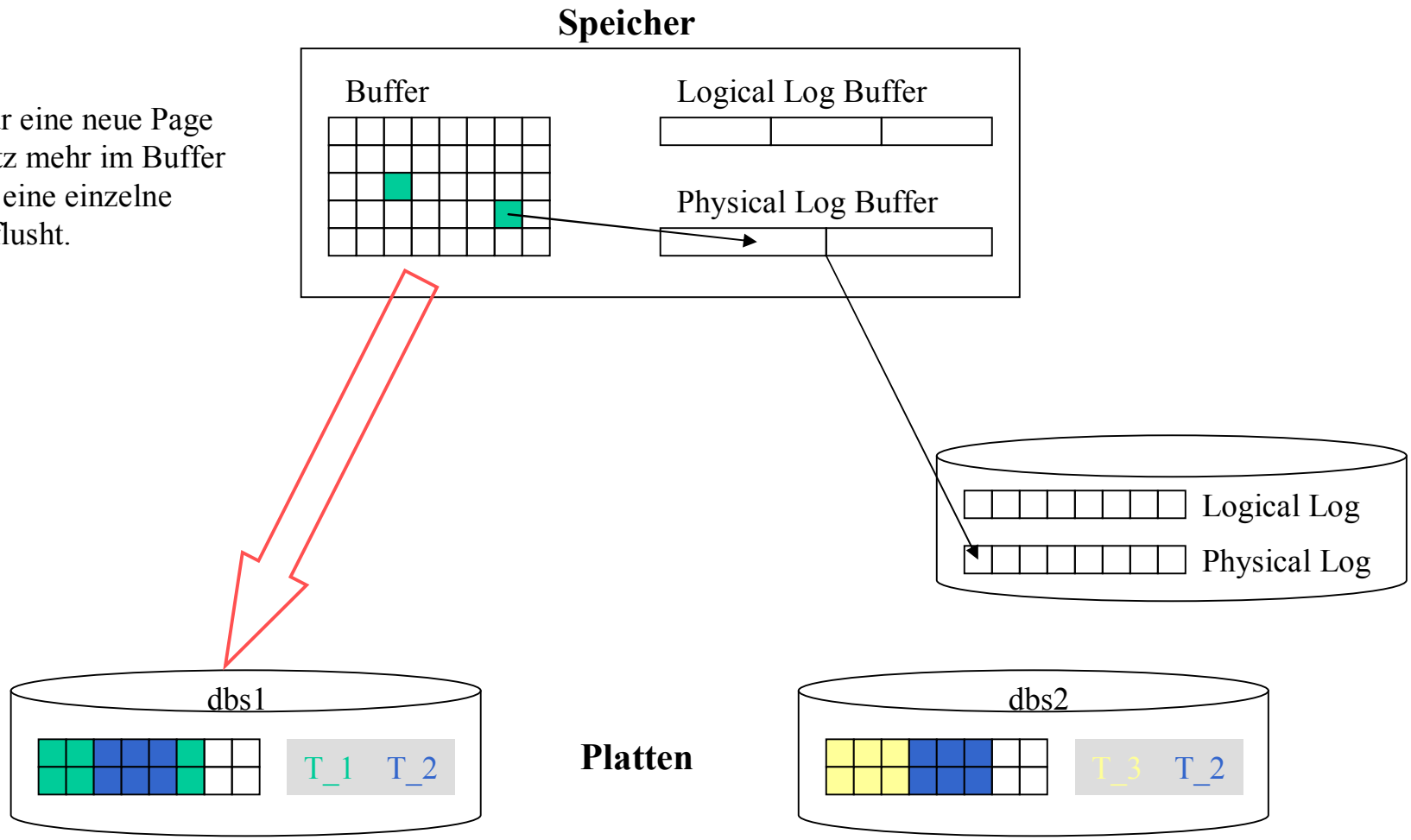
Flushing: LRU-Writes

In Abhängigkeit von dem Erreichen eines Schwellwertes an modifizierten Pages im Buffer wird ein Ceaning-Prozeß bis zum Erreichen eines unteren Schwellenwertes angestoßen.



Flushing: Foreground-Writes

Wenn für eine neue Page kein Platz mehr im Buffer ist, wird eine einzelne Page geflusht.



Datenbankkennzahlen

Plattenspeicherreserven

- Extent-Überwachung
- DBSpace-Überwachung

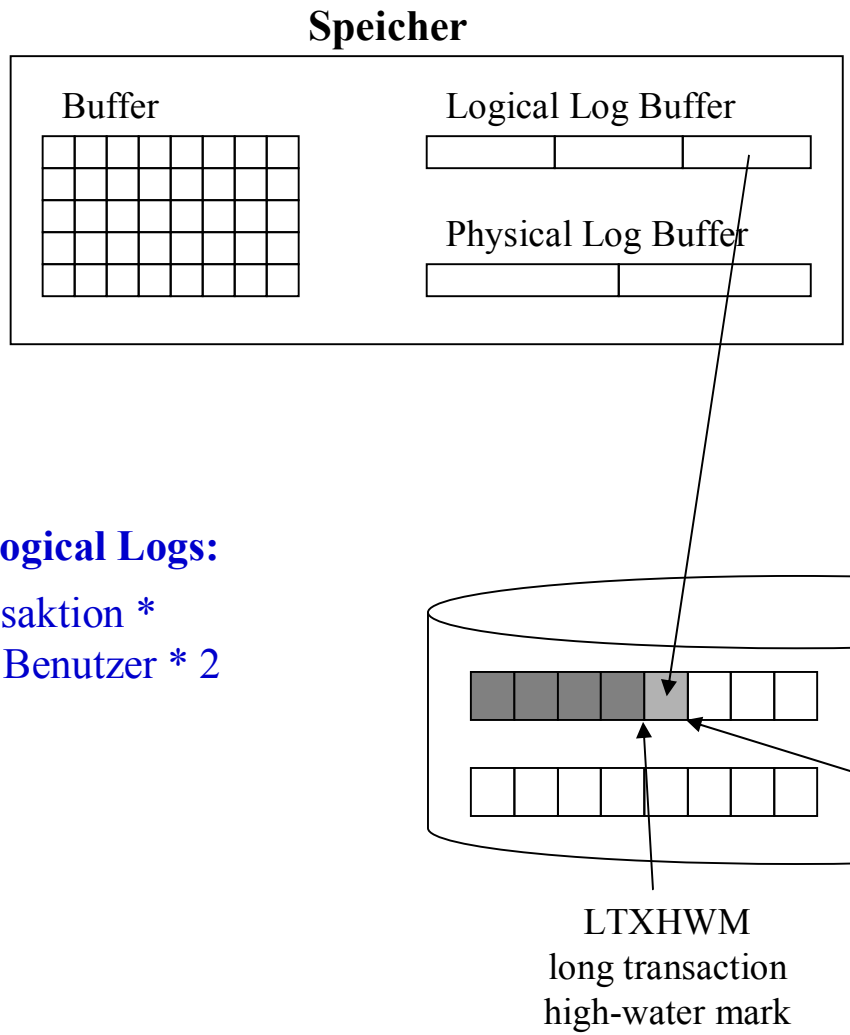
I/O-Leistung

- Cache-Rate
- Read-Ahead
- Foreground-Writes

Datenbankmonitoring und Systemtuning

- Transaktionslängen & LOG-Space-Bedarf
- Wait-Statistics
- SQL-Optimizer

Transaktionen & Logical Log



Beim Erreichen der Hochwassermarke werden Lange Transaktionen zurückgefahren.
Mit Erreichen der exklusiven Hochwassermarke bekommen nur noch commit- und rollback-threads einen Zugang zum Log.

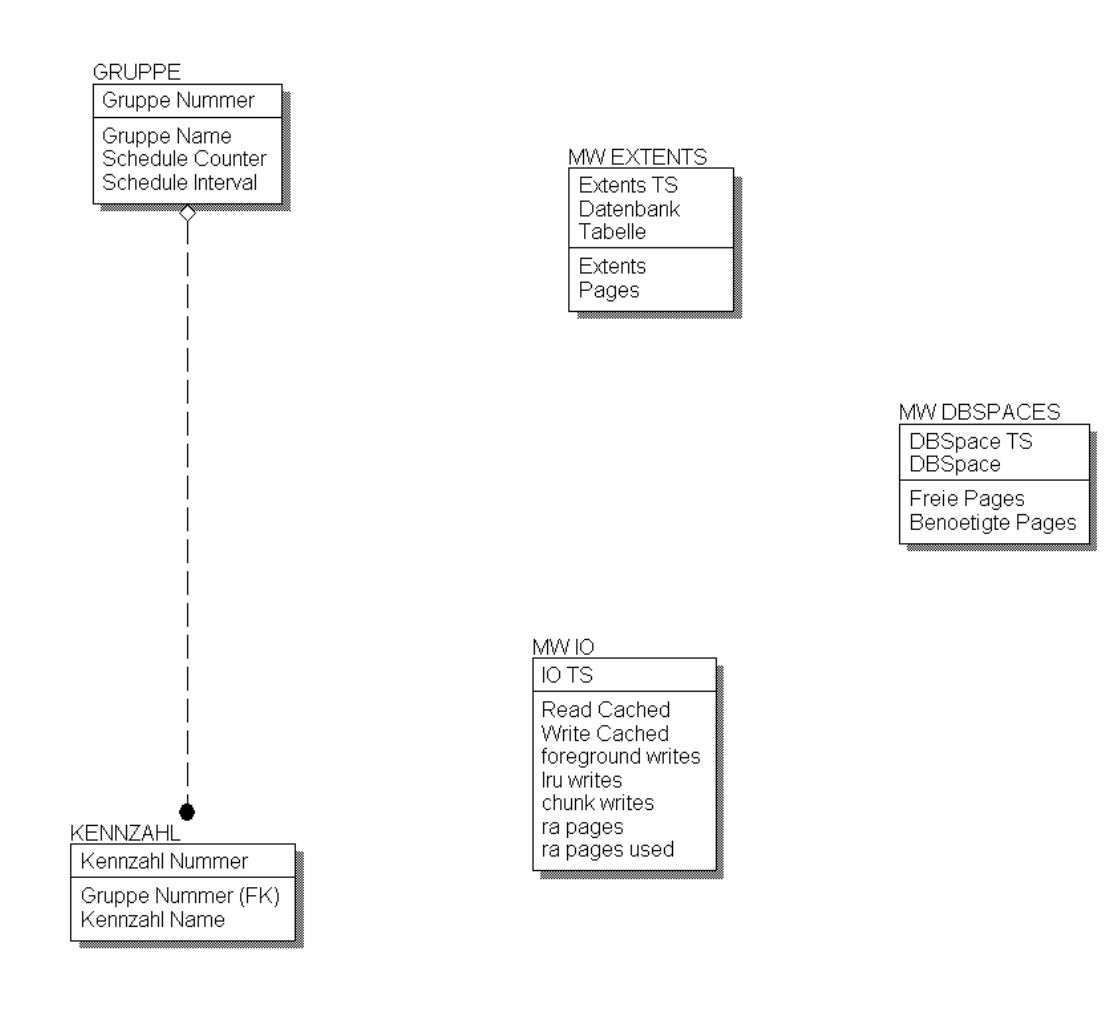
Größe des Logical Logs:

Längste Transaktion *
max. Anzahl Benutzer * 2

Idee der Admin-Datenbank

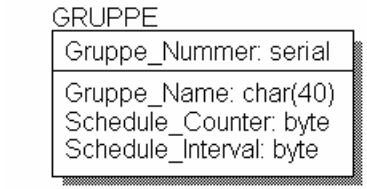
- **Die Informationen, welche zur Ableitung der Datenbank-Kennzahlen benötigt werden, sind über das Sys Master Interface erhältlich.**
- **Diese Informationen können automatisiert regelmäßig abgefragt werden.**
- **Verschiedene Kennzahlen bedingen auch verschiedene Abfrageintervalle.**
- **Die Verwaltung der Abfrageintervalle kann flexibel über eine Datenbank gelöst werden.**
- **Alle Meßergebnisse werden in einer zentralen Meßdatenbank abgelegt.**
- **Basierend auf diesen Meßergebnissen können Schwellwert-Alarmgeber und andere Mechanismen zur Unterstützung der Datenbankadministration erstellt werden.**

Datenmodell der Admin-Datenbank

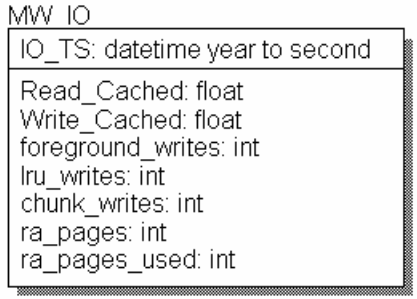


Stored Procedure zur Meßwerterfassung

```
create procedure admin_watch()
update GRUPPE set Schedule_Counter=mod(Schedule_Counter,Schedule_Interval)+1 ;
end procedure ;
```



Beispiel

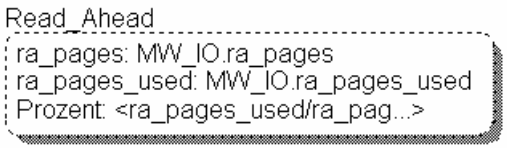



```
create trigger todowatch UPDATE OF
                    Schedule_Counter on GRUPPE
  REFERENCING OLD AS pre NEW AS post
  FOR EACH ROW
-- ERwin Builtin Thu Oct 16 18:56:09 1997
-- default body for todowatch
when ( (post.Gruppe_Nummer=1) and
    (post.Schedule_Counter=post.Schedule_Interval) )
    ( call mess1(), let post.Schedule_Counter=0 )
;
```

```
create procedure mess1()
define ts datetime year to second ;
define a like Read_Cached ;
define b like Write_Cached ;
define c like foreground_writes ;
define d like lru_writes ;
define e like chunk_writes ;
define f like ra_pages ;
define g like ra_pages_used ;
define t1,t2,t3,t4 int ;
let ts = today ;
let t1 = select value from sysmaster:sysprofile where name='dskreads' ;
let t2 = select value from sysmaster:sysprofile where name='bufreads' ;
let t3 = select value from sysmaster:sysprofile where name='dskwrites' ;
let t4 = select value from sysmaster:sysprofile where name='bufwrites' ;
let a = 100*(t2-t1)/t2 ;
let b = 100*(t4-t3)/t4 ;
let c = select value from sysmaster:sysprofile where name='fgwrites' ;
let d = select value from sysmaster:sysprofile where name='lruwrites' ;
let e = select value from sysmaster:sysprofile where name='chunkwrites' ;
let f = select sum(value) from sysmaster:sysprofile
where name in ('btradata','btraidx','dpra') ;
let g = select value from sysmaster:sysprofile where name='rapgs_used' ;
insert into MW_IO values (ts,a,b,c,d,e,f,g) ;
end procedure ;
```

Views und kritische Kennzahlen-Werte


```
CREATE VIEW Cache_Rate (Read_Cached, Write_Cached) AS
        SELECT MW_IO.Read_Cached, MW_IO.Write_Cached
        FROM MW_IO
        WHERE Read_Cached<98 or Write_Cached<98
```



Beispiel

```
CREATE VIEW Read_Ahead (ra_pages, ra_pages_used, Prozent) AS
        SELECT MW_IO.ra_pages, MW_IO.ra_pages_used, ra_pages_used/ra_pages*100
        FROM MW_IO
        WHERE Prozent<98
```